Clause Aggregation:
An Approach to Generating Concise Text

James C. Shaw

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2002

Clause Aggregation:
An Approach to Generating Concise Text

James C. Shaw

This dissertation identifies and resolves constraints related to the task of combining related clauses to formulate fluent and concise sentences. To incorporate complex linguistic constructions into text generation systems, novel algorithms were designed to systematically generate conjunction, ellipsis, and quantification constructions. CASPER a submodule in a text generation system, was designed and implemented. It can convey the same information using fewer words by taking advantage of redundancies in the input based on syntactic, semantic, and discourse information. In addition to these symbol approaches, my research also employs corpus-based statistical approaches to enhance the fluency of the generated text. By employing advance linguistic constructions and removing redundancies through clause aggregations, the generated text or speech is more fluent and concise and thus improves human-computer interface.

Contents

Acknowledgments

I would like to thank first the members of my thesis committee: Steve Feiner, Luis Gravano, Karen Kukich, Johanna Moore, and especially my advisor, Kathleen McKeown, without whose patience, support, and guidance, the present work would not have been possible.

Kathy also deserves thanks for providing a stimulating environment for carrying out natural language processing research and has kept me focus on important issues. I want to thank Karen in particular – she is the person who first provided me with lots of raw data that kindled my interest in aggregation. Much appreciation to Judith Klavans for providing feedback and encouragement over the years. Throughout the years at Columbia, I have benefited from discussions with colleagues: Regina Barzilay, Vasileios Hatzivassiloglou, Hongyan Jing, Min-Yen Kan, Shimei Pan, and Michelle Zhou. I also thankful to other members of the natural language group: Pablo Duboue, Noemie Elhadad, David Evans, Ani Nenkova, Smaranda Muresan, Carl Sable, and Barry Schiffman. They have made my stay at Columbia much more enjoyable.

I thank Michael Elhadad and Jacques Robin for creating SURGE as the foundation for me to explore clause aggregation operators.

I would like to thank my parents, Shang-Chuan Shaw and Chiu-Hwa Wu, for giving me the opportunities to pursue my interest in language and science, I feel most indebted to them for their help and love. My wife, Phyllis Lin, deserves special mention. She once complained that I am the worst project she has ever managed. Just for the record, I am sorry.

Finally, I would like to thank various agencies for providing me with financial support: Bellcore project #CU01403301A1, DARPA Contract DAAL01-94-K-0119, and NSF Grants GER-90-2406. This research is also supported in part by the National Library of Medicine under grant R01-LM06593-01 and LM06593-02, and Columbia University Center for Advanced Technology in High Performance Computing and Communications in Healthcare (funded by the New York State Science and Technology Foundation).

Chapter 1

Introduction

This dissertation studies the process of generating concise and fluent text through clause aggregation. *Clause aggregation* is the process in which two or more linguistic structures are merged together to formulate a single sentence. This study incorporated linguistic knowledge into generation systems to synthesize many syntactic constructions systematically. It is an effort to bridge the generation gap and to satisfy Meteer's *expressibility* criterion: "A text plan is expressible if there are linguistic resources for realizing the elements in the plan and their composition conforms to the syntactic rules of composition in the language" (Meteer, 1991a, p. 297). Despite various linguistic theories, such as the Revised Extended Standard Theory, Head-driven Phrase Structure Grammar (HPSG), Lexical Functional Grammar (LFG), Lexicalized Tree Adjoining Grammar (LTAG), and Categorial Grammar, which have all provided in-depth analyses of coordination and subordination constructions, there has been little treatment of these constructions in generation systems to date. In their survey paper on aggregation, Reape and Mellish (1999) noted the dearth of linguistic theories in the aggregation literature, especially in syntactic aggregation. The current work describes algorithms to systematically formulate complex sentences through clause aggregation operations.

1.1 Clause Aggregation

Clause aggregation employs linguistic constructions to combine clauses. Simple constructions include simple conjunctions (e.g., the conjoined objects in "The patient received Fentanyl <u>and</u> Protomaine") and adjective phrase attachment (e.g., the adjective in "Fido is an <u>intelligent</u> dog"). Advanced linguistic constructions include quantification, ellipsis, and gapping (e.g., the deletion of a second verb in

"John likes Mary and Phil ϕ^1 Sue"). The term *clause aggregation* is the title of the present study because the basic linguistic units to be aggregated are clauses. In a text generation system, the basic unit to be combined is a *proposition*, which is also known as a *message* in other natural language generation (NLG) literature (Reiter and Dale, 2000; McKeown, Kukich, and Shaw, 1994). The main difference between a "clause" and a "proposition" is that "clause" is a linguistic/syntactic concept while a "proposition" is a semantic concept. In this dissertation, we will use the term "proposition" to refer to a relation that represents an event or a state in a domain. The term "clause" will be used when the syntactic issues are involved.

Previous generation systems generated sentences containing coordinating conjunctions and relative clauses, but these complex linguistic structures were directly provided to the generation systems or created through domain-specific rules. Early generation systems such as PROTEUS (Davey, 1979) employed coordinating conjunctions based on Systemic Functional Grammar (Halliday, 1994). Although various researchers, including McDonald (1983a) among others, noted that linguistic theories were pertinent to generate complex sentences, they did not discuss specific aggregation issues, such as linear ordering between aggregated constituents. Mann and Moore (1980; 1981) coined the term "aggregation" for clause combining operations and identified several related issues. They proposed various aggregation rules that combine clauses through logical derivations. In the early 1990s, simple clause aggregation operators were applied to discourse structure without using syntactic information. In EPICURE, Dale (1992) described clause aggregation as an optimization. Meteer (1993) identified the problem of *expressibility* and proposed a level of representation called *text structure* that permits abstract characterization of linguistic resources in order to bridge the generation gap – the problem that clause aggregation also addresses. In the late 1990s, syntax started to play a more prominent role in clause aggregation (Robin, 1995; Dalianis and Hovy, 1993; Shaw, 1998b). Chapter 8 describes these works in more detail and provides in-depth comparisons between them.

1.1.1 The Need for Aggregation

Two main developments triggered the recent surge in research on aggregation. As more NLG systems started using real-world data encoded in representations that were not originally designed for NLG purposes, it became difficult for underlying applications to provide text generation systems with the right amount of informa-

[1] ϕ indicates a constituent is deleted.

tion to form a sentence in the output text. As a result, mapping each proposition into a sentence often resulted in a text with redundant expressions and disfluencies. The other trigger was the need for researchers to produce more complex sentences tailored to a particular situation and discourse. In the 1990s, researchers were familiar with such technology as schema and rhetorical structure theory and gained much experience using such NLG tools as KPML (Matthiessen and Bateman, 1991), SURGE (Elhadad and Robin, 1997), or RealPro (Lavoie and Rambow, 1997). Because producing a text was no longer a difficult issue, researchers turned their attention to making the text more fluent and more similar to that produced by humans. Robin (1995) pointed out the ongoing gap in sentence complexity between computer-generated and human generated sentences. For example, no current NLG system comes close to producing sentences similar to those on the front pages of the New York Times in terms of complexity. The following sample consists of 51 words:

(1) All decked out in bunting and civic pride for the forthcoming Republican National Convention, this city today suffered the ignominy of seeing its police on national television beating and kicking a man accused of stealing a police car and shooting at an officer during a running gun battle through the streets. (the New York Times, July 14, 2000, p. A1)

Research in clause aggregation attempts to produce fluent and complex sentences to emulate human linguistic performance. To achieve this, it is necessary to identify what information must be encoded in order for a system to produce sentences with these constructions, and to determine algorithms to mechanically synthesize these constructions with the appropriate information.

1.1.2 Benefits of Aggregation

Incorporating aggregation into a text generation system improves the following three aspects of a text:

- **conciseness**: A text is more concise if it can convey the same information using fewer words. This involves removing redundant and inferable information.

- **cohesion**: A text is more cohesive if it is more tightly integrated and acts as a whole. Cohesion is a property that makes a text a semantic unit rather than a jumble of unconnected phrases.

- **fluency**: A text is more fluent if it flows more quickly and takes less effort to read. Related factors include variety in sentence structure and lexical items, unambiguousness, and adhering to communicative convention (Grice, 1975).

These aspects are not independent but interact in a complex relationship. As Robin (1995) pointed out, one very successful strategy to achieve conciseness is to use complex sentences, although this may seem paradoxical. Many aggregation operations remove redundant or inferable constituents from surface forms, thus expressing the sentence in fewer words. The effect of clause aggregation on conciseness is clearly seen when the sentences in (2a) are transformed into (2b), a complex sentence that conveys the same information but using fewer words.

(2) a. The patient was admitted on Monday.
The patient was discharged on Friday.

b. The patient was admitted on Monday and discharged on Friday.

The aggregated version is more concise than the unaggregated ones because the references to repeated entities are removed. Halliday and Hasan (1976) listed various constructions as cohesive devices – pronouns, ellipsis, substitution of generals for specifics, and elision of redundant information – which make the text seem to be a whole instead of unrelated parts. Since aggregation operators synthesize many of these linguistic constructions, they are also cohesive devices.

Fluency is the ease with which a hearer attempts to understand the text. Currently, fluency is hard to measure objectively. Factors that directly impact fluency include sentence complexity, variety of syntactic structure and lexical items, and text structure. Aggregation contributes to fluency by increasing the variety of syntactic structures and adhering to communication convention (Grice, 1975). By combining structurally similar propositions using coordinating conjunctions, aggregation operators reduce the occurrences of syntactically similar sentences. Humans use aggregated constructions in everyday language; therefore, without their incorporation into a generation system, communication conventions would be violated and cause undesirable implicatures. Without aggregation, sentences such as "Someone ate apples. Someone ate oranges." would be ambiguous. The normal interpretation of these sentences would be that one person ate apples while a different person ate oranges. But when the existentially quantified references in both sentences refer to the same person, the unaggregated version is semantically anomalous with respect to normal reading. Another aspect of clause aggregation that improves fluency is the linear ordering of aggregated constituents. Since words in a sentence must be ordered, once constituents are aggregated, the system must specify a linear ordering among the constituents. Native speakers would consider that both sentences (3a) and (3b) are grammatical, but (3a) is more fluent.

(3) a. John ate a large red apple.

b. John ate a red large apple.

By ensuring that a generation system use the same linear ordering as humans do, aggregated sentences are more natural and again, avoid undesirable implicatures. Thus, incorporating aggregation into a generation system can improve conciseness, cohesion, and fluency of the generated text.

1.2 Issues in Clause Aggregation

To make sense of the complex clause aggregation phenomenon, clause aggregation operators must be identified and categorized first. In our effort to build NLG systems that incorporate clause aggregation to make generated concise and fluent text, additional issues were identified and addressed.

1.2.1 What are the Clause Aggregation Operators?

Clause aggregation can be categorized into four major types: *interpretive*, *referential*, *syntactic*, and *lexical*. In general, *interpretive aggregation* uses common sense knowledge and domain-specific knowledge; thus it is application-specific and not portable. Interpretive aggregation is outside the scope of current work. *Referential aggregation* takes advantage of specific knowledge the hearer has, such as discourse and ontological information, to combine multiple propositions. An example of a referential aggregation operation is quantification, such as "`every`" and "`all`," which will be the focus of Chapter 6. *Syntactic aggregation* includes both paratactic and hypotactic constructions. *Parataxis* is a term referring to a construction in which elements of equal status are linked together. Examples of paratactic constructions are coordinations, such as "John and Mary like school". *Hypotaxis* describes a nucleus-satellite or subordinate relationship between the linked elements. Hypotactic constructions include modifying relationships such as adjective phrases, prepositional phrases, and relative clauses; for example, "John, who enjoys sports, was injured." All logical structures in language are either (a) paratactic or (b) hypotactic (Halliday, 1994, p. 218). Algorithms for synthesizing coordinating conjunction are covered in Chapter 5, and premodifiers in Chapter 4. *Lexical aggregation* combines multiple lexical items to express them more concisely. Although it is an interesting topic, the current work does not cover lexical aggregation in depth because proper treatment of the topic deserves a separate thesis, if not more. Such effort should incorporate lexical semantics, an active research topic (Pustejovsky, 1995), and capture interactions between lexical items which are still not well understood.

We focus on three aggregation operations: hypotactic aggregation involving adjective phrases; paratactic aggregation involving coordinating conjunctions and ellipsis; and referential aggregation involving quantifiers. Linguists have explored each of these operations. In performing hypotactic aggregation involving adjective phrases, the problem of adjective ordering arises when multiple adjectives modify the same head noun. For example, the sentence "John ate a large red apple" is more natural than "John ate a red large apple." The current work uses corpus-based approaches to resolve this problem. To perform paratactic aggregation involving coordinating conjunction, a unified algorithm was developed to handle coordinating conjunctions, gapping, and ellipsis. These phenomena have been known to be difficult to model well in most grammar formalisms. The referring aggregation involving quantifiers extends the algorithm proposed in coordination conjunctions. This algorithm takes advantage of discourse and ontological information to combine propositions. Both hypotactic and paratactic aggregation operators are syntactic in nature because they have a direct impact on the ordering of the aggregated constituents. Referential aggregations are more semantic in nature because quantifiers in natural language are considered a major part of semantics. A unified picture of different types of clause aggregation operators is presented in Chapter 3.

1.2.2 What are the Constraints?

Clause aggregation is not an unconstrained process. A generation system cannot randomly put two clauses together into one and expect the resulting linguistic structure to be grammatically correct and convey the same information as the original unaggregated propositions. To synthesize the kind of sentences that a human expert can write, the generation system must model information needed by such operators and observe linguistic constraints. One particularly interesting issue is the linear ordering between aggregated entities. Since the propositions do not provide ordering information, the clause aggregation procedure must determine the linear ordering among them. This ordering issue applies to both paratactic and hypotactic aggregation. In a coordinating conjunction, the ordering between conjoined constituents is influenced by preferences such as chronological ordering (e.g., "years 1998, 1999, and 2000," not "years 1999, 2000, and 1998"), or based on other pragmatic factors such as importance. In hypotactic aggregation, uncommon orderings between premodifiers modifying the same head noun create disfluencies. For example, the expression "a happy old man" is more fluent than "an old happy man." These preferences must be captured in clause aggregation to ensure fluency and avoid undesired implicatures. Chapter 4 describes how CASPER a system we developed, obtained the ordering information among premodifiers in MAGIC a multimedia generation project.

In the corpus analysis in Section 3.4, coordinating conjunction is the most common aggregation construction used.

(4) a. John ate *an apple* and *an orange*. (= NP and NP)

b. John ate *in the morning* and *in the evening*. (= PP and PP)

c. *[2] John ate *an apple* and *in the evening*. (= NP and PP)

d. * John ate *in the evening* and *an apple*. (= PP and NP)

One obvious constraint in coordinating conjunction is that the constituent being conjoined must be of the same syntactic status, such as (4a) and (4b). Those conjoined constructions with different syntactic type are ungrammatical, as (4c) and (4d). But, there are ungrammatical sentences with conjoined constituents with the same syntactic type, such as the conjoined NPs in the following sentence:

(5) John broke the window.
A hammer broke the window.
→ *John and a hammer broke the window.

In addition, there are grammatical sentences with conjunction with different syntactic categories as in (6) with a conjunction of NP with a prepositional phrase.

(6) He is Nobel prize winner (NP) and at the peak of his career (PP).

Chapter 5 identifies and discusses the constraints involved in the synthesis of coordinated conjunctions and shows why sentence (6) is grammatical while (5) is not. In this section, we describe several examples showing constraints affecting the validity

[2] We use "*" in front of sentence to denote that the particular sentence is ungrammatical, as commonly done in linguistic literature.

of the conjunction construction. There are many constraints that affect other aggregation constructions, such as adjective phrase attachment, prepositional phrase attachment, or relative clause attachment. These constraints will be explored further in later chapters.

1.2.3 What are the Algorithms to Synthesize these Constructions?

Once the features affecting the synthesis of specific linguistic constructions are identified, algorithms to synthesize them are developed to construct the sentences. Researchers employed algorithms for various constructions to increase text fluency. For example, in (Dale, 1992; Dale and Reiter, 1995; Horacek, 1997) the researchers focused on the generation of referring expressions, while McKeown (1985) and Hovy (1988) focused on text structuring. Others were interested in improving the efficiency of the algorithms in lexical choosers (Elhadad, McKeown, and Robin, 1997; Bateman et al., 1998). Even when linguistic constructions have been identified for synthesis, the algorithm for constructing them might still not be obvious. For example, in much linguistic literature, gapping ("John likes Mary and Phil Sue") and right-node-raising ("John likes and Phil hates Mary") are considered related, yet different syntactic constructions, and it is not clear if separate algorithms are needed for each. After exploring the issue, a unified algorithm was developed to handle simple coordinating conjunction, gapping, right-node-raising, and paratactic ellipsis. Similarly, although the algorithm for synthesizing multiple quantifiers in the same sentence was a mystery at first, the newly devised algorithms provided insights into these linguistic constructions.

1.2.4 Do Aggregated Sentences Convey the Original Meaning?

An aggregated sentence is *meaning-preserving* if readers can recover the exact entities and relations in the original input propositions. Given the sentences in (7a), sentence (7b) is meaning-preserving because a reader would give the same answers to questions regarding the situation described in (7b) as if they had read (7a).

(7) a. John likes Mary.
Paul likes Mary.

b. John and Paul like Mary.

c. All the boys like Mary.

d. John and Paul like the same girl.

e. John and Paul like a girl.

Sentence (7c) is meaning-preserving if readers have the appropriate context in which the only boys are John and Paul. On the other hand, in (7d), the substitution of a definite noun phrase for "Mary" might make recovery of who the girl is difficult for the readers; as a result, (7d) is not meaning-preserving. In (7e), the information that both "John" and "Paul" like the same girl is not explicit. Depending on individual interpretation, a reader might answer "one" or "two" to the question "how many people do John and Paul like?" Reusable clause aggregation operators should have the property that the aggregated sentences convey the same information as the unaggregated propositions. Without such guarantee, aggregation operators might alter the meaning of the input propositions, thus invalidating the output text and making such operators untrustworthy.

When propositions are combined, the resulting sentence might convey information that does not exist in the original propositions. Because certain constituents are shared in the resulting sentence, multiple interpretations of the resulting sentence might be possible. For example, the combined sentence (8a) can be interpreted to be either that the boys separately lifted the piano as in (8b) or that they together lifted the piano as in (8c):

(8) a. John and Paul lifted the piano.

b. John lifted the piano.
Paul lifted the piano.

c. John and Paul together lifted the piano.

In (8b), the coordination of smaller units is logically equivalent to coordination of clauses. But the sentence (8c) cannot be analyzed as separate clauses. Chapter 5 describes linguistic devices that can ensure that aggregated sentences are not ambiguous with respect to these two readings.

Avoiding ambiguity is also a problem in combining clauses by using quantified expressions. In addition to distributive reading versus collective reading problems in coordinating conjunctions, a generation system also needs to ensure that when multiple quantifiers are synthesized in the same sentence, the scope of the quantifiers is not ambiguous. These ambiguity issues related to quantifiers are discussed in detail in Chapter 6.

1.2.5 How Good are the Aggregation Algorithms?

To verify that the present algorithms generate these linguistic constructions correctly, they needed to be evaluated. Aggregation improves the conciseness, cohesion, and fluency of a text (these benefits were described earlier in Section 1.1.2). Since aggregation operators remove redundancies from the meaning representations, the claim that the resulting text became more concise is easier to sustain. The most obvious approach is simply to count the number of words. The number of redundancies that aggregation operators can remove depends on the nature of information specified by the underlying application. Using a small corpus, Dalianis (1996) estimated that his aggregation operators reduced 10–34% of an unaggregated text. For cohesion and fluency, the evaluation problem is more difficult. Subjective evaluation can be used to determine if aggregation improves these aspects of a text, but this approach is problematic because many factors might affect text quality. For example, a reader might think the generated text is too verbose even after aggregation operations have removed many of the redundancies from the unaggregated version. The problem in such text is that it contains too much detailed information, which is in fact a problem with the content planner, not with clause aggregation. In addition to such confounding effects, it is difficult to determine how much clause aggregation actually contributed to text cohesion and fluency. Callaway and Lester (1997) performed a subjective evaluation demonstrating that in a blind prose study with a panel of four judges, the majority of readers preferred the aggregated texts over the unaggregated ones.

Instead of evaluating aggregation using a black-box approach based on examining the whole text, it seems more informative to use a glass-box approach. In this study, each clause aggregation operation was evaluated individually using a corpus containing the related constructions. By limiting the scope of each evaluation, it was possible to find objective criteria to determine how well each aggregation operation performed. The evaluations for each construction will be provided in the corresponding chapters. By demonstrating that each individual linguistic construction can be generated correctly to show the relevance of each individual linguistic construction to both cohesion and fluency, it is then possible to claim with confidence that aggregation improves the conciseness, cohesion, and fluency of the synthesized text.

1.3 Two Applications

In order to support the generation of aggregated sentences, two language generation systems that provide the appropriate infrastructure are now presented: MAGIC, in the medical domain, and PLANDOC, in the telecommunications domain, are two experimental settings for clause aggregation. These applications provide operational environments to experiment with clause aggregation operations. The underlying applications for these systems were developed by experts in the corresponding domains before natural language generation modules were considered as an extension of the original systems. Successful deployments of the generation systems can provide users with timely information not previously available. Despite its facilitation for improving the quality of the text produced in these systems, clause aggregation was not a factor in choosing domains for NLG research. As a result, aggregation seems to be a general process that can improve the quality of automatically-generated text in many domains.

1.3.1 MAGIC

MAGIC is a joint project between the Columbia-Presbyterian Medical Center and Columbia University. Instead of one underlying application, MAGIC interfaces with a database to gather the information to be conveyed. The goal of the system is to provide a standardized multimedia presentation (speech and graphics) to healthcare providers in a timely fashion. When a cardiac patient reaches the Intensive Care Unit (ICU), a variety of information concerning the patient's condition and status must be summarized for the ICU medical team, including existing medications, ventilation parameters, laboratory results, demographics, and past medical history. This summary is usually given verbally by a physician, or anesthesia resident to another physician and nurses in the ICU. Because the participating caregivers are extremely busy, the information is provided only once and only after the patient arrives. The report's quality varies with the experience and preferences of the reporting physician and the questions asked by the recipients. Personnel currently provide some critical information about the patient via telephone during the operation, but this information is cursory and the personnel are usually rushed. Communicating an interim postoperative status report to the ICU personnel prior to the patient's arrival is believed to have a beneficial effect on patient care by giving ICU personnel more lead time to address otherwise unanticipated patient needs. Thus, the goal in developing MAGIC is to produce a full interim report automatically, eliminating the need for interim telephone calls.

MAGIC exploits the extensive online data available through the Columbia-Presbyterian Medical Center (CPMC) as its source of content for its briefing. Operative events during surgery are monitored through the LifeLog database system (Modular Instruments Inc.), which polls medical devices (ventilators, pressure monitors, and the like) from the very start of the case to the final recording of such information as vital signs. In addition, physicians (anesthesiologist and anesthesia residents) enter data throughout the course of the patient's surgery, including the start and end of cardiopulmonary bypass as well as subjective clinical factors such as heart sounds and breath sounds that cannot be retrieved by medical devices. In addition, CPMC main databases provide information from the online patient record (e.g., medical history). From this large body of information, the data filter selects information that is relevant to bypass surgery and ICU patient care. For example, generating a sentence for each fact in a medical domain might result in the following text:

(9) The patient is Jones.
The patient is 80-year old.
The patient is female.
The patient has hypertension.
The patient has diabetes.
The patient's doctor is Smith.
The patient is undergoing coronary heart bypass surgery.

The multiple appearances of the noun phrase "the patient" are clearly redundant and should be deleted. Clause aggregation operators can express them in a more concise sentence:

(10) Jones is an <u>80-year old</u> <u>hypertensive</u> <u>diabetic</u> <u>female</u> patient <u>of Doctor Smith undergoing coronary heart bypass surgery</u>.

The clauses in (9) are transformed into (10), a reduction of 48% in terms of word count (15/31). With respect to clause aggregation, the main difference between PLANDoc and Magic is that Magic contains a more diverse set of rhetorical relations among the propositions being aggregated. The main rhetorical relations in PLANDoc are limited to Addition and Sequence while in Magic, the rhetorical relations include also Elaboration which enable a clause to be transformed into an adjective phrase, prepositional phrase, and relative clause. As a result, Magic handles more varieties of the aggregation operations than a coordinating conjunction, the only aggregation operator in PLANDoc.

1.3.2 PLANDoc

PLANDoc was a joint project by researchers from Bellcore and Columbia University (Kukich et al., 1994; McKeown, Kukich, and Shaw, 1994). The underlying application was a forecast system in the telecommunications domain. The goal of the system was to produce documentation describing the interactions between telephone network planning engineers and the forecast system. The job of telephone network planning engineer was to derive a capacity expansion (relief) plan specifying when, where, and how much new copper, fiber, multiplexing and other equipment to install in the local network to avoid facilities exhaustion. Planning engineers have the benefit of a powerful software tool, the Bellcore LEIS-PLAN^{TM} system, with which they can derive a 20-year relief plan that optimizes the timing, placement, and cost of new facilities for a route in the network. Until now, they have not had the benefit of a tool to help them with the equally important, but often tedious, task of documenting their planning decisions; thus, in many busy shops, this crucial writing task has been upstaged by more pressing planning tasks. Documentation is needed primarily to provide a record of the planner's activities and reasoning that can be used for future network studies, for informing managers who are responsible for authorizing project plans, and for justifying expenditures to internal auditors and external regulators. PLANDoc enhances LEIS-PLAN by providing a natural language text generation system.

The information specified by the underlying application in PLANDoc concerns actions, equipment, location, and time. A particular session between the planning engineer and the system might have up to forty actions tried by the engineers. Because of the nature of the application, the same equipment might participate in multiple events, or different equipment might be placed in the same location or time. Since some entities and actions might be mentioned multiple times in the output text, generating a sentence for each action can result in a verbose and repetitive document. Conjunction was an obvious linguistic device to remove redundancies in the generated text. Because entities and actions can reappear in equipment, location, and time slots in various combinations, simple rules limited to conjunction of a single constituent such as object grouping (e.g., "PLAN 1001 removed MUX1 <u>and</u> MUX2 in CSA 1234 in year 2005.") were not sufficient to formulate conjoined sentences. Furthermore, certain combinations of the events from the underlying system created difficulties for Systemic Functional Grammar (Halliday, 1994), the linguistic formalism of the present system. Specifically, the issue of non-constituent coordination was very puzzling. For example, the sentence

(11) PLAN 1002 activated MUX1 in CSA 1234 and MUX2 in CSA 5678.

contains a conjunction of two constituents, "MUX1 in CSA 1234" and "MUX2 in CSA 5678." Since NP-PP does not form a basic syntactic category, the conjunction of such derived constituents is non-constituent coordination. Despite the fact that Systemic Functional Grammar uses function roles instead of syntactic categories, the way to handle such syntactic constructions was not obvious.

1.4 Contributions

The present dissertation studied and proposed algorithms to automate the process of creating concise sentences with complex linguistic constructions. Various linguistic insights were extracted from linguistic research and applied in natural language generation systems. In addition, abstract knowledge resources were systematically incorporated into the generation systems to enhance the quality of the generated text, e.g., ontological and discourse information.

From an engineering viewpoint, the goal was to create computational systems that mimic human language performance:

1. **Using statistical approaches to provide constraints and preferences to improve fluency.** By ensuring that the ordering of aggregated premodifiers appears in the same order that a human would order them, generation systems can improve the fluency of generated text.

2. **Using symbolic approaches to enhance both the conciseness and cohesion of the generated text.** A unified algorithm to synthesize various types of coordinating conjunction constructions, such as gapping, right-node-raising, and paratactic ellipsis, was proposed. These constructions were considered related but separate phenomena. Instead of proposing multiple rules, a generalized algorithm was designed to treat these constructions as variations of the same phenomena.

3. **Providing an abstraction for other NLG researchers.** The aggregation operations studied in this study were domain independent and reusable. This was demonstrated by employing them in two applications for different domains, MAGIC and PLANDOC. NLG researchers can take these operators for granted and focus on other techniques for making text more concise and fluent.

4. **Using more abstract knowledge resources systematically to enhance the quality of the generated text.** Traditionally, discourse history and ontology have been considered more abstract resources in language processing. The synthesis of quantified referring expressions was a clear attempt to incorporate more abstract resources in order to influence the surface forms.

1.5 Overview of the Dissertation

Chapter 2 describes a system architecture that provides the appropriate infrastructure to implement clause aggregation operators. In addition to the system architecture, Chapter 2 describes the representations used between these components. A predicate-argument structure was chosen as the underlying linguistic representation for clause aggregation.

Chapter 3 examines clause aggregation in detail. First, clause aggregation operators are categorized in order to consolidate previous work in this area by using both information source and complexity of the operations. Second, general issues in clause aggregation are discussed.

Chapters 4, 5, and 6 focus on specific types of aggregation operators. In Chapter 4, various hypotactic aggregation operators are described, including adjective phrase attachment, prepositional phrase attachment, and relative clause attachment. The specific constraint analyzed in this chapter is identifying the linear ordering among aggregated adjectives modifying the same noun. A corpus-based approach to model and obtain such information is also presented. Chapter 5 discusses coordinating conjunction. In doing so, I proposed a unified algorithm to generate simple conjunction and non-constituent coordination (conjunction of constituents of different syntactic categories). Chapter 6 describes the generation of referring quantified expressions. In addition to computing appropriate contexts for generating universal and existential quantifiers, the proposed quantification algorithm ensures that the scopes of the quantifiers are obvious in the generated sentences. The avoidance of generating ambiguous sentences is discussed in both coordinating conjunction and quantification.

Chapter 7 discusses the ordering between clause aggregation operators, while Chapter 8 examines related work in clause aggregation. The final chapter, Chapter 9, discusses conclusions and future extensions to this work.

Chapter 2

System Architecture and Representation

This chapter describes a system architecture that provides an appropriate infrastructure to implement clause aggregation operators and a framework for discussing aggregation. Attention is then turned to the representation used in the system. The chapter ends with a description of the representations produced by each module and their involvement with clause aggregation operators.

2.1 The System Architecture

Though not all natural language generation (NLG) researchers agree on how to break down the generation process, it was common in the 1980s to divide the generation process into two stages. The two-stage model consists of *strategic* and *tactical* components to handle different aspects of the generation process (Thompson, 1977). The *strategic* component addresses what information needs to be conveyed and the *tactical* component decides how to convey the selected information. Significant progress was made in the mid-1980s and early 1990s in the strategic aspect of the generation process (McKeown, 1985; Hovy, 1993). But due to the complexity of the task and limited theoretical understanding of the strategic planning, domain independent and domain dependent knowledge are often interwoven, which makes strategic components difficult to reuse across different applications. On the other hand, tactical components deal with syntax and morphological issues which are relatively domain independent. Several surface realizers for English have been developed as a general tool for tactical components, such as KPML (Matthiessen and Bateman, 1991), SURGE (Elhadad and Robin, 1997), and RealPro (Lavoie and Rambow, 1997). They have been successfully incorporated into various applications. For example, SURGE has been used in PLANDoc (McKeown, Kukich, and Shaw, 1994; Kukich et al., 1994), KNIGHT (Lester and Porter, 1997), Revisor (Callaway and Lester, 1997), and Magic (Dalal et al., 1996).

The two-stage process is often implemented as a pipeline. In pipelined architectures, the information produced in later components does not pass back to earlier components to influence decisions. As a result, the potential interactions between the components are ignored. Despite the fact that a pipelined approach limits a system's ability to handle certain linguistic phenomena, most applied NLG systems implement it for reasons of efficiency, simplicity, and maintainability. Rubinoff (1992) proposed an interleaved model which can take advantage of feedback in an NLG system. As a consequence of distinguishing between strategic and tactical components, Meteer (1991b; 1993) introduced the notion of a "generation gap" between the two components. She designed a representation called *text structure* in order to ensure that the language-independent representation from the strategic component can be expressed by the tactical component.

As researchers gained more experience with surface realizers, the tactical component was further divided into two smaller modules, the sentence planner and the surface realizer. In the late 1990s, the consensus architecture of a three-stage pipelined model was employed in various applied NLG systems (Rambow and Korelsky, 1992; Reiter, 1994; Wanner and Hovy, 1996; Reiter and Dale, 2000), as shown in Figure 2.1:

- **Content planner**: Selects the information to be conveyed and determines the text structure to convey the selected information. Other common terms for this module include *text planner*, *strategic planner*, and *macroplanner*.

- **Sentence planner**: Selects lexical items and sentence structures to convey the concepts and relations from the content planner. Other common terms for this module include *microplanner*.

- **Surface realizer**: Transforms a lexicalized linguistic structure into a linearized string. This module was known as *tactical planner*, but in current analysis, the tactical planner includes both the sentence planner and the surface realizer.

Rambow (1992) and Reiter (1994) described several applied NLG systems which include a "sentence planning" component. Wanner (1996) and Reiter (2000) discussed various subtasks inside the sentence planning module.

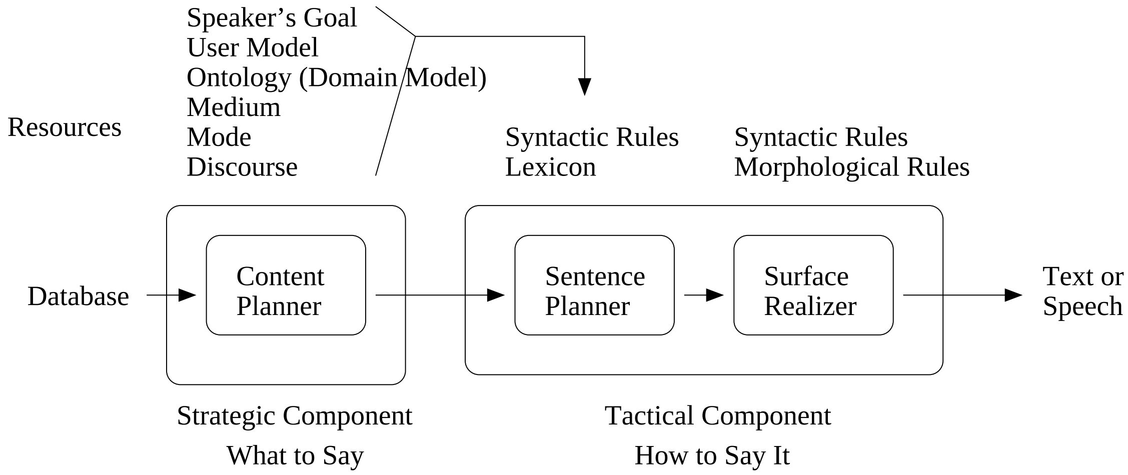

Figure 2.1: The Consensus NLG System Architecture

2.1.1 Strategic Planner

A strategic planner is responsible for deciding on what information to communicate and how to structure such information for presentation. In general, this component and the discourse plan produced are relatively domain-specific and language-independent (Rambow and Korelsky, 1992; Shaw, 1995). Since medical decisions are made in the same way, regardless of which natural language will be used to formulate the report, strategic planners can be considered a language-independent component. By integrating with different tactical components, the reports can be generated in English, French, or Chinese. Researchers usually consider the strategic planner to be a single integrated module–*content planner*. In practice, the strategic planner can be broken down into two parts: (1) a *host application* for which the domain experts are fully responsible and (2) a *content planner* in which NLG researchers are interested. Both PLANDoc and Magic have these two modules. In PLANDoc, the host application was developed by experts in telecommunications. Similarly, for the Magic system, healthcare professionals created the databases. The following are the tasks of a host application:

- **Performing data collection**: The information to be conveyed needs to be collected from various sources. In Magic, the data are collected from mainframe databases containing patients' medical records and results from various monitoring machines in the operating room.

- **Making domain-specific inferences**: The medical domain requires performing inference as is done in expert systems (e.g., MYCIN (Shortliffe and Buchanan, 1984)). Since the generated reports might influence treatments of patients, validating those rules is essential. An evaluation of the inference rules in MAGIC was performed in Jordan et al. (2001).

Since acquiring the knowledge needed to build these applications might require many years of training, most language researchers treat host applications like black boxes and rely on domain experts to provide the necessary modifications to adapt the underlying applications for text generation.

Compared to the other modules, the content planner is a relatively language-independent module. Though evidence demonstrates that incorporating language-specific decisions in the content planner can improve the overall quality of the generated text (Rubinoff, 1992), doing so consistently in applied systems is complex and difficult to maintain. To simplify implementation and ensure reasonable system performance, the content planners in PLANDOC and in MAGIC do not access the lexicon. They perform the following tasks:

- **content selection**: Based on the data provided by the host application, the content planner determines what information should be presented to the users of the system. Since the system only presents a subset of the information provided by the host system, domain experts are crucial in the design.

- **proposition formulation**: Once the content selection is performed, the content planner transforms selected information into propositions, each of which roughly corresponds to an English clause. Since the representation in the host application is application-specific and language-independent, it is first transformed into a linguistic representation which can be processed by NLG modules. Predicate argument representation was chosen as the meaning representation for these propositions.

- **text structuring**: The content planner figures out a way to present the information to the users of the system in a coherent manner. This includes deciding the sequential ordering between the propositions and specifying rhetorical relationships between them.

All these tasks are performed simultaneously in the content planner. Based on the raw data provided by the host system, the content planner performs its tasks while satisfying constraints from communicative goals, the user model, media, mode, and discourse. Two main approaches can implement the content planner. The first approach is based on schemas (McKeown, 1985), which specify what information will be conveyed and what order to convey them in a pre-specified structure. To create a discourse plan, the schema is traversed and the discourse plan is instantiated. The other approach is to use operators similar to the ones in rule-based languages such as OPS5 (Brownston et al., 1985) or CLIPS (Giarratano and Riley, 1998). Instead of using a pre-specified structure to create a discourse plan, the discourse plan is computed dynamically using a top-down planning mechanism (Hovy, 1988; Moore and Paris, 1989; Hovy, 1993). Since the discourse plan is computed dynamically, the second approach has the advantage that a more tailored report is possible and is better suited for interactive systems. An operator-based version of the content planner was build for MAGIC in the summer of 1996. It was based on UCPOP(Penberthy and Weld, 1992), a partial order planner. Implementation was discontinued after realizing that the search space was too large and the execution time for a partial-ordered plan, more than 20 minutes, was unacceptable. This can be especially problematic when adding operators which increase the search space exponentially.

Currently in MAGIC, a variation of the schema approach (McKeown, 1985) is implemented to perform content selection, text structuring, and proposition formulation. Based on the user model, data from the databases and inference engine, an appropriate schema is selected. While traversing the selected schema, the content planner populates entities and relations in the discourse plan. The text structuring was performed by linking propositions with rhetorical relationships; this operation is application-specific. For example, in the MAGIC system, the drugs given after a medical abnormality can be classified as either a CONSEQUENCE or SEQUENCE relation, depending on which inference rule is fired. The resulting hierarchical structure is then sent to the sentence planner, which has access to language-specific resources such as a lexicon and grammar.

2.1.2 Tactical Component

The content planner only performs operations that do not require syntactic or lexical information. This constraint was imposed to simplify implementation and to ensure the efficiency of the content planner. The tactical component is divided into two submodules, a sentence planner and a surface realizer.

2.1.2.1 Sentence Planner

The sentence planner is the focus of this dissertation. Since it performs its tasks by accessing language-specific resources, such as syntax and lexical knowledge, the sentence planner is a part of the tactical component. The sentence planner used in PLANDoc and Magic is called Casper (**C**lause **A**ggregation in **S**entence **P**lann**ER**). Casper takes the hierarchical plan created by the content planner and refines it into a sequence of lexicalized and aggregated propositions for the surface realizer to transform each lexicalized proposition into a sentence. The sentence planner is further divided into three modules:

- **referring expression generation module**: It decides what attributes to use to identify a particular entity to users, and requires a user model and discourse history to make its decisions.
- **clause aggregation module**: It combines multiple propositions into a sentence while satisfying various linguistic constraints.
- **lexical chooser**: It chooses lexical items and syntactic structures to realize the concepts and relations in a proposition. This module also ensures fluency by employing paraphrasing.

Referring expression generation was pursued by Appelt (1985) as a research topic in the early 1980s. Dale, Reiter, and Horacek have also published a number of papers on the topic (Dale, 1992; Dale and Reiter, 1995; Horacek, 1997). Although the idea of combining multiple sentences to formulate one complex sentence has been around for a long time in linguistics (e.g., generalized transformations in (Chomsky, 1957)), the first NLG literature discussing clause aggregation appeared in 1981 (Mann and Moore, 1981). A detailed discussion of clause aggregation is provided in the next chapter. Goldman (1975) first used discrimination networks to implement lexical choice. More recent work in lexical choice includes (Mel'čuk and Polguère, 1987; Nogier and Zock, 1991; Stede, 1996; Elhadad, McKeown, and Robin, 1997). In addition to choosing words to express concepts and relations, the lexical chooser is also in charge of paraphrasing, an essential process to make the output text more fluent. In the lexicalization process, the most important resource is the lexicon. Encoding the lexicon for a particular domain is a very costly process and the result often is not portable. Because of domain specificity and paraphrasing requirements, the lexicons for PLANDoc and Magic were developed separately with no reuse of the implementation. Some efforts have provided a general lexicon for text generation (Jing and McKeown, 1998; Knight and Luk, 1994) based on

WordNet (Miller et al., 1990) and COMLEX (Grishman, Macleod, and Meyers, 1994). How successful these efforts will be in minimizing development time has yet to be determined.

Many interactions occur between the three submodules in the sentence planner. The decision made by one submodule might have a direct impact on the operations and result of the final sentence. The following examples illustrate the interactions:

- **Referring expression generation affects clause aggregation**: Though a coordinating conjunction can be used to delete recurring entities among the propositions, it cannot delete the recurring entities which have different referring expressions; e.g., "John$_i$ stole the money and the bastard$_i$ is never coming back."[1]

- **Clause aggregation affects referring expression generation**: Using the coordination conjunction as an example, the system can remove recurring entities from the surface level. As a result, the computation for the referring expression will be directly impacted because fewer entities need referring expression computation.

- **Lexical choice affects clause aggregation**: By combining multiple lexical items into fewer ones, the sentence complexity might decrease. For example, originally a sentence might have multiple PPs attached. By transforming one of the PPs into an adjective, the newly formed proposition might allow combination with other propositions.

- **Clause aggregation affects lexical choice**: PLANDoc includes cases where passive transformations should not be applied to the aggregated linguistic structure after clause aggregation. For example, while the following active sentence is grammatical, "This refinement used a cutover strategy of ALL for CSAs 1111 and 1112, of MIN for CSA 2221, and of GROWTH for CSA 3331", the passive version of the same sentence is much less fluent: "?A cutover strategy of ALL was used for CSAs 1111 and 1112, of MIN for CSAs 2221 and 2222, and of GROWTH for CSA 3331."

These examples show many potential interactions between these modules. Currently, Magic first invokes the referring expression module, then the clause aggregation module, then the referring expression module again, and finally the lexical chooser. This arrangement worked well for both PLANDoc and Magic.

[1]The index i indicates that both expressions refer to the same entity.

At present, no reusable sentence planner exists in the public domain. Although various algorithms were proposed for referring expression generation and lexical choice in the sentence planner, there is no such reusable module available. Many of the clause aggregation operators proposed in CASPER are domain-independent and reusable. This dissertation is a step toward creating a reusable sentence planning module to further shorten the development time of NLG systems and to increase the robustness of NLG systems.

2.1.2.2 Surface Realizer

Once propositions have been lexicalized and aggregated by the sentence planner, each aggregated proposition is sent to the surface realizer to be transformed into a sentence. It processes one sentence at a time and transforms the lexicalized proposition into a grammatical sentence. At this stage of the generation process, discourse information and domain specific knowledge no longer have an impact on the surface realizer. The surface realizer performs the following tasks:

- Linear order between constituents: The surface realizer decides the order of constituents for a particular language. For example, in English, the subject appears before the verb phrase, and both appear before the object in an active declarative sentence.

- Agreement enforcement: Realizers can propagate gender and number agreement to ensure the grammaticality of the generated sentence. In addition, the coordinating conjunction can only be applied to constituents that are of the same syntactic category.

- Morphology: Various morphological issues are taken care of by the realizer, such as subject-verb agreement.

- Punctuation: Appropriate punctuation will be inserted into the surface form.

These tasks are generally language-specific but domain-independent. Since a surface realizer is built based on domain-independent information, it can be used in different applications. When a generation system uses one of the publicly available surface realizers such as KPML (Matthiessen and Bateman, 1991), SURGE (Elhadad and Robin, 1997), or RealPro (Lavoie and Rambow, 1997), it can take advantage of the functionalities provided by these packages and concentrate on other more problematic areas in NLG. In general, the appropriate surface realizer to pick for developers depends on their familiarity with the particular linguistic formalism embodied in a realizer. Dale and Reiter (2000) provide a good overview of various realizers in their work.

Both PLANDoc and Magic use SURGE as the surface realization module. SURGE is based on Systemic Functional Grammar (Halliday, 1994) and Functional Unification Grammar (Kay, 1984). SURGE grammar is an independent body of grammatical statements. SURGE has wide coverage of English grammar and imposes linguistic constraints by limiting combinations of features that can be used together in a feature structure. Given a correctly specified complex feature structure consisting of relative clauses and prepositional phrases, SURGE can output corresponding grammatical sentences.

2.2 The Linguistic Representation

The previous section described the three major modules of a standard applied generation system: a content planner, a sentence planner, and a surface realizer. This section focuses on the basic unit of meaning representation used in Casper: a proposition. The representation chosen for the proposition should have the following properties:

- Must be a linguistic representation.
- Must be compatible with Systemic Functional Grammar.
- Must facilitate aggregation operations.
- Should be easy to specify and understand by system implementors.

Chapter 1 notes that clause aggregation operators combine linguistic structures. To convey information, an NLG system must express the information in sentences using linguistic constructions. In addition to Systemic Functional Grammar and Functional Unification Grammar, researchers have used Tree Adjoining Grammar (Joshi, 1986) and Meaning Text Theory (Mel'čuk and Polguère, 1987) as the underlying linguistic theories for text generation. No thorough comparative evaluation has been made of these different formalisms with regard to generation. The one chosen by a researcher seems to be based on who he or she studied with and what was available locally (McDonald, 1992). Systemic Functional Grammar (SFG) is popular for language generation. It places a central focus on grammar as a complex resource for achieving communicative and social goals. Within SFG, grammatical description is organized around features appropriate for the expression of specific meanings. These functions, rather than the structural regularity of syntax, determine the organization of the grammar. In this formalism, a surface form is viewed as the consequence of selecting a set of features from this systemic network. SFG, a theory of meaning as choice, explicitly represents three general types of functionalities:

- *ideational*: The meaning representing the world; it largely corresponds to *propositional content*. Examples include "actor" and "process".
- *interpersonal*: The meaning concerned with a linguistic form as an action between speaker and hearer. Examples includes "mood" such as interrogative or imperative.
- *textual*: The resource for using language appropriate to a particular context in a text. Examples include "theme" and "rheme".

To specify a clause, all three distinct structures are used. Many linguistic formalisms, such as Transformational Grammar, do not have much to say about interpersonal and textual aspects. The representation used in SFG is called a Functional Description, or FD. It is a special type of *feature structure*. Various aspects of feature structure and unification have been studied in (Shieber, 1986; Kasper and Rounds, 1986; Johnson, 1988; Carpenter, 1992).

Although Systemic Functional Grammar is well developed, it is difficult for people not familiar with the formalism to specify or create SFG representations. In addition, not all thematic roles are useful in clause aggregation operations. For example, the feature structures in (12a) are the propositions corresponding to sentences in (12b) and (12c). Despite the fact that the entity "John" appears in different thematic roles in these propositions (agent versus carrier), conjunction operators still delete the second occurrence of "John" to remove redundancy:

(12) a.

$$\text{S1:} \begin{bmatrix} \text{process} & \begin{bmatrix} \text{type} & \text{material} \\ \text{concept} & \text{eat} \\ \text{tense} & \text{past} \end{bmatrix} \\ \text{participants} & \begin{bmatrix} \text{agent} & \begin{bmatrix} \text{concept} & \text{person} \\ \text{name} & \text{“John”} \end{bmatrix} \\ \text{affected} & \begin{bmatrix} \text{concept} & \text{dinner} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\text{S2:}\begin{bmatrix} \text{process} & \begin{bmatrix} \text{type} & \text{ascriptive} \\ \text{mode} & \text{attributive} \\ \text{tense} & \text{past} \end{bmatrix} \\ \text{participants} & \begin{bmatrix} \text{carrier} & \begin{bmatrix} \text{concept} & \text{person} \\ \text{name} & \text{“John”} \end{bmatrix} \\ \text{attribute} & \begin{bmatrix} \text{concept} & \text{satisfied} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

b. John ate dinner.
John was satisfied.

c. John ate dinner and was satisfied.

Since the feature-structure in SFG is difficult to specify and not at the appropriate level for clause aggregation operations, non-essential information for clause aggregation was stripped away to simplify aggregation operations. The current representation for CASPER is a predicate-argument structure influenced by Lexical Functional Grammar (LFG) (Kaplan and Bresnan, 1983) and Semantic Structure (Jackendoff, 1985; Jackendoff, 1990). In this representation, the thematic roles and number of arguments associated with each predicate are stored in the lexical entry of each predicate in the lexicon. As a result, system implementors no longer have to encode them in propositions. The following predicate-argument structures correspond to the FDs in (12a):

(13)

$$\text{S1':}\begin{bmatrix} \text{pred} & \begin{bmatrix} \text{type} & \text{event} \\ \text{concept} & \text{eat} \\ \text{tense} & \text{past} \end{bmatrix} \\ \text{arg1} & \begin{bmatrix} \text{type} & \text{entity} \\ \text{concept} & \text{person} \\ \text{name} & \text{“John”} \end{bmatrix} \\ \text{arg2} & \begin{bmatrix} \text{type} & \text{entity} \\ \text{concept} & \text{dinner} \end{bmatrix} \end{bmatrix}$$

$$\text{S2'}: \left[\begin{array}{ll}
\text{pred} & \left[\begin{array}{ll}\text{type} & \text{state}\\ \text{concept} & \text{has-attribute}\\ \text{tense} & \text{past}\end{array}\right]\\
\text{arg1} & \left[\begin{array}{ll}\text{type} & \text{entity}\\ \text{concept} & \text{person}\\ \text{name} & \text{“John”}\end{array}\right]\\
\text{arg2} & \left[\begin{array}{ll}\text{type} & \text{attribute}\\ \text{concept} & \text{satisfied}\end{array}\right]
\end{array}\right]$$

Compared to feature-structures used in SFG, this predicate argument representation does not have the problem with different role names which complicates clause aggregation operators. If thematic roles information is needed, aggregation operators can obtain it from the lexicon. Each concept in the proposition can be further modified by attaching modifiers, such as adjectives or relative clauses:

(14) a.

$$\text{S3}: \left[\begin{array}{ll}
\text{pred} & \left[\begin{array}{ll}\text{type} & \text{state}\\ \text{concept} & \text{is-an-instance}\end{array}\right]\\
\text{arg1} & \left[\begin{array}{ll}\text{type} & \text{entity}\\ \text{concept} & \text{person}\\ \text{name} & \text{“John”}\end{array}\right]\\
\text{arg2} & \left[\begin{array}{ll}
\text{type} & \text{entity}\\
\text{concept} & \text{patient}\\
\text{mods} & \begin{array}{l}
\left[\begin{array}{ll}\text{type} & \text{attribute}\\ \text{concept} & \text{male}\end{array}\right],\\
\left[\begin{array}{ll}
\text{pred} & \left[\begin{array}{ll}\text{type} & \text{event}\\ \text{concept} & \text{undergo}\\ \text{tense} & \text{present-progressive}\end{array}\right]\\
\text{arg2} & \left[\begin{array}{ll}\text{type} & \text{entity}\\ \text{concept} & \text{CABG}\end{array}\right]
\end{array}\right]
\end{array}
\end{array}\right]
\end{array}\right]$$

b. John is a male patient undergoing coronary heart bypass surgery.

In this example, the feature structure after MODS is a list. In this representation, the order between the modifiers is not yet specified. Complement and adjunct should be distinguished as this distinction affects coordinated conjunction operators. More restrictions on the representation related to clause aggregation operators will be noted in later chapters.

Other representations were also considered as the underlying representation. In the machine translation community, the use of logical forms as the semantic representation for generation is still common (Shieber et al., 1990; van Eijck and Alshawi, 1992). Using this representation, the ambiguities in quantifier scope are preserved. Since it is desirable to preserve the ambiguities during translation, the difficult task of resolving quantifier scope is often not warranted (Kay, Gawron, and Norvig, 1994, p. 95). By comparison, the representations used in both PLANDoc and Magic do not contain quantified variables. From the structured data from which the current representations are derived, no directly encoded quantified variables were found.

2.3 Representations between NLG Modules

In Section 2.1, three major modules of a standard applied generation system were described: a content planner, a sentence planner, and a surface realizer. This section focuses on the representations being passed between these modules. Specific information related to clause aggregation is pointed out.

In general, there are no specific constraints on the input representation to the content planner. As long as the representation is understandable to the domain experts and can be transformed into propositions by the content planner, the representation issues at this stage are domain-specific and handled in an ad-hoc manner.

2.3.1 Output from Magic's Content Planner

In Magic, the output of the content planner is a *presentation graph*. A portion of a presentation graph is shown in Figure 2.2. The whole discourse starts from a top-level node which is called "`discourse`." All non-leaf nodes are "`group`" nodes which only contain links to their children and have no proposition of their own. Both "discourse" and "group" nodes are shown in rectangular boxes in Figure 2.2. "`Atomic`" nodes contain one or more propositions, and are shown in ovals in the Figure. A node in the graph contains the following information:

- **level**: In Magic, the choices are either `discourse`, `group`, or `atomic`.

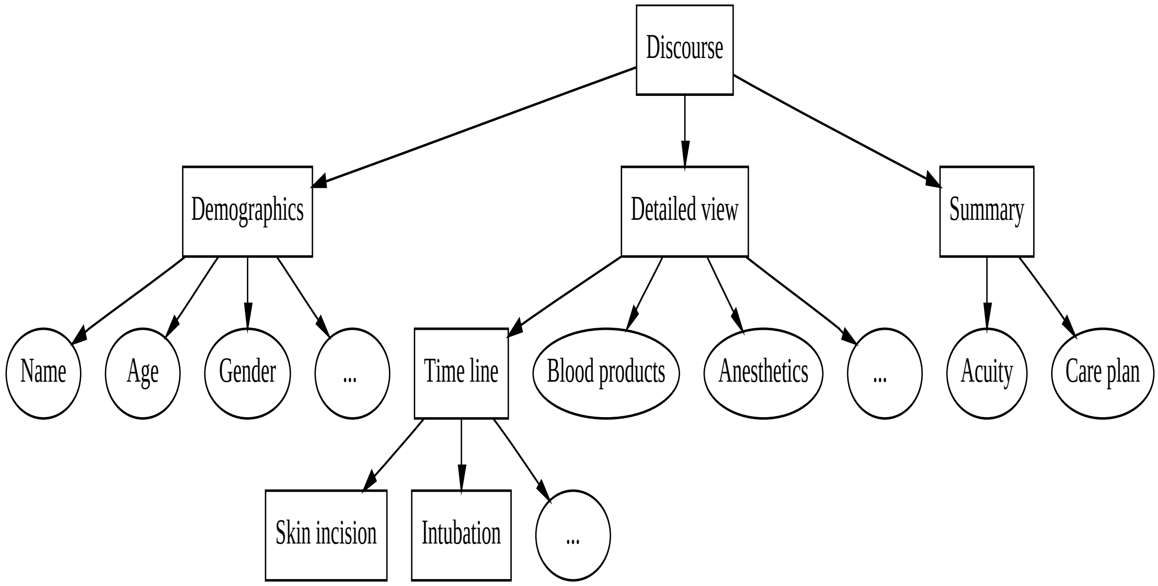

Figure 2.2: Top portion of a presentation graph in MAGIC.

- **ordering**: This feature only applies to group nodes and indicates the presentation ordering of the current node compared to others.
- **propositions**: The propositional content to be conveyed by the system.
- **rhetorical relations between nodes**: Examples of rhetorical relations include ELABORATION, SEQUENCE, and ADDITION.
- **paragraph and sentence boundaries**: This feature indicates which adjacent nodes should not be combined.

In the presentation graph, the orderings and rhetorical relations between nodes are specified by the selected schema. The schema is knowledgeable about paragraph boundaries. But since the content planner does not have access to lexical information, the specified sentence boundaries are estimated. Not all sentence delimitation decisions are related to sentence complexity. Other factors include avoiding undesirable implicatures, or information between two propositions that is not related enough to be mentioned in the same sentence. The content planner is the only module in the whole NLG system which has the necessary pragmatic information and capability to detect such implications and compute relatedness.

2.3.2 Output from Magic Sentence Planner

The output from the sentence planner is a sequence of aggregated propositions. Since each proposition has been lexicalized, the exact words have been chosen to convey the concepts and relations. In Magic, SURGE (Elhadad and Robin, 1997) was selected as the surface realizer. The representation used as input is an abstract syntactic representation based on Systemic Functional Grammar (Halliday, 1994). In this abstract syntactic representation, the order between the constituents are not specified. It is up to the grammar in SURGE to determine their order.

2.3.3 Output from Magic Surface Realizer

The surface realizer in Magic can either produce text directly as the final output of the system or send the linearized constituents to a text-to-speech module to produce speech. Currently, the text output produced is in HTML and can be displayed in standard Web browsers. The speech output is further processed by the concept-to-speech synthesizer.

Producing fluent and natural sounding speech is a difficult task. Despite commercial systems which produce speech from a written text, it is not difficult for a person to distinguish between a computer-generated speech and a human-generated one. To make the speech output of Magic acceptable to users, Pan has been working on a concept-to-speech module (McKeown et al., 1997; Pan and McKeown, 1997; Pan and McKeown, 1998; Pan and McKeown, 1999; Pan and Hirschberg, 2000). In her work, Pan takes advantage of the information provided by the surface realizer, which is much more accurate than that in state-of-the-art text-to-speech systems. For example, a text-to-speech system can use part-of-speech and syntactic information from parsers to model tone, rhythm, and melody. The information provided by the parser is inferred and not always reliable. This situation is particularly problematic when the sentences are complex – the very type of sentences that Casper creates. Since parsers have difficulties with complex sentences and Magic generates complex sentences using clause aggregation, using off-the-shelf text-to-speech systems in Magic produces awkward-sounding speech output. Pan (2000) showed that incorporating accurate information from a surface realizer can improve the quality of the generated speech. Readers interested in the topic are referred to her dissertation.

Chapter 3

Clause Aggregation

Chapter 1 explained the phenomenon of clause aggregation and why mastery of this process allows computational systems to produce text that is more concise, cohesive, and fluent. In this chapter, the first section presents an overview of clause aggregation operators and specific research issues. Section 3.2 discusses the pervasive roles of rhetorical relations in the aggregation process. In Section 3.3, clause aggregation operators are divided into four major categories: interpretive, referential, syntactic, and lexical. The rationale for such categorization is discussed. Section 3.4 investigates the frequencies of clause aggregation operators and the type of aggregation employed in a medical corpus. Later chapters will go into implementation details of aggregation operators.

3.1 Clause Aggregation

Clause aggregation is the process in which two or more linguistic structures are merged to form a single linguistic structure. At an abstract level, clause aggregation can be viewed as a two-stage process. In the first stage, a decision of which propositions might be aggregated is made. In other words, related propositions are partitioned into groups which roughly correspond to sentences. In the second stage, transformations[1] are applied to the propositions within each group to form one or more complex linguistic structures. The first stage is not concerned with surface form. It does not use syntax or lexical information and no transformation is applied to the propositions. In contrast, the second stage focuses on surface forms.

The division of labor corresponds to the division between the content planner

[1]The term *transformation* is not related to Transformation Grammar unless specifically noted.

and sentence planner. The content planner makes two decisions related to clause aggregation. First, can two propositions be combined without causing undesirable implicatures? Second, if two propositions can be aggregated, then what is the rhetorical relation or relations between the propositions? Propositions are grouped together for the following reasons:

- **The events or propositions are discourse related**: Certain propositions often appear together or in specific order. For example, doctors in the medical domain often state a patient's medical conditions followed by the drugs given to the patient to treat such conditions. Such relations between propositions are specified in the schema and can be identified or inferred based on an analysis of a domain-specific corpus. Discourse related propositions includes ELABORATION, NON-VOLITIONAL RESULT, SEQUENCE, and ADDITION, which were proposed in (Mann and Thompson, 1987; Mann and Thompson, 1988).

- **Multiple propositions obtained from the same schema rule**: Applying a schema rule might result in multiple propositions being conveyed, not just one. For example, a schema rule might convey the drugs given to a patient during a specific time period. Since there might be multiple drugs given, there will be multiple propositions being conveyed. Since these propositions are derived from the same schema rule and they must be linked to create a cohesive discourse, currently they are linked by ADDITION relations.

- **Opportunistic aggregation due to similarity between adjacent propositions in the discourse plan**: Sometimes there are similarities between adjacent propositions. Since combining two propositions might result in undesirable implicatures, the pragmatic knowledge needed for such decisions is only available in the content planner. To prevent undesirable implicatures, the content planner specifies in the discourse plan if two propositional groups must be generated as separate sentences, or if they might be further aggregated.

Depending on the desired complexity of the target sentences, each group might be realized as one or multiple sentences. The only operations applied to the propositions in the content planner are linking propositions using rhetorical relations and sentence delimitations.

In the second stage, transformations are applied to the linked propositions to formulate a more complex linguistic structure. Three aspects of the combining process need to be address: what are the transformations, what are the algorithms to carry them out, and what information must a computational system capture to support the transformations? Many grammar books describe the linguistic constructions related to these transformations (Quirk et al., 1985; McCawley, 1988). The current study focuses on the following three linguistic constructions: coordinating conjunctions, nouns modified by multiple premodifiers, and quantified noun phrases.

Compared to previous linguistic work, this analysis provides new insights into these linguistic phenomena for the following two reasons. First, the starting point of the analysis is different. Previous linguistic analyses started from surface forms and tried to deduce their meaning representations. This analysis starts from a meaning representation to derive a surface form which conveys the same information. In both PLANDoc and Magic, the meaning representations are derived from data taken from preexisting host applications. Presenting such stored data promptly and appropriately to decision makers can improve their productivity. Although the current representation is not exactly the meaning representation used by humans, it is an approximation which is reasonably captured in computational systems. Using this extra information, NLG research can provide new insights into the phenomena under study. The second reason the current study is innovative is that computational models have been built to synthesize these linguistic phenomena, which was not possible earlier. While humans use these constructions, linguists have difficulty characterizing them based on formal models such as TG, HPSG, LTAG, and LFG. To automate the computational process to synthesize these constructions, different issues were encountered and resolved. These algorithms support the claim that these linguistic phenomena are well understood.

3.2 The Role of Rhetorical Relation in Aggregation

Inferring the rhetorical relations between any two propositions is at best an educated guess. As a result, the relations must be specified by the host application or the content planner. These modules have access to the domain model and inference engine. Consider the following two clauses:

(15) John abused the duck.
The duck buzzed John.

Depending on the rhetorical relations between the two clauses, the following four aggregated sentences might result:

(16) a. John abused the duck that had buzzed him.

b. The duck buzzed John who had abused it.

c. The duck buzzed John and he abused it.

d. John abused the duck and it buzzed him.

In the first two sentences, (16a) and 16b), the main rhetorical relation connecting the nucleus clause (the main clause) and the satellite clause (the relative clause) is ELABORATION. In addition, SEQUENCE relations connect them, as manifested by the past perfect tense. The sentences in (16a) and (16b) describe very different situations. In (16a), John was the victim first before he became an aggressor, while in (16b), the duck was the victim first. In (16c) and (16d), the likely rhetorical relation between the two clauses can be either SEQUENCE, NON-VOLITIONAL RESULT, or ADDITION. Depending on which proposition in (15) is conveyed initially, either the duck is first the victim and then the aggressor or vice versa. The propositions in (15) are insufficient for a listener to determine who instigated the exchange of unpleasantries. These examples show that propositional content alone is insufficient to represent meaning, and rhetorical relations are an essential aspect of such a meaning representation which must be specified and not inferred.

The point that rhetorical relations cannot be inferred by the sentence planner is further illustrated by the following example. In the current analysis, the decision about which clauses should be combined and the exact relations between the propositions are application-specific. In MAGIC, there are situations where the drugs given after a medical condition cannot be guaranteed to relate to the treatment of that condition. As a result, sentence (17b) can be generated for the two propositions in (17a), but not (17c).

(17) a. The patient had an episode of hypertension before bypass.
The patient received Fentanyl.

b. The patient had an episode of hypertension before bypass and received Fentanyl.

c. *The patient had an episode of hypertension before bypass and was treated with Fentanyl.

In (17a), the rhetorical relation is SEQUENCE while in (17b), the rhetorical relation is NON-VOLITIONAL RESULT. This example clearly shows that it is risky for the clause aggregation module to "infer" or "interpret" the relationships between propositions. In example (18a), since syntactically the two propositions share the same subject, generating sentence (18c) would convey the propositional content concisely. But if the propositions in (18a) are linked by NON-VOLITIONAL RESULT, generating (18b) is more appropriate and clear.

(18) a. John ate sushi.
John took Pepto-Bismol.

b. Because John ate sushi, he took Pepto-Bismol.

c. *John ate sushi and took Pepto-Bismol.

These examples demonstrated that the rhetorical relationship between propositions is situation-specific and should be dictated by the content planner. Once the rhetorical relations are specified, different aggregation operators can be selected to realize the rhetorical relations.

3.3 Four Categories of Aggregation Operators

Categorizing clause aggregation operators provides a generalization of the phenomena and is helpful in defining the scope of the process. The operators are divided into the following four categories:

1. **Interpretative Aggregation**: Propositions are transformed into fewer ones based on domain-specific knowledge or operations that involve entities in multiple thematic roles. This type of operator is often not meaning-preserving.

2. **Referential Aggregation**: Propositions are collapsed based on contextual information obtained from discourse history, user model or ontology. There are two types of referential aggregation operators. The better-known type is closely related to referring expression generation (Appelt, 1985; Dale, 1992). In such operations, propositions containing the attributes necessary for identifying an entity are merged into the nucleus proposition as restrictive modifying constructions. The lesser-known type is quantification based on discourse history or ontology.

3. **Syntactic Aggregation**: Propositions are combined using syntactic constructions. The two main types of syntactic aggregation are *hypotactic* and *paratactic* aggregation.

4. **Lexical Aggregation**: Lexical items in the propositions are combined into fewer lexical items. This operation is related to paraphrasing; an example is nominalization.

These operators are categorized mainly based on the type of information needed by the operator and the resulting linguistic constructions as a consequence of applying the operators, such as quantifiers, conjunctions or relative clauses. For example, some operators use discourse and ontological information extensively, others only use syntactic information. Table 3.1 lists the categories with corresponding clause

Category	Operators	Resources	Surface Markers
Interpretive	summarization inference	common sense knowledge ontology	
Referential	ref expr gen. quantified expr.	ontology discourse	each, all, every both, some, a
Syntactic	paratactic hypotactic	syntactic rules lexicon	and with, who, which
Lexical	paraphrasing	lexicon	

Table 3.1: Four categories of clause aggregation operators.

aggregation operators, the resources used for such operators, and the surface markers which identify the operators. The details of each type of aggregation operator are presented in the next four sections.

3.3.1 Interpretative Aggregation

This is the most general form of clause aggregation. Some researchers distinguish between conceptual and semantic aggregation (Reape and Mellish, 1999). They consider *conceptual aggregation* as a language-independent operation while *semantic aggregation* is domain dependent. As Reape and Mellish (1999) state, the distinction between semantic, conceptual, and lexical aggregation is not a clear-cut process. Because of the confusion, the terms semantic aggregation and conceptual aggregation are not used in our classification. Instead, except for referential aggregation, all aggregation operations which make no use of syntactic knowledge or lexicon are considered interpretive.

Interpretive aggregation operators perform inferences over conceptions and relations across propositions. Currently, they are divided into two subcategories: ad-hoc rules and logical derivations. Two general types of ad-hoc aggregation operations are identified. One is based on complex and domain-specific transformation rules, the other is based on deletion. The first type of interpretation aggregation replaces entities in multiple thematic roles among the propositions. If only the entities in a single thematic role are substituted, then the operation is *referential aggregation*, the topic of next section. To transform the sentences in (19a) into (20a), the interpretive operator collapses the entities in ARG1 (John, Mary, John), PRED (kick, punch), and ARG2 (Mary, John, Mary).

(19) a. John punched Mary.
Mary kicked John.
John kicked Mary.

b. John fought with Mary.

In general, transformations involving multiple thematic roles are situation-specific and not meaning-preserving. Sentences (19a) and (19b) have different meanings, as demonstrated by the fact that if people read (19a), they can answer the question: "Who started the fight?". However, if they only read (19b), they cannot answer the question. For this type of clause aggregation operation, it is not enough to simply look into the ontology and confirm that "kick" and "punch" are specific types of fighting, as the concept "fight" implies. It also requires that the agent and patient arguments of both "kick" and "punch" include both parties. For example, the system should not generate the sentence (20b) based on (20a):

(20) a. John punched Mary.
John kicked Mary.

b. *John fights with Mary.

In addition, if the entity "John" occurs much more frequently as the agent of "kick" and "punch", it is not clear what the appropriate threshold should be to change the concept of "fighting" to "beat up" or "abuse." In MAGIC, an example of ad-hoc aggregation is a domain-specific operator on "cardiotonic therapy," which replaces the propositions containing the following three propositions if they are encountered in the discourse plan:

(21) The patient received drips.
The patient's devices include pacemaker.
The patient's devices include balloon pump.

They are replaced by the new proposition, "The patient received cardiotonic therapy."

Pushing the concept of interpretive aggregation to an extreme makes it similar to text summarization (Luhn, 1958; Paice, 1990; Mani and Maybury, 1999) and consists of deleting less important or informative propositions and reformulating the important propositions into concise fluent sentences. Transforming a text by deleting less important propositions is also known as sentence extraction (Rath, Resnick, and Savage, 1961; Jacobs and Rau, 1990; Brandow, Mitze, and Rau, 1995). Statistical approaches based on the positional information in the article and term frequency/inverse document frequency (TF.IDF) seem to work quite well for getting the gist of the articles in the newspaper domain. For example, taking the first two sentences of newspaper articles seems to provide adequate summaries (Brandow, Mitze, and Rau, 1995). For summarizing research papers, this simple approach has not yet proven to work adequately. For the text generation system, the reduction of information is performed in the content planner. In MAGIC, many megabytes of data are sent to the medical inference engine, but only a subset of the original input was deemed worth conveying. Instead of using statistical techniques, the deletion of non-essential propositions is based on domain-specific rules acquired from interactions with medical experts. The deletion of less important constituents share similar problems as with deletion of propositions.

Logical derivation is a domain-independent transformation. The system can generate the sentence "A implies C" from these two propositions, "A implies B" and "B implies C." By taking advantage of the entailment relation, a logical derivation rule removes B from the surface level. Aggregation operators of this type were proposed in Mann et al. (1980) and Huang and Fiedler (1996). Other logical derivations include transforming the two propositions "Liz is Sue's sister" and "John is Sue's uncle" into "Liz is John's niece." Such transformations could be either lexical transformations or logical derivations. In the current study, the operation is classified based on whether the operator uses an ontology or a lexicon. Since this dissertation focuses on referential and syntactic aggregation operators, detailed analysis of interpretive aggregation operators falls outside the scope of the current work.

3.3.2 Referential Aggregation

There are two types of referential aggregation. The first type is similar to the *referring expression generation* task, as described in the generation literature (Dale, 1992; Dale and Reiter, 1995; Horacek, 1997). It selects attributes to uniquely identify entities in the discourse for listeners. In MAGIC, each attribute of a patient is represented as a separate proposition as in (22).

(22) The patient is Jones$_{name}$.
The patient is female$_{gender}$.
The patient is 80 years old$_{age}$.
The patient has hypertension$_{medical_history}$.

The content planner does not aggregate the attributes together into a sentence earlier because it is unclear exactly how much information can fit into a sentence. Linguistic knowledge, such as lexical choice and syntactic structure, is needed before sentence boundaries can be determined. Since attributes for entities are encoded uniformly in the input propositions, the selection of attributes for referring expression and the use of such attributes affect the number of propositions for clause aggregation. MAGIC uses the "name" attribute to refer to a particular human (including patient and doctor) when the entity first appears in the discourse. As a result, the "name" proposition is removed from the input and will not be involved in the later aggregation process. Since various researchers have studied how to compute unique attributes to identify a particular entity, the quantification aspects of the referential aggregation will be the main focus.

The second type of referential aggregation is known as *quantification*. Quantification replaces a set of entities in the propositions with a reference to their type as restricted by a quantifier. The main operation in the process is replacing distinct entities with a reference to their type based on ontology. This operator is also known as *generalization* or *subsumption*. An interesting issue in this quantification process is the computation of the context for universal quantifiers (e.g., '`each`', '`every`'), using discourse history and ontological information. Various ways to obtain the context have been proposed in Chapter 6. Given the following propositions in (23a) and the context in which the only patients are John and Mary, the system can use this information to generate the sentence (23b).

(23) a. John is doing well.
Mary is doing well.

b. All the patients are doing well.

In another example, using the information in the ontology that a person has only two arms, the entities "the patient's left arm" and "the patient's right arm" can be referred to as "each arm."

Although both interpretive and referential aggregation operators use the ontology, they can be distinguished by the number of thematic roles involved in the aggregation operation and the type of operations used to access the ontology. As described in Section 3.3.1, interpretive operations involving entities from multiple thematic roles, are often domain- and application-specific. In contrast, referential aggregation operators are domain independent. Instead of unrestricted operations on the ontology, such as the capability to infer that "Liz is John's niece" from "Liz is Sue's sister" and "John is Sue's uncle," referential aggregation only sends queries to the ontology about instance-class relations (e.g., "`John`" is an instance of a type of "`patient`"), inheritance relations (e.g., "`penguin`" is a subtype of "`bird`"), and part-of relations (e.g., a person has maximally two arms). In addition, the number of thematic roles being affected by referential aggregation is usually one. Chapter 6 will describe the specific conditions in which the entities in two thematic roles are transformed into quantified expressions. The operation of subsumption has been studied in knowledge representation systems such as KL-ONE-like languages (Brachman and Schmolze, 1985; Borgida et al., 1989), and in NLP (Passonneau et al., 1996). In referential aggregation, the quantified entities are limited to a single thematic role. By limiting the operators to entities in universal quantifiers and limited cases of existential quantifiers, the information conveyed in the original propositions is preserved in the aggregated sentences with quantified expressions. In contrast, interpretive aggregations do not have this meaning-preserving property. In interpretive aggregation, generalization operations are applied multiple times to entities in multiple roles. Since the interactions between multiple generalization operations are not well understood yet, interpretive aggregation operators are not reusable. In contrast, the referential aggregation operations studied in the current work are domain-independent and reusable. Domain dependency is delegated to ontology, which, in any event, has to be tailored for each new application.

3.3.3 Syntactic Aggregation

There are two types of syntactic aggregation operations: *paratactic* and *hypotactic* constructions, both of which are distinguished by the syntactic constructions employed. The aggregated constituents in a paratactic construction are of equal syntactic status (e.g., "John likes <u>Mary and Sue</u>" contains a conjunction of NP

"Mary" and NP "Sue"). In contrast, the constituents in hypotactic constructions have unequal syntactic status, such as a noun phrase modified by a prepositional phrase (e.g., "John likes Mary who is a nurse"). The main *paratactic* aggregation operator is the coordinating conjunction. Coordinating conjunctions involve combining propositions linked by the rhetorical relations ADDITION, SEQUENCE, and NON-VOLITIONAL RESULT. The respective examples containing these relations are shown in (24a):

(24) a. The patient received Fentanyl and Protomaine.

b. Before start of bypass, the patient had hypertension and received Fentanyl.

c. Before start of bypass, the patient had hypertension and was treated with Fentanyl.

The hypotactic aggregation operators in the current study focus on transformations of propositions sharing common entities and linked by ELABORATION relations. In this type of transformation, the propositions in satellite position are transformed into modifying constructions, such as adjectival phrases, prepositional phrases, reduced relative clauses, or relative clauses. Although syntactically linguists considered *apposition* to be a paratactic construction, it is treated as a reduced relative clause with deleted verb phrase and is classified as an hypotactic aggregation operation in CASPER. A more detailed treatment of hypotactic aggregation will be provided in Chapter 4.

The operations in syntactic aggregation are based on language-specific syntax. Since to a large extent syntactic information is domain independent, these syntactic aggregation operators are reusable in different applications. During syntactic aggregation, lexical information is used to determine if the result of hypotactic aggregation will not violate any syntactic or lexical constraints. After syntactic aggregation, the aggregated proposition contains coordinated constructions in paratactic cases or a nucleus proposition with modifiers attached in hypotactic cases. Since some lexical checking is done when doing the hypotactic aggregation, preliminary assignments of syntactic categories have been performed and CASPER can guarantee that aggregated linguistic structures are expressible. The assignments of exact lexical items for the concepts in the aggregated proposition are carried out in the lexical chooser, the topic of the next section.

3.3.4 Lexical Aggregation

Similar to hypotactic aggregation operators, lexical aggregation operators also use lexical knowledge when combining clauses. The main difference between them is that in hypotactic aggregation, the satellite propositions are transformed into a modifying construction, such as an adjectival phrase, a prepositional phrase, or a relative clause. In lexical aggregation, the concepts in the satellite clause do not have to observe such restrictions. For example, in (25), the hypotactic operator first combines the clauses in (25a) into sentence (25b) using the apposition construction. The aggregated sentence can be further optimized into sentence (25c) using lexical aggregation.

(25) a. The Index closed at 510.85.
510.85 is a record-high close.

b. The Index closed at 510.85, a record-high close.

c. The Index set a record-high close of 510.85.

The phrase "510.85," which is originally in the argument position of the nuclear proposition, becomes a modifier of the phrase "a record-high close," a constituent of the satellite proposition, in the newly aggregated sentence. Compared with hypotactic aggregation, lexical aggregation operators use more detailed lexical information. Other lexical aggregation examples include the following:

- a dog used by the police → a police dog
- the book written by Stephen King → the Stephen King book
- the book about Clinton → the Clinton book
- the doctor is a woman → the woman doctor
- the doctor is for a woman → the woman doctor

In the first example, the transformation deletes the reduced relative clause and transforms it into a prenominal modifier. The other examples also require lexical information beyond the one used in syntactic aggregations. We consider the above examples a special type of lexical aggregation because the original explicit relations between the modifiers and the entities being modified are deleted from the combined expression, i.e., the relations "used by," "written by," and "about."

The patient's past medical history is significant for bladder carcinoma$_1$, status post cystectomy with a urostomy tube insertion$_2$, left nephrolithiasis$_3$, status post surgery$_4$, recurrent syncope$_5$, questionable vagovagal$_6$, a neurological workup was negative$_7$, and the EPS was negative$_8$, abdominal aortic aneurysm approximately 5 cm$_9$, high cholesterol$_{10}$, exertional angina$_{11}$, past tobacco smoker, quit about one year ago$_{12}$.

Figure 3.1: The sentence with the maximum number of propositions in the corpus

Another major type of lexical aggregation combines multiple lexemes into one. Such lexical aggregation operations were performed in Robin's generation system, STREAK(1995):

- rise sharply → shoot.
- drop sharply → plunge.

Unification was employed extensively to search through lexical space to compute various paraphrases to realize concepts (Robin, 1995; Elhadad, McKeown, and Robin, 1997). Currently both MAGIC and PLANDOC perform limited lexical aggregation through unification.

3.4 Corpus Analysis

A corpus analysis was conducted to understand how often aggregation operations are used and which ones are more popular. Because syntactic aggregations, including both hypotactic and paratactic operators, leave clear surface markers, they are the focus of this analysis. The corpus used in this study consisted of the first few sentences in the discharge summaries of 54 patients in the medical domain. These sentences describe patients' demographics and medical conditions pertinent to patient care in the Intensive Care Unit (ICU). In this study, the first step was to find out how many propositions were combined in each sentence. A proposition was defined as a piece of information that the physician (the speaker) might choose to convey in a stand-alone sentence to the nurses in the ICU (the listener). For example, a sentence "The patient is a 40-year-old female admitted for heart surgery" contains three propositions: "The patient is a female," "The patient is 40 years old," and "The patient was admitted for heart surgery."

The small corpus contained 121 sentences with 2262 words. From the 121 sentences, 418 propositions were obtained after manual decomposition, with a maximum twelve propositions in a single sentence, as shown in Figure 3.1. On average, there were 3.5 propositions per sentence. Out of 54 summary sentences (the first sentence in each discharge summary) for each patient, doctors preferred to use prepositional phrases (PPs) ("with aortic stenosis") rather than relative clauses ("who likely has endocarditis...") to insert medical conditions into a sentence (35 occurrences versus 4). In only two cases were both PPs and relative clauses used; all others have neither. The study revealed the following observations:

- Physicians produce complex sentences.
- Coordinated constructions are the most popular aggregation operations, followed by PPs and then adjectives. Present and past participle clauses are less common; relative clauses are rare.
- These aggregation operations result in long-distance dependencies and non-constituent coordinations (conjunction of constituents of different syntactic categories).

The analysis also indicated that people preferred using linguistic devices that were simpler (e.g., words over phrases over clauses) (Scott and de Souza, 1990; Hovy, 1993).

There are sentences from the corpus which could be formulated more concisely. The doctors did very little editing of the discharge summaries. In this respect, the summaries are somewhat similar to speech. As a result, doctors prefer to use more flexible linguistic constructions, such as PPs, instead of producing the most concise sentences. Concepts such as "hypertension" and "diabetes" have both noun and adjective forms. Even though the noun form is longer (it is always used together with other words as in "patient with hypertension" or "patient who has hypertension"), the shorter adjective form ("hypertensive patient") did not appear in the corpus. In only one case was the adjective "obese" used instead of the PP "with obesity" to indicate medical conditions. Since many medical conditions have no adjective forms, such as "peptic ulcers," the speaker is more likely to use noun forms to group together all medical conditions. Furthermore, additional information can be attached to nouns but not to adjectives. In the noun form, the medical condition "diabetes" might be modified, as in "type 1 diabetes with extensive end organ damage" and "borderline diabetes." Such flexibility with nouns explains the popularity of its usage over adjectives in the corpus.

In summary, this analysis shows that a high level of aggregation is typical in the domain. Judging from the number of PPs in comparison to relative clauses used, clause aggregation using simpler syntactic constituents is preferred. Doctors generate summaries in real-time without examining all the information before them. As a result, they might not generate the most concise sentences. MAGIC, on the other hand, generates text off-line with all the conveying information available. This would allow MAGIC to generate more concise text by taking advantage of linguistic opportunities. In addition to medical corpus, our experience with financial corpus and PLANDOC system also indicates that employing clause aggregation in text generation system can result in a more concise and fluent text.

Chapter 4

Hypotactic Aggregation

At an abstract level, clause aggregation is a two-stage process. In the first stage, a content planner specifies rhetorical relations among the propositions to create a cohesive discourse. The content planner decides which propositions are more important to the communicative goal of the speaker (nucleus) and which are less important (satellite), and what the exact rhetorical relations are between them. In the second stage, a sentence planner, such as CASPER, applies language-specific operations to linked propositions and transforms them into sentences as part of a cohesive text. Many language specific transformations result in hypotactic or subordinate constructions. The current work focuses on combining propositions linked by ELABORATION relations, as shown in Example (26):

(26) Rhetorical Relation: ELABORATION
N: John likes Mary.
S: Mary came yesterday.
→ John likes Mary, who came yesterday.

The ELABORATION relation often occurs in generation systems which provide factual descriptions and properties of an entity or a situation such as MAGIC.

This chapter first describes the relationship between rhetorical relations and hypotactic constructions. Section 4.2 presents four issues related to hypotactic aggregation. These issues include identifying the range of linguistic realizations for these rhetorical relations; defining a set of hypotactic operators to realize rhetorical relations using the identified linguistic realizations; choosing a specific construction among the alternatives; and satisfying linguistic constraints during the transformation process. Section 4.3 provides an example from MAGIC to demonstrate the details of hypotactic aggregation operations. Section 4.4 is an in-depth study of a particular linguistic constraint related to hypotactic aggregation: the linear ordering among aggregated premodifiers. It describes several approaches to obtain the linear ordering between premodifiers which modify the same noun. Producing the expected ordering among such premodifiers help make the aggregated expressions fluent. Section 4.5 summarizes the chapter and describes several possible extensions of current work in hypotactic aggregation.

4.1 The Relationship between Nucleus-Satellite Rhetorical Relations and Hypotactic Constructions

Early work in discourse structure has noted that there is no straightforward one-to-one mapping between surface forms and the rhetorical relations that link the clauses (Ballard, Conrad, and Longacre, 1971; Longacre, 1983; Grimes, 1975). Mann and Thompson (1988) and Matthiessen and Thompson (1988) pointed out that nuclearity in text structure is a plausible communicative basis for hypotactic clause combining. Matthiessen and Thompson (1988) proposed the following hypothesis about a fundamental analogy between clause combination and the rhetorical organization of a text:

> Clause combining in grammar has evolved as a grammaticalization of the rhetorical units in discourse defined by rhetorical relations. (p. 301)

They supported their claim by showing that nucleus-satellite relations are often grammatically coded as hypotaxis, though not always. Of the 18 short texts in their database, the frequency of the usage of nucleus-satellite units and LIST[1] units are shown in Table 4.1. Since the ratio for the mapping of nucleus-satellite relations to

	Hypotactic	Paratactic
Nucleus-Satellite	45 (92%)	4 (8%)
Multi-Nucleus (LIST)	3 (11%)	24 (89%)

Table 4.1: Coding correlations between the type of rhetorical relations and grammaticalization (Matthiessen and Thompson, 1988, p. 308)

[1]The general term used in Matthiessen and Thompson (1988) for a multi-nucleus relation.

hypotactic constructions and to paratactic constructions is roughly 9:1, Matthiessen and Thompson's hypothesis accounts for about 90% of the data. The analysis shows that hypotactic constructions are effective linguistic means to communicate nucleus-satellite rhetorical relations explicitly.

Nucleus-satellite		Multi-nucleus
CIRCUMSTANCE	VOLITIONAL CAUSE	SEQUENCE
SOLUTIONHOOD	NON-VOLITIONAL CAUSE	CONTRAST
ELABORATION	VOLITIONAL RESULT	JOINT
BACKGROUND	NON-VOLITIONAL RESULT	COMPARISON
ENABLEMENT	PURPOSE	DISJUNCTION
MOTIVATION	INTERPRETATION	
EVIDENCE	EVALUATION	
JUSTIFY	RESTATEMENT	
ANTITHESIS	SUMMARY	
CONCESSION	OTHERWISE	
CONDITION		

Table 4.2: Rhetorical relations proposed in Mann and Thompson (1988)

The set of rhetorical relations proposed in Mann and Thompson (1988) is listed in Table 4.2. As Mann and Thompson noted, this is not an exhaustive list. Various researchers have added other relations to adapt RST for a new genre or for their specific applications. Readers interested in the variety of rhetorical relations should consult Hovy (1990a) and Knott (1996). Many rhetorical relations can be marked using cue phrases, as illustrated by CONCESSION in (27a) and EVIDENCE in (27b):

(27) a. Rhetorical Relation: CONCESSION
N: He was fine.
S: He just had an accident.
→ Although he had an accident, he was fine.

b. Rhetorical Relation: EVIDENCE
N: My car is not British.
S: My car is a Renault.
→ My car is not British because it is a Renault.

In Example (27a) and (27b), the unaggregated propositions seem to describe unrelated facts. By explicitly conveying the rhetorical relations using cue phrases, the aggregated sentences connect facts together to create cohesion. In addition to cohesion, applying hypotactic aggregation operations based on ELABORATION relations can also produce more concise text:

(28) Rhetorical Relation: ELABORATION
N: My car is not British.
S: My car is expensive.
→ My expensive car is not British.

In this example, there is a reduction of 33% based on the number of words used. Such a benefit increases as more clauses are combined into the same sentence. The hypotactic operators analyzed in the current work include transforming a propositions into one of the following modifying constructions: adjective phrase, apposition, preposition phrase (PP), reduced relative clause (ReducedRCl), and relative clause (RCl). A better understanding of these aggregation operators will allow text generation systems to generated text that is more concise and cohesive.

4.2 Issues

As mentioned earlier, the content planner specifies which propositions are linked by rhetorical relations. For CASPER to signal these rhetorical relations, the range of linguistic realizations to realize them must first be identified.

4.2.1 Identifying Linguistic Realizations

The first step in identifying linguistic constructions is to collect a corpus annotated with rhetorical relations. Unfortunately, the process to annotate a corpus with rhetorical relations is not straightforward since the mapping between textual marks and rhetorical relations is not one-to-one but many-to-many. Although many rhetorical relations are explicitly realized, some of them are implicit in a text. The problem is also complicated by the ambiguity of the textual markers. Some textual markers are *strong* because they explicitly identify a rhetorical relation between two propositions, such as "because," while other cue markers which are particularly *weak*, such as "and," which can be used to realize ELABORATION, JUSTIFY or SEQUENCE.

Rhetorical relations can be marked textually by either using lexical or phrasal *cue phrases*, such as "but" and "in order to," or using syntactic constructions. Vander Linden and Martin (1995) specifically studied various possible realizations of the PURPOSE relation in instructional text:

(29) Rhetorical Relation: PURPOSE
N: Lift the cover.
S: Install battery.

These realizations are shown in Table 4.3. In addition to cue phrases, rhetorical

	%	Examples
To-Infinitive	59.6%	To install battery, lift the cover.
For-Nominalization	7.5%	Lift the cover for battery installation.
For-Gerund	2.5%	Lift the cover for installing battery.
For-Goal-Metonymy	5.0%	For battery installation, lift the cover.
By-Purpose	10.0%	Install battery by lifting the cover.
Adjoined-Purpose	3.3%	Install Battery. Lift the cover first.
So-That-Purpose	8.4%	Lift cover so that battery can be installed.
Others	3.3%	

Table 4.3: Realizations for PURPOSE relation identified in Vander Linden and Martin (1995)

relations can also be realized by syntactic constructions, which is how ELABORATION relations are often realized textually. These constructions include adjective phrases, appositions, prepositional phrases (PP), reduced relative clauses (ReducedRCl), and relative clauses (RCl). Mann and Thompson (1988) described specific relations which are realized as ELABORATION: set/member, abstract/instance, whole/part, process/step, object/attribute, and generalization/specific. In the current study, these relations are considered too restrictive for classifying a relation such as ELABORATION. CASPER focuses on combining propositions linked by the ELABORATION relation, in which the satellite propositions simply present additional details about an entity in the nucleus propositions. This entails that the linked propositions share a common entity, and resembles the focus-based move aspect of the ELABORATION relation, proposed recently in Moser and Moore (1995) and Knott (2000).

4.2.2 Defining Hypotactic Operators

Before focusing on hypotactic operators for realizing the ELABORATION relation, the hypotactic operators for realizing nucleus-satellite relations will first be discussed in general. When expressing rhetorical relations using cue phrases, if the propositions do not share any entities in common such as in Example (30), the operator can simply join the clauses together and insert the cue phrase.

(30) Rhetorical relation: NON-VOLITIONAL CAUSE
N: John is cold.
S: The window is open.
→ Because the window is open, John is cold.

But in many cases, the linked propositions do share common entities, as in (31a) and (31b).

(31) a. Rhetorical Relation: PURPOSE
N: John stopped hunger.
S: John ate an apple.
→ To stop hunger, John ate an apple.

b. Rhetorical Relation: EVIDENCE
N: John was hungry.
S: John did not eat dinner.
→ John was hungry because he did not eat dinner.
→ Because John did not eat dinner, he was hungry.

In such cases, the internals of the linked propositions might undergo modifications to minimize repetition, such as the infinitive transformation in Example (31a), and the use of pronoun "he" in Example (31b). To ensure fluency, either a referring expression generation module chooses different expressions for the recurring shared entities or some transformations are applied to remove the recurring entities. In Example (31a), the infinitive transformation removed the recurring entity in the subject position of the satellite proposition; in Example (31b), the second reference to "John" in the aggregated sentence was pronominalized. Syntactic constraints, such as C-Command and long-distance dependency, sometimes play a role in these operations. This thesis does not provide an in-depth study of hypotactic aggregation operators which use cue phrases to realize rhetorical relations. Instead, the focus is on realizing the ELABORATION relation, the most common nucleus-satellite rhetorical relation used in the MAGIC system. It is often realized textually using syntactic constructions.

The MAGIC system provides status information to healthcare providers which is descriptive in nature. It uses ELABORATION relations often to connect propositions. Table 4.4 lists six syntactic constructions used to express an ELABORATION relation. The current analysis made some assumptions about the types

	verbosity	M-direction†	examples
RCl	short	before	an apple which weighs 3 ounces
ReducedRCl	shorter	before	an apple weighing 3 ounces
PP	shorter	before	an apple in the basket
apposition	shortest	before	an apple, a small fruit
prenominalization	shortest	after	a 3-ounce apple
adjective	shortest	after	a dark red apple

Table 4.4: Syntactic constructions for realizing Elaboration relations. M-direction† stands for "modifying direction."

of hypotactic constructions addressed. First, the current study does not distinguish between embedding and hypotactic constructions, as did Cheng, Mellish, and O'Donnell (1997), because in Casper, a satellite proposition can be transformed into a non-clausal constituent in the nucleus clause, as exemplified by adjective and PP transformations. Second, the current work does not provide separate treatment for either restrictive or non-restrictive modifiers, as shown in Example (32):

(32) a. `Restrictive`: The <u>famous</u> man is an analyst.

b. `Non-restrictive`: The man is a <u>famous</u> analyst.

The specification of restrictive nucleus-satellite relations is performed by the referring expression module as a part of referential aggregation. The referential aggregation module determines which propositions uniquely identify an entity to the users, attaches them to the entity, and marks the satellite proposition as a restrictive modifier. Except for adding commas for non-restrictive postmodifier constructions, Casper performs the same syntactic operations on both restrictive and non-restrictive hypotactic aggregations. Currently, the complexity of the satellite propositions is restricted to simple clauses to avoid complex issues with extraction constraints.

Of the six syntactic constructions for the Elaboration relation listed in Table 4.4, the relative clause (RCl) attachment uses minimal lexical information in its operation. Like all other syntactic constructions for realizing an Elaboration relation, the relative clause attachment operator requires that the nucleus and the satellite propositions share a common entity. If the shared entity is in the ARG1 position of the satellite proposition, a simple relative pronoun replaces it, as in Example (33):

(33) N: John likes Mary.
S: Mary is an analyst.
→ John likes Mary who is an analyst.

If the shared entity is in the ARG2 of the satellite proposition, a passive transformation is applied first to move the position of the shared constituent to the front of the clause at the surface level before substituting a relative pronoun for it.

(34) N: John likes Mary.
S: A car hit Mary.
→ John likes Mary who was hit by a car.

In the process of transforming the satellite proposition into a relative clause, there is no need to access the lexicon to guarantee that the result of the clause-combining process is expressible. Similar to the RCl operator, the reduced relative clause (ReducedRCl) operator also does not use lexical information. When connecting two propositions using ReducedRCl, the satellite clauses being transformed contain present or past progressive tenses as in Example (35); the phrases "who/which is/are/was/were" are deleted at the surface level.

(35) N: Ms. Jones is a patient.
S_2: The patient is undergoing heart surgery.
→ Ms. Jones is a patient <u>undergoing heart surgery</u>.

One possible explanation for the existence of such a deletion is that the surface forms "who/which is/are/was/were" are very common so that even without conveying them, the hearer can infer their existence after realizing something is missing at the surface level. Similar deletions also apply to apposition constructions, i.e., "Bill, <u>the President</u>, sneezed."

Using relative clauses to connect propositions increases cohesion, but it does not necessarily make text more concise since substitutions of nouns with relative pronouns might result in the same number of words in both non-aggregated and aggregated propositions. Reduction of text using hypotactic operators mainly comes from the other four hypotactic operators: apposition, adjective, prenominalization, and PP constructions. In contrast to RCl and ReducedRCl, these transformations use lexical information to ensure the expressibility of the combined linguistic structure. The apposition operator requires that the predicate of the satellite proposition be "`C-IS-IDENTIFIED-AS`,"[2] and that the concept in ARG2 be realizable as a noun before the operator can be applied:

[2] "`C-`" stands for "`CONCEPT-`." It is used to distinguish a concept from the lexical forms realizing such a concept.

(36) N: Bill Clinton sneezed.
S: Bill Clinton is the President.
→ Bill Clinton, the President, sneezed.

Although apposition is a syntactically paratactic construction, the operation is similar to other hypotactic operators and thus categorized as such. Similarly, the adjective and prenominalization operator requires that the predicate of the satellite proposition be "c-has-attribute." In the case of an adjective operator, the concept in ARG2 must be realizable using an adjective before the adjective operator can be applied:

(37) N: The man is an analyst.
S: The man is happy.
→ The man is a happy analyst.

In the case of a prenominal operator, the concept in ARG2 must be realizable, as a prenominal modifier. Such restrictions are coded in a lexicon and verified by the prenominalization operator before the transformation takes place.

Prepositional phrase attachment is a powerful but ambiguous aggregation operator. Many semantic relations (i.e., (38a) and (38b)) can be mapped to the same preposition (i.e., "of" in Example (38c)).

(38) a. a book which is written by Stephen King.

b. a book which describes Stephen King.

c. → a book <u>of</u> Stephen King.

Such ambiguity is not addressed in the current work because doing so would require taking into account pragmatics and common sense knowledge which are difficult to capture in computers. The PP operator in Casper is specific to the predicate and to the attribute of the predicate arguments.

(39) N: Ms. Jones is a patient.
S: The patient's doctor is Dr. Smith.
→ Ms. Jones is a patient <u>of Dr. Smith</u>.

In Example (39), an *of*-genitive (Quirk et al., 1985) operator was used. It requires that the POSSESSOR in the ARG1 of the satellite proposition contain the shared entity, "the patient," and that the predicate be "C-HAS-ATTRIBUTE" while the concept in ARG2 be realizable as a noun, i.e. "Dr. Smith." The proposition of the

```
((pred ((pred c-has-attribute) (type EVENT)
        (tense present)))
 (arg1 ((pred c-doctor)       (type THING)
        (mod ((pred c-patient) (type THING)
              (modify-type POSSESSOR)
              (entity-id ID1)))))
 (arg2 ((pred c-name) (type THING)
        (last-name "Smith"))))
```

Figure 4.1: Semantic representation for "The patient's doctor is Smith."

satellite proposition is shown in Figure 4.1. This particular operator can be also applied to advisor/advisee and boss/employee relations.

Usually the attachment of a modifier to the entity being modified is straightforward. But when the entity being modified is also a referring expression, special care needs to be taken to minimize unintended restrictive reading. For example, if the nucleus proposition is "`C-IS-INSTANCE`" as in Example (40), the adjective is attached to ARG2 of the "`C-IS-INSTANCE`" relation instead of ARG1.

(40) N: John is a patient.
S: John is handsome.
→ John is a handsome patient.
→ *Handsome John is a patient.

Except for apposition, the same attachment rule applies to other hypotactic operators as well. Attaching an apposition to the subject of a nucleus proposition does not create a restrictive reading: "John, a carpenter, likes pineapple."

4.2.3 Choosing a Hypotactic Operator

A rhetorical relation can be either realized or not realized in a text. Scott and de Souza (1990) suggested that every rhetorical relation should be realized because explicitly conveying the rhetorical relations seems to improve the comprehension of the text. Although this is true for many rhetorical relations, other researchers (Rösner and Stede, 1992; Knott, 1996) have identified that certain relations, such as Background, Elaboration, and Circumstance, are often not realized by using either syntax forms or cue phrases. They suggested that propositional content and contextual information were sufficient for the readers to infer the intended rhetorical relations. Once a decision to realize a rhetorical relation is made, a generation system must choose a specific realization if there are alternatives. Elhadad and McKeown (1990) focused on cue selection (e.g., '`but`', '`since`', '`because`', '`although`') based on pragmatic constraints. Vander Linden (1995) tried to identify the context under which a particular linguistic form is used to realize PURPOSE relation with some success.

Many issues, such as the relative order of nucleus and satellite clauses, were explored in Moser and Moore (1995) and Oberlander and Moore (1999). Rösner and Stede (1992) suggested that for certain relations such as ELABORATION, BACKGROUND, and CONDITION, the order is almost fixed while others are flexible, but there is a strong preference for one order, i.e., ENABLEMENT and SOLUTIONHOOD. They noted that a few relations, such as PURPOSE, CONTRAST, MOTIVATION, and PRECONDITION, order the nucleus and satellite propositions freely. Mann and Thompson (1988) observed that a text rewritten to conform to canonical order often improves it while the opposite is true of converting canonical order to non-canonical.

Since multiple hypotactic constructions can be used to realize an ELABORATION relation, CASPER needs to decide which one to use. Scott and de Souza (1990) proposed a heuristic in which simple syntactic constructions are preferred over more complex ones to express embedding (ELABORATION) relations. In Section 3.4, the corpus analysis of a human-written corpus in a medical domain also supported this heuristic. Simple constructions, such as PPs, are preferred over RCls to express medical conditions (35 occurrences versus 4). As many other researchers have noted, this heuristic rule produces more concise text because simpler constructions use fewer words. CASPER also prefers to use adjectives over PPs, PPs over ReducedRCls, and ReducedRCls over RCls. But this preference might be affected by various linguistic constraints, as the next section describes.

4.2.4 Satisfying Linguistic Constraints

From the description of the hypotactic operators for realizing ELABORATION relations, it is clear that transforming clauses into modifying constructions and attaching them to constituents in a nucleus proposition is not an unconstrained process. These operators must satisfy various linguistic constraints which facilitate communication of intended meaning. Three major types of constraints have been identified and are discussed below: lexical constraints, syntactic constraints, and linear ordering constraints.

4.2.4.1 Lexical Constraints

Except for ReducedRCl and RCl attachments, transforming a proposition into an adjective, an apposition, or a PP requires that the ARG2 of the satellite proposition be of a specific syntactic type (an adjective, a noun, or a PP, respectively). This lexical check takes care of the expressibility of the modifier with its noun, but it does not guarantee that the combination of the modifier and the noun it modifies conveys the intended meaning. For example, "a beautiful dancer" can mean either "a dancer who is beautiful" or "a dancer who dances beautifully." Example (41) further demonstrates the potential pitfall of not taking account of the interactions between a modifier and the noun it modifies:

(41) N: John is a runner.
S: John is fast.
→ ?John is a fast runner.

The aggregated sentence has a meaning which does not exist before the nucleus and satellite propositions are combined; thus, ideally, a generation system should consider such ambiguity before a specific hypotactic operator is chosen. Pustejovsky (1991) proposed Qualia Structure to capture such information in a lexicon to allow a computational system to detect similar ambiguity. By taking account of the potential interactions between a modifier and the noun it modifies, the system can ensure that the lexical choices made during both the transformation and the attachment process convey the intended meaning. Because the lexicon used in Casper currently does not support the capability to detect such lexical ambiguity, Casper assumes that the interactions between the modifier and its head are minimal.

4.2.4.2 Scope of the Modifiers

The attachment of a modifier to the noun it modifies is a simple task when there is only one modifier. Of the six hypotactic constructions in Table 4.4, only prenominalization and adjective constructions precede the noun they modify while the other four follow the noun. When multiple modifiers modify the same noun, attachment decisions become more complicated. This section addresses the problem of making the scoping of modifiers obvious to readers. Two issues are identified: first, deciding the ordering of postmodifiers, and second, avoiding scope ambiguity from interactions between modifier scope and conjunction.

When different hypotactic constructions are used to modify the same noun, the modifiers need to be in a certain linear order to avoid disfluencies and incorrect scope readings by readers. For premodifiers and apposition, identifying the noun that is modified is usually straightforward. But for multiple postmodifier constructions containing other noun phrases, a modifier might be mistaken as modifying an entity in another modifying construction instead of modifying the noun which is farther away, as in Example (42):

(42) N: Ms. Jones is a patient.
S_1: The patient's doctor is Dr. Smith.
S_2: The patient is undergoing heart surgery.
→ Ms. Jones is a patient <u>of Dr. Smith</u> <u>undergoing heart surgery</u>.

In Example (42), the reduced relative clause could modify either "a patient" or "Dr. Smith," but to the users of the healthcare application MAGIC, the reading that "undergoing surgery" modifies "Dr. Smith" is very remote. Because preventing such ambiguous expressions requires common-sense knowledge and domain-specific pragmatics, it is not treated systematically in the current system.

To avoid scope ambiguity resulting from interactions between scope of modifiers and conjunctions, CASPER modifies the linear ordering of the conjoined constituents to ensure that the scope of the modifier is obvious. If the first constituent in a conjunction contains a modifier in the premodifier position while later constituents have no modifier, the conjoined constituents will be switched so that the scope of the modifier is clear, as in Example (43):

(43) <u>Old men and women</u> should board the ship first.
→ <u>Women and old men</u> should board the ship first.

Similarly, CASPER will move the last conjoined constituent with a postmodifier to the beginning of the conjoined construction to clarify the scope of the postmodifier.

4.2.4.3 Linear Ordering between Aggregated Modifiers

Random ordering of multiple postmodifiers might create sentences that are difficult to read or understand. For example, since a PP usually modifies the noun it directly follows, the sentences in Example (44), with a PP not directly following the noun "a man," seem awkward.

(44) a. N: John met a man.
S_1: The man had a mustache.
S_2: The man was wearing a suit.
S_3: The man drank heavily.

b. → John met a man with a mustache wearing a suit who drank heavily.

c. → ?John met a man with a mustache who drank heavily wearing a suit.

d. → ?John met a man wearing a suit with a mustache who drank heavily.

e. → ?John met a man wearing a suit who drank heavily with a mustache.

f. → ?John met a man who drank heavily with a mustache wearing a suit.

g. → ?John met a man who drank heavily wearing a suit with a mustache.

From sentences (44b-g), the preferred ordering seems to be ReducedRCl before RCl. Upon inspecting (44b), it seems that the relative pronoun "who," which signals the relative clause at the end of the sentence, is modifying "a man" instead of "a suit." In (44c), the ReducedRCl does not have a relative pronoun to signal the ELABORATION relation. It is possible to interpret sentence (44c) as having a CONCURRENT relation between "the man drank heavily" and "wearing a suit." The other orderings of postmodifiers (44d-44g) are not as fluent as sentence (44b), which orders postmodifiers as PP-ReducedRCl-RCl. To produce sentences containing much information, CASPER allows the generation of sentences with constituents modified by both a ReducedRCl and a RCl. The generation of a constituent with multiple ReducedRCls or multiple relative clauses are disallowed because they sound strange, as demonstrated by Example (45):

(45) a. → ?John met a man <u>w</u>ho drank heavily <u>w</u>ho was wearing a suit.

b. → ?John met a man <u>w</u>earing a suit <u>d</u>rank heavily.

To convey multiple ELABORATION relations fluently, a generation system can emulate the linear order used by human-writers by obtaining the ordering information from a corpus. However, establishing the scope of postmodifiers and disambiguating postmodifier attachments is beyond the capability of state-of-the-art parsing systems. Obtaining a corpus for learning the linear ordering between postmodifiers requires tremendous effort in manual annotation. The only known large corpus with manual annotation of modifier attachments is the Penn TreeBank (Marcus, Santorini, and Marcinkiewicz, 1993). Unfortunately, since the corpus is genre-specific (financial domain), the results obtained from such analysis may not be applicable to different domains. On the other hand, the situation is not as dire for premodifiers. Because identifying the noun being modified by multiple premodifiers from a corpus is much simpler, obtaining the ordering information for premodifiers is

```
1a. The patient has name - Jones.
1b. The patient has gender - female.
1c. The patient has age - 80 year.
1d. The patient has hypertension.
1e. The patient has diabetes.
1f. The patient's doctor has name - Smith.
1g. The patient is undergoing CABG.
```

Figure 4.2: Input propositions for "Ms. Jones is an 80 year old hypertensive diabetic female patient of Doctor Smith undergoing CABG."

possible. Section 4.4 presents a corpus-based approach to obtain linear orderings between premodifiers and incorporate such information to improve the fluency of generated text.

4.3 A Detailed Example in MAGIC

An example from MAGIC is used to demonstrate how hypotactic operators work. The surface forms of the propositions from the content planner are shown in Figure 4.2. In addition to the propositions, the content planner also indicates that the first proposition, (*1a*), is the nucleus proposition and contains the focus entity "the patient." Since the entity in focus is assumed to be given and should appear as early as possible to provide a context, the proposition (*1a*) is transformed from "The patient has name - Jones" into the semantic representation for "Jones is a patient." In addition to switching the entities in ARG1 and ARG2, the PRED of the proposition is changed from `C-HAS-ATTRIBUTE` to `C-IS-INSTANCE`. Each proposition is represented similarly to the one shown earlier in Figure 4.1. The concept `C-HAS-ATTRIBUTE` indicates that the entity in ARG1 has the attribute stored in ARG2. Depending on the lexical properties of the attribute in ARG2, the proposition *1e* can be realized as "the patient has diabetes$_{\texttt{noun}}$" or "the patient is diabetic$_{\texttt{adj}}$."

To transform a proposition into an adjective, a proposition must satisfy the following two preconditions. First, the slot PRED of the proposition being transformed must be `C-HAS-ATTRIBUTE` (the patient *has* age - 80 years). The other requirement is that the ARG2 of the proposition (*age - 80 years*) can be mapped to an adjective, as permitted in the lexicon. Using a similar procedure, propositions (*1b*), (*1c*), (*1d*), (*1e*) can all be transformed into adjectives and attached to proposition (*1a*), resulting in "Jones is an 80-year-old hypertensive diabetic female patient." Two interesting items can be noted here. First, because the PRED of the nucleus proposition is `C-IS-INSTANCE`, the transformed modifiers (age, gender, etc.) are attached to the ARG2 slot of the dominant proposition ("a patient") instead of ARG1 ("Jones"). Second, the sequential order of the modifiers is not yet determined at this stage. The goal of CASPER is to produce a concise linguistic representation for a set of propositions and to guarantee at least one way to express the aggregated linguistic structure in later generation modules. To guarantee expressibility (Meteer, 1991b), CASPER looks ahead into the lexicon, but does not make detailed lexical decisions for efficiency reasons. The exact lexical and syntactic decisions, including the ordering between modifiers, are made later in the lexical chooser.

Consider aggregating another proposition: "the patient has peptic ulcers." This proposition cannot be transformed into an adjective because there is no adjective form for `C-PEPTIC-ULCER` in the lexicon. A proposition can be transformed into a PP with a general preposition "with" if the PRED of the proposition is `C-HAS-ATTRIBUTE` and the concept in its ARG2 can be mapped into a noun phrase. If the PP operator is applied, the combined sentence can be realized as "Jones is an 80 year old hypertensive diabetic female patient with peptic ulcers." CASPER currently uses an ontology which identifies that `C-PEPTIC-ULCER`, `C-HYPERTENSION`, and `C-DIABETES` are all medical disorders and groups them together for cohesion. Since all these medical conditions can be mapped to nouns but not to adjectives, they will all be realized as PPs: "Jones is an 80-year-old female patient with hypertension, diabetes and peptic ulcers."

In (*1f*) in Figure 4.2, "The patient's doctor has name - Smith," is transformed into a PP ("of Smith") using an *of*-genitive operator, and proposition (*1g*) is combined as a reduced relative clause. The result of the hypotactic operators is a linguistic structure for "Jones is an 80-year-old hypertensive diabetic female patient of Smith undergoing CABG."

4.4 A Study of Linear Ordering: Aggregated Premodifiers

The rest of this chapter focuses on studying a specific constraint in hypotactic aggregation: the linear ordering of aggregated premodifiers modifying the same noun. Sequential ordering among premodifiers affects the fluency of text, e.g., "large foreign financial firms" or "zero-coupon global bonds" are desirable, while "foreign large financial firms" or "global zero-coupon bonds" sound odd. The difficulties in specifying a consistent ordering of adjectives have already been noted by linguists (Whorf, 1956; Vendler, 1968). During the process of generating complex sentences by combining multiple clauses, situations arise where multiple adjectives or nouns modify the same noun. The text generation system must order these modifiers in a similar way as domain experts use them to ensure text fluency. For example, the description of the age of a patient precedes his ethnicity and gender in medical domain as in "a 50-year-old white female patient." Yet, general lexicons such as WordNet (Miller et al., 1990) and COMLEX (Grishman, Macleod, and Meyers, 1994) do not store such information.

The current work presents automated techniques for addressing this problem of determining the preferred ordering between two premodifiers A and B. Our methods rely on and generalize empirical evidence obtained from large corpora, and are evaluated objectively on such corpora. They are informed and motivated by our practical need for ordering multiple premodifiers in the MAGIC system (Dalal et al., 1996). MAGIC utilizes co-ordinated text, speech, and graphics to convey information about a patient's status after coronary bypass surgery; it generates concise but complex descriptions that frequently involve four or more premodifiers in the same noun phrase.

To demonstrate that a significant portion of noun phrases have multiple premodifiers, all the noun phrases (NPs, excluding pronouns) were extracted from a two-million-word corpus of medical discharge summaries and a 1.5-million-word Wall Street Journal (WSJ) corpus (see Section 4.4.3 for a more detailed description of the corpora). In the medical corpus, out of 612,718 NPs, 12% have multiple premodifiers and 6% contain solely multiple adjectival premodifiers. In the WSJ corpus, the percentages are a little lower: 8% and 2%, respectively. These percentages imply that one in ten NPs contains multiple premodifiers while one in 25 contains just multiple adjectives.

Traditionally, linguists study the premodifier ordering problem using a *class-based* approach. Based on a corpus, they propose various semantic classes, such as color, size, or nationality, and specify a sequential order among the classes. However, it is not always clear how to map premodifiers to these classes, especially in domain-specific applications. This justifies the exploration of empirical, corpus-based alternatives, where the ordering between A and B is determined either directly from prior evidence in the corpus or indirectly from other words whose relative order to A and B has already been established. The corpus-based approach lacks the ontological knowledge used by linguists, but uses a much larger amount of direct evidence, provides answers for many more premodifier orderings, and is portable to different domains.

In the next section, prior linguistic research on this topic is described. Sections 4.4.2 and 4.4.3 describe the methodology and corpus used in this analysis, while the results of these experiments are presented in Section 4.4.4. In Section 4.4.5, the ordering results are incorporated into a general text generation system.

4.4.1 Related Work

The order of adjectives (and, by analogy, nominal premodifiers) seems to be beyond grammar; it is influenced by factors such as polarity (Malkiel, 1959), scope, and collocational restrictions (Bache, 1978). Linguists (Goyvaerts, 1968; Vendler, 1968; Quirk and Greenbaum, 1973; Bache, 1978; Dixon, 1982) have performed manual analyses of (small) corpora and pointed out various tendencies such as underived adjectives often precede derived adjectives and shorter modifiers precede longer ones. Given the difficulty of adequately describing all factors that influence the order of premodifiers, most earlier work is based on placing premodifiers into broad semantic classes, and specifying an order among these classes. More than ten classes have been proposed, with some further broken down into subclasses. Although not all of these studies agree on the details, they demonstrate a fairly rigid regularity in the ordering of adjectives. Goyvaerts (1968) proposed the order `quality` $\prec$ `size/length/shape` $\prec$ `old/new/young` $\prec$ `color` $\prec$ `nationality` $\prec$ `style` $\prec$ `gerund` $\prec$ `denominal` (p. 27);[3] Quirk and Greenbaum (1973) the order `general` $\prec$ `age` $\prec$ `color` $\prec$ `participle` $\prec$ `provenance` $\prec$ `noun` $\prec$ `denominal` (p. 404); and Dixon (1982) the order `value` $\prec$ `dimension` $\prec$ `physical property` $\prec$ `speed` $\prec$ `human propensity` $\prec$ `age` $\prec$ `color` (p. 24).

Researchers have also looked at adjective ordering across languages (Dixon, 1982; Frawley, 1992). Frawley (1992), for example, observed that English, German, Hungarian, Polish, Turkish, Hindi, Persian, Indonesian, and Basque all order value before size and both of those before color. As with most manual analyses, the corpora used in these analyses are relatively small compared with modern corpora-based studies. Furthermore, different criteria were used to arrive at the classes. To illustrate, the adjective "beautiful" can be classified into at least two different classes because the phrase "beautiful dancer" can be transformed from either the

[3] Where $A \prec B$ stands for "A precedes B."

phrase "dancer who is beautiful" or "dancer who dances beautifully."

Several deep semantic features have been proposed to explain regularity among the positional behavior of adjectives. Teyssier (1968) first proposed that adjectival functions, i.e., identification, characterization, and classification, affect adjective order. Martin (1970) carried out psycholinguistic studies of adjective ordering. Frawley (1992) extended the work by Kamp (1975) and proposed that intensional modifiers precede extensional ones. However, while these studies offer insights into the complex phenomenon of adjective ordering, they cannot be directly mapped to a computational procedure.

On the other hand, recent computational work on sentence planning (Bateman et al., 1998; Shaw, 1998b) indicates that generation research has progressed to a point where hard problems such as ellipsis, conjunctions, and ordering of paradigmatically-related constituents are addressed. Computational corpus studies related to adjectives were performed by Justeson and Katz (1991) and Hatzivassiloglou and McKeown (1993; 1995), but none was directly on the ordering problem. Knight and Hatzivassiloglou (1995) and Langkilde and Knight (1998) have proposed models for incorporating statistical information into a text generation system, an approach that is similar to our way of using the evidence obtained from the corpus in our actual generator.

In addition to using direct evidence and transitive closure to identify ordering among aggregated adjectives in the current study, Malouf (2000) proposed and evaluated two additional approaches. One is a memory-based learning method using morphological forms of the adjectives; the other is based on positional probabilities based on an independence assumption which gave quite good results. Malouf noted that the errors made by the two approaches do not completely overlap; thus, he combined the two approaches and achieved 91.85% accuracy. The results reported by Malouf were in line with the results published in Shaw and Hatzivassiloglou (1999). In addition to the difference in Malouf's approach and the approach described in this chapter, the corpus used for Malouf's analysis has different characteristics. The 100-million-word British National Corpus used in Malouf's analysis is much larger than the one used in the current analysis, but it is not as domain-specific as the corpora used by Shaw and Hatzivassiloglou (1999).

4.4.2 Methodology

This section discusses how to obtain premodifier sequences from the corpus for analysis as well as the three approaches used to establish ordering relationships:

direct corpus evidence, transitive closure, and clustering analysis. The result of the analysis is embodied in a function, $compute\_order(A, B)$, which returns the sequential ordering between two premodifiers, word A and word B.

To identify orderings among premodifiers, premodifier sequences are extracted from simplex NPs. A simplex NP is a maximal noun phrase that includes premodifiers such as determiners and possessives, but not post-nominal constituents such as prepositional phrases or relative clauses. A part-of-speech tagger (Brill, 1992) and a finite-state grammar was used to extract simplex NPs. The extracted noun phrases start with an optional determiner (DT) or possessive pronoun (PRP$), followed by a sequence of cardinal numbers (CDs), adjectives (JJs), and nouns (NNs), and then end with a noun. The extracted NPs include cardinal numbers to capture the ordering of numerical information such as age and amounts. Gerunds (tagged as VBG) or past participles (tagged as VBN), such as "heated" in "heated debate," are considered to be adjectives if the word preceding them is a determiner, possessive pronoun, or adjective, thus separating adjectival and verbal forms that are conflated by the tagger. A morphology module transforms plural nouns and comparative and superlative adjectives into their base forms to ensure maximization of frequency counts. There is a regular expression filter which removes obvious concatenations of simplex NPs, such as "takeover bid last week" and "Tylenol 40 milligrams."

After simplex NPs are extracted, sequences of premodifiers are obtained by dropping determiners, genitives, cardinal numbers, and nouns. Subsequent analysis operates on the resulting premodifier sequences, and involves three stages: direct evidence, transitive closure, and clustering. Each stage is described in more detail in the following subsections.

4.4.2.1 Direct Evidence

This analysis proceeds on the hypothesis that the relative order of two premodifiers is fixed and independent of context. Given two premodifiers A and B, there are three possible underlying orderings, and the current system should strive to find which is true in this particular case: either A comes before B, B comes before A, or the order between A and B is truly unimportant. The first stage relies on frequency data collected from a training corpus to predict the order of adjective and noun premodifiers in an unseen test corpus.

To collect direct evidence on the order of premodifiers, all the premodifiers are extracted from the corpus, as described in the previous subsection. The premodifier sequences are first transformed into *ordered pairs*. For example, the phrase "well-known traditional brand-name drug" has three ordered pairs: "well-known $\prec$ traditional," "well-known $\prec$ brand-name," and "traditional $\prec$ brand-name." A phrase with n premodifiers will have $\binom{n}{2}$ ordered pairs. From these ordered pairs, we construct a $w \times w$ matrix *Count*, where w is the number of distinct modifiers. The cell $[A,\ B]$ in this matrix represents the number of occurrences of the pair "A $\prec$ B" (in that order) in the corpus.

Assuming a preferred ordering between premodifiers A and B, one of the cells $Count[A, B]$ and $Count[B, A]$ should be much larger than the other, at least if the corpus becomes arbitrarily large. However, given a corpus of a fixed size, there will be many cases where the frequency counts will be small. This data-sparseness problem is exacerbated by the inevitable occurrence of errors during the data extraction process, which will introduce some spurious pairs (and orderings) of premodifiers. Therefore, probabilistic reasoning was applied to determine when the data were strong enough to decide that $A \prec B$ or $B \prec A$. Under the null hypothesis that the two-premodifiers order is arbitrary, the number of times one of them is seen follows the binomial distribution with parameter $p = 0.5$. The probability of seeing the actually observed number of cases with $A \prec B$, say m, among n pairs involving A and B is

$$\sum_{k=m}^{n} \binom{n}{k} \cdot p^k \cdot (1-p)^{(n-k)} \tag{4.1}$$

which, for the special case $p = 0.5$, becomes

$$\sum_{k=m}^{n} \binom{n}{k} \cdot 0.5^k \cdot 0.5^{(n-k)} = \sum_{k=m}^{n} \binom{n}{k} \cdot 0.5^n \tag{4.2}$$

If this probability is low, the null hypothesis is rejected and it can be concluded that A indeed precedes (or follows, as indicated by the relative frequencies) B.

4.4.2.2 Transitivity

As mentioned before, sparse data is a serious problem in this analysis. For example, the matrix of frequencies for adjectives in the training corpus from the medical domain is 99.8% empty—only 9,106 entries in the 2,232 × 2,232 matrix contain non-zero values. To compensate for this problem, the transitive properties between ordered pairs was explored by computing the transitive closure of the ordering relation. Utilizing transitivity information corresponds to making the inference that $A \prec C$ follows from $A \prec B$ and $B \prec C$, even if there is no direct evidence for the pair (A, C), but only provided that there is no contradictory evidence to this inference. This approach filled 15% (WSJ) to 30% (medical corpus) of the entries in the matrix.

To compute the transitive closure of the order relation, the underlying data were mapped to special cases of *commutative semi-rings*. Each word was represented as a node of a graph, while arcs between nodes corresponded to ordering relationships and were labeled with elements from the chosen semi-ring. This formalism can be used for a variety of problems, using appropriate definitions of the two binary operators ((Pereira and Riley, 1997) called them *collection* and *extension*) that operate on the semi-ring's elements.

For example, the all-pairs shortest-paths problem in graph theory can be formulated in a *min-plus* semi-ring over real numbers, with the operators *min* for collection and + for extension. Similarly, finding the transitive closure of a binary relation can be formulated in a *max-min* semi-ring or a *or-and* semi-ring over the set $\{0, 1\}$. Once the proper operators are chosen, the generic Floyd-Warshall algorithm (Aho, Hopcroft, and Ullman, 1974) can be used to solve the corresponding problem without modifications.

Three semi-rings appropriate to the present problem were explored. First, the statistical decision procedure of the previous subsection were applied and each pair of premodifiers were assigned either 0 (if missing information about their preferred ordering) or 1 (if sufficient evidence). Then the *or-and* semi-ring was used over the {0,1} set; in the transitive closure, the ordering $A \prec B$ would be present if at least one path connecting A and B via ordered pairs existed. Note that it is possible for both $A \prec B$ and $B \prec A$ to be present in the transitive closure.

This model involved conversions of the corpus evidence for each pair into hard decisions on whether one word in the pair preceded the other. To avoid such early commitments, a second, refined model for transitive closure was used where the arc from A to B was labeled with the probability that A indeed preceded B. The natural extension of the ({0, 1}, *or*, *and*) semi-ring, when the set of labels was replaced with the interval [0, 1], was then ([0, 1], *max*, *min*). The estimated probability was that A preceded B as one minus the probability of reaching that conclusion in error, according to the statistical test of the previous subsection, i.e., one minus the sum specified in equation (4.2). Similar results were obtained with this estimator and with the maximal likelihood estimator (the ratio of the number of times A appeared before B to the total number of pairs involving A and B).

Finally, a third model was considered which explored an alternative to transitive closure. Rather than treating the number attached to each arc as a probability, the number was treated as a *cost*, the cost of erroneously assuming that the corresponding ordering exists. Each edge (A, B) was assigned the negative logarithm of the probability that A preceded B; probabilities were estimated as in the previous paragraph. Then the problem became identical to the all-pairs shortest-path problem in graph theory; the corresponding semi-ring was $((0, +\infty), min, +)$. Logarithms were used to address computational precision issues stemming from the multiplication of small probabilities, and the logarithms were negated in order to cast the problem as a minimization task (i.e., finding the path in the graph which minimizes the total sum of negative log probabilities, and therefore maximizes the product of the original probabilities).

4.4.2.3 Clustering

As noted earlier, earlier linguistic work on the ordering problem placed words into semantic classes and generalized the task from ordering between specific words to ordering the corresponding classes. A similar, but evidence-based approach was developed to resolve the linear ordering of premodifier pairs that neither direct evidence nor transitivity can resolve. An *order similarity* measure can be computed between any two premodifiers, to reflect whether the two words share the same pattern of relative order with other premodifiers for which there is sufficient evidence. For each pair of premodifiers A and B, every other premodifier is examined in the corpus, X; if both $A \prec X$ and $B \prec X$, or both $A \succ X$ and $B \succ X$, one point is added to the similarity score between A and B. If, on the other hand, $A \prec X$ and $B \succ X$, or $A \succ X$ and $B \prec X$, one point is subtracted. X does not contribute to the similarity score if there is no sufficient prior evidence for the relative order of X and A or of X and B. This procedure closely parallels non-parametric distributional tests such as Kendall's τ (Kendall, 1938).

The similarity scores were then converted into dissimilarities and fed into a non-hierarchical clustering algorithm (Späth, 1985), which separated the premodifiers in groups. This was achieved by minimizing an *objective function*, defined as the sum of within-group dissimilarities over all groups. In this manner, premodifiers that were closely similar in terms of sharing the same relative order with other premodifiers were placed in the same group.

Once classes of premodifiers were induced, every pair of classes were examined to decide which preceded the other. For two classes C_1 and C_2, all pairs of premodifiers (x, y) were extracted with $x \in C_1$ and $y \in C_2$. With evidence (either direct or through transitivity) that $x \prec y$, one point was added in favor of $C_1 \prec C_2$; similarly, one point was subtracted if $x \succ y$. After all such pairs were considered, it was possible to predict the relative order between words in the two clusters which were not seen together previously. This method leads to (weak) predictions for any pair (A, B) of words, except if (a) both A and B are placed in the same cluster; (b) no ordered pairs (x, y) with one element in the class of A and one in the class of B are identified; or (c) the evidence for one class preceding the other is in the aggregate equally strong in both directions.

4.4.3 The Corpus

Two corpora were used for the analysis: hospital discharge summaries from 1991 to 1997 from the Columbia-Presbyterian Medical Center, and the January 1996 part of the Wall Street Journal corpus from the Penn TreeBank (Marcus, Santorini, and Marcinkiewicz, 1993). To facilitate comparisons across the two corpora, the analysis was intentionally limited to only one month of the WSJ corpus, so that approximately the same amount of data would be examined in each case. The text in each corpus was divided into a training part (2.3 million words for the medical corpus and 1.5 million words for the WSJ) and a test part (1.2 million words for the medical corpus and 1.6 million words for the WSJ).

All domain-specific markup was removed, and the text was processed by the MXTERMINATOR sentence boundary detector (Reynar and Ratnaparkhi, 1997) and Brill's (1992) part-of-speech tagger. Noun phrases and pairs of premodifiers were extracted from the tagged corpus according to the methods described in Section 4.4.2. From the medical corpus, 934,823 simplex NPs were retrieved, of which 115,411 had multiple premodifiers and 53,235 multiple adjectives only. The corresponding numbers for the WSJ corpus were 839,921 NPs, 68,153 NPs with multiple premodifiers, and 16,325 NPs with only multiple adjectives.

Two groups of premodifiers were analyzed separately: adjectives, and adjectives plus nouns modifying the noun. Although the techniques are identical in both cases, the division was motivated by the expectation that the task would be easier when modifiers were limited to adjectives, because nouns tend to be harder to match correctly with finite-state grammar and the input data are sparser for nouns.

4.4.4 Results

The three ordering algorithms proposed in this chapter were applied separately to the two corpora for both adjectives and adjectives plus nouns. For the first technique of directly using evidence from a separate training corpus, the *Count* matrix (see Section 4.4.2.1) was filled with the frequencies of each ordering for each pair of

Corpus	Test pairs	Direct evidence	Transitivity (max-min)	Transitivity (min-plus)
Medical/ adjectives	27,670	**92.67%** (88.20%–98.47%)	**89.60%** (94.94%–91.79%)	**94.93%** (97.20%–96.16%)
Financial/ adjectives	9,925	**75.41%** (53.85%–98.37%)	**79.92%** (72.76%–90.79%)	**80.77%** (76.36%–90.18%)
Medical/ adjectives and nouns	74,664	**88.79%** (80.38%–98.35%)	**87.69%** (90.86%–91.50%)	**90.67%** (91.90%–94.27%)
Financial/ adjectives and nouns	62,383	**65.93%** (35.76%–95.27%)	**69.61%** (56.63%–84.51%)	**71.04%** (62.48%–83.55%)

Table 4.5: Accuracy of direct-evidence and transitivity methods on different data strata of the test corpora. In each case, overall accuracy is listed first in bold, and then, in parentheses, the percentage of the test pairs for which the method has an opinion (rather than randomly assign a decision because of lack of evidence) and the accuracy of the method within that subset of test cases.

---

premodifiers using the training corpora. Then, a calculation was made to identify which of those pairs correspond to a true underlying order relation, i.e., passed the statistical test of Section 4.4.2.1 with the probability given by equation (4.2) less than or equal to 50%. Each *instance* of ordered premodifiers was then examined in the corresponding test corpus to count how many the direct evidence method could predict correctly. Note that if A and B occur sometimes as $A \prec B$ and sometimes as $B \prec A$, no prediction method can get all those instances correct. This evaluation approach, which lowered the apparent scores of the method, was chosen rather than force each pair in the test corpus into one unambiguous category ($A \prec B$, $B \prec A$, or arbitrary).

Under this evaluation method, stage one of the current system achieves for adjectives in the medical domain 98.47% correct decisions on pairs for which a determination of order could be made. Since 11.80% of the total pairs in the test corpus involved previously unseen combinations of adjectives and/or new adjectives, the overall accuracy was 92.67%. The corresponding accuracy on data for which predictions could be made and the overall accuracy were 98.35% and 88.79% for adjectives plus nouns in the medical domain; 98.37% and 75.41% for adjectives in the WSJ data; and 95.27% and 65.93% for adjectives plus nouns in the WSJ data. Note that the WSJ corpus was considerably more sparse, with 64.24% unseen combinations of adjective and noun premodifiers in the test part. Using lower thresholds in equation (4.2) resulted in a lower percentage of cases for which the system had an opinion but a higher accuracy for those decisions. For example, a threshold of 25% resulted in the ability to predict 83.72% of the test adjective pairs in the medical corpus, with 99.01% accuracy for these cases.

Subsequently, the transitivity approach was applied. The three semi-ring models were tested, as discussed in Section 4.4.2.2. Early experimentation indicated that the *or-and* model performed poorly; this can be attributed to the extensive propagation of decisions (once a decision in favor of the existence of an ordering relationship is made, it cannot be revised even in the presence of conflicting evidence). Therefore, results are reported below for the other two semi-ring models. Of those, the *min-plus* semi-ring achieved higher performance. That model offered additional predictions for 9.00% of adjective pairs and 11.52% of adjective-plus-noun pairs in the medical corpus, raising the overall accuracy of the predictions to 94.93% and 90.67%, respectively. The overall accuracy in the WSJ test data was 80.77% for adjectives and 71.04% for adjectives plus nouns. Table 4.5 summarizes the results of these two stages.

Finally, the third, clustering approach on each data stratum was applied. Due to data sparseness and computational complexity issues, the most frequent words in each set of premodifiers (adjectives or adjectives plus nouns) were clustered, only those that occurred at least 50 times in the training part of the corpus being analyzed were selected. Results were reported for the adjectives selected in this manner (472 frequent adjectives from the medical corpus and 307 adjectives from the WSJ corpus). For these words, the information collected by the first two stages of the system covered most pairs. Of the 111,176 (=472·471/2) possible pairs in the medical data, the direct evidence and transitivity stages made predictions for 105,335 (94.76%); the corresponding number for the WSJ data was 40,476 out of 46,971 possible pairs (86.17%).

The clustering technique made ordering predictions for a part of the remaining pairs—on average, depending on how many clusters were created, this method produced answers for 80% of the ordering cases that remained unanswered after the first two stages in the medical corpus, and for 54% of the unanswered cases in the WSJ corpus. Its accuracy on these predictions was 56% on the medical corpus, and slightly worse than the baseline 50% on the WSJ corpus; this latter, aberrant result was due to a single, very frequent pair, *chief executive*, in which *executive* was consistently mistagged as an adjective by the part-of-speech tagger.

Qualitative analysis of the third stage's output indicated that it identified many interesting relationships between premodifiers; for example, the pair of most similar premodifiers on the basis of positional information was *left* and *right*, which clearly fell into a class similar to the semantic classes manually constructed by linguists. Other sets of adjectives with strongly similar members included {*mild*, *severe*, *significant*} and {*cardiac*, *pulmonary*, *respiratory*}.

In these analyses, we have obtained better results for medical corpus over WSJ. We believe there are two possible explanations. One is that the amount of training and testing corpus for these domains is not the same. As stated in Section 4.4.3, the ratio of our training corpus to testing corpus in medical domain is roughly 2:1 while in WSJ, the ratio is roughly 1:1. This might cause some bias when we compare the two results. But more importantly, we believe that the better results in medical corpus is likely caused by using a more restrictive, or domain-specific, medical corpus in comparison with the corpus taken from WSJ. Because WSJ discusses much broader topics than discharge summaries, such as politics, personnel, technology, and financial issues, it is expected that there are more variety in the orderings of the premodifiers in such corpus and it is much harder to obtain a good coverage.

The empirical analysis was concluded by testing whether a separate model was needed to predicting adjective order in each different domain. We trained the first two stages of the present system on the medical corpus and tested them on the WSJ corpus, obtaining an overall prediction accuracy of 54% for adjectives and 52% for adjectives plus nouns. Similar results were obtained when we trained on the financial domain and tested on medical data (58% and 56%). These results were not much better than what would have been obtained by chance, and were clearly inferior to those reported in Table 4.5. Although the two corpora shared a large number of adjectives (1,438 out of 5,703 total adjectives in the medical corpus and 8,240 in the WSJ corpus), they shared only 2% to 5% of the adjective *pairs*. This empirical evidence indicated that adjectives were used differently in the two domains, and hence domain-specific probabilities must be estimated, thereby increasing the value of an automated procedure for the prediction task.

4.4.5 Using Ordered Premodifiers in Text Generation

Extracting sequential ordering information of premodifiers is an off-line process, the results of which can be easily incorporated into the overall generation architecture. The function $compute\_order(A, B)$ was integrated into the multimedia presentation system MAGIC (Dalal et al., 1996), in the medical domain and it resolved numerous premodifier ordering tasks correctly. Example cases where the statistical prediction module was helpful in producing a more fluent description in MAGIC included placing age information before ethnicity information and the latter before gender

(a) “`John is a diabetic male white 74-`
`year-old hypertensive patient with a red swollen mass in the left groin.`”

(b) “`John is a 74-year-old hypertensive diabetic white male patient with a swollen red mass in the left groin.`”

Figure 4.3: (a) Output of the generator without our ordering module, containing several errors. (b) Output of the generator with our ordering module.

information, as well as specific ordering preferences such as “thick” before “yellow” and “acute” before “severe.” MAGIC’s output was evaluated by medical doctors, who provided us with feedback on different components of the system, including the fluency of the generated text and its similarity to human-produced reports.

Lexicalization is inherently domain-dependent, so traditional lexica cannot be ported across domains without major modifications. Our approach, in contrast, was based on words extracted from a domain corpus and not on concepts; therefore, it can be easily applied to new domains. In our MAGIC system, aggregation operators, such as conjunction, ellipsis, and transformations of clauses to adjectival phrases and relative clauses, were performed to combine related clauses together and increase conciseness (Shaw, 1998a; Shaw, 1998b). We wrote a function, *reorder_premod(. . .)*, which is called after the aggregation operators, takes the whole lexicalized semantic representation, and reorders the premodifiers right before the linguistic realizer is invoked. Figure 4.3 shows the difference in the output produced by our generator with and without the ordering component.

4.5 Summary

After describing the association between nucleus-satellite rhetorical relations and hypotactic constructions, various issues relevant to synthesizing hypotactic constructions were identified and discussed. The current chapter focuses on realizing the ELABORATION relation by using syntactic devices such as adjectives, PPs, and relative clauses. An in-depth analysis of one specific issue—the linear ordering between multiple premodifiers—was performed and the result of the analysis was incorporated into MAGIC to improve the fluency of generated text. Three techniques for exploring prior corpus evidence in predicting the order of premodifiers within noun phrases were presented. The current methods expanded on observable data by inferring new relationships between premodifiers even for combinations of premodifiers that do not occur in the training corpus. We have empirically validated our approach, showing that we can predict order with more than 94% accuracy when enough corpus data are available. We have also implemented our procedure in a text generator, producing more fluent output sentences.

Despite the current successful effort in realizing the ELABORATION relation, there remain many open problems in hypotactic aggregation that need to be addressed. One pressing issue is developing an algorithm for extracting the shared entity in the transformation of the satellite proposition before attaching the transformed constituent to the nucleus proposition. Currently, the satellite proposition to be transformed must be simple, and the shared entity in the satellite proposition must be in either ARG1 or ARG2. Both of these constraints can be relaxed when a robust extraction algorithm is developed. Many grammatical formalisms have provided devices to model extraction (Bouma, Malouf, and Sag, in press), but they have not been implemented into generation systems. Other issues include determining sentence boundaries and preventing undesirable implicatures using lexical information similar to the approach taken by Pustejovsky (1991). In addition, taking account of collocation information (Smadja and McKeown, 1991) in the selection of modifiers can further enhance the fluency of the generated text.

Chapter 5

Coordinating Conjunctions

Coordination is a common linguistic construction in language usage. In both the medical and financial corpora that we studied,[1] one out of three sentences contains the word '`and`.' A *coordinating conjunction* is the linguistic construction which uses one of the *coordinators*, i.e., '`and`,' '`or`,' '`but`,' to link two or more linguistic units of equal syntactic status, such as a series of clauses, phrases, or words. The conjoined elements are referred to as *conjuncts* or *conjoins* (Crystal, 1997; Quirk et al., 1985). In this chapter, we focus mainly on the coordinator '`and`' because in descriptive domains, it is the most common of the coordinators. To build a text generation system that produces grammatical sentences with coordinated conjunctions, in addition to obeying pertinent linguistic constraints, we need to identify and model information necessary for a computational system to generate such constructions. By automating the process, we demonstrate our understanding of the phenomenon and identify the issues that are still open to further research.

The most significant contribution of the current work in the generation of coordinated conjunction constructions is our conjunction algorithm. It is similar to the deletion approach proposed by Van Oirsouw (1987) and unifies several related constructions: gapping, right-node-raising, non-constituent conjunction, and ellipsis. The details of these constructions will be discussed in Section 5.5. In addition to proposing a unified algorithm which handles various coordination constructions, the current work also points out the importance of surface ordering information and peripherality (a deletion target must be peripheral to its construction) in the process of synthesizing coordinating conjunctions. Conjunction involves all the major

[1] A description of the corpus used for this analysis is provided in Section 4.4.3. In our medical corpus, 96,985 out of 268,962 sentences (36%) contain the word '`and`.' In WSJ corpus, 50,003 out of 155,613 sentences (32%) contain the word.

modules in the content planner, sentence planner, and surface realizer. The content planner, based on pragmatic and discourse information, specifies which propositions are linked by rhetorical relations Sequence, Addition, and Non-Volitional Result. These rhetorical relations trigger the application of coordinating conjunction operators in the sentence planner. The sentence planner takes advantage of semantic information to identify recurring entities in the propositions being combined. The surface realizer uses the linear ordering information of the recurring constituents to correctly delete redundant expressions at the surface level to realize the coordination conjunction constructions.

Section 5.1 describes various benefits of employing coordinating conjunction constructions in a natural language generation system. Section 5.2 describes linguistic approaches to analyze coordinating conjunctions. Two orthogonal dimensions of coordination constructions are identified. Section 5.3 describes modifications to the Systemic Functional Grammar (SFG) (Halliday, 1994) representation to simplify our unified conjunction algorithm. We describe and illustrate the unified conjunction algorithm with an extensive example in Section 5.4. In Section 5.5, we discuss the details of the various linguistic constraints involved. In Section 5.6, potential ambiguities which might result from the coordinating conjunction operations are described and analyzed. Section 5.7 identifies the coverage of our proposed algorithm in two ways. First, difficult cases cited in the linguistic literature are used to demonstrate the wide coverage of our algorithm. Then, sample sentences from the medical and financial domains are analyzed to identify the coverage of our algorithm. Section 5.8 describes extensions to the proposed algorithm to handle some related syntactic constructions such as “respectively” and “each other.”

5.1 Motivation

Incorporating coordinating conjunction constructions into a generation system improves three aspects of synthesized text: naturalness, conciseness, and cohesion. Humans use coordinated conjunctions in their everyday communication process. In fact, the coordinating conjunction is one of the most common marked syntactic aggregation constructions found in medical discharge summaries. Without handling coordinating conjunctions, generating data in a medical database might result in the following sentences:

(46) The patient received one unit of packed red blood cells. The patient received six units of fresh frozen plasma. The patient received two units of cell savers.

Although the sentences report true facts in a domain and all of them are grammatical, readers of a text containing such sentences bunched together immediately notice awkwardness in its flow. Without coordinating conjunctions, a generation system violates one of Grice's (1975) conversational maxims, Maxim of Manner: Avoid obscurity of expression and ambiguity. Because humans often use coordinating conjunctions in normal communication, sentences without conjunctions might create conversational implicatures. The repetition of the surface expression "the patient" in (46) is likely to be interpreted as an abnormal emphasis or a request to readers to look more deeply into the situation than is warranted by the facts reported. A person is more likely to communicate the information using the following sentence:

(47) The patient received one unit of packed red blood cells, six units of fresh frozen plasma, and two units of cell savers.

Readers of the sentence in (47) will obtain the desired meaning without the undesirable side-effects of the sentences in (46). A more problematic example of a text consisting of unaggregated sentences involves the usage of indefinite pronouns:

(48) a. A patient is awake. A patient is hungry.

b. A patient is awake and hungry.

Without aggregation, as in (48a), the sentences usually imply that two different patients are involved. When the two references to "a patient" specify the same person, using coordinating conjunctions is important to communicate clearly the intended reading, as in (48b).

In addition to naturalness, texts using coordinating conjunction constructions also satisfy Grice's Maxim of Quantity: make your contribution as informative as is required, but not overly informative, such as excessive verbosity. The improvement of the conciseness of the text is a result of deleting repeated expressions in the original sentences. In Example (47), the recurring references to "the patient" and "received" were deleted from the unaggregated sentences in (46). As a result, because fewer words were generated using conjunctions, the time required for the system to convey the information to the listener is shortened. In a high-pressure, fast-paced environment such as an intensive care unit, being concise is a virtue.

Coordinating conjunctions also introduce cohesion in a text (Halliday and Hasan, 1976). Of the four constructions Halliday and Hassan listed as cohesive devices (reference, ellipsis and substitution, conjunction, and lexical cohesion), our conjunction algorithm handles both conjunction and a certain type of ellipsis. These two constructions relate together various entities in a discourse, thus making the text tightly integrated, functioning as a whole. To automatically produce texts that are more similar to texts produced by humans, a text generation system needs to incorporate coordinating conjunction constructions as one device which can make a text more natural, concise, and cohesive. By achieving this, we are one step closer to creating a more transparent and natural human-computer interface.

5.2 Background

The coordinating conjunction is one of the most studied linguistic phenomena (Ross, 1967; Tai, 1969; Dougherty, 1970; Gleitman, 1965; Gazdar, 1981; van Oirsouw, 1987; Steedman, 2000). Just like other well-known syntactic phenomena such as extraction and long-distance dependency, coordinating conjunctions must be handled correctly before a grammatical formalism is considered robust. Many grammatical formalisms describe how such constructions fit into their framework, including Lexical Functional Grammar (LFG) (Kaplan and Maxwell III, 1988; Kehler et al., 1999); Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994; Sag and Fodor, 1994; Sag and Wasow, 1999); Lexical Tree-Adjoining Grammar (LTAG) (Joshi and Schabes, 1990; Jorgensen and Abeille, 1992; Sarkar and Joshi, 1996; Sarkar, 1997); and Combinatory Categorial Grammar (CCG) (Steedman, 1985; Steedman, 1990; Steedman, 2000). Linguists have long used coordination to test constituency in a language. Chomsky (1957) stated that "the possibility of conjunction offers one of the best criteria for the initial determination of phrase-structure" (p. 36). Based on the principle that only identical categories can be conjoined, linguists can determine constituent boundaries and identify the basic categories in a language, as demonstrated by (49):

(49) a. John ate *an apple* and *an orange.* (= NP and NP)

b. John ate *in the morning* and *in the evening.* (= PP and PP)

c. * John ate *an apple* and *in the evening.* (= NP and PP)

d. * John ate *in the evening* and *an apple.* (= PP and NP)

In (49a) and (49b), both conjoined constituents are of the same syntactic categories. In (49c) and (49d), since the conjuncts are not of the same syntactic type, the conjoined expression is not grammatical.

Two orthogonal dimensions can be used to analyze coordinating conjunctions. One is based on the syntactic properties of the conjuncts — either the conjuncts are of the same basic syntactic category (also known as *constituent conjunction*), or the conjuncts are of different or non-basic syntactic types (also known as *non-constituent conjunction*). The other is based on intended readings of the conjoined expression — either distributive or collective. Both of these dimensions are pertinent to our conjunction algorithm. We will first look at the syntactic property of conjunctions. Linguists often distinguish between *simple coordination* or *ordinary coordination*, and *complex coordination* (Quirk et al., 1985; Radford, 1988) . Simple coordinations are coordinations of single grammatical constituents, such as clauses, predications (VPs), phrases, and words.

(50) John opened the door and Mary closed the window. (conjunction of clauses)
John opened the door and closed the window. (conjunction of predications)
John opened the red door and the left window. (conjunction of phrases)
John closed the door and window. (conjunction of words)

Complex coordinations are less common. The conjuncts in complex coordination are combinations of syntactic constituents rather than single constituents. They include non-constituent coordination, gapping, right-node-raising, and ellipsis, as shown in Example (51):

(51) a. **non-constituent coordination**: John ate <u>fish on Monday</u> and <u>rice on Tuesday</u>.

b. **gapping**: <u>John ate fish</u> and <u>Bill rice</u>.

c. **right-node-raising**: <u>John caught</u> and <u>Mary killed</u> the spider.

In (51a), the conjuncts have a combined syntactic type, "object + adverbial modifier." In a gapping coordination such as (51b), a medial constituent is deleted to make the constituents in the latter conjunct noncontiguous. In the right-node-raising (RNR) construction, an identical constituent at the end of the conjoined clauses (the object "the spider" in (51c)) is shared by the conjuncts. Although our analysis of coordinating conjunctions distinguishes between simple and complex conjunctions, there are some distinctions which will be clarified after the description of the algorithm in Section 5.4.

The second dimension by which to analyze conjunction constructions is based on whether the construction has a *distributive reading* or a *collective reading*.[2] In

[2] In Quirk et al. (1985), the terms *segregatory* and *combinatory* were used instead of *distributive* and *collective*, respectively.

distributive coordination, the coordination of smaller units is logically equivalent to coordination of clauses; for example, "John and Paul like Mary" is logically equivalent to "John likes Mary" and "Paul likes Mary." Sentences containing conjunctions with collective readings cannot be analyzed as separate clauses; for example, "Mary and Sue are sisters" is not equivalent to "Mary is a sister" and "Sue is a sister."

(52) a. John and Mary sneezed.

b. John and Mary are a couple.

c. John and Mary bought furniture.

d. John ate apples and Mary oranges.

In looking at the sentences in (52), it is clear that (52a) and (52d) have only distributive readings and (52b) has only a collective reading, while (52c) can have either. Classifying coordinating conjunctions based on distributive and collective readings is useful in our analysis and implementation of the phenomenon.

The distinction between distributive and collective readings is closely related to the two linguistic approaches to analyze coordinating conjunctions: transformation rules and phrase-structure rules. In the analysis based on transformation rules, sentences with conjunctions are derived from two or more sentences which are identical except for the constituents to be coordinated. Chomsky (1957) provided an early transformational account of such a rule:

> If S_1 and S_2 are grammatical sentences, and S_1 differs from S_2 only in that X appears in S_1 where Y appears in S_2 (i.e., S_1 = ..X.. and S_2 = ..Y..), and X and Y are constituents of the same type in S_1 and S_2, respectively, then S_3 is a sentence, where S_3 is the result of replacing X by X + and + Y in S_1 (i.e., S_3 = ..X + "`and`" + Y..). (p. 37)

The transformation-based approach can be found in Tai (1969), Gleitman (1965), Ross (1970), and van Oirsouw (1987). Since transformation rules involve deletion of identical elements, it is also known as a deletion-based approach. The other approach, phrase-structure rules (PS-rule), was advocated by Dik (1968), Smith (1969), Dougherty (1970), and Gazdar (1981). According to PS-rule analysis, coordinating constructions are not derived from two or more sentences. Instead, a grammar containing rules like the following can adequately handle all coordinated structures:

$$\mathrm{XP} \Rightarrow \mathrm{XP}_1, \mathrm{XP}_2, ..., \text{ and } \mathrm{XP}_n, \text{ where XP = any syntactic category}$$

This rule claims that any number of constituents of the same category except determiner can be combined to form a conjoined constituent. For coordinations with a collective reading, PS-rule analysis provides a much more satisfying description than transformation rules. For example, sentences that have, or potentially have, collective readings, such as (52b) and (52c), can be derived using PS-rules to expand a NP into a conjunction: "John and Mary." Even in sentences with a distributive reading, such a derivation also seems to be adequate, as "John and Mary" in (52a). But, in (52d), while the conjunct to the left of the conjunctor '`and`' is an S (clause), the conjunct "Paul oranges" to the right of '`and`' is not even a valid syntactic constituent. Although the original PS-rule approach has been extended to handle RNR and maybe gapping conjunction through the use of SLASH category (Gazdar, 1981; Gazdar et al., 1985; Sag et al., 1985), these coordination constructions can be explained more simply using transformation rules. Section 5.6.1 describes a text generation architecture which incorporates both transformation rules and phrase-structure rules to provide an adequate description of the coordinating conjunction phenomenon.

5.3 Input Representation and Assumptions

Before presenting the details of our conjunction algorithm, we need to first describe the information used and the input representation. In Section 2.2, we motivated the use of Systemic Functional Grammar (SFG) for text generation. In SFG, grammatical description is organized around features appropriate for the expression of specific meanings. These functions, rather than the structural regularity of syntax, determine the organization of the grammar. SURGE, the surface realizer used in both MAGIC and PLANDOC, is based on SFG and Functional Unification Grammar (Kay, 1984). It uses feature structures as its underlying data structure to store linguistic information. In Section 2.2, we discussed removing non-essential features from the feature representation. For the purpose of synthesizing conjunction constructions, one such non-essential feature is the thematic roles for subjects and objects. In SFG, different processes have different names for their thematic roles (e.g., the MENTAL process has the role SENSER as agent, while the INTENSIVE process has the role IDENTIFIED). Using thematic roles in the representation for conjunction has a drawback: entity equivalence determination becomes complicated. One major task of generating coordination expressions is identifying identical elements in the propositions being combined. Identifying identical elements under various thematic roles requires first looking at the predicate (or "process" in

```
((pred ((pred c-lose) (type EVENT)
        (tense past)))
 (arg1 ((pred c-name) (type THING)
        (first-name ``John'')))
 (arg2 ((pred c-laptop) (type THING)
        (specific no)
        (mods ((pred c-expensive)
               (type ATTRIBUTE)))))
 (mods ((pred c-yesterday)
        (type TIME))))
```

Figure 5.1: Semantic representation for "John lost an expensive laptop yesterday."

SFG) in order to figure out which thematic roles should be checked for redundancy. To simplify the process, we decided to use the argument structure in LFG as the underlying representation instead of the rich thematic representation used in SFG. The thematic roles, which are important in linguistic processing (Levin, 1993), are delegated to and stored in the lexicon, not in the input.

In our representation, the roles for each event or state are PRED, ARG1, ARG2, ARG3, and MOD. The slot PRED stores the predicate, or the verb concept. Depending on the concept in PRED, ARG1, ARG2, and ARG3 can take on different thematic roles such as Actor, Beneficiary, and Goal, respectively in the example "John gave Mary a red book yesterday." The optional slot MODS stores modifiers of the PRED. It can have one or multiple circumstantial elements, including MANNER, PLACE, or TIME. Each argument slot can contain a MODS slot to store additional information such as POSSESSOR or ATTRIBUTE. An example of the semantic representation is provided in Figure 5.1.

In addition to the linguistic representation used for our conjunction algorithm, there are a few assumptions about the input that the clause aggregation module imposes on its input:

- Rhetorical relationships are specified by the content planner. With regard to coordinating conjunctions, the system is concerned with three rhetorical relations: ADDITION, SEQUENCE, and NON-VOLITIONAL RESULT. In Section 5.6.2, we briefly presented eight rhetorical relationships implied by coordination conjunctions (Quirk et al., 1985). CASPER only realizes a subset of them using '`and`.' The ADDITION relation connects the propositions resulting from a single query to the database which returns multiple answers, such as multiple drug giving events which occurred during surgery. When multiple propositions are linked by the ADDITION relation, the order between them can be altered by the aggregation operators based on pragmatic constraints. Propositions linked by SEQUENCE and NON-VOLITIONAL RESULT relations cannot be reordered by the sentence planner. They are specified in the content planner.

- Because of potential ambiguities resulting from clause aggregation, it is dangerous to combine propositions for the sole purpose of producing a concise text. Combining the sentences “John ate an apple” and “John died” into the sentence “John ate an apple and died” creates a causal relationship which might or might not be true in that particular situation. Detecting such undesirable implicature in general is beyond the current state of art and probably unsolvable for forseeable future. The sentence planner currently assumes that the content planner does not ask it to aggregate clauses that will result in undesirable implicatures. The issue related to undesirable implicatures will be further discussed in Section 5.6.2.

- The input must provide information so that the aggregation module can determine if two entities are the same or not. This is a semantic requirement. By knowing that “the car” refers either to the same car or to different ones in “John likes the car” and “Mary likes the car,” different combined sentences can result, such as “John and Mary like the same car” or “John and Mary like different cars.” Entity identity affects the aggregation results.

- The referring expression module has decided on the surface expressions that refer to the entities. In the sentence “Fidel Castro stood in an open car and the Cuban Prime Minister was enthusiastically welcomed,” both subjects refer to the same entity; however, because different referring expressions were selected, neither of them is deleted. If their surface expressions were the same, they would have been deleted. This example demonstrates that for the conjunction algorithm, certain referring expression decisions must be made before conjunction. On the other hand, other referring expression decisions can only be taken after clause aggregation operations have been carried out because combining operations might delete recurring entities at the surface level. With fewer entities to make referring expression decisions, clause aggregation, in some sense, also simplifies and influences referring expression decisions.

In the current implementation, CASPER will not expand a sentence with conjoined constituents into multiple clauses and reformulate the clauses. When a constituent comes in as a conjoined constituent, the system assumes there is a particular reason why the entities are conjoined in the conjoined constituent and will not break them apart, although additional entities can be added to such a constituent.

5.4 The Conjunction Algorithm

We have divided the algorithm into four steps. The first three steps take place in the sentence planner and the last step takes place in the surface realizer.

Step 1: Group propositions and order them according to their similarities while satisfying pragmatic and contextual constraints.

Step 2: Determine recurring elements in the ordered propositions being combined.

Step 3: Create a sentence boundary when the combined clause reaches *a-priori* thresholds.

Step 4: Decide which recurring elements are redundant and should be deleted.

In the following sections, we provide details on each step. Instead of using terms familiar to telecommunication engineers, as in McKeown, Kukich, and Shaw (1994)) to illustrate the algorithm, we use an imaginary employee report generation system for a human resource department in a supermarket.

5.4.1 Step 1: Group and Order Propositions

Grouping propositions together with similar elements is desirable because these elements are likely to be inferable and redundant at the surface level. There are many ways to group and order propositions based on similarities. For the propositions in Figure 5.2, the semantic representations have the following slots: PRED, ARG1, ARG2, MOD-PLACE, and MOD-TIME. To identify which slot has the most similarity among its elements, we calculate the number of distinct elements in each slot across the propositions, which we call NDE (number of distinct elements). For the purpose of generating concise text, it is beneficial to group propositions so that the resulting propositions have as many slots with NDE = 1 as possible. For the propositions shown in Figure 5.2, the NDEs of both PRED and ARG1 are 1 because all the actions are "re-stock" and all the agents are "Al"; the NDE for ARG2 is 4 because it contains 4 distinct elements: "milk," "coffee," "tea," and "bread"; similarly, the NDE of MOD-PLACE is 3 and the NDE of MOD-TIME is 2 ("on Monday" and "on Friday").

```
Al re-stocked milk in Aisle 5 on Monday.
Al re-stocked coffee in Aisle 2 on Monday.
Al re-stocked tea in Aisle 2 on Monday.
Al re-stocked bread in Aisle 3 on Friday.
```

Figure 5.2: A sample of input propositions in surface form.

```
Al re-stocked coffee in Aisle 2 on Monday.
Al re-stocked tea in Aisle 2 on Monday.
Al re-stocked milk in Aisle 5 on Monday.
Al re-stocked bread in Aisle 3 on Friday.
```

Figure 5.3: Propositions in surface form after Step 1.

The algorithm re-orders the propositions by sorting the elements in each slot using comparison operators which can determine that Monday comes before Friday, or Aisle 2 is before Aisle 3. Starting from the slots with the largest NDE to the lowest, the algorithm re-orders the propositions based on the elements of each particular slot. In this example, propositions will be ordered according to ARG2 first, followed by MOD-PLACE, MOD-TIME, ARG1, and PRED. The sorting process will put similar propositions adjacent to each other, as shown in Figure 5.3.

5.4.2 Step 2: Identify Recurring Elements

Based on the linear ordering computed in Step 1, a coordination operator combines only two propositions at any one time. The *aggregated proposition* is a semantic representation which represents the result of combining multiple input propositions. One task of the sentence planner is to find a way to combine the next proposition in the ordered propositions into the aggregated proposition. Step 2 is concerned with how many slots have distinct values and which slots they are. When multiple adjacent propositions have only one slot with distinct elements, these propositions are *one-distinct*. A special optimization can be carried out between the one-distinct propositions by conjoining their distinct elements into a coordinating structure, such as conjoined verbs, nouns or adjectives. McCawley (1981) described this phenomenon as *Conjunction Reduction*: "whereby conjoined clauses that differ only in one item can be replaced by a simple clause that involves conjoining that item." In our example, the first and second propositions are one-distinct at ARG2, and are combined into a semantic structure representing "Al re-stocked *coffee* and *tea* in Aisle 2 on Monday." If the third proposition were one-distinct at ARG2 with respect to the result proposition, the element "milk" in ARG2 of the third proposition would be similarly combined. In the example provided, it is not. As a result, the third proposition cannot be combined using only conjunctions within a syntactic constituent.

When the next proposition and the resulting proposition have more than one distinct slot, or when their one-distinct slot differs from the previous one-distinct slot, the two propositions are said to be *multiple-distinct*. Our approach in combining multiple-distinct propositions differs from the previous linguistic analysis. Instead of removing recurring entities right away based on transformations or substitutions, the current system generates *every* conjoined multiple-distinct proposition. During the generation process of each conjoined proposition, the recurring elements might be prevented from appearing at the surface level by preventing the realization component from generating any string for such redundant elements. Our multiple-distinct coordination produces what linguists describe as non-constituent conjunction, gapping, and ellipsis. Figure 5.4 shows the result of combining two propositions that will produce the sentence "Al re-stocked tea on Monday and milk on Friday." Some readers might notice that PRED and ARG1 in both propositions are marked as RECURRING, but only subsequent recurring elements are deleted at the surface level. The reason for this will be explained in Section 5.4.4.

5.4.3 Step 3: Determine Sentence Boundary

Unless combining the next proposition into the result proposition exceeds the predetermined parameters for the complexity of a sentence, the algorithm will continue to combine more propositions into the resulting proposition using one-distinct or multiple-distinct coordination. After examining a small sample of generated sentences and finding some of them too complex to read, the maximum number of propositions conjoined using multiple-distinct coordination, by default, is limited to two. In Figure 5.5, the first and second propositions are multiple-distinct in ARG2 and MOD-LOC slots. The third proposition is three-distinct, or multiple-distinct to the aggregated propositions of 1 and 2 in ARG2, MOD-LOC, and MOD-TIME. As a result, a sentence break is inserted and the third propositions will be generated as a separate sentence. In special cases where the same slots across mul-

```
((pred c-and) (type LIST)
 (elts
    ~(((pred ((pred "re-stock") (type EVENT)
              (tense past) (status RECURRING)))
       (arg1 ((pred "Al") (TYPE THING)
              (status RECURRING)))
       (arg2 ((pred "tea") (type THING)))
       (mods ((pred "on") (type TIME)
              (arg2 ((pred "Monday")
                     (type TIME-THING))))))
      ((pred ((pred "re-stock") (type EVENT)
              (tense past) (status RECURRING)))
       (arg1 ((pred "Al") (TYPE THING)
              (status RECURRING)))
       (arg2 ((pred "milk") (type THING)))
       (mods ((pred "on") (type TIME)
              (arg2 ((pred "Friday")
                     (type TIME-THING)))))))))
```

Figure 5.4: The simplified semantic representation for "Al re-stocked tea on Monday and milk on Friday." Note: $\sim() \equiv$ a list.

tiple propositions are multiple-distinct, the pre-determined limit is ignored. By taking advantage of parallel structures, these propositions can be combined using multiple-distinct procedures without making the coordinate structure more difficult to understand. For example, the sentence "John took aspirin on Monday, penicillin on Tuesday, and Tylenol on Wednesday" is long but understandable. Similarly, conjoining a long list of three-distinct propositions also produces understandable sentences: "John played tennis on Monday, drove to school on Tuesday, and won the lottery on Wednesday." These constraints allow CASPER to produce complex sentences that contain a lot of information, yet are easy to understand.

5.4.4 Step 4: Delete Redundant Elements

Step 4 handles non-constituent conjunction, gapping, and ellipsis, some of the most difficult phenomena to handle in syntax. In the previous steps, elements that occur more than once among the propositions are marked as RECURRING, but actual deletion decisions have not been made because CASPER lacks the necessary information to do so. The importance of the surface linear ordering can be demonstrated by the following example. In the sentence "On Monday, Al re-stocked coffee and [on Monday,] [Al] removed rotten milk," the elements in MOD-TIME *delete forward*

```
Al re-stocked coffee and tea in Aisle 2 on Monday.
Al re-stocked milk in Aisle 5 on Monday.

—

Al re-stocked bread in Aisle 3 on Friday.
```

Figure 5.5: Propositions in surface form after Step 3.

(i.e., the subsequent occurrence of the identical constituent disappears). When MOD-TIME elements are realized at the end of the clause, the same elements in MOD-TIME *delete backward* (i.e., the antecedent occurrence of the identical constituent disappears): "Al re-stocked coffee [on Monday,] and [Al] removed rotten milk on Monday."

Our algorithm uses the linear ordering of the recurring element for making deletion decisions. In general, if a slot is realized at the front or medial position of a clause, the recurring elements in that slot delete forward. In the first example, MOD-TIME is realized as the front adverbial while ARG1, "Al," appears in the middle of the clause, so that elements in both slots delete forward. On the other hand, if a slot is realized at the end position of a clause, the recurring elements in such a slot delete backward, as the MOD-TIME indicates in the second example. The extended directionality constraint also applies to conjoined premodifiers and postmodifiers as well, as demonstrated by "in Aisle 3 and [in Aisle] 4," and "at 3 [PM] and [at] 9 PM."

Using the algorithm just described, the result of the supermarket example is concise and easily understandable: "Al re-stocked coffee and tea in Aisle 2 and milk in Aisle 5 on Monday. Al re-stocked bread in Aisle 3 on Friday." Further discourse processing will replace the second "Al" with the pronoun "he," and the adverbial "also" may be inserted as well.

Casper has been used in an upgraded version of PLANDoc (McKeown, Kukich, and Shaw, 1994), a robust, deployed system which generates reports that justify costs to management in the telecommunications domain. Some of the reports generated by PLANDoc are presented in Figure 5.6. In the figure, "CSA" is a location; "Q1" stands for first quarter; "multiplexor" and "working-pair transfer" are telecommunications equipment. The first example is a typical simple proposition in the domain, which consists of PRED, ARG1, ARG2, MOD-PLACE, and MOD-

TIME. The second example shows one-distinct coordination at MOD-PLACE, where the second CSA has been deleted. The third example demonstrates coordination of two propositions with multiple-distinct coordination in MOD-PLACE and MOD-TIME. The fourth example shows multiple items: ARG1 became plural in the first proposition because multiple placements occurred, as indicated by the simple conjunction in MOD-PLACE; the gapping of the PRED "was projected" in the second clause, based on multiple-distinct coordination. The last example demonstrates the deletion of MOD-PLACE in the second proposition because of its location at the front of the clause at the surface level; therefore, MOD-PLACE deletes forward.

It is interesting to note that coordination of VPs, traditionally considered a simple conjunction, is formulated by the multiple-distinct operation. For example, in our analysis, the sentence "John *finished his work* and [John] *went home*" is the result of combining two clauses with deleted ARG1, "John," in the second clause. Earlier in Section 5.2, we mentioned that although there is a parallel between the simple and complex conjunctions proposed by linguists (Quirk et al., 1985; Radford, 1988), and the one-distinct and multiple-distinct conjunctions in our analysis, these two analyses are not exactly the same.

5.4.5 Nested Conjunction

Nested conjunction refers to a coordinated conjunction construction that contains another coordinated conjunction as a constituent. Sentences with nested conjunctions are derived from applying multiple-distinct operators to clauses which have already undergone coordination using one-distinct operators. For example, in (4), the algorithm first collapses three propositions by combining different locations, "CSA 1160," "CSA 1335," and "CSA 1338," using the one-distinct operator, and then applies the multiple-distinct operator to combine the propositions regarding 150mb_mux and 200mb_mux multiplexor placements. Nested coordination is common in everyday language usage: "John likes apples and pears and Mary hates oranges."

5.4.6 Recursive Conjunction

The coordinating conjunction algorithm can apply again to coordinated sentences to produce *recursive* conjunctions. For example, given the four propositions,

- An unaggregated proposition in PLANDoc:

  (1) The Base Plan called for one new fiber activation at CSA 1061 in 1995 Q2.

- An aggregated sentence formed by using one-distinct operator:

  (2) New 150mb_mux multiplexor placements were projected at CSA 1160 and 1335 in 1995 Q2.

- Sentences formed by using multiple-distinct operators:

  (3) New 150mb_mux multiplexors were placed at CSA 1178 in 1995 Q4 and at CSA 1835 in 1997 Q1.

  (4) New 150mb_mux multiplexor placements were projected at CSA 1160, 1335 and 1338 and one new 200mb_mux multiplexor placement at CSA 1913 in 1995 Q2.

  (5) At CSA 2113, the Base Plan called for 32 working-pair transfers in 1997 Q1 and four working-pair transfers in 1997 Q2 and Q3.

Figure 5.6: Text generated by Casper in PLANDoc.

(53) John likes donuts.
John likes ice cream.
Mary likes donuts.
Mary likes ice cream.

applying one-distinct to propositions in (53) results in either (54a) or (54b):

(54) a. John likes donuts and ice cream.
Mary likes donuts and ice cream.

b. John and Mary like donuts.
John and Mary like ice cream.

c. → John and Mary like donuts and ice cream.

Reapplying the coordinating conjunction algorithm, the system can generate sentence (54c).

Sometimes a sentence like (54c) can also be derived from (55):

(55) John likes donuts.
Mary likes ice cream.

Such a transformation violates the premise that clause aggregation operators should preserve their meaning in the original clauses. In this case, a reader of sentence (54c) will answer "yes" when asked "Does John like ice cream?" But a reader of the sentences in (55) would not be able to answer such a question. This type of transformation is used in everyday life where precision is often not an issue, but in medical and financial domains, such an operator is not precise enough and is not used in the system.

5.4.7 Complexity of the Algorithm

The algorithm presented in Section 5.4 is not optimal with respect to conciseness. We intentionally presented a simplified version for exposition purposes. In this section, the complexity of the algorithm and various related issues are addressed. Let us assume that t represents the number of thematic roles in a proposition, and n represents the number of propositions. In Step 1, sorting propositions according to similarity, the complexity of the algorithm as described is $O(tn \log n)$—sorting the entities inside each thematic role takes $n \log n$ and this operation occurs t times. Based on the input data for PLANDoc, the range of t is between 2 to 7 (2 for sentences with intransitive verbs, and 7 for sentences with many PP attachments), and the usual value is between 3 and 5. The value of n usually is between 1 to 4, with values almost always less than 10. The largest value of n we have encountered in our applications is greater than 30 in PLANDoc.[3] Since the potential number of propositions n might be much larger than t, we will simply ignore t in our complexity analysis. The complexity of Step 1 is $O(n \log n)$. In Step 2, identifying recurring elements, the algorithm simply has to look at adjacent propositions to determine if the entities in their respective thematic roles are recurring. This happens for each role. As before, assuming t is small, this is still an $O(n)$ operation. Step 3, determining the sentence boundary, can be carried out as a side effect of Step 2, so its complexity is also $O(n)$. Step 4, deleting redundant elements, requires looking at entities across propositions, and is $O(n)$ too. Overall, the complexity of the algorithm is $O(n \log n)$.

By using the sequential ordering of propositions as the basis of checking whether propositions are one-distinct or multiple-distinct, the proposed algorithm produces concise text, but it is not optimally concise. It is possible that combining the last proposition in a multiple-distinct conjunction with later propositions might produce more concise text, as in (56). Example (56b) contains 21 words while Example (56a), based on the algorithm in Section 5.4, uses 25 words.

(56) a. Al re-stocked coffee in Aisle 4 on Monday and Tuesday and milk in Aisle 5 on Wednesday.
He rearranged cereals in Aisle 5 on Wednesday.

b. Al re-stocked coffee in Aisle 4 on Monday and Tuesday. He restocked milk and rearranged cereals in Aisle 5 on Wednesday.

At an abstract level, the algorithm described in Section 5.4 finds a local maximum instead of a global one. As a result, situations occur where the generated text is not the most concise one possible. Although more systematic approaches can be used for producing a more concise text, the ones we developed are much more expensive in terms of complexity and execution time. One such approach is to identify a global maximum by partitioning the proposition in a systematic manner and then testing all of the partitions to see which one gives the best desired results. Given a set of n propositions, the number of possible ways to partition the propositions is equivalent to Stirling numbers of the second kind (Graham, Knuth, and Patashnik, 1991, p. 244) (Knuth, 1973). For example, when $n = 3$,

[3] Whether a conjunction with 30 or more conjuncts is desirable or not is an application-specific issue. A grammar does not limit the number of conjuncts in a conjunction.

there are 5 different partitions: {(1),(2),(3)}, {(1,2),(3)}, {(1,3),(2)}, {(1),(2,3)}, {(123)}. Let function $f(n)$ be the number of possible partitions for n elements. For $n = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, f(n) = \{1, 1, 2, 5, 15, 52, 203, 877, 4140, 21147\}$. It is exponential with respect to n. In MAGIC and PLANDOC, the number of clauses being considered for aggregation seldom exceeds 10. But for n= 9, $f(9)$ is 21,147, a rather large number of possibilities to go through. After the partitioning, the propositions in each partition with $n \geq 2$ are permuted to obtain all the possible orderings of the propositions inside a partition. Each permutation is called a proposition-set. Since the permutation of n propositions is $n!$ proposition-sets, for $n = 9$, there are 362,880 proposition-sets. The partitioning process will allow the system to explore various sentence boundaries and avoids the pitfall of producing a non-optimal solution. But there is a trade-off. The complexity of going through every permutation in every partition is clearly an exponential process—too costly during run-time.

Although one of the benefits of clause aggregation is conciseness, it is not the main goal. There are pragmatic constraints in the domain which override the conciseness criteria for clause aggregation. For example, in PLANDOC, chronological ordering of the events overrides the preference for conciseness. Although efficiency will be an issue only when the number of propositions becomes large, such situations do happen. When they do, users might have to wait for 25 minutes instead of the usual 3 minutes. Before the report is generated, it is difficult for users to tell if the system is running or if the system has failed. To minimize user complaints, both MAGIC and PLANDOC used an $O(n \log n)$ algorithm, as described in Section 5.4, instead of a more exhaustive, but more concise approach.

5.5 The Constraints

Section 5.4 described the major steps of the algorithm without going into great detail. In this section, we will focus on the linguistic constraints which play a role in the design of the algorithm. As demonstrated by the following examples, coordinating conjunctions are not solely a surface-based phenomenon:

(57) John ate quickly. John ate a grilled sandwich.
$\rightarrow$ *John ate quickly and a grilled sandwich.

(58) The baker baked. The bread baked.
$\rightarrow$ *The baker and the bread baked.

(59) John took a nap. John took an umbrella.
→ *John took a nap and an umbrella.

To synthesize grammatical sentences with coordinating conjunctions, a generation system not only needs surface information, but also semantic and lexical information. In the next few sections, the constraints pertaining to the conjunction algorithms will be discussed: linear ordering of the conjuncts, constituent identity, directional constraints, and peripherality. The linear ordering of the conjuncts has an impact on Step 1; the constituent identity pertains to the operations in Step 2; directional constraint and peripherality are important in Step 4.

5.5.1 Linear Ordering of Conjoined Constituents (affects Step 1)

The linear ordering of the propositions determines which propositions are adjacent; thus, such a decision has a drastic impact on the structure of coordinating conjunctions. Linguists (e.g., (Lakoff, 1971)) have noted that the used of the asymmetric "`and`" imposes an order on the sentences that it conjoins. One such use of "`and`" is equivalent to "`and then`" in either the temporal or causal sense in which the order of the sentences cannot be reordered. Inappropriate ordering of conjoined constituents is not ungrammatical or semantically incorrect, but it can cause dysfluencies, misconceptions, and false implications. Section 5.4 uses the conciseness criteria to order the propositions in a way that produces concise sentences. In practice, domain constraints and pragmatic preferences often override the conciseness criteria. Currently, Casper linearizes the propositions aiming to produce concise sentences while satisfying pragmatic preferences.

In the development of Magic and PLANDoc, several strategies were identified to linearize the propositions: chronology, distance, urgency, importance, and alphabet ordering. In her writing manual, Hacker (1994) listed other strategies for arranging information at the paragraph level: climax, complexity, familiarity, and audience appeal. Her strategies seem applicable to conjoined constituents at the sentence level also. Some of them, such as climax, familiarity, and audience appeal, are difficult to implement in a computational system since they are difficult to model. These strategies are summarized below:

1.	**chronology**	from oldest to newest, or vice versa.
2.	**distance**	from closest to listener/speaker to farthest from listener/speaker, or vice versa.
3.	**urgency**	from most urgent to least urgent.
4.	**importance**	from most important to least important.
5.	**alphabet ordering**	from A-Z, alphabetically.
6.	**complexity**	from simple to complex.
7.	**climax**	from most to least dramatic.
8.	**familiarity**	from most familiar to least familiar.
9.	**audience appeal**	from "safe" ideas to those that may challenge the audience's views.

Chronological order is appropriate for telling a story or describing a process, such as cooking recipes. In both MAGIC and PLANDOC, it is the dominant feature in linearizing propositions. In these domains, temporal information is recorded with each event, such as the time when drugs are given or the expected dates for placing various telephone equipment. Even with numerous time formats, it is generally simple to implement a numerical procedure to determine which events occur earlier and to use such information to order temporal constituents. In contrast, the second strategy, to compare or to sort constituents according to distance information, is much more difficult to implement. For example, the information that New Jersey is much closer to New York City in comparison to California is not obvious to a computer system. Although the use of Global Positioning System (GPS) might become more popular, neither MAGIC nor PLANDOC have it available to estimate distance information about installation sites or hospital rooms. Instead, an alphabet ordering strategy was used to linearize constituents which convey locational information. Alphabetical ordering has a particular benefit. In PLANDOC, if a reader of a report is interested in a particular type of equipment, he/she can quickly identify whether it is present in a conjoined expression using a strategy similar to binary search by taking advantage of the fact that equipment is alphabetically ordered. Alphabetical ordering is the default ordering used in CASPER when no other information is available. Other features for ordering propositions, such as urgency and importance, are domain-specific and must be specified by the application. For example, in sports the number of gold medals a team won is likely to be mentioned before the silver and bronze ones.

It is surprising that the complexity strategy, from simple to complex, is also applicable in the linear ordering of conjoined constituents. Goodall (1987) stated that there is a general preference for the final conjunct to be equal or greater in "heaviness" to previous ones (p. 42). It would be more natural to list the simplest constituent first, as in (60a) and (61a), than the more complex ones, as shown in (60b) and (61b).

(60) a. John likes to eat the fish he catches in San Pablo Bay and toast.

b. John likes to eat toast and the fish he catches in San Pablo Bay.

(61) a. Mary talks to John and Tom, who is standing next to the elevator.

b. Mary talks to Tom, who is standing next to the elevator, and John.

A check in CASPER re-orders constituents in such cases if the reordering does not violate other pragmatic constraints.

In the implementation of CASPER, the code that compares and sorts constituents occupies a significant portion of the overall code. Two functions, "`equal`" and "`smaller`," are needed. The "`equal`" function allows the system to determine if two propositions are multiple-distinct or one-distinct; the "`smaller`" allows the system to reorder constituents. Many special cases need to be handled. For example, when ordering two prepositional phrases which have the same argument, such as "before bypass" and "after bypass," it is preferable to generate "before and after bypass" instead of "after and before bypass." To order them correctly, the system needs to know that "before" precedes "after." Currently, the "smaller" function always returns false when comparing two proper names.

5.5.2 Identical Constituents (affects Step 2)

Van Oirsouw (1987) proposed a general rule, deletion by identity, to handle coordinating conjunctions. The algorithm described in Section 5.4 is also based on the deletion of identical expressions. Most deletion accounts of coordinating conjunctions prior to van Oirsouw's work assumed that deletion is performed under identity of words, but did not clarify what it means for two words to be identical. Sag (1976) and Van Oirsouw (1987) explored the concept of identity in detail. Sag specifically looked into the "sloppy identity" problem in a sentence like "John loves his mother, and Bill does too." This sentence can be derived from either (62a) or (62b), where the pronoun "his" refers to different entities.

(62) a. John$_i$ loves his$_i$ mother.
Bill$_j$ loves his$_i$ mother.

b. John$_i$ loves his$_i$ mother.
Bill$_j$ loves his$_j$ mother.

In the process of generating "John loves his mother, and Bill does too," an algorithm can use *identity of the indexicals* to combine the two clauses in (62a; but in (62b, the concept of *alphabetic identity* provides a more appropriate account. Sag used λ-calculus to analyze this phenomenon. But earlier, Examples (57), (58), and (59) have already shown that the coordinating conjunction is not merely a surface-level construction. Van Oirsouw also discussed morphological and referential identity. Our approach extends the concept of identity further and is described in detail in this section.

In our analysis, the task of identifying identical constituents can be implemented using two equivalence tests:

- *alphabet equivalence*: Two constituents are equivalent if they have the same surface expressions. The examples in (62b) passed this test.
- *sense equivalence*: Two constituents are equivalent if they have the same sense. This is also known as *identity of the indexicals*. In short, for nouns, their entity identifiers are tested; for verbs and adjectives, their lexical senses are tested; and for prepositions, their thematic roles are tested.

Constituents that pass both equivalence tests are considered identical and will be marked as "recurring" in Step 2 of the algorithm in Section 5.4. Of course, those constituents which do not pass either tests are considered not identical and not deleted. The more interesting cases in which one test is passed while the other test fails are discussed next.

When two constituents pass the sense equivalence test but fail the alphabet equivalence test, they should not be considered for deletion. In the following sentence, even though the subjects in both clauses refer to the same person, none of them are deleted:

(63) Fidel Castro$_i$ stood in an open car and the Cuban Prime Minister$_i$ was enthusiastically welcomed.

The reason for referring to the same entity with different surface expressions is a stylistic decision and not based on conciseness criteria. This example suggests that referring expression decisions affect the coordinating conjunction process. Magic has examples in which, depending on the decisions of the referring expression module, different coordinating conjunction operations are applied. If the referring expression module chooses to map the time points associated with each medication, as in (64a), to a specific time period during a surgery, as in (64b), CASPER can use a one-distinct operation to produce (64c), with a non-constituent coordination.

(64) a. The patient received Fentanyl at 13:15:34.
The patient received Protamine at 14:20:43.

b. The patient received Fentanyl before induction.
The patient receive Protamine before end of bypass.

c. The patient received Fentanyl before induction and Protamine before end of bypass.

(65) a. The patient received Fentanyl during surgery.
The patient received Protamine during surgery.

b. The patient received Fentanyl and Protamine during surgery.

But instead, if the referring expression module uses an imprecise time frame, such as "during surgery" in (65a), CASPER can use a one-distinct operation to produce a more concise sentence, as in (65b), with a conjoined object. Both (64c) and (65b) describe the same events. But, depending on how the entities are referred to in the unaggregated propositions, different coordinated conjunction constructions result.

When two constituents pass the alphabet equivalence test but fail the sense test, CASPER also considers them non-identical and no deletions are applied. Sentences violating these constraints are also known as *zeugma* and are considered ungrammatical. In general, the *sense* of a basic constituent is a unique identifier which represents its meaning. To synthesize grammatical conjunction constructions, the definition of "sense" is similar to the concept of "synset" in WordNet (Miller et al., 1990), but with modifications. To simplify the discussion below, we will assume that the constituents being tested have no modifiers. If they do have modifiers, which might include an AP, PP, or relative clause, the equivalence check can be performed recursively.

Using senses of nouns to determine object identity first seems reasonable. For example, in (66a), the word "bank" in the first clause refers to a financial institution and the word "bank" in the second clause refers to a river bank.

(66) a. The bank$_i$ opens on Monday.
The bank$_j$ filled with garbage.

b. *The bank opens on Monday and filled with garbage.

Since their senses are different, deleting one of them based on surface expression alone would produce ungrammatical sentences as in (66b). Using senses of nouns would have worked in this case. Testing using entity identifiers is more general, as demonstrated by the indefinite NPs in (67a).

(67) a. John saw an apple.
John saw an apple.

b. John saw the same apple.

c. John saw two apples.

d. John saw apples.

Using sense alone, Casper would not able to distinguish whether the indefinite NPs, apples, in (67a) are the same or different, thus cannot decide which one of (67c), (67d), (67d) is appropriate realization of the unaggregated clauses in (67a). Instead of using sense to verify that two nouns are equivalent, Casper uses entity identifiers.

For verbs, adjectives, and prepositions, using identifiers for a sense equivalence test is not appropriate. In Casper, unless two entities refer to the same entity, every entity has a unique identifier. Similar rules apply to events, states, and attributes. This policy would allow a referring expression to refer to any entity or relation mentioned previously in the discourse. Because events usually occur at different time points, it would be strange to assume that all C-RECEIVE events or all C-HAS-ATTRIBUTE statements have the same identifier. When the system is determining whether two verbs are equivalent, the system looks only at the sense of the verb. This allows the system to combine the sentences in (68a) into (68b), even though the verbs "receive" have different identifiers.

(68) a. John received Fentanyl.
John received Protamine.

b. John received Fentanyl and Protamine.

For verbs that are polysemous, the sense equivalence test works very well. Given (69a), the system would detect that the verb senses are distinct and generate (69c).

(69) a. John took an umbrella.
John took a nap.

b. *John took an umbrella and a nap.

c. John took an umbrella and took a nap.

(70) a. *Mary will make lunch and a scene.
Mary will make lunch and will make a scene.

b. *Mary has an apple and a cold.
Mary has an apple and has a cold.

c. ?John opened a letter and the door.
John opened a letter and opened the door.

The sense of "took" in the first clause of (69a) is a physical action, while the "took" in the second clause is an empty verb. Since they have different senses, the multiple-distinct operation is applied to produce (69c). In similar examples shown in (70), deleting based on both alphabet and sense equivalence tests are grammatical, while those based only on the alphabet equivalence test are not. The issue of identity is further complicated by the special treatments required by verbs. In English, the deletion of recurring verbs is not affected by person or number agreement. In (71c), although the verb "vote" in the second clause is third-person-plural instead of first-person-singular, as is the verb in the first clause, it is still deleted.

(71) a. I like apples and John oranges.

b. John likes apples and my friends oranges.

c. Tom votes for Jerry and the other committee members Ben.

Other features of a verb, such as tense or mode, do matter and cannot be deleted, e.g., (72).

(72) a. John ate apples.
John will eat oranges.

b. John ate apples and will eat oranges.

c. *John ate apples and ϕ oranges.

Casper can handle the examples just described. The following complex example was described in Kaplan and Maxwell (1988).

(73) a. The girl promised John to go.
The girl persuaded John to go.

b. *The girl promised and persuaded John to go.

c. The girl promised John to go and persuaded him to go too.

In the first sentence in (73a), "the girl" is the unrealized subject of the clause "to go," while in the second sentence, the unrealized subject of the clause "to go" is "John." Although Casper does not yet generate such sentences, the proposed algorithm can be extended to handle unrealized entities at semantic level to ensure correct synthesis of such a conjunction construction.

For adjectives, strictly speaking, a sense equivalence test should also be used. For example, the adjective "beautiful" in "a beautiful dancer" can either describe the physical appearance of a dancer, or describe the impression of a dancer's dance performance. Since these two meanings are distinct, they should not be considered sense equivalent and should not be deleted when they occur in a conjunction construction. Since all adjectives in Magic and PLANDoc have only one sense, sense equivalence tests for adjectives are always true when the alphabet equivalence test is true.

For testing the sense equivalence of PPs, the situation becomes rather interesting. WordNet does not contain entries for prepositions; thus, it does not define any synset for them. For synthesis of conjunction constructions, the prepositions obtain their sense by the particular relationships they play between the entities they connect. In other words, the sense of a preposition is directly related to its thematic roles in a proposition. In (74a) and (75a), the unaggregated prepositions have very similar surface expressions, but because both thematic roles for "for" in (74a) are BENEFICIARY, the forward deletion is applied to the subsequent references to "for." In (75a), the "for" in the second clause signals PURPOSE, not BENEFICIARY. Since they have different senses, deletion cannot be applied. Instead of "for Mary and fun," as in (75b), the preposition with a different thematic role must be repeated, as in (75c) or (76b).

(74) a. John cooked dinner for Mary.
John cooked dinner for Paul.

b. John cooked dinner for Mary and Paul.

(75) a. John cooked dinner for Mary.
John cooked dinner for fun.

b. *John cooked dinner for Mary and fun.

c. John cooked dinner for Mary and for fun.

(76) a. *Mary came in a hurry and a taxi.

b. Mary came in a hurry and in a taxi.

In addition to thematic roles of adverbial constructions, the difference between complement and adjunct also affect conjunction constructions:

(77) a. John put the chair in the house.
John slept in the house.

b. ?John slept ϕ and put the chair in the house.

c. *John put the chair and slept in the house.

d. John put the chair in the house and slept in there.

In (77b) and (77c), the first occurrence of "in the house" was deleted. These sentences are strange or ungrammatical because "in the house" is a complement in the first sentence of (77a) while in the second sentence of (77a), it is an adjunct. As a result, they cannot be deleted even though, on the surface level, they could have been. Replacing the second recurring reference to "in the house" with a locational anaphoric expression "in there", as in (77d), is both concise and grammatical.

Based on the identity information, Step 2 in the conjunction algorithm determines which constituents are recurring. Now, we will turn our attention to constraints which affect Step 4.

5.5.3 Directionality (affects Step 4)

The proposal for directional constraints was first offered by Ross (1970) for gapping constructions:

> The order in which GAPPING operates depends on the order of elements at the time that the rule applies; if the identical elements are on left branches, GAPPING operates forward; if they are on the right branches, it operates backwards. (p. 251)

Later, Tai (1969) extended the directional constraint to cover other deletions under identity in coordinated structures. For identical constituents which appear in the beginning or medial position of a clause, the recurring constituent *deletes forward* (i.e., the subsequent occurrence of the identical constituent disappears). When identical constituents are realized at the end of the clause, the recurring constituent *deletes backward* (i.e., the antecedent occurrence of the identical constituent disappears). Both van Oirsouw (1987) and the current research incorporate a unified rule to transform sentences into a conjoined one, based extensively on deletion and directional constraint.

Example (78) demonstrates the importance of the directional constraint. If CASPER only deletes forward or deletes backward when conjoining the sentences in (78a), ungrammatical sentences occur, such as those in (78b) or (78c).

(78) a. John baked apple on Monday.
John ate orange on Monday.

b. *Baked apple and John ate orange on Monday.

c. *John baked apple on Monday and ate orange.

d. John baked apple and ate orange on Monday.

Only when the directional constraint is observed does the grammatical sentence (78d) result. To perform this operation in a text generation system, the surface realizer SURGE uses the identity information computed by CASPER in Step 2 and the constituent ordering information which becomes available only during surface realization to perform the deletion correctly in Step 4.

It is interesting to note that when multiple adjacent constituents are marked identical in a proposition, they should be merged and considered as one unit. Then, based on whether such a unit appears in the beginning, medial, or end position of a clause, the deletion direction can subsequently be determined. In the examples shown in Table 5.1, both entities in ARG2 and CIRCUM-TIME are identical. If the adjacent identical units are not merged, an algorithm based on directional constraints would produce the sentence "*John washed apples and Phil peeled on Monday," which is ungrammatical. In the table, the value in the "Status" column indicates the grammaticality of the sentence, with "*" signaling ungrammatical, "?" signaling questionable grammatical, and "G" standing for grammatical. The "Slot" column contains four letters, each letter referring to the identity condition of the entities in each argument in the propositions being coordinated. "D" stands for "Distinct." In the case of ARG1, the entities are "John" and "Phil"; thus, they are distinct and "D" is given as the first letter. "S" stands for "the Same"; e.g., both the entities in ARG2 and CIRCUM-TIME are the same across the propositions.

	Status	Slots	ARG1	PRED	ARG2	CIRCUM-TIME
1.	*	ddss	John	washed	apples	on Monday and
			Phil	peeled	apples	on Monday.
2.	G	ddss	John	washed	apples	on Monday and
			Phil	peeled	apples	on Monday.

Table 5.1: One-distinct operation is an optimization of multiple-distinct, based on directional constraints.

Because adjacent constituents being deleted should be merged, the unit "apples on Monday" is considered as appearing at the end of a clause and thus deletes backward, as the second example in Table 5.1 shows.

Another interesting observation is that one-distinct operations can be treated as merely computational optimizations, not as a special case of coordinating conjunctions. This observation is supported by the fact that all the conjunctions in Table 5.2 can be derived from both a one-distinct operation and a multiple-distinct operation using directional constraints. Looking at the letter in the "Slots" column, it is clear that all of the clauses being coordinated satisfy the precondition of one-distinct operation: they all have only one slot that contains distinct entities. With the correct handling of directional constraints, Casper can synthesize coordinating conjunction constructions systematically and correctly. Some readers might notice that if the tense is present, the analysis might have a minor problem, namely, that the number agreement aspect of the verb and conjoined subjects is

	Status	Slots	ARG1	PRED	ARG2	CIRCUM-TIME
1.	G	dsss	John	washed	apples	on Monday and
			Phil	washed	apples	on Monday.
2.	G	sdss	John	washed	apples	on Monday and
			John	peeled	apples	on Monday.
3.	G	ssds	John	washed	apples	on Monday and
			John	washed	oranges	on Monday.
4.	G	sssd	John	washed	apples	on Monday and
			John	washed	apples	on Tuesday.

Table 5.2: One-distinct operation is an optimization of multiple-distinct operation based on directional constraints.

not captured using the multiple-distinct algorithm. For example, "*John ϕ ϕ and Phil washes apples" is an ungrammatical sentence. We consider this a morphological issue which can be taken care of by passing on the proper features to later linearization and morphology modules.

Linguists noted that directional constraint seems to be a universal phenomenon. For example, depending on the default ordering of a constituent in a language, gapping of verbs seems to behave according to directional constraints. English is a Subject-Verb-Object (SVO) language, where subject usually appears before a verb and a verb appears before an object. When two sentences are combined using a gapping operation, the resulting structure is "SVO and SO," where the medial verbs delete forward. In languages which have a different default ordering, such as SOV in Japanese, the result of gapping would be "SO and SOV," where the verbs at the end of the clauses delete backward. Readers interested in language universals should refer to the work by Neijt (1979), van Oirsouw (1987), and Steedman (2000).

5.5.4 Peripherality (affects Step 4)

In the current implementation, directional constraints are used to determine which recurrent elements to delete. Section 5.5.2 already established that the deletion process is not just a surface-based phenomenon. An interesting question is which thematic roles should be considered for the identity test. As defined in the conjunction algorithm in Section 5.4, identity tests are applied to the entities in the same thematic role among aggregated propositions, and deletions are applied to recurring constituents based on directional constraints. A more general algorithm would take account of the peripherality of the constituents when performing identity tests. *Peripherality* is defined as follows: a constituent is peripheral if it is immediately left-adjacent or right-adjacent to its S-boundary. In (79), although the entity in the subject of the first active clause and the subject of the second passive clause are identical, their thematic role is not the same. In the first clause, the subject is in ARG1, while in the second clause, the subject is in ARG2.

(79) The Dodgers beat the Red Sox and were beaten by the Giants.

To handle such cases, the algorithm needs to first use the passive transformation to put the ARG2, "the Dodgers," into the subject position, making the constituent available at the left peripheral position and then deleting it based on directional constraints. Casper currently detects that the voice is passive in the second clause and applies an identity test to the entities in different thematic roles. Other examples of deletions based on peripheral aspects are:

(80) Mary likes ϕ and Jane would go anywhere to find antique horse-brasses from the workshop of that genius in metalwork, Sam Small. (Hudson, 1976, p. 535)

The recurring entity *antique horse-brasses...* is in ARG2 at the nucleus of the first clause, but in the second clause, it is in a less dominant position as in the ARG2 of the adjunct CIRCUM-PURPOSE phrase. Example (81) is another coordinated example which requires proper handling of peripherality:

(81) John enjoyed and Mary believed that Susan hated the dinner.

The constituent "the dinner" is in ARG2 of the first proposition, but in the second proposition, it is the ARG2 of the embedded proposition "Susan hated the dinner" in the second proposition. CASPER currently does not incorporate peripheral constraints. Incorporating peripheral constraints into the algorithm would have a negative impact on efficiency.

5.6 Convey Original Meaning

When multiple clauses are combined to create a sentence with coordinating conjunctions, it is important that the combined sentence convey exactly the same information as in the unaggregated propositions. Unfortunately, this is not always possible. Sometimes, there is a price to pay for the capability of generating concise sentences: ambiguities might be produced as a result. The goal of this section is to identify what ambiguities might result and what measures a generation system can take to minimize them. But before going into details of the different ambiguities, it is important to note that not all ambiguities in the generated sentences need to be avoided. People tolerate the ubiquitous PP-attachment ambiguity used in everyday language. Instead of resolving such ambiguities, humans often employ context to resolve them. Since people do employ sentences with ambiguous PP-attachments, the fact that CASPER generates sentences with ambiguous sentence structures should be considered a feature, not a bug.

We identified three types of ambiguities which can result from using coordinating conjunctions in a sentence: (1) distributive reading versus collective reading, (2) undesirable implicatures, and (3) interactions between other linguistic constructions with coordination. Whenever entities are conjoined together in the same constituent, there is always the possibility that these entities act as one or act individually, resulting in collective versus distributive readings. We will address the ambiguity between the two readings first and list linguistic devices to eliminate such ambiguities. Then, we will look into the possible implicatures which might result from using coordinating conjunctions. Section 5.6.3 discusses the interaction between coordinating conjunctions and other linguistic constructions, such as modification constructions, negation, quantifiers, and ellipsis.

5.6.1 Collective versus Distributive Readings

In Section 5.2, a sentence with coordinating conjunctions can have either a *distributive reading* or a *collective reading*. In sentences with a distributive reading, the aggregated sentence can be analyzed as a conjunction of multiple sentences, a construction which is also known as S-coordination (S stands for Sentence or a clause, as a common practice in linguistics). Those sentences which cannot be analyzed as conjunctions of clauses have collective readings and are also known as non-S-coordinations. The sentence "Dr. Smith and Dr. Edward examined the patient" can be interpreted either as both doctors examined the patient at same time (a collective reading), or separately at different times (a distributive reading). The conjunction algorithm described in Section 5.4 specifically creates sentences which are conjunction of clauses; thus, it only synthesizes sentences with distributive readings. This does not imply that Casper cannot generate sentences with conjunctions with collective readings. Since the conjoined structures with collective readings are not derived from a conjunction of clauses, these structures should be specified or synthesized in the content planner. Currently, neither the content planners in Magic and PLANDoc are capable of systematically synthesizing conjunction constructions with collective readings. The distinction between the two readings corresponds to the different treatments of coordinating conjunctions using transformation rules and phrase structure rules. We agree with Lakoff and Peters (1969) that different approaches should be used to handle each reading. Casper generates sentences with both collective and distributive readings, but such operations are simply performed in different modules.

English has several markers which can signal which meaning is intended by the speaker. In cases of NPs, the speaker can append the word "each" or "separately" after the last conjoined constituent to signal a distributive reading, as in (82a) and (82b). This seems to work well for subjects, but not for objects, as in (82c) and (82d).

(82) a. John and Mary each ate an apple.

b. John, Paul, and Mary separately lifted the table.

c. *John ate an apple and an orange each.

d. *The teacher reprimanded John and Mary each.

Consistently inserting such markers after every conjunction structure unfortunately makes a text very redundant. As a result, even though Casper can signal the intended reading, it does not do so in Magic or PLANDoc. For sentences resulting from multiple-distinct operations, the readings are almost always distributive, as in (83b), so they do not pose a problem.

(83) a. John chopped parsley and Mary onions.

b. John ate dinner and drank beer.

To signal a collective reading, the system can use "together" or "simultaneously," as in "John and Mary together chopped onions."

As many linguists have noted, the issue with collective readings and distributive readings affect not only coordination conjunctions, but also plural constructions.

(84) a. The doctors examined the patient.

b. The students ate apples.

In these cases, it is unclear if the doctors acted together or if the students ate at the same time. Such *vagueness* or *underspecification* in natural language usage is common. Currently Casper only generates markers if they are provided in the input.

5.6.2 Implicatures

In the newsgroup rec.humor.funny, a post with the subject line "coincidence?" was sent by Kevin Kelleher:

> Two successive headlines from the AP wire service:
>
> Victoria's Secret Webcasts Show
> Iraq To Get Internet Access

As this example shows, randomly joining sentences together might cause undesirable implicatures. Quirk et al. (1985) list eight semantic implicatures of coordination by "`and`." In the following examples, the second clause is related to the first clause by the rhetorical relation specified below:

1. CONSEQUENCE-OR-RESULT: He heard an explosion and he (therefore) phoned the police.
2. SEQUENT: I washed the dishes and (then) I dried them.
3. CONTRAST: Robert is secretive and (in contrast) David is candid.
4. CONCESSIVE: She tried hard and (yet) she failed.
5. CONDITION: Give me some money and (then) I'll help you escape.
6. SIMILAR: A grade agreement should be no problem, and (similarly) a cultural exchange could be easily arranged.
7. PURE-ADDITION: He has long hair and (also) he often wears jeans.
8. COMMENT-OR-EXPLANATION: They disliked John – and that's not surprising in view of his behavior.

In these implicatures, the first clause can only be swapped positionally with the second clause when they are linked by CONTRAST, SIMILAR, or PURE-ADDITION relationships. Depending on the domain, the number of occurrences of each rhetorical relation appearing in the input might have very different distributions. In both MAGIC and PLANDOC, the rhetorical relations that are mapped to coordinating conjunctions are PURE-ADDITION, SEQUENT, and CONSEQUENCE-OR-RESULT.

In a descriptive domain, the propositions linked by PURE-ADDITION are the result of retrieving facts from the database, such as "What drugs are given to the patient in a certain time period?" or "What are the laptop models between $1,500-$2,000?" Multiple propositions might be returned as valid answers and they will be linked by the PURE-ADDITION relation among them. Propositions linked by the PURE-ADDITION relation might be reordered to produce the most concise text. For example, in the medical domain, the propositions might be ordered according to the amount of drugs given, the method of drugs given (injecting or swallowing), or the chronological order of the events. In contrast, propositions linked by SEQUENT and CONSEQUENCE-OR-RESULT relations cannot be reordered.

It is interesting to note that in the medical domain, it is difficult to confirm that two propositions have a CONSEQUENCE-OR-RESULT relationship. Given that a patient had hypertension at a certain time period and he/she received Fentanyl and Protomaine in the 10 minutes right after the hypertension, the system first generated the sentence, "The patient had an episode of hypertension before end of bypass and was treated with Fentanyl and Protomaine." The physician on our team noticed that not all the drugs given during the 10-minute period were related to the treatment of hypertension. As a result, we had to change the lexical chooser to reflect the system's confidence of the CONSEQUENCE-OR-RESULT relationship. Now the sentence, "The patient had an episode of hypertension before end of bypass and received Fentanyl and Protomaine," is generated instead. The implicature of CONSEQUENCE-OR-RESULT relation still exists, but it is not as strong as in the previous version. Although it is not impossible to improve the inference engine to ensure that only the relevant drugs are inserted in the second proposition, given the potential interactions between drugs and the number of drugs in medicine, we decided not to pursue this further. In critical applications, implicatures as a result of aggregation should be taken into account and minimized.

For SEQUENT and CONSEQUENCE-OR-RESULT, the clause aggregation module can always insert "then" or "as a result," respectively, into the second clause to clarify the rhetorical relation. The problem is that in normal text, humans do not always make such insertions. As a result, consistently inserting such markers will make the text verbose and strange. Using the pragmatic constraints to enable the sentence planner to delete the markers at appropriate places is still an open research topic.

5.6.3 Scope and Coordination

Similar to other advanced linguistic constructions like quantifiers and prepositions, coordinators also have a scope which might result in multiple interpretations. This section lists such constructions and points out some remedies to eliminate ambiguity.

5.6.3.1 Modifiers

When NPs with modifiers are conjoined, ambiguities might result. The NP "tall men and women" can be derived from either (85a) or (85b).

(85) a. tall men
women

b. tall men
   tall women.

In (85b), since the conjoined NPs have the same modifier in front of the head, the system would use directional constraints to delete the recurring adjective in the second NP. Due to the deletion, coordinating the semantically distinct nouns in (85a) or (85b) would result in the same surface expression. We identified two approaches to prevent the modifiers scoping across the coordinator:

- reorder the conjoined elements
- signal the boundary of NP with a modifier or an article

For example, we can eliminate the ambiguities in (86a), "intelligent students and faculty members," by reordering the conjoined constituents and moving the noun with a premodifier to the last position, as in (86b).

(86) a. intelligent students and faculty members

b. faculty members and intelligent students

For NPs with postmodifiers, such as PP or relative clause, the system would reorder the conjoined elements (e.g., (87a)) by moving those with a postmodifier to the front of the conjoined NP (e.g., (87b)).

(87) a. faculty members and students with gowns

b. students with gowns and faculty members

In (86a) and (87a), the modifiers might describe both elements in the conjoined expressions. In other words, the scope of the modifier might be wider than the conjunctor. Based on this observation, if deletion of modifiers occurred, then reordering the conjoined elements is not necessary. Otherwise, the system moves the elements based on predictions from directional constraints. In certain cases, despite possible scope ambiguities between modifiers and conjunctors, humans can disambiguate the scope based on semantic information, such as "pregnant women and children." CASPER currently reorders constituents according to directional constraints without taking advantage of such lexical information.

Another way to eliminate modifier ambiguity is to signal the boundary of the NPs involved in the construction. To do this, we can either insert a premodifier or an article into the second NP in the premodifier case, "intelligent students and *excited* faculty members" or "faculty members and *the* students with gowns"), or insert a postmodifier into the first NP to signal the boundary of the first NP: "intelligent students *with gowns* and faculty members."

5.6.3.2 Negation

In addition to modifiers, negation is another device in natural language which also has scope as shown in (88b) and (89b).

(88) a. I smoke.
I don't drink.

b. *I don't drink and smoke.

c. I smoke and don't drink.

(89) a. This dish contains pepper.
This dish contains no cinnamon.

b. *This dish contains no cinnamon and pepper.

c. This dish contains pepper and no cinnamon.

For coordination constructions in which some, but not all, elements contain negation, we can reorder the elements to make the scope clear. Since negation occurs in the premodifier position, the elements with negation are moved toward the end of the conjoined list. In such cases, since there is a change in the polarity of the conjoined element, using the conjunctor "but" is more fluent. The same observations apply to multiple-distinct cases, as shown in (90a).

(90) a. I chew tobacco.
I don't drink alcohol.

b. *I don't drink alcohol and chew tobacco.

c. I chew tobacco and don't drink alcohol.

d. I chew tobacco but don't drink alcohol.

When all the conjoined elements contain negation, deleting recurring negations might cause ambiguity. Syntactically, it is not clear whether (91c) contains one or two negated clauses. We can either not perform a deletion on the negation, as in (91c) and (92c), or use the conjunctor "nor," as in (91d) and (92d).

(91) a. I don't smoke.
I don't drink.

b. ?I don't drink and smoke.

c. I don't drink and don't smoke.

d. I don't drink nor smoke.

(92) a. I don't drink alcohol.
I don't chew tobacco.

b. ?I don't drink alcohol and chew tobacco.

c. I don't drink alcohol and don't chew tobacco.

d. I don't drink alcohol nor chew tobacco.

Naive gapping of negation is disallowed in English, as in (93b). The conjunctor "nor" should be used instead, as in (93c) (Ross, 1970).

(93) a. Alice didn't eat fish.
Bill didn't eat rice.

b. *Alice didn't eat fish and Bill ϕ rice.

c. Alice didn't eat fish nor Bill ϕ rice.

5.6.3.3 Quantifier

Analysis of quantifiers is a well-known complex problem in semantics. In Chapter 6, we propose an algorithm to generate quantified expressions from multiple clauses. In this section, the focus is to avoid the generation of ambiguous coordinating conjunctions involving existential quantifiers and cardinals.

Because universal quantifiers exhaust all entities of a set within a context, they do not create ambiguities with coordinating conjunctions in general:

(94) a. All the rules are explicit.
All the rules are easy to read.

b. All the rules are explicit and easy to read.

But for sentences with existential quantifiers and conjunctions, these two constructions do interact and ambiguities can arise. Depending on which entities these existential quantified expressions represent, different coordinating operations will be performed to clarify the original meaning in the unaggregated proposition.

(95) a. Few rules are explicit.
Few rules are easy to read.

b. Few rules are explicit and easy to read.

c. Few rules are explicit and few rules easy to read.

(96) a. Some men are married.
Some men are happy.

b. Some men are married and happy.

c. Some men are married and some men happy.

In both (95a) and (96a), though the surface expressions of the existentially quantified expressions are the same, it is not clear if they refer to the same set of entities. In Section 5.3, we explained that the content planner must provide semantic identifiers for the entities. Based on such information, it is a simple task to determine if the two sets of entities referred to in the existentially quantified expression are identical. If they are, (95b) and (96b) would be generated; if they are not, the system would generate (95c) and (96c). By mentioning the same surface expression multiple times at the surface level, the generation system indicates that the existentially quantified expressions do not refer to the same set of entities. In addition, because quantifiers sometimes appear in a similar relative position as premodifiers to their head nouns, the same reordering rules which apply to premodifiers also apply to quantifiers.

(97) a. Some men must leave the area.
Women must leave the area.

b. → *Some men and women must leave the area.

c. → Women and some men must leave the area.

Given the two unaggregated propositions in (97a), Casper generates (97c) instead of (97b). Issues with quantifiers and coordinations are studied in Partee (1970) and McCawley (1981).

Another major source of ambiguity involving existential quantifier[4] and coordination occurs in collective and distributive readings. Because of the numerous

[4]We consider cardinal quantifiers as special cases of existential quantifiers.

possible interpretations of existential quantified expressions, such constructions are a particular source of ambiguity. We have identified five possible operations to handle the interactions between conjunction and existential quantifiers. The algorithm to synthesize quantified expressions is described in Chapter 6. For now, we will simply describe the operations at an abstract level.

(98) a. John was carrying three baskets.
Mary was carrying a basket.

b. John was carrying a basket.
Mary was carrying a basket.

Given the two propositions in (98a) and assuming these are different baskets, a generation system has three possible realizations:

1. use a multiple-distinct operation without deletion: John was carrying three baskets and Mary a basket.

2. merge existential quantified expressions and use a one-distinct operation: John and Mary were carrying baskets.

3. add existential quantified expressions and use a one-distinct operation: John and Mary were carrying four baskets.

If the existential quantified expressions are the same but refer to different entities, as in (98b), the system has two other possible realizations:

4. use the same existential quantified expression and use a one-distinct operation: John and Mary were carrying a basket.

5. use the same existential quantified expression and use a one-distinct operation with cue phrase "each" appended: John and Mary each were carrying a basket.

There are other possibilities, such as inserting "another" or "some other" in the second existential quantified expression. Depending on the context of a particular application, one or many of those possible realizations might be appropriate. This parameter needs to be identified and set for each application.

5.6.3.4 Ellipsis

Because ellipsis deletes various elements at the surface level, one might wonder if the semantics in the original propositions can always be recovered unambiguously as a result of the ellipsis operation.

(99) a. Dawn gave Billy some milk and Dawn gave Edward some squash.

b. Dawn gave Billy some milk and Edward gave Billy some squash.

c. → Dawn gave Billy some milk and Edward some squash.

Unfortunately, (99c) can be derived from either applying forward deletion to (99a) or to (99b). As a result, this example shows that ellipsis can create ambiguity. We assume that contextual information provided by the underlying application is sufficient for the readers to disambiguate such sentences, and CASPER currently does not have any mechanism to minimize such ambiguity.

5.7 Coverage of the Algorithm

5.7.1 Coverage in Linguistic Literature

This section presents challenging examples from diverse linguistic literature (Quirk et al., 1985; van Oirsouw, 1987) and shows how the algorithm developed in Section 5.4 generates them. Linguists have categorized coordination into *simple* and *complex*. Simple coordination conjoins single clauses or clause constituents, while complex coordination involves multiple constituents. In our algorithm, the one-distinct procedure can generate all simple coordinated structures with distributive readings, including coordinated verbs, nouns, adjectives, PPs, etc. With simple extensions to the algorithm, propositions with relative clauses could be combined and coordinated also. This is the easy part. The more difficult problem is how a system generates coordinated structures involving multiple constituents. Since coordination traditionally involves elements of equal syntactic status, those constructions are abnormal. But they do exist and informants have no problem understanding sentences with such constructions.

Based on the literature on coordinated deletions, van Oirsouw (1987) identified a number of rules which result in deletion under identity: gapping, which deletes a verb; *Right-Node-Raising* (RNR), which deletes identical right-most constituents in a syntactic tree; and *VP-deletion* (VPD), which deletes identical verbs

Gapping: John ate fish and Bill ϕ rice.
RNR: John caught ϕ, and Mary killed the rabid dog.
VPD: John sleeps, and Peter does ϕ, too.
CR1: John gave ϕ ϕ, and Peter sold a record to Sue.
CR2: John gave a book to Mary and ϕ ϕ a record to Sue.

Figure 5.7: Four coordination rules for identity deletion, as described by van Oirsouw.

and handles post-auxiliary deletion (Sag, 1976). van Oirsouw used the term *Conjunction Reduction* (CR), which deletes identical right-most or left-most material. He pointed out that these four rules reduce the length of a coordination by deleting identical material, and serve no other purpose. We will describe how our algorithm handles the examples van Oirsouw used in Figure 5.7.

The algorithm described in Section 5.4 can use the multiple-distinct procedure to handle all cases except VPD. The generation of VPD and ellipsis constructions involving "too" and "also" are not addressed in this chapter. In the gapping example, the PRED deletes forward. In RNR, ARG2 deletes backward because it is positioned at the end of the clause. In CR1, even though the medial slot ARG2 should delete forward, it deletes backward because it is considered to be at the end position of a clause. In this case, once ARG3 (the BENEFICIARY "to Sue") deletes backward, ARG2 is at the end position of a clause. In CR2, it is straightforward to delete forward because both ARG1 and PRED are medial. The current algorithm does not address VPD. For such a sentence, the system would have generated "John and Peter slept" using a one-distinct operation.

Complex coordinations involving ellipsis and gapping are much more challenging. In multiple-distinct coordination, each conjoined proposition is generated, but recurring elements among the propositions are deleted, depending on the extended directionality constraints mentioned in Subsection 5.4.4. This works because it takes advantage of the parallel structure at the surface level.

Non-constituent coordination phenomena, the coordination of elements that are not of equal syntactic status, are challenging for syntactic theories. The following non-constituent coordination can be explained nicely with the multiple-distinct procedure. In the sentence, "The spy *was in his forties*, *of average build*, and *spoke with a slightly foreign accent*," the coordinated constituents are VP, PP, and VP. Based on our analysis, the sentence could be generated by combining the first two clauses using the one-distinct procedure, and the third clause is combined using the multiple-distinct procedure, with ARG1 ("the spy") deleted forward.

(100) The spy was in his forties, [the spy] [was] of average build, and [the spy] spoke with a slightly foreign accent.

Other conjunction constructions with unlike categories, such as (101), can also be handled nicely by our algorithm.

(101) She walked slowly and with great care.

This sentence can be best analyzed as "She walked slowly and she walked with great care" by generating both clauses and then deleting both ARG1 and PRED. This is a much more satisfying account of the sentence than applying the PS-rule, which results in the conjunction of unlike categories, an adverb and a PP.

In the rest of this section, we will evaluate the correctness of the proposed algorithm by applying it to all possible configurations of two conjoined clauses with three and four thematic slots. In the "Status" column, "G" stands for "grammatical," "*" stands for "ungrammatical," and "?" stands for "questionable grammatical." In column "Slot," "d" stands for "Distinct" and "s" stands for "the Same." These letters refer to whether the entities in a particular slot are the same or distinct. They are exhaustively permuted to ensure that all possible cases are tested. Table 5.3 contains only three slots. Sentence (1) is a conjunction of two full clauses. Sentence (2) is an instance of right-node-raising. Sentence (3) is a case of gapping. Sentence (4) contains a conjunction of VPs. Sentences (5), (6), and (7) contain simple conjunctions. It is unclear what it means to conjoin two clauses containing the same semantics, as in Sentence (8). Based on the resulting sentences shown in Table 5.3, the proposed conjunction algorithm works well with propositions with three slots.

In the next three tables, we attempted to apply our conjunction algorithm to propositions containing four thematic slots. Although most of the resulting sentences are grammatical, there are cases which the proposed algorithm failed. We speculated why they might have failed and will discuss these possibilities later. The main reason for using three different tables is that the propositions conjoined in each table have different syntactic, semantic, or lexical properties. In Table 5.4, the fourth slot is an optional adjunct "CIRCUM-TIME," which can appear at the beginning or end of a clause. In Table 5.5, although the fourth column "ARG3" also contains a PP, it is a complement, not an optional adjunct as in Table 5.4. Fortunately, if our conjunction algorithm failed a case in Table 5.4, the corresponding case in Table 5.5 also failed. It seems that whether a constituent is a complement

	Status	Slots	ARG1	PRED	ARG2
1.	G	ddd	John Phil	likes adores	Mary and Sue.
2.	G	dds	John Phil	likes adores	Mary and Mary.
3.	G	dsd	John Phil	likes likes	Mary and Sue.
4.	G	sdd	John John	likes adores	Mary and Sue.
5.	G	dss	John Phil	likes like	Mary and Mary.
6.	G	sds	John John	likes adores	Mary and Mary.
7.	G	ssd	John John	likes likes	Mary and Sue.
8.	*	sss	John John	likes likes	Mary and Mary.

Table 5.3: Coordination conjunctions involving three thematic slots.

or an adjunct does not affect the grammaticality in these cases. The sentences in Table 5.6 are semantically equivalent to the sentences in Table 5.5. The transformation of the adjunct PP phrase into a complement creates many more ungrammatical sentences. But interestingly enough, all cases that failed in Table 5.5 also failed in Table 5.6.

In the cases that failed in Tables 5.4 and 5.5, the problematic cases (3), (4), and (10) all involved medial deletions. Sentence (10) was especially troublesome because two non-adjacent deletions were in the same clause. Multiple deletions in the same clause are likely to require more processing in terms of memory and cause difficulties in recovering the deleted entities. In Table 5.6, both (4) and (7) contain three nouns concatenated in a sequence without any verb or prepositional phrases to signal constituent boundaries. Overall, these tables have shown that the proposed algorithm works quite well. Before we can develop a better algorithm to explain why those cases failed, we can incorporate the current findings into a better conjunction algorithm by encoding all the configurations that produce grammatical sentences and using special operations when the system encounters the configurations that failed in those tables.

5.7.2 Corpus Coverage

We identify how often conjunction constructions are used in a corpus by dividing the number of sentences containing the conjunction "and" by the number of sentences in a corpus. For medical discharge summaries, it is 36% (33,302/92,688), for WSJ, it is 32% (23,959/75,616). The corpus used for this analysis is described in Section 4.4.3 in Chapter 4. After combining the two corpora, 20% of the sentences with the conjunctor `and` contains two `and`'s, 4% has three `and`'s, and 1% has 4 or more conjunctor `and`'s. The maximum number of conjunctor `and`'s found in the combined corpus is nine, shown in Figure 5.8. Clearly, conjunction constructions are often used in human-to-human communication. A subset of the corpora used for premodifier ordering in Chapter 4 was analyzed to determine the generality of the conjunction algorithm. One hundred sentences containing the conjunctor '`and`' are extracted from each of the medical discharge summaries and The Wall Street Journal articles. Based on the manual analysis of these two hundred sentences, we want to see if the proposed algorithm can generate them and the breakdown of the types of operations used for generating them.

The current analysis is based on manual inspection of the sentences containing conjunctor '`and`' and categorization of the sentence according to the conjunction

	Status	Slots	ARG1	PRED	ARG2	CIRCUM-TIME
1.	G	dddd	John Phil	washed peeled	apples oranges	on Monday and on Tuesday.
2.	G	ddds	John Phil	washed peeled	apples oranges	on Monday and on Monday.
3.	*	ddsd	John Phil	washed peeled	apples apples	on Monday and on Tuesday.
4.	?	dsdd	John Phil	washed washed	apples oranges	on Monday and on Tuesday.
5.	G	sddd	John John	washed peeled	apples oranges	on Monday and on Tuesday.
6.	G	ddss	John Phil	washed peeled	apples apples	on Monday and on Monday.
7.	G	dsds	John Phil	washed washed	apples oranges	on Monday and on Monday.
8.	G	sdds	John John	washed peeled	apples oranges	on Monday and on Monday.
9.	G	dssd	John Phil	washed washed	apples apples	on Monday and on Tuesday.
10.	*	sdsd	John John	washed peeled	apples apples	on Monday and on Tuesday.
11.	G	ssdd	John John	washed washed	apples oranges	on Monday and on Tuesday.
12.	G	dsss	John Phil	washed washed	apples apples	on Monday and on Monday.
13.	G	sdss	John John	washed peeled	apples apples	on Monday and on Monday.
14.	G	ssds	John John	washed washed	apples oranges	on Monday and on Monday.
15.	G	sssd	John John	washed washed	apples apples	on Monday and on Tuesday.
16.	*	ssss	John John	washed washed	apples apples	on Monday and on Monday.

Table 5.4: Coordination conjunctions involving four thematic slots.

	Status	Slots	ARG1	PRED	ARG2	ARG3
1.	G	dddd	John Phil	gave lent	a pen a book	to Mary and to Sue.
2.	G	ddds	John Phil	gave lent	a pen a book	to Mary and to Mary.
3.	*	ddsd	John Phil	gave lent	a pen a pen	to Mary and to Sue.
4.	?	dsdd	John Phil	gave gave	a pen a book	to Mary and to Sue.
5.	G	sddd	John John	gave lent	a pen a book	to Mary and to Sue.
6.	G	ddss	John Phil	gave lent	a pen a pen	to Mary and to Mary.
7.	G	dsds	John Phil	gave gave	a pen a book	to Mary and to Mary.
8.	G	sdds	John John	gave lent	a pen a book	to Mary and to Mary.
9.	G	dssd	John Phil	gave gave	a pen a pen	to Mary and to Sue.
10.	*	sdsd	John John	gave lent	a pen a pen	to Mary and to Sue.
11.	G	ssdd	John John	gave gave	a pen a book	to Mary and to Sue.
12.	G	dsss	John Phil	gave gave	a pen a pen	to Mary and to Mary.
13.	G	sdss	John John	gave lent	a pen a pen	to Mary and to Mary.
14.	G	ssds	John John	gave gave	a pen a book	to Mary and to Mary.
15.	G	sssd	John John	gave gave	a pen a pen	to Mary and to Sue.
16.	*	ssss	John John	gave gave	a pen a pen	to Mary and to Mary.

Table 5.5: Coordination conjunctions involving four thematic slots.

	Status	Slots	ARG1	PRED	ARG3	ARG2
1.	G	dddd	John Phil	gave lent	Mary Sue	a pen and a book.
2.	G	ddds	John Phil	gave lent	Mary Sue	a pen and a pen.
3.	*	ddsd	John Phil	gave lent	Mary Mary	a pen and a book.
4.	*	dsdd	John Phil	gave gave	Mary Sue	a pen and a book.
5.	G	sddd	John John	gave lent	Mary Sue	a pen and a book.
6.	G	ddss	John Phil	gave lent	Mary Mary	a pen and a pen.
7.	*	dsds	John Phil	gave gave	Mary Sue	a pen and a pen.
8.	G	sdds	John John	gave lent	Mary Sue	a pen and a pen.
9.	G	dssd	John Phil	gave gave	Mary Mary	a pen and a book.
10.	*	sdsd	John John	gave lent	Mary Mary	a pen and a book.
11.	G	ssdd	John John	gave gave	Mary Sue	a pen and a book.
12.	G	dsss	John Phil	gave gave	Mary Mary	a pen and a pen.
13.	G	sdss	John John	gave lent	Mary Mary	a pen and a pen.
14.	G	ssds	John John	gave gave	Mary Sue	a pen and a pen.
15.	G	sssd	John John	gave gave	Mary Mary	a pen and a book.
16.	*	ssss	John John	gave gave	Mary Mary	a pen and a pen.

Table 5.6: Coordination conjunctions involving four thematic slots.

Combination of Defense Giants Northrop Grumman Key Systems: B-2 stealth bomber; E-8 Joint Stars battlefield communications plane; E-2C Hawkeye; subcontractor for McDonnell Douglas Corp. F/A-18 Hornet; makes commercial airframes, nacelle systems **and** components for Boeing **and** others; defense-electronics **and** systems integration; data systems **and** services Westinghouse Defense & Electronics Key Systems: Radars for next-generation Air Force F-22 air-superiority fighter **and** Lockheed Martin Corp. F-16; electronics for Boeing-built AWACs **and** Northrop Grumman E-8 Joint Stars; air defense radars, torpedo **and** anti-submarine programs, such as torpedo-recognition systems; air traffic control radars; electronic countermeasures Source: Company reports The nation's most formidable political figure, intellectually **and** politically acute **and** facing a splintered opposition, is poised to dominate the year.

Figure 5.8: A sentence with nine "`and`" found in a corpus.

operations that can be used to synthesize such constructions. The sentences are divided into two major categories, *collective*, and *distributive*. In Section 5.4, we have already described an algorithm which synthesizes sentences containing conjunctions with distributive readings from separate propositions. Despite the surface similarity between conjunctions with collective readings and distributive readings, the proposed algorithm for distributive readings cannot be extended easily to handle the synthesis of coordinated conjunctions with collective readings. Because the decision of whether a group of entities acted collectively is a rather semantic/pragmatic decision, only the content planner has enough information to synthesize sentences containing conjunctions with collective readings. Currently, CASPER does not systematically generate conjunction constructions with collective readings. In this analysis, by default a coordinating construction is distributive. Only in specific cases where it is obvious that a conjunction construction contains only collective readings, then it is marked as collective. Examples of collective readings include "If x and y, then...," "four minutes and 5 seconds," and "between 25% and 30%." The analysis shows that in the medical corpus, 123 (96%) conjunction constructions have distributive readings and only 5 (4%) constructions have collective readings. The number of conjunction constructions is more than the number of sentences in the corpus because some of them contain multiple conjunction constructions. In the WSJ corpus, 104 (83%) constructions have distributive readings and 21 (17%) have the collective reading. From the analysis, it is clear that the majority of the coordination constructions have distributive readings and can be synthesized using our algorithm described in Section 5.4.

Our analysis further breaks down the conjunctions into four smaller types: one-distinct operation, conjunction of VPs, gapping, and medial deletion. Conjunction of VP, gapping, and medial deletion are special cases of multiple-distinct operations in our algorithm. Constructions included in medial deletions are right-node-raising and other non-constituent conjunctions. The result of the analysis is shown in Table 5.7. The majority of the coordinating conjunction constructions can be derived using simple conjunction or one-distinct operations (77%) while all the multiple-distinct operations make up the remaining 23%. Conjunction of VPs, or the deletion of subject in the second clause in the process of synthesizing conjunction construction is quite common (18%). Of those, 9 out of 47 (19%) use passive voice to move the constituents around to enable deletion of subjects. Interestingly, 8 out of those 9 cases occurred in the medical corpus. It is not clear if this is coincidence or caused by genre. In addition to categorizing by the type of conjunction operations, we also categorized them by the syntactic category of the conjuncts. For conjuncts that are not of a basic syntactic category, either NP-PP or S are used to describe them. An example of such a non-constituent conjunction is a conjunction of cardinals and adjectives (e.g., "pulse 60 and regular"). The results are shown in Table 5.8, sorted by the frequency. We also tried to infer the rhetorical relations between the conjuncts. Most of them (86%, or 216 out 252) are ADDITION. Other inferred rhetorical relations include SEQUENCE, NON-VOLITIONAL RESULT, and COMPARATIVE. Their numbers are quite small.

Operation Type	Medical	WSJ	Medical + WSJ
Simple	90 (70%)	103 (83%)	193 (77%)
conj. of VP	28 (22%)	19 (15%)	47 (19%)
gapping	0	1 (1%)	1 (<1%)
medial	10 (8%)	1 (1%)	11 (4%)
Total	128	124	252

Table 5.7: Categorization of conjunctions according to the operations.

For most of the sampled sentences, if the appropriate input representation is given, CASPER would be able to generate the desired coordinated conjunctions. Because the input representation for generating these sentences is not available and very costly to encode manually, we did not test the algorithm directly by generating all the sampled sentences using CASPER. Most of the conjunction constructions can

Conjunct Type	Medical	WSJ	Medical + WSJ
NP	61	85	146 (58%)
VP	28	19	47 (19%)
S	29	11	40 (16%)
ADJ	3	3	6 (2%)
V	0	4	4 (2%)
NP-PP	3	0	3 (1%)
ordinal	2	0	2 (1%)
ADV	1	0	1
PREP	1	0	1
PP	0	1	1
cardinal	0	1	1
Total	128	124	252

Table 5.8: Categorization of conjunctions according to the syntactic type of conjuncts.

be synthesized using one-distinct operations (77%, shown in the top right column of Table 5.7). Many instances of VP, S, and NP-PP can be synthesized using multiple-distinct operations, which are also covered in our conjunction algorithm. Except for a few cases of comparative constructions, i.e., "His I's and O's were roughly even" in medical corpus, Casper can delete redundant expressions using the conjunction algorithm to make the text more concise, fluent, and cohesive.

5.8 Related Constructions

Several linguistic constructions are closely related to coordinating conjunctions. They include apposition, comparatives, "respectively," "each other," and "do too" constructions. Apposition is a well-known paratactic construction since it is of the same syntactic category as the noun it is attached. Because the transformation of a proposition into apposition constructions is very similar to a reduced relative clause construction, it was treated earlier in Chapter 4 as a hypotactic operator. Comparative constructions are closely related to deletion operations in our algorithm. In "John likes Mary more than Sue," the second proposition "John likes Sue" is somehow reduced to "Sue" during the transformation. Similarly, the synthesis of constructions "too" and "also" also involves deletion, but Casper does not handle such constructions yet. In this section, our algorithm is extended to handle the "respectively" and "each other" constructions, which are often analyzed by linguists together with coordinated conjunctions.

5.8.1 The "`respectively`" Construction

Our algorithm for generating "`respectively`" corresponds to McCawley's and Dougherty's observation: sentences with "`respectively`" can be analyzed as a conjunction of propositions. It is particularly suitable for transformational analysis.

(102) a. John and Phil kissed Mary and Sue, respectively. (subj, obj)

b. John and Phil hugged and kissed Mary, respectively. (subj, verb)

c. John hugged and kissed Mary and Sue, respectively. (verb, obj)

d. John and Phil hugged and kissed Mary and Sue, respectively. (subj, verb, obj)

e. *John and Phil kissed Mary, Sue, and Lisa, respectively.

The examples in (102) show that "respectively" sentences require a pairing of the entities in the conjoined constituents. Without one-to-one pairing, as in (102e), the resulting sentence is ungrammatical.

To generate a sentence with a "respectively" construction, the system has to ensure the following conditions:

- more than one role in the propositions contain distinct elements;
- all the entities in each conjoined role are distinct;
- the number of distinct elements is the same in each distinct role.

When these conditions are met, the system can generate one sentence for the propositions by generating surface expressions for each role. If a particular role contains distinct entities, a conjoined constituent is generated; if all the entities in a role are the same, only one of them will be generated. Of course, the order of the entities in the conjoined constituents must be preserved and cannot be changed across different roles. Once the surface expressions for all the roles are realized, then the system appends the expression "respectively" at the end of the sentence.

Because a "respectively" marker occurs at the end of the sentences, it does not have special intonation to alert listeners that they need to remember the entire sentence verbatim. As a result, van Oirsouw (1987) pointed out that listeners are almost systematically unable to answer a simple question like "Who seemed to be trying to kiss Mary?" after hearing :

> John, Bill, Harry and Paul seemed to be trying to kiss Harriet, Selina, Mary and Sue, respectively.

It is quite easy to figure out the answer if the sentence is written, but not if it is only spoken once. For this reason, "respectively" sentences are extremely rare in spontaneous speech and are usually restricted to no more than three conjoined constituents. The "respectively" construction is regarded as an optional stylistic rule applying to surface structure.

5.8.2 The "`each other`" Construction

Since "each other" is often used together with conjunction constructions, linguists interested in conjunction constructions have studied such constructions. Semantically, the "each other" construction is quite interesting. The propositions in (103a) can be transformed into (103b) by using coordinated conjunctions and "each other" constructions.

(103) a. John hit Paul.
Paul hit John.

b. John and Paul hit each other.

To derive sentences with "`each other`" from input propositions, we originally came up with a *complete directed graph criteria.* In a complete directed graph, each node in the graph is connected directly with every other node. In (103), there is a complete directed graph with John and Paul as nodes in the graph. When we apply this constraint to (104a), we realize that this requirement is too strict for any real-life situation involving more than four entities. To create a complete directed graph with n nodes, $n \times (n-1)$ propositions are needed. The criteria did not work well when given many people "hitting" one another; the constraint still says that these people are not hitting "each other" because there is no complete directed graph as in the examples in (104)

(104) a. John hit Paul.
Paul hit John.
Mark hit Paul.
John hit Mark.

b. John hit Paul.
Paul hit John.
Mark hit Paul.

c. John hit Paul.
Paul hit Mark.
Mark hit John.

d. John, Mark, and Paul hit each other.

Instead of using a complete graph, we relaxed the constraint for applying "`each other`" construction. To produce the sentence "John, Mark, and Paul hit each other," a reasonable semantics can be the set of entities in one distinct role is the same as the entities in another distinct role. When this constraint is satisfied as in (104a) and (104c), a conjoined constituent with distinct entities can be generated as the subject and "each other" as the object, as in (104d). In (104b), "Mark" did not appear in the object role; thus, it did not satisfy our new criteria. In such a case, the sentence "John and Mark hit Paul, and Paul John" is generated.

5.9 Summary

To produce coordinating conjunction constructions which express the intended meaning for the speaker, pragmatic, semantic, thematic, and syntactic information all play clear roles in the process:

- **pragmatic**: The pragmatics provide information to order conjoined constituents.
- **discourse**: The rhetorical relations between propositions provide the motivation for clause aggregation.
- **semantic** and **lexical**: Semantics provides information about object identity and sense.
- **syntactic**: The linear ordering of the constituents in the conjoined sentence determine the legal target sites for deletion.

The construction encompasses a content planner, a sentence planner, and a surface realizer. The content planner interacts with the sentence planner to determine the preferred ordering for conjoined constituents and specifies rhetorical relations between propositions. The sentence planner uses semantic and thematic role information to identify identical entities as candidates for deletion. The surface realizer, based on the recurring information provided by the sentence planner, decides which surface expressions are truly redundant based on their surface ordering and deletes them. Coordinating conjunctions are a remarkable complex linguistic phenomenon.

The current research examined the phenomenon of coordinating conjunctions and proposed an algorithm to create sentences with such constructions. Given a set of propositions, Casper can compute or predict grammatical sentences which can convey the same information that is more concise, fluent, and coherent by using coordinating conjunction constructions. Various linguistic constraints involved in the synthesis, such as constituent identity and directional constraints, were identified and discussed in detail. Most important, separate but related linguistic constructions—gapping, right-node-raising, and certain types of ellipsis—are all unified into one algorithm. Numerous examples have shown how our algorithm can generate complex coordinated constructions from clause-sized semantic representations. Both the representation and the algorithm have been implemented and used in two different text generation systems (McKeown, Kukich, and Shaw, 1994; McKeown et al., 1997).

Chapter 6

Quantification

This chapter describes how quantifiers can be generated in a text generation system. By using discourse and ontological information, quantified expressions can replace entities in a text, making it more fluent and concise. In addition to avoiding ambiguities between distributive and collective readings in the process of generating quantifiers, we will also show how different scope orderings between universal and existential quantifiers will result in different quantified expressions in our algorithm.

6.1 Introduction

To convey information concisely and fluently, text generation systems often perform opportunistic text planning and employ advanced linguistic constructions (Robin, 1995; Harvey and Carberry, 1998; Shaw, 1998b). But a system can also take advantage of quantification and ontological information to generate a concise reference to entities at the discourse level (Shaw and McKeown, 2000). For example, a sentence such as "The patient has an infusion line in each arm" is a more concise version of "The patient has an infusion line in his left arm. The patient has an infusion line in his right arm." Quantification is an active research topic in logic, language, and philosophy (Carpenter, 1997; de Swart, 1998). Since understanding systems requires as few interpretations as possible from the text, researchers have studied quantifier scope ambiguity extensively (Woods, 1978; Grosz et al., 1987; Hobbs and Shieber, 1987; Pereira, 1990; Moran and Pereira, 1992; Park, 1995). Research in quantification interpretation first transforms a sentence into predicate logic, then raises the quantifiers to the sentential level, and permutes these quantifiers to obtain as many readings as possible related to quantifier scoping. Finally, invalid readings are eliminated using various constraints.

Ambiguity in quantified expressions is caused by two main culprits. The first type of ambiguity involves the distributive reading versus the collective reading. In universal quantification, a referring expression refers to multiple entities. Potential ambiguity exists between whether the aggregated entities act individually (distributive) or act as one (collective). In the distributive reading, the sentence "All the nurses inspected the patient" implies that each nurse individually inspected the patient. In the collective reading, the nurses inspected the patient together as a group. The other ambiguity in quantification involves multiple quantifiers in the same sentence. The sentence "A nurse inspected each patient" has two possible scope orderings. In one of these, $\forall$patient$\exists$nurse, the universal quantifier $\forall$ has a wide scope, outscoping the existential quantifier $\exists$. This ordering means that each patient is inspected by a nurse who might not be the same in each case. In the other scope order, $\exists$nurse$\forall$patient, a single particular nurse inspected every patient. In both types of ambiguities, a generation system should make the desired reading clear.

Fortunately, the difficulties of quantifier scope disambiguation faced by the understanding community do not apply to text generation. For generation, the problem is the reverse: given an unambiguous representation of a set of facts as input, how can a quantified sentence be generated that unambiguously conveys the intended meaning? In this chapter, an algorithm is proposed which selects the appropriate quantified expression to refer to a set of entities using discourse and ontological knowledge. The algorithm first identifies the entities for quantification in the input propositions. Then, an appropriate concept in the ontology is selected to refer to these entities. Using discourse and ontological information, the system determines if quantification is appropriate and, if it is, which particular quantifier can be used to minimize the ambiguity between distributive and collective readings. More important, with multiple quantifiers in the same sentence, the algorithm generates different expressions for different scope orderings. The current study focused on generating referring quantified expressions for entities which were mentioned earlier in the discourse or were inferred from an ontology. Certain quantified expressions do not refer to particular entities in a domain or discourse, i.e., the negation in "The patient has *no allergies*" or a general statement like "All bears are mammals." The synthesis of such quantifiers is outside the scope of the dissertation.

The next section compares our approach with previous work in the generation of quantified expressions. The algorithm for generating universal quantifiers is detailed in Section 6.3, including how the system handles ambiguity between the distributive and collective readings. Section 6.4 describes how our algorithm generates sentences with multiple quantifiers.

6.2 Related Work

Because a quantified expression refers to multiple entities in a domain, our work can be categorized as referring expression generation (Dale, 1992; Reiter and Dale, 1992; Horacek, 1997). Previous work in this area did not directly address the generation of quantified expressions. The present study is interested in how to systematically derive quantifiers from input propositions, discourse history, and ontological information. Recent work on the generation of quantifiers (Gailly, 1988; Creaney, 1996; Creaney, 1999) follows the analysis viewpoint, extensively discussing scope ambiguities. Although the algorithm described in this chapter generates different sentences for different scope orderings, it does not achieve this through scoping operations as Gailly and Creaney did. Creaney also discussed various imprecise quantifiers, such as "`some`," "`at least`," and "`at most`."

Researchers (van Eijck and Alshawi, 1992; Copestake et al., 1999) proposed representations in a machine translation setting which allow underspecification with regard to quantifier scope. Our work is different in that we perform quantification directly on the instance-based representation obtained from database tuples. Our input does not have the information about which entities are quantified as in the case of machine translation, where quantifiers are already specified in the input from a source language.

6.3 Quantification Algorithm

We implemented our quantification algorithm as a part of MAGIC (Dalal et al., 1996; McKeown et al., 1997) sentence planner which contains a referring expression generation module, a clause aggregation module, and a lexical choice module. The sentence planner takes a set of rhetorically structured predicate-argument structures from the content planner and uses linguistic information to make decisions about how to convey the propositions fluently. Each predicate-argument structure is represented as a feature structure (Kaplan and Bresnan, 1982; Kay, 1979) similar to the one shown in Figure 6.1. The input to our quantification algorithm is a set of predicate-argument structures output by the referring expression module (Dale, 1992; Reiter and Dale, 1992) after it has selected the properties to identify the

```
((TYPE EVENT)
 (PRED ((PRED receive) (ID id1)))
 (ARG1 ((PRED patient) (ID PT1)))
 (ARG2 ((PRED aprotinin) (ID AP1)
 (MODS ((PRED after)) (id2)
       (TYPE TIME)
       (ARG2 ((PRED critical-point)
             (NAME intubation) (ID C1)))))
```

Figure 6.1: The predicate-argument structure of *"After intubation, a patient received aprotinin."*

entities, but without carrying out the assignment of quantifiers. Our quantification algorithm first identifies the set of distinct entities which can be quantified in the input propositions. A *generalization* of the entities in the ontology is selected to potentially replace the references to these entities. If universal quantification is possible, then the replacement is made and the system must select which particular quantifier to use. Our system has five realizations for universal quantifiers: "`each`," "`every`," "`all`," "`both`," and "`any`," and two for existential quantifiers: the indefinite article "`a`" and cardinal `n`.

The current study does not claim that the quantified expression and the conjoined entities are necessarily semantically equivalent. The quantified expression might provide more additional information than the conjunction of the entities. For example, knowing that "John ate all the apples" indicates that there is no apple left, while "John ate the three apples" does not contain such information. In this situation, the current system would generate the sentence with the universal quantifier because universal quantifiers have the more difficult constraint to satisfy, and thus are more informative than cardinal quantifiers. For a similar reason, when given a conjunction of entities and an expression with a cardinal quantifier, the system, by default, would use the conjunction. This default can be superseded by directives from the content planner which are application-specific.

6.3.1 Identify Thematic Roles with Distinct Entities

Our algorithm identifies the roles containing distinct entities among the input propositions as candidates for universal and existential quantification. Suppose the system is given two propositions similar to the one in Figure 6.1—"After intubation, Alice received aprotinin" and "After start of bypass, Alice received aprotinin"—each with four roles: PRED, ARG1, ARG2, and MODS-TIME. By computing similarity among entities in the same role, the system determines that the entities in ARG1, PRED, and ARG2 are identical in each role, and only the entities in MODS-TIME are different. Based on this result, the distinct entities in MODS-TIME, "after intubation" and "after start of bypass," are candidates for quantification.

6.3.2 Generalization and Quantification

We used the axioms in Figure 6.2 to determine if the distinct entities can be universally or existentially quantified. Although the axioms are similar to those used in the Generalized Quantifier (Barwise and Cooper, 1981; Zwarts, 1983; de Swart, 1998), the semantics of set X and set D are different. In the previous step, the entities in set X have been identified. To compute set D in Figure 6.2, we introduce a concept, Class-X. Class-X is a *generalization* of the distinct entities in set X. Quantification can replace the distinct entities in the propositions with a reference to their type as restricted by a quantifier, thereby accessing discourse and ontological information to provide a context. Our ontology is implemented in CLASSIC (Borgida et al., 1989) and is a subset of WordNet (Miller et al., 1990), with an online medical dictionary (Cimino et al., 1994) designed to support multiple applications across the medical institution. Given the entities in set X, queries to CLASSIC determine the class of each instance and its ancestors in the ontology. Based on this information, the generalization algorithm identifies Class-X by computing the most specific class which covers all the entities. Earlier work (Passonneau et al., 1996) has provided a framework for balancing specificity and verbosity in selecting appropriate concepts for generalization. However, given the precision needed in medical reports, our generalization procedure selects the most specific class.

Set D represents the set of instances of Class-X in a context. Our system currently computes set D for three different contexts:

- **discourse**: Previous references can provide an appropriate context for universal quantification. For example, if "Alice" and "Bob" were mentioned in the previous sentence, the system can refer to them as "both patients" in the current sentence.

- **domain model**: The domain model provides a closed world from which we can obtain set D by matching all instances of a concept in the knowledge base,

- '`both`': $|D - X| = 0$ and $|X| = 2$, can have collective reading
- '`every`', '`all`': $|D - X| = 0$ and $|X| > 2$, can have collective reading
- '`each`': $|D - X| = 0$ and $|X| \geq 2$, only distributive reading
- '`any`': $|D - X| = 0$, when interact with negation scope
- '`a`': $|D \cap X| > 0$ and $|X| = 1$
- `n` (cardinal): $|D \cap X| > 0$ and $|X| = n$

Figure 6.2: Axioms of quantifiers discussed in this chapter

such as "every patient." In addition, certain concepts in the ontology have limited types. For example, knowing that cell savers, platelets, and packed red blood cells are the only possible types of blood products in the ontology, the quantified expression "every blood product" can be used instead of referring to each entity.

- **domain knowledge**: The possessor of the distinct entities in a role might contain a maximum number of instances allowed for Class-X. For example, because a person has only two arms, the entities "the patient's left arm" and "the patient's right arm" can be referred to as "each arm."

When the computed set D matches set X exactly ($|D - X| = 0$), a quantified expression with either "`each`," "`all`," "`every`," "`both`," and "`any`" replaces the entities in set X.

6.3.3 Selecting a Particular Quantifier

In general, the universal quantification of a particular type of entity, such as "every patient," refers to all such entities in a context. As a result, readers can recover what a universally quantified expression refers to. In contrast, readers cannot pinpoint which entity has been referred to in an existentially quantified expression, such as "a patient" or "two patients." Because a universally quantified expression preserves the original semantics and is more concise than listing each entity, it is the focus of our quantification algorithm. The universal quantifiers implemented in our system include the five possible realizations of $\forall$ in English: "`each`," "`all`," "`every`," "`both`," and "`any`." The only existential quantifiers implemented in our system are the singular indefinite quantifier "`a`" and the cardinal quantifier `n`, such as `two`, `three`, and etc. These are used in sentences with multiple quantifiers and only when the entities being referred to do not have a proper name. A more developed pragmatic module is needed before quantifiers, such as "`some`," "`most`," "`at least`," and "`few`," can be systematically generated. Indiscriminate application of imprecise quantification can result in vague or inappropriate text in our domain, such as *"The patient received **some** blood products."* In our application, knowing exactly what blood products are used is very important. To avoid generating such inappropriate sentences, the system only performs generalization on the entities which can be universally quantified. If the distinct entities cannot be universally quantified, the system might realize these entities using coordinated conjunctions.

Once the system decides that a universally quantified expression can be used to replace the entities in set X, it must select a universal quantifier. Because our sentence planner opportunistically combines distinct entries from separate database entries for conciseness, it is not the case that these aggregated entities act together (the collective reading). Given such input, the referring expression for aggregated entities should have only a distributive reading.[1] The universal quantifier "`each`" always imposes a distributive reading when applied. Thus, it is the default universal quantifier in our algorithm. Of course, indiscriminate use of "`each`" can result in awkward sounding text. For example, the sentence "Every patient is awake" sounds more natural than "Each patient is awake." However, since quantified expressions with the universal quantifiers "`all`" and "`every`" can have collective readings (Vendler, 1967), our system generates "`all`" and "`every`" under two conditions when the collective reading is unlikely. First, if the proposition is a state as opposed to an event (Jackendoff, 1990), only a distributive reading is possible. The quantifier "`every`" is used in "Every patient had tachycardia" because the proposition contains the predicate *has-attribute*, an attributive relation. Second, when the concept being universally quantified is not an animated object. Since the universally quantified unanimated entities are unlikely to act collectively, the quantifier "`every`" will be used instead of "`each`." For example, the sentence "I tried every dish on the table" has a more natural sounding than "I tried each dish on the table." These quantifiers make the quantified sentences more natural because they do not pick out the redundant distributive meaning.

[1]For our system to generate noun phrases with collective readings, the quantification process must be performed at the content planner level, not in the clause aggregation module.

6.3.4 Examples of a Single Quantifier

Given the four propositions, "After intubation, Mrs. Doe had tachycardia," "After skin incision, Mrs. Doe had tachycardia," "After start of bypass, Mrs. Doe had tachycardia," and "After coming off bypass, Mrs. Doe had tachycardia," the algorithm first identifies roles with similar entities (ARG1, PRED, ARG2) and removes them from further quantification processing. Meanwhile, the distinct entities in the role MODS-TIME, "after intubation," "after skin incision," "after start of bypass," and "after coming off bypass," are further processed for universal quantification. The role MODS-TIME is further separated into two smaller roles, one role with the prepositions and the other role with different critical points. Since the prepositions are all the same, universal quantification is only applied to the distinct entities in set X, in this case, the four critical points. Queries to the CLASSIC ontology indicate that the entities in set X—"intubation," "skin-incision," "start-of-bypass," and "coming-off-bypass"—match all possible types of the concept `critical-point`, satisfying the domain model context in Section 6.3.2. Since set D and set X match exactly, generalization and universal quantification can be used to replace the references to these entities: "After **each** critical point, Mrs. Doe had tachycardia." The system currently does not perform generalization on entities which fail the universal quantification test. In such cases, a sentence with conjunction will be generated, i.e., "After intubation and skin incision, Mrs. Doe had tachycardia."

In addition to "`every`," the system generates "`both`" when the number of entities in set X is two. In our application, "`both`" is used in the following discourse context: "Alice had episodes of bradycardia before induction and start of bypass. In **both** episodes, she received Cefazolin and Phenylephrine."

When a universal quantifier is under the government of negation, "`each`," "`all`," "`every`," and "`both`" are inappropriate, and " "`any`" should be used instead. Given that the patient went on bypass without complications, the system should generate "The patient went on bypass without **any** problem." Our system currently uses "`any`" as a universal quantifier when the propositions contain negation in the predicate, such as "The patient denied any drug allergy," or when a negative preposition, such as "`without`," is used.

6.4 Generation of Multiple Quantifiers

With two distinct roles across the propositions, the algorithm tries to use a universal quantifier for one role and an existential quantifier for another. We intentionally ignore cases where two existential quantifiers or two universal quantifiers are generated in the same sentence. The sentence with $\exists\exists$ quantifiers is too vague and inappropriate for our medical report generation system. For sentences with $\forall\forall$, the likelihood for such input to a text generation system is slim; we did not encounter such cases in MAGIC.

We differentiate between cases where there is or is no dependency between the two distinct roles. Two roles are independent of each other when one is not a modifier of the other. For example, the roles ARG1 and ARG2 in a proposition are independent. In "Each patient is given a high severity rating," performing universal quantification on the patients (ARG2) is a separate decision from the existential quantification of the severity ratings (ARG3). Similarly, in "An abnormal lab result was seen in each patient with hypertension after bypass," the quantification operations on the abnormal lab results and the patients can be performed independently.

When the roles being quantified are not independent, the quantification process of each role might interact because modifiers restrict the range of the entities being modified. We found that when universal quantification occurs in the MODS role, the quantification of PRED and MODS can be performed independently, just as in cases without dependency. Given the input propositions "Alice has IV-1 in Alice's left arm. Alice has IV-2 in Alice's right arm," the distinct roles are ARG2 "IV-1" and "IV-2," and ARG2-MODS "in Alice's left arm" and "in Alice's right arm." The ARG2-MODS is universally quantified based on domain knowledge that a patient is a human and a human has a left arm and a right arm. In this example, universal and existential quantifications are independent. But in 'Every patient with a balloon pump had hypertension," the existentially quantified expression "with a balloon pump" is a restrictive modifier of its head. In this case, set D does not include all the patients, but only the patients "with a balloon pump." When computing set D for universal quantification, the algorithm takes this extra restriction into account by eliminating all patients without such a restriction. Once a role is universally quantified and the other is existentially quantified, our algorithm replaces both roles with the corresponding quantified expressions. Figure 6.3 shows sentences with multiple quantifiers that were generated by applying our algorithm.

6.5 Ambiguity Revisited

Section 6.3.3 described ways to minimize the ambiguity between distributive and collective readings when generating universal quantifiers. What about scope ambi-

- Roles without dependency, $\forall$ Role-1,$\exists$ Role-2
  Each patient is given a high severity rating.
- Roles without dependency, $\exists$ Role-1, $\forall$ Role-2
  An abnormal lab result was seen in each patient with hypertension after bypass.
- Roles with dependency, $\forall$ PRED, $\exists$ MODS
  Every patient with a balloon pump had hypertension.
- Roles with dependency, $\exists$ PRED, $\forall$ MODS
  Alice has an IV in each arm.

Figure 6.3: Sentences with multiple quantifiers.

guity when the same sentence has multiple quantifiers? If we look at the roles which are being universally and existentially quantified in our examples in Figure 6.3, the universal quantifiers always have wider scope than the existential quantifiers. In the first example, the scope order is $\forall$patient$\exists$high-severity-rating, the second example is $\forall$patient$\exists$lab-result, the third is $\forall$patient$\exists$balloon-pump, and the fourth is $\forall$arm$\exists$IV. The scope orderings are all $\forall\exists$.

What happens if a sentence contains an existential quantifier which has a wider scope than a universal quantifier? In "A doctor operated on each patient," the normal reading is $\forall$patient$\exists$doctor. But if the existentially quantified noun phrase "a doctor" refers to the same doctor, as in $\exists$doctor$\forall$patient, the system would generate "(A particular/The same) doctor operated on each patient." In an applied generation system, the input for such a sentence is likely to be "Dr. Rose operated on Alice," "Dr. Rose operated on Bob," and "Dr. Rose operated on Chris." Given these three propositions, the entities in ARG1 and PRED are identical, and only the distinct entities in ARG2 ("Alice," "Bob," and "Chris"), will be quantified. With an appropriate context, the sentence "Dr. Rose operated on each patient" will be generated. If the name of the doctor is not available but the identifiers for the doctor entities across the propositions are the same, the system will generate "The same doctor operated on each patient." As this example indicates, when $\exists$ has a wider scope than $\forall$, the first step in our algorithm—identifying roles with distinct entities—would eliminate the roles with identical entities from further quantification processing. Based on our algorithm, sentences with $\exists\forall$ readings are taken care of by the first step, identifying roles with distinct entities, while $\forall\exists$ cases are handled by quantification operations for multiple roles, as described in Section 6.4.

6.6 Summary

Though imprecise quantifiers such as "`few`," "`many`," and "`some`" are not addressed, the current study is an important advance in synthesizing quantified expressions. We have succeeded in making the text more concise while preserving the original semantics in the input propositions. We have described an algorithm which systematically derives quantifiers from input propositions, discourse history, and ontological information. We identified three types of information from ontology and discourse to determine if a universal quantifier can be applied. We also minimized the ambiguity between distributive and collective readings by selecting the appropriate universal quantifier. Most important, for multiple quantifiers in the same sentence, we have shown how our algorithm generates different quantified expressions for different scope orderings.

Chapter 7

Ordering Among Aggregation Operators

7.1 Introduction

The previous three chapters described referential operators (i.e., universal quantifiers), paratactic operators (i.e., conjunctions and related ellipsis constructions) and hypotactic operators (i.e., relative clauses, prepositional phrases, and adjective phrases). As a general planning task in AI, the sequential ordering of applying multiple operators is an issue because aggregation operators are not commutative – applying one of the operators to the input propositions might prevent applications of others. For example, if relative clause operators are applied before adjective or prepositional phrase operators, there wouldn't be any propositions left for adjective or prepositional phrase operators to combine since there are less restrictions for applying relative clause operators. In addition, depending on the ordering of applying the aggregation operators, different meanings might result. In Example (105a), the first two propositions are linked by a ADDITION relation and the second and third propositions are linked by a CONTRAST relation. Applying the hypotactic operator first before conjunction, Sentence (105b) is produced. In this sentence, the modifying proposition only modifies the proposition "John ate apples" and not "John drank orange juice." Applying the operator in the reverse ordering, Sentence (105c) results. In this case, the modifying proposition has a wide scope and modifies both propositions.

(105) a. John drank orange juice.
John ate apples.
(even though) John did not like fruits.

b. John drank orange juice and even though he did not like fruits, he ate apples.

c. Even though John did not like fruits, he drank orange juice and ate apples

Clearly, the ordering of operators can have an impact on the meaning of the aggregated sentences. This chapter explores the interactions between aggregation operators and uses a corpus-based approach to evaluate a specific ordering of these operators based on our understanding of their characteristics.

Before incorporating the operators analyzed in this thesis into a natural language generation system, researchers and developers need to know how these operators interact and the appropriate order for applying them. Without such information, the grammaticality and fluency of the generated sentence might not be guaranteed. In this chapter, we will focus mainly on the ordering among the three types of aggregation operators studied in this thesis: quantification operators, paratactic operators, and hypotactic operators. Since paratactic operators can be divided into (1) simple conjunction and (2) complex conjunction, and hypotactic operators can be divided into (1) adjective, (2) prepositional phrase, and (3) reduced relative clause (this also includes apposition) and (4) relative clause operators, there are seven operators to be ordered. Theoretically, if these operators can only be applied once, then there are 5040 (or 7!) ways to order these operators. As it turns out, operators in each type of clause aggregation can be applied together one after another as a group. By treating each type of aggregation as one single generalized operator, only three groups (quantifications, paratactic operators, and hypotactic operators) of operators are to be sequentialized. In this scenario, the possible ordering among them is six, a much smaller search space. Attempts to apply all paratactic operators before hypotactic ones or vice versa for different sets of input propositions did not work well. There seem to be other factors which are not captured in such simple operator orderings. In particular, it seems that some operators might be applied more than once to input propositions. This situation invalidates the above permutation analysis, and the search space is much larger.

The current analysis focuses on the operator ordering among different types of operators. When the same operator is applied multiple times to obtain aggregated constituents, such as a sequence of adjectives, the ordering decision among the same operator is outside the scope of the current annotation effort. Chapter 4 addresses such linearization issue using a different approach. Among different type of hypotactic operators, we assumed that the operators which are applied earlier should be closer to the head than the constituent results from operators applied later. For example, if the reduced relative clause operator is applied before the relative clause operator, at the surface level, the reduced relative clause will appear closer to the head than the relative clause.

Section 7.2 describes the corpus-based methodology used to evaluate the effectiveness of our proposed sequential ordering of the aggregation operators. The corpus-based approach requires that complex human-written sentences be de-aggregated manually and annotated with rhetorical relations and other information related to clause aggregation. The markup language used for this annotation is described in Section 7.3.1, with a complex example sentence analyzed and annotated in Section 7.3.3. The guidelines for performing the annotation are in Appendix A. Section 7.4 presents the result of the analysis and evaluates our proposed ordering based on the annotated corpus.

7.2 Methodology

In earlier chapters, we have defined a set of aggregation operators and would like to find a sequential ordering among them so that NLG researchers and system builders can incorporate them into various NLG systems. Humans perform clause aggregation tasks well, as demonstrated by the complex sentences in literature and news articles. Since how human perform these tasks is still a mystery and cannot be examined directly, we run into the same problem faced by linguists. The concrete evidence that we have is the sentences written by humans which contain only the result of the linguistic process, not the process itself. Besides, our focus is not that the text generation systems perform the generation tasks as humans do, but rather, the NLG system should be able to generate sentences with similar complexity as humans. If we can identify an ordering which allows computers to produce many of the sentences in human-written text, then we have succeeded.

Section 7.1 mentioned that some aggregation operators seem to be applied multiple times throughout the whole aggregation process, so we cannot exhaustively permute the operators to find all possible sequential orderings and then figure out which one provides the best result. Instead of finding the optimal ordering, the current approach will show that our proposed ordering works well in reconstructing human-written sentences. Given a specific aggregation operator ordering, an evaluation will be performed using a manually annotated corpus to determine if applying the operators in the specific order can reconstruct the original sentences. The original ordering of aggregation operators implemented in MAGIC is given below:

1 Quantification
2 Simple conjunction
3 Adjective
4 Prepositional phrase
5 Reduced relative clause
6 Relative clause
7 Simple conjunction
8 Complex conjunction
9 Transformations for other rhetorical relations

This ordering is similar to, but not exactly the same as the operator ordering used in the current evaluation:

1 Quantification
2 Adjective (maybe with conjunction)
3 Prepositional phrase (maybe with conjunction)
4 Reduced relative clause, including apposition (maybe with conjunction)
5 Relative clause (maybe with conjunction)
6 Transformations for other rhetorical relations (maybe with conjunction)
7 Simple conjunction
8 Complex conjunction

If such an ordering works well for the annotated corpus, then researchers can be confident that their NLG systems using such an ordering will work well. The rationale for using the revised ordering is provided in Section 7.5.

In our evaluation, a special corpus was used. To increase the chance of encountering sentences that underwent both paratactic and hypotactic transformations, only sentences contain the conjunctor "and" are selected for annotation. Sentences containing the word "and" already used paratactic transformations and might have used other aggregation operators as well. Since our goal is to analyze the ordering between hypotactic and paratactic operators, these sentences are more likely to contain both types of operators than those without. By intentionally making the set of sentences to be analyzed more complex, it is more likely that evidence either supporting or negating our proposed ordering will be found. The current analysis uses the medical and financial corpus use for the adjective ordering analysis in Chapter 4. Due to the amount of effort needed to annotate the complex sentences, only one hundred sentences from each domain are analyzed. During the annotation stage, several tasks are performed:

1. **De-aggregate original sentence**: the original sentences are broken down into smaller propositions. This is basically an ellipsis recovering process.

2. **Specify rhetorical relations between propositions**: identify the rhetorical relations between the de-aggregated propositions. This is necessary because sentences are aggregated based on the fact that they are related pragmatically or rhetorically.

3. **Specify a sequence of transformation operators to combine de-aggregated propositions back into the original sentence**. This is used to evaluate the applicability of our proposed ordering.

For each annotated sentence, if the sequence of the transformation operators does not violate our proposed ordering, it is considered as positive evidence supporting our proposed ordering. If the sequence of transformations used violate our proposed ordering, it is a negative evidence. After tallying the number of positive and negative evidence, we will know how well the proposed ordering works. The result is presented in Section 7.4.

7.3 The Annotation

Most of the effort in this analysis went into the annotation of the corpus, which is quite complex. We will first describe the markup language used for annotation in Section 7.3.1. Section 7.3.2 will provide details on the concept of *proposition set* or *propset*, which was identified as a useful device to facilitate the annotation effort during the early exploratory stage of the annotation effort. Section 7.3.3 contains an extended example which demonstrates the whole annotation procedure. A guideline for performing the annotation is presented in Appendix A.

7.3.1 The Markup Language

The de-aggregated sentences are annotated using an XML-like notation (Prescod and Goldfarb, 1999). This representation is chosen because such a representation is easy for humans to understand and also simple for computer programs to process. Scripting code for extracting all the rhetorical relations annotated in the corpus can be easily written. Each sentence entry consists of five parts. The first part is the original sentence. The second part is a list of de-aggregated propositions after manual reconstruction of the ellided constituents. These propositions are enclosed in a propset[1] , which might contain nested propsets. The third section specifies the rhetorical relations which linked the de-aggregated propositions or propsets.

[1]The concept of propset is discussed in Section 7.3.2.

The number of rhetorical relations in a sentence entry is always one less than the number of propositions. The fourth section is a sequence of transformations that can be applied to reconstruct the original sentence from the de-aggregated propositions. The fifth section contains annotator comments. One of them, `seqordering` tag, indicates whether the sequence of the transformations in the transformation annotation section violates or adheres to the proposed aggregation operator ordering. The `conj` tag indicates whether a conjunctor "and" in the original sentence contains a collective or distributive reading. An example is taken from the corpus to illustrate the annotation.

```
<sentence id=s32>
  Local sports fans themselves, long known for their passive
  demeanor at games and propensity to leave early, don't resist
  the image.
  <propset id=pset32-1>
    <prop id=p32-1>
      Local sports fans don't resist the image.    </prop>
    <prop id=p32-2>
      Local sports fans are long known for their passive
      demeanor at games.    </prop>
    <prop id=p32-3>
      Local sports fans are long known for their propensity to
      leave early.    </prop>
  </propset>
  <focus entity='local sports fans'/>
  <rst-rel id=r32-1 name=elab
           nuc=p32-1   sat=p32-2   ref=no  />
  <rst-rel id=r32-2 name=elab
           nuc=p32-1   sat=p32-3   ref=no  />
  <trans id=tx32-1 name=conj-simp   nuc=p32-2 sat=p32-3 />
  <trans id=tx32-2 name=rel-reduced-del-wh-be nuc=p32-1 sat=tx32-1 />
  <seqorder valid=true />
  <conj id=c32-1 type=dist />
</sentence>
```

In this example, the original sentence is broken into three propositions, with proposition p32-2 and p32-3 modifying p32-1 with ELABORATION relation, as indicated by rhetorical relations r32-1 and r32-2. Two transformations are applied to the propositions to reconstruct the original sentence. After the first simple conjunction transformation, the propositions became the following:

```
<prop id=p32-1>
  Local sports fans don't resist the image. </prop>
<prop id=tx32-1>
  Local sports fans are long known for their passive
  demeanor at games and propensity to leave early. </prop>
```

with <prop id=tx32-1> contains the combined result of applying transformation simple conjunction to proposition <prop id=p32-2> and <prop id=p32-3>. The combined result <prop id=tx32-1> undergoes further transformation in <trans id=tx32-2> as a satellite proposition to nucleus proposition <prop id=p32-1>. The reduced relative clause transformation deletes "who" and "be" and the original sentence is reproduced:

```
<prop id=tx32-2>
  Local sports fans, long known for their passive
  demeanor at games and propensity to leave early,
  don't resist the image. </prop>
```

In the annotated corpus, the resulting sentence after each transformation step is not recorded, as we have done here. Otherwise, the annotated document would be several times the current size. We do not believe such effort would have a noticeable impact on the result. Please note that the order of the transformations listed is based on the proposed ordering proposed in the Section 7.2.

7.3.2 The Concept of Proposition Set or *Propset*

In our preliminary effort to annotate the selected complex sentences, we run into several problems. Simply specifying rhetorical relations among the de-aggregated propositions did not seem to provide enough information to reproduce the original sentence. For example, the propositions in <propset id=pset1-1> below can be realized as either Sentence (106a) or (106b) depending on whether the hypotactic operator or the conjunction operator is applied first. In <propset id=pset1-1>, the first two propositions are linked by a ADDITION relation and the second and third propositions are linked by a CONTRAST relation.

```
<sentence id=s2>
    <propset id=pset1-1>
      <prop id=p1-1>
        John drank orange juice.  </prop>
      <propset id=pset1-1>
        <prop id=p1-2>
          John ate apples.  </prop>
        <prop id=p1-3>
          (even though) John did not like fruits.  </prop>
      </propset>
    </propset>

    <focus entity='John'/>
    <rst-rel id=r1-1 name=join
```

```
            nuc=p1-1   sat=pset1-2   ref=no  />
    <rst-rel id=r1-2 name=contrast
            nuc=p1-2   sat=p1-3   ref=no  />
</sentence>
```

(106) a. John drank orange juice and even though he did not like fruits, he ate apples.

b. Even though John did not like fruits, he drank orange juice and ate apples.

In Sentence (106a), the proposition p1-3 only modified proposition p1-2, not p1-1. The join relation between the p1-1 and p1-2 describes merely events that happened, and one of the event is contrasted by the fact "John did not like fruits." While in Sentence (106b), the proposition p1-3 has a wide scope and modifying both propositions p1-1 and p1-2. To clarify the scope of such modifying construction, we came up with the concept of *proposition set*, or *propset* which facilitates the specification of the scope of modifying proposition, as shown below in two specifications using the markup language defined in Section 7.3.1. The first case specified a narrow scope for the modifying proposition, as in Sentence (106a):

```
<sentence id=s2>
    <propset id=pset1-1>
      <prop id=p1-1>
        John drank orange juice.  </prop>
      <propset id=pset1-1>
        <prop id=p1-2>
          John ate apples.  </prop>
        <prop id=p1-3>
          (even though) John did not like fruits.  </prop>
      </propset>
    </propset>
    <focus entity='John'/>
    <rst-rel id=r1-1 name=join
            nuc=p1-1   sat=pset1-2   ref=no  />
    <rst-rel id=r1-2 name=contrast
            nuc=p1-2   sat=p1-3   ref=no  />
</sentence>
```

To specify that a modifying proposition modifies both p1-1 and p1-2, the concept of propset is used to group the two propositions before they are jointly modified by p1-3:

```
<sentence id=s1>
    <propset id=pset1-1>
```

```
      <propset id=pset1-2>
        <prop id=p1-1>
          John drank orange juice.  </prop>
        <prop id=p1-2>
          John ate apples.  </prop>
      </propset>
      <prop id=p1-3>
        (even though) John did not like fruits.  </prop>
    </propset>
    <focus entity='John'/>
    <rst-rel id=r1-1 name=join
             nuc=p1-1   sat=p1-2   ref=no  />
    <rst-rel id=r1-2 name=contrast
             nuc=pset1-2   sat=p1-3   ref=no  />
</sentence>
```

As a result of making the scope clear in the de-aggregated proposition, CASPER can generate either Sentence (106a) or (106b) and ensures correct scoping of modifier is conveyed.

Incorporating the concept of propset into annotation provided several benefits:

- **Specifying certain propositions are more tightly related**. For example, events related to a patient's smoking habit, such as "he was a smoker" and "he quit 10 years ago", are grouped together in a propset. As a result of such specification, the propositions in a propset will be treated as one proposition. The system will combine them first before aggregating the combined proposition with other propositions.

- **Simplifying the annotation process for certain constructions**. Information contained in the embedded S-structure of verbs like "said" or "believe" can be extracted and analyzed as a propset. Without using such a mechanism, the subject and verb of the main clause would appear multiple times in the de-aggregated propositions. By eliminating such recurrence of the same main subjects and verbs, aggregation analysis and aggregation operators are simplified. The transformations which extract the embedded S-structure are annotated as "arg" transformations. They are just cosmetic artifacts and unlikely to have any impact on the analysis of the ordering of the operators. The first aggregation operation after combining all the propositions in a propset of an embedded S-structure is always "arg" transformation, which attaches the aggregated embedded S-structure to the main subject and verb.

- **Minimizing redundant specification of multiple modifying rhetorical relations**. When a proposition modifies multiple propositions at the same time, the propositions being modified can be grouped under a propset so that only one rhetorical relation need be specified between the modifying proposition and the propset being modified. If the concept of propset is not available, one rhetorical relation needs to be specified between the modifying proposition and each of the propositions being modified. Normally, other than the nucleus proposition itself, each new proposition is linked to the nucleus proposition directly or indirectly through a single rhetorical relation to create a connected graph (which make a sentence cohesive). But since in the case of a subordinate proposition modifying multiple propositions, the number of rhetorical relations for de-aggregated propositions can be larger than the number of propositions. In the above example, if the proposition "(even though) John does not like fruits" has wide scope, without the use of propset, there would be two contrast rhetorical relations, each attaching to p1-1 and p1-2. There would be a total of four rhetorical relations, not just three. Since transformation operators are directly related to rhetorical relations, extra or redundant specifications of rhetorical relations would introduce complications to the implementation of the aggregation operators.

- **Minimizing scope ambiguity**. The earlier example, "John ate apples even though he did not like fruits, and he drank orange juice" illustrates this point well. By using propset, the scope of the modifying proposition is made explicit.

The ability to eliminate redundant specification of multiple rhetorical relations is particularly important because it makes one transformation operator corresponds to one rhetorical relation directly or indirectly. This simplification makes the aggregation task more manageable.

7.3.3 An Illustrative Example

In Section 7.3.1, a simple example was used to describe the tags use for annotation. In this section, we will look at a more complex example which also includes propsets.

```
<sentence id=s99>
  As above, status post left hemispheric cerebrovascular
  accident fifteen years ago with residual right hemiparesis and
  expressive aphasia, seizure disorder following the
```

```
  cerebrovascular accident, former smoker quit ten years ago.
    <propset id=pset99-1>
      <prop id=p99-1>
        (As above) the patient has status post left hemispheric
        cerebrovascular accident fifteen years ago.    </prop>
      <prop id=p99-2>
        The accident caused residual right hemiparesis.    </prop>
      <prop id=p99-3>
        The accident caused expressive aphasia,    </prop>
      <prop id=p99-4>
        The patient had seizure disorder following the
        cerebrovascular accident.    </prop>
      <propset id=pset99-2>
      <prop id=p99-5>
        He is a former smoker. </prop>
      <prop id=p99-6>
        He quit ten years ago.  </prop>
      </propset>
    </propset>
    <focus entity='-X-'/>
    <rst-rel id=r99-1 name=elab
             nuc=p99-1   sat=p99-2   ref=no  />
    <rst-rel id=r99-2 name=elab
             nuc=p99-1   sat=p99-3   ref=no  />
    <rst-rel id=r99-3 name=elab
             nuc=p99-1   sat=p99-4   ref=no  />
    <rst-rel id=r99-4 name=elab
             nuc=p99-1   sat=pset99-2   ref=no  />
    <rst-rel id=r99-5 name=elab
             nuc=p99-5   sat=p99-6   ref=no  />
    <trans id=tx99-1 name=rel-reduced-del-wh   nuc=p99-5 sat=p99-6 />
    <trans id=tx99-2 name=conj-simp   nuc=p99-2 sat=p99-3 />
    <trans id=tx99-3 name=pp-with     nuc=p99-1 sat=tx99-2 />
    <trans id=tx99-4 name=conj-mult   nuc=p99-4 sat=tx99-1 />
    <trans id=tx99-5 name=rel-reduced-del-wh-verb nuc=tx99-3 sat=tx99-4 />
    <seqorder valid=true />
    <conj id=c99-1 type=dist />
</sentence>
```

In this example, the two propositions about smoking are grouped as a propset because of their close relation. Most of the other propositions are satellite propositions modifying the first proposition through the ELABORATION relation. Below is a step-by-step description of each transformation.

1. **Applying the transformation operators to nested propset first.** Proposition p99-5 and p99-6 are combined using a reduced relative clause transformation with deleted "who" (rel-reduced-del-wh).

```
<prop id=tx99-1>
He is a former smoker quit ten years ago. </prop>
```

2. **Apply conjunction operators to similar modifying propositions at the top level which modify the same entity in their nucleus proposition.**

   Notice that the rhetorical relations between proposition p99-2 and p99-1, and proposition p99-3 and p99-1 are both ELABORATION, but because they both modify p99-1 and their structure are similar, 1-distinct conjunction is applied to these propositions even though they are satellite propositions of ELABORATION relations.

   ```
   <prop id=tx99-2>
   The accident caused residual right hemiparesis and expressive
   aphasia   </prop>
   ```

3. **Apply hypotactic PP operator.** The previous paratactic operator combines two satellite propositions of ELABORATION relations. Now, the combined result should be attached to the nucleus proposition before any other operators are applied. The transformation of the "caused" predicate into "with" is a lexical or domain specific transformation.

   ```
   <prop id=tx99-3>
   The patient has status post left hemispheric cerebrovascular
   accident fifteen years ago with residual right hemiparesis and
   expressive aphasia.  </prop>
   ```

4. **Apply paratactic operators to combine the next two propositions which can be realized as reduced relative clauses.** Since both p99-4 and tx99-1 are somewhat similar and both modify proposition p99-1, they are combined using the conjunction operator first before hypotactic transformation is applied in the next step.

   ```
   <prop id=tx99-4>
     He has seizure disorder following the cerebrovascular accident
     and is a former smoker quit ten years ago.    </prop>
   ```

5. **Apply reduced relative clause to combine the previous proposition tx99-4 to the nucleus proposition tx99-3**, but with deleted pronoun and verb deleted in tx99-4. The use of the verbs "be" and "have" are very common in this domain. As a result, they both are deleted although they are not the same verb. Putting proposition <prop id=tx99-3> and <prop id=tx99-4> together,

```
<prop id=tx99-3>
  The patient has status post left hemispheric
  cerebrovascular accident fifteen years ago with residual
  right hemiparesis and expressive aphasia.  </prop>
<prop id=tx99-4>
  [He has] seizure disorder following the cerebrovascular
  accident and [he is] a former smoker quit ten years ago.
  </prop>
```

the original sentence is reproduced:

```
<prop id=tx99-5>
  The patient has status post left hemispheric cerebrovascular
  accident fifteen years ago with residual right hemiparesis
  and expressive aphasia, seizure disorder following the
  cerebrovascular accident, and a former smoker quit ten years
  ago. </prop>
```

Although the result is not 100% the same as the original sentence, they are very similar. Many sentences with similar complexity were present in the corpus and they have been analyzed in similar fashion. One interesting aspect of this analysis is the use of paratactic operators to combine modifying propositions before they are attached to the nucleus proposition. By first performing paratactic operations on the modifying propositions that have similar structures and modify the same entity in the nucleus proposition, the annotator avoids applying hypotactic transformations multiple times. One clear example of this phenomenon is the combining of propositions p99-2 and p99-3 to the nucleus proposition p99-1. In our analysis, only two transformations are applied, with the conjunction transformation first followed by PP transformation. If hypotactic operators are applied first followed by conjunction, then there would be three transformation operations – two PP transformation followed by a conjunction to insert the "and" conjunctor. The current approach eliminates extra PP transformation operation and produces the same result.

7.4 Evaluation

After specifying the transformation operators for 200 sentences in the corpus according to our proposed sequential ordering, the majority of the sentences can be resynthesized from the de-aggregated propositions using our ordering of aggregation operators (195 out of 200). The percentage is quite high because the incorporation of *propset* in the annotation takes care of many cases which would have violated our proposed ordering. This result provides evidence supporting our claim that the proposed ordering is effective. By ensuring the content planner can specify propositions, propset, and rhetorical relations as the markup provided in the annotated corpus, researchers incorporating our proposed ordering of aggregation operators in their generation systems can be assured that the complex sentences resulting from the clause combining operations will be concise and grammatical.

The full annotated corpus is provided in Appendice C and D. In the annotated corpus, excluding "arg" transformations which are not involved in the ordering of aggregation operators, there are 523 transformations used, roughly 2.6 transformations for each sentence. Of the 523 transformations, 417 (80%) of them are transformations studied and analyzed in the current work, shown in Figure 7.1, while 106 (20%) of them are not implemented at all, shown in Appendix B. The transformations which were not implemented in Casper include "or," parenthesis, using "with" for paratactic operation, or any transformations which involve extraction. In the analysis, we did not remove sentences containing transformations which Casper does not handle. Doing so would eliminate many complex sentences containing multiple hypotactic and paratactic transformations — the type of sentence appropriate for our analysis. Instead, such implemented transformations are categorized as either hypotactic and paratactic, and they are mapped to the closest type of transformations when we evaluate if such transformations violate our proposed ordering. Since the sentences selected for analysis are not a random sample because all of them contains the word "and," this bias might create a tendency to select sentences with transformations that Casper handles well, such as paratactic transformation (62% of the transforms analyzed contains conjunctor "and"). Given that our goal was to find as much interaction between hypotactic and paratactic operators as possible, this bias is reasonable. Because of such a bias and more importantly, a sample size of 200 sentences is very small, it is difficult to reliably quantify the percentage of the clause aggregation transformations in English that Casper currently handles. Figure 7.1 shows a list of the operators handled by Casper. Some of these transformations are the same, such as conj-mult, conj-mult-gap, conj-mult-vp, or conj-mult-sent. They are all of the same paratactic operator, but when performing annotation, the author chose to mark the different syntactic forms just in case others might find such information useful. It is always easier to collapse detailed categories into coarser ones than going the other way.

In our analysis of rhetorical relations, excluding "arg" relations, there are 20 different types of rhetorical relations identified in the corpus shown in Figure 7.2, with a total of 523 rhetorical relations. The interesting one for our analysis are

10	`adj`
16	`apposition`
40	`arg`
32	`conj-mult`
1	`conj-mult-gap`
1	`conj-mult-passive`
19	`conj-mult-sent`
29	`conj-mult-vp`
203	`conj-simp`
3	`conj-simp-neg`
6	`conj-simp-nested`
2	`pp-as-np`
1	`pp-between`
2	`pp-for`
1	`pp-from`
4	`pp-in`
1	`pp-like`
5	`pp-of`
1	`pp-on`
3	`pp-to`
9	`pp-with`
1	`pp-without`
2	`prenominal`
7	`prenominal-title`
1	`rel-reduced-del-wh`
24	`rel-reduced-del-wh-be`
11	`rel-reduced-ing`
5	`rel-that`
16	`rel-wh`

Figure 7.1: Types of transformation handled by CASPER in the annotated corpus.

```
  2 alternate
 41 arg
  2 circum-location
  1 circum-purpose
 11 circum-time
  1 comparative
  5 concession
  1 concurrent
  4 condition-except
  2 condition-if
 12 contrast
  1 disj-simp-neg
215 elab
  1 evaluation
  9 evidence
210 join
  2 join-collective
 20 non-volitional-cause
  2 non-volitional-result
  6 purpose
 15 sequence
```

Figure 7.2: Rhetorical relations used to annotated the corpus.

Elaboration(elab), Addition(join), and Sequence(sequence). Together, these three rhetorical relations made up of 440 of 523 rhetorical relations. Except for a few of them (e.g., join-collective, alternative, and comparative), the other rhetorical relations are hypotactic in nature. These subordinate rhetorical relations are usually realized using relative clause transformation or no transformation at all.

Of the 5 failed cases, two of them (c42 and c60 in medical corpus) involved application of relative clause transformation (rel-wh) before reduced relative clause with -ing (rel-reduced-ing). The full annotated sentence, c42, is shown below. Casper assumes that hypotactic operators that are applied earlier should create constituents closer to the head they modifies, while the constituents in the original sentences indicated that rel-wh transformations are applied before rel-reduced-ing, which violates our proposed ordering. The other failed cases involved realizing join relations using hypotactic constructions (c43 and c63 in WSJ corpus) or realizing elaboration relations using paratactic construction (c90 in medical corpus). Since these transformations are not the expected transformation operators for the rhetorical relations, they violated our proposed ordering. Overall, these failed cases are rare.

```
<sentence id=c42>
  The patient was a 38-year-old woman from the Dominican
  Republic who presented to the Cardiology Clinic in 11/90
  complaining of dyspnea on exertion and palpitations.
    <propset id=pset42-1>
      <prop id=p42-1>
        The patient was a woman.   </prop>
      <prop id=p42-2>
        The patient was a 38-year-old. </prop>
      <prop id=p42-3>
        The patient is from Dominican Republic.  </prop>
      <prop id=p42-4>
        The patient presented to the Cardiology Clinic in 11/90.
         </prop>
      <prop id=p42-5>
        The patient complained of dyspnea on exertion.  </prop>
      <prop id=p42-6>
        The patient complained of palpitations.
        </prop>
    </propset>
    <focus entity='the patient'/>
    <rst-rel id=r42-1 name=elab
             nuc=p42-1   sat=p42-2   ref=no  />
    <rst-rel id=r42-2 name=elab
             nuc=p42-1   sat=p42-3   ref=no  />
    <rst-rel id=r42-3 name=elab
             nuc=p42-1   sat=p42-4   ref=no  />
    <rst-rel id=r42-4 name=elab
             nuc=p42-1   sat=p42-5   ref=no  />
    <rst-rel id=r42-5 name=elab
             nuc=p42-1   sat=p42-6   ref=no  />
    <trans id=tx42-1 name=adj   nuc=p42-1 sat=p42-2 />
    <trans id=tx42-2 name=pp-from   nuc=tx42-1 sat=p42-3 />
    <trans id=tx42-3 name=rel-wh   nuc=tx42-2 sat=p42-4 />
    <trans id=tx42-4 name=conj-simp   nuc=p42-5 sat=p42-6 />
    <trans id=tx42-5 name=rel-reduced-ing   nuc=tx42-3 sat=tx42-4 />
    <seqorder valid=false />
    <conj id=c42-1 type=dist />
    <comment text="- rel-wh is applied before rel-reduced-ing, if
                  the ordering between same class of construction
                  does not matter, such as adj-order, then this is
                  not a violation." />
</sentence>
```

7.5 Rationale for the Proposed Ordering

Before the current annotation effort was underway, the ordering of operators used in CASPER were paratactic operators, hypotactic operators, and then paratactic operators again. It was not clear why paratactic operators are applied multiple times while hypotactic operators only once. It was not clear if there were different types of join relations connecting the propositions which resulted in multiple applications of paratactic operators.

After some preliminary annotation of the corpus, the answer becomes clear – the first application of paratactic operators is a sub-step of the hypotactic operations which combine satellite propositions with subordinate rhetorical relations (including ELABORATION) that have similar structures and modify the same entity in their nucleus proposition. The high level ordering of these operators should be hypotactic operators followed by paratactic operators, but paratactic operators are also applied to specific configurations of satellite propositions inside hypotactic operators as an optimization. Subordinate constructions are generally used to modify entities or relations. The operations to insert a modifying constituent into a sentence are often local operations. The operation is local because such insertion can be done often without considering other constituents in the sentence which are not being modified by the newly inserted modifying constituent. For example, attaching a prepositional phrase "with hypertension" to the sentence "Ms. Jones is an 80-year old patient undergoing heart bypass surgery" can be performed without considering how "with hypertension" interacts with adjectives or the relative clause, or where the entity being modified, "patient", appears in the sentence. In contrast, paratactic operators are global in nature because they are very sensitive to constituents that are identical across all the propositions being combine. The deletion of identical constituents cannot be made locally but must wait until the surface ordering of the identical constituents is known (this is related to *directional constraint* which is discussed in Chapter 5, Section 5.5.3).

Based on our understanding of the characteristics of the aggregation operators, the following sequence of aggregation operators are applied to the unaggregated propositions for evaluation:

1 Quantification
2 Adjective (maybe with conjunction)
3 Prepositional phrase (maybe with conjunction)
4 Reduced relative clause, including apposition (maybe with conjunction)
5 Relative clause (maybe with conjunction)
6 Transformations for other rhetorical relations (maybe with conjunction)

7 Simple conjunction
8 Complex conjunction

Quantification operations are applied before hypotactic and paratactic operators because quantification operations depend on discourse and contextual information. Since discourse and contextual information is more closely related to the overall structure of the whole document, operations based on such information should be first applied before syntactic operations, such as hypotactic and paratactic operators. Earlier in this section, an argument has been made for hypotactic operators to be applied before paratactic ones. The proposed sequence reflects this preference by applying adjective, prepositional, reduced relative clause (including apposition), and relative clause transformations before paratactic operations. The ordering for intra-hypotactic operators is chosen to produce the most concise sentence by applying operators which produce shortest transformed constituents first. This preference was described in detail in Chapter 4. Similarly, simple conjunction operator is applied before complex conjunction operator because the simple conjunction operator produces more concise expressions. Other hypotactic transformations for non-ELABORATION relation are treated as similar to relative clause transformations. Since there is little or no deletion result from such constructions ("Because John likes fruit, he ate apples." which has no deletion), they are low on the priority and thus becomes the last transformation of the hypotactic transformations.

Sentence (107) demonstrates the global versus local tendencies between paratactic and hypotactic operators.

(107) a. John borrowed a book1.
The book1 is about airplane.
Phil borrowed a book2.
The book2 is about train.

b. John and Phil borrowed books.
The book1 is about airplane.
The book2 is about train.

c. John borrowed a book1 about airplane.
Phil borrowed a book2 about train.

d. John borrowed a book about airplane and Phil borrowed a book about train.

If paratactic operators are applied first, then the grammatical sentence “John and Phil borrowed books” is derived first, as in (107b). Applying hypotactic operators next, the original simple conjunction construction must be undone because there is only a constituent “books” left in the combined proposition that can be modified by the two modifying propositions, one is 'about train” and the other is “about airplane.” As a result, the result of the simple paratactic operator needs to be undone, hypotactic operators are applied, followed by applying the multiple-distinct conjunction operator to the propositions in (107c). In this particular ordering of the operators, the processing of the first paratactic operation is wasted. In contrast, if the hypotactic operator is applied first, the paratactic operators can correctly use multiple-distinct operator on the first try. Since paratactic operators requires a global view of all the constituents in order to make its deletion decisions, it is to our advantage to apply all the hypotactic operators to make all local information known before applying global operations.

7.6 Conclusion

The goal of this chapter is to identify an ordering of the aggregation operators for synthesizing grammatical and concise sentences. By using such ordering information together with aggregation operators analyzed in the earlier chapters, the research community can incorporate clause aggregation operations into natural language generation systems and expect grammatical, concise sentences to be automatically generated. By imposing our proposed ordering onto de-aggregated propositions and trying to re-synthesize the original sentences, we were able to determine how well the proposed ordering works based on a human-written corpus.

In the analysis, we discovered several interesting things:

- The author did not encounter major difficulties in de-aggregating complex sentences written by humans. This was a little unexpected because the task is quite complex and it was not clear if all complex sentences can be broken down into smaller propositions easily. In both MAGIC's domain and Wall Street Journal articles, such de-aggregation process worked well. The generality of the de-aggregation process is an encouraging sign that the research in clause aggregation is on the right track.

- The discovery of *proposition set* or *propset* for annotating propositions during the de-aggregation process is also unexpected. The importance of rhetorical relations in clause aggregation operations were noted by many researchers (Scott and de Souza, 1990; Moser and Moore, 1995; Rösner and Stede, 1992), but the concept of propset was not mentioned in previous literature. It facilitates annotation during the de-aggregation process and allows annotator to ensure that both the number of transformations and rhetorical relations is always one smaller than the number of propositions. This kept both the de-aggregation and aggregation process simple.

- In the original ordering of the aggregation operators in CASPER, there was no good explanation for why paratactic operators are applied twice while hypotactic operators only applied once to a set of propositions. After the annotation, we realized that the earlier application of paratactic operators are for combining satellite propositions with similar structure and they modify the same entity in the nucleus proposition with the same rhetorical relation. This eliminated the possibility that there are two different types of ADDITION relations which cause the two separate applications of paratactic operators.

Conjunction is a global operator which involves all the constituents in the propositions and requires their positional information in the surface form to synthesize correct surface form. In contrast, hypotactic operators in general does not require information across multiple propositions and thus are local in nature. As a result, hypotactic operators should be applied first to pack as much information into propositions, followed by the applications of paratactic operators which delete redundant constituents from these complex propositions. The end result is complex sentences that are also concise.

Chapter 8

Related Work

In this dissertation, a chapter is devoted to each of the following aggregation operations: coordinating conjunctions (Chapter 5), premodifiers (Chapter 4), and quantification (Chapter 6). Because many researchers have analyzed these phenomena and detailed discussions of them tend to be technical, the related work of these phenomena was presented in each chapter devoted to these topics.

This chapter begins with a brief description of the early efforts in the linguistic community in clause aggregation at the syntactic level during the 1950s and 1960s, and the renewed effort at the rhetorical level in 1980s. The remainder of this chapter focuses on related work on clause aggregation operators in the natural language generation community. Section 8.3 describes various generation systems which incorporated aggregation operations since the early 1970s. In the early systems, the details of such aggregation operations were scarce. In the 1980s, several generation systems used aggregation operations as discourse level optimizations to improve the conciseness of the generated text. The type of aggregation operations proposed in such work is syntactically simple. In the early 1990s, several published papers described using revision operators to improve the generated text; these were basically clause aggregation operators.

Section 8.4 describes related work using the framework of clause aggregation operators as proposed in Chapter 3. Section 8.5 compares the current work of Robin and Dalianis, both of which focused on clause aggregation. Section 8.6 compares the current framework with those proposed by other researchers.

8.1 Related Linguistic Work

Since clause aggregation involves manipulations at the pragmatic, discourse, semantic, and syntactic levels, linguistic research in these areas is relevant to the current study. Two significant efforts in the linguistic community in aggregation should be noted. The first effort was at the syntactic level in the 1960s, based on Transformational Grammar. The second effort was at the rhetorical level, resulting in the development of Rhetorical Structure Theory (Mann and Thompson, 1987; Mann and Thompson, 1988). Information from both levels is used in the current study to constrain the clause aggregation process.

In the early days of Transformational Grammar, Chomsky (1957) used the term "generalized transformation" to describe the process of creating a new sentence by combining smaller ones. He proposed transformation rules such as conjunction transformation (Chomsky, 1957, p. 37) and nominalizing transformation T_{adj} in which the sentence "the boy is tall" becomes "the tall boy" (Chomsky, 1957, p. 72). In his later work (Chomsky, 1965), generalized transformation was eliminated and replaced by phrase-structure rules which introduced recursion. Complex sentences, such as "The cat that we loved died," are derived from the phrase-structure rule NP $\rightarrow$ Det N S, after which transformational machinery is applied to tidy up the surface form (Brown, 1991). The published arguments for and against using transformational or phrase-structure rules to perform coordinating conjunction were quite heated (Gleitman, 1965; Wierzbicka, 1980). By analyzing these difficult and complex constructions, linguists can identify inadequacies in linguistic formalisms and improve the theories accordingly. This led to a shift away from transformation towards phrase structure and lexicalized grammar. One of the early GPSG papers, (Gazdar, 1981), specifically described how a phrase-structure grammar can provide a formal description for gapping constructions. CCG can handle non-constituent coordination constructions better than any other formalisms. Numerous linguistic work was also related to quantifiers and premodifiers; these studies were discussed earlier in each respective chapter.

In the 1980s, computational linguists studied the rhetorical relations between propositions and proposed Rhetorical Structure Theories (Mann and Thompson, 1987; Mann and Thompson, 1988). They proposed 13 rhetorical relations as a starting point, including several studied in the current work (e.g., ELABORATION, SEQUENCE, ADDITION, and CONSEQUENCE). As discussed in Chapter 3, Section 3.2, rhetorical relations are essential to determine how propositions are combined. Research in syntactic transformations and rhetorical relations forms the basis of the clause aggregation operations.

8.2 System Architecture Related to Clause Aggregation

Both PLANDoc and Magic implement a pipelined architecture consisting of a content planner, a sentence planner, and a surface realizer. Many NLG systems have incorporated such an architecture into their applied generation systems (Reiter, 1994; Rambow and Korelsky, 1992; Panaget, 1994; Shaw, 1995). In Chapter 2, the sentence planner was divided into three submodules: referring expression generation, clause aggregation, and lexical choice. In contrast to the general acceptance of a pipeline architecture for the overall system architecture, researchers working on sentence planners have not yet agreed on a particular order among the submodules of the sentence planner. Wanner and Hovy (1996) suggested using a blackboard approach which allows flexibility between the processes. Because aggregation operators are non-monotonic, it is not clear how a blackboard approach can minimize interactions between the operators and ensure efficiency. In their default ordering, clause aggregation occurs first, followed by both lexical choice and referring expression performed in parallel (Wanner and Hovy, 1996, p. 5). In STREAK, Robin (1995) used an integrated approach to perform lexical choice and aggregation simultaneously. As a result, STREAK produced highly complex sentences but took more than half an hour to generate a single sentence (Robin, 1995).[1]

Reiter and Dale (2000) provided a good overview of sentence planners containing lexical choosers, referring expression modules, and clause aggregation modules. Their proposed ordering among the submodules was lexicalization, aggregation, and referring expression generation. In Magic, the ordering was referring expression generation, clause aggregation, referring expression generation again, and lexical choice. From the different ordering of submodules among these systems, it was clear that many issues in the sentence planner require further exploration.

An aspect of the current work worth noting is that the system architecture proposed in Casper is compatible with Systemic Functional Grammar, the linguistic formalism used by the surface realizer SURGE. The motivation for using

[1]Disclaimer: Back in 1994, computers were also much slower.

Systemic Functional Grammar (SFG) as the underlying formalism of a generation system is provided in Chapter 2. In SFG, a surface form is constructed as a consequence of selecting a set of features from this systemic network. As a result, it was an challenge to incorporate syntactic transformations into such formalism. In addition to the current work, SURGE (Elhadad and Robin, 1997) also incorporated various syntactic constructions into SFG. For example, given two conjoined clauses, SURGE can detect repeated subjects and delete the second one to generate a sentence with conjoined VPs, such as "John ate breakfast and took a shower." CASPER and SURGE encourage users to specify only the functional aspects of the propositions and take care of syntactic details internally within CASPER and SURGE.

8.3 Early Work

Aggregation has been employed since early generation systems. In PROTEUS, Davey (1979) described a procedure which "decides how many moves to describe in the next sentence, what conjunctions to use between move descriptions ..." PROTEUS is a computer program which gives a commentary on a tic-tac-toe game. Conjunctions were used to express SEQUENCE and CONTRASTIVE relations (e.g., "You forked me and I blocked your diagonal" and "I threatened you by taking a corner, but you blocked my edge," respectively). To generate a sentence with non-constituent coordination using Systemic Functional Grammar, such as "He is a mathematician and very clever," Davey proposed treating coordinated constituents as the equivalent in terms of functions instead of grammatical types (Davey, 1979, p. 80). Although the coordinating algorithm proposed in Chapter 5 can generate such sentences, a different approach compatible with Systemic Functional Grammar was used. Derr and McKeown (1984) showed how focus of attention is a factor in deciding whether to use a sequence of simple sentences or a complex one. Similar to Davey, they proposed combining tests based on a constituent's function within the sentence. ANA (Kukich, 1983) also formulated complex sentences up to 34 words in financial domain. The combination was based on rhetorical relations. Because ANA used a phrasal lexicon, the combining operators were domain specific. Another documented system which used aggregation operations is MUMBLE (McDonald, 1983b; McDonald, 1984). MUMBLE did not assume that an intermediate representation always maps to sentence-sized chunks. A unit can be incorporated into an already realized phrase-structure or become a separate sentence depending on its relation to previous text and stylistic considerations (McDonald, 1987, p. 176).

Similar to Davey, McDonald referred to Chomsky's work and clearly stated that linguistic constraints had a direct impact on aggregation, but he did not discuss the issues in detail (McDonald, 1983a, p.534).

KDS (Mann and Moore, 1980; Mann et al., 1982) used aggregation rules to combine propositions to formulate complex sentences, but these rules are not based on general linguistic principles. Mann and Moore used the term *aggregation* to describe clause-combining rules and focused on related issues. Mann and Moore considered their aggregation rules to be autonomous and meaning-preserving — essential properties in making an operator reusable across different domains. They proposed aggregation rules similar to logical derivation rules in interpretive aggregation. Examples included *common cause*, which reduced "Whenever C, then X," "Whenever C, then Y" into "Whenever C, then X and Y"; *delete mid-state* which reduced "Whenever X, then Y" and "Whenever Y, then Z" into "Whenever X, then Z." They used hill-climbing algorithms based on numerical quality scores to decide which aggregation rule to apply. They pointed out that since a rule that produces a large gain may preclude even more gain, the interaction among the rules was a problem. The difficulties in figuring out reliable scores for choosing a preferred aggregation rule were similar to identifying disfluencies in a text, a very difficult problem related to parsing. Mann later turned his attention to the rhetorical aspects of clause-combining operations (Mann, 1984; Mann and Thompson, 1987; Mann and Thompson, 1988).

The syntactic aspects of clause aggregation were ignored until Meteer (1991a; 1993) pointed out a *generation gap* between text planning and realization components. She defined the *expressibility criterion* as follows:

> A text plan is expressible if there are linguistic resources for realizing the elements in the plan and that their composition conforms to the syntactic rules of composition in the language. (Meteer, 1991b)

Most generation systems do not satisfy Meteer's expressibility criteria because in such systems, aggregation decisions are made without lexical information. Much aggregation work in the 1980s and early 1990s was performed on the discourse plan inside the content planner without access to a lexicon. Mellish (1988) proposed a set of clause aggregation rules called "message optimization." Although his rules removed redundancies in the plan structure, they were not linguistically-oriented. Other researchers incorporated coordinating conjunctions into their systems and described the construction as "grouping" or "optimization" (Dale, 1992). As shown in earlier chapters, performing clause aggregation without syntactic and lexical knowledge does not guarantee expressibility.

Some NLG researchers have studied clause aggregation under the topic "revision" (Meteer, 1991b; Robin, 1995; Callaway and Lester, 1997). Revision is a generate-and-test method in which drafts are iteratively evaluated and improved upon. Although revisions do not necessarily involve the combining of clauses, revision operations that combine clauses seem to be very effective for improving computer-generated text. Due to the difficulty of modeling complex pragmatic, semantic, and syntactic constraints which are needed for evaluating a text, identifying disfluencies in a text is as difficult as the general parsing problem, if not even more complex. Meteer's revision operators improve the following aspects of a text: they make the text more concise, make the text more explicit (or clear), change the emphasis, and make the text more active. Her revision work is extended in Robin's (1995) dissertation. Much work in revision operators was never fully implemented (Meteer and McDonald, 1986; Yazdani, 1987), while others were application-specific and only worked on a few examples (Gabriel, 1988; Inui, Tokunaga, and Tanaka, 1992). In systems that successfully used revision operators such as Streak (Robin, 1995) and REVISOR (Callaway and Lester, 1997), revision operations were applied to an intermediate representation, not to a generated text. Callaway and Lester (1997) performed a small experiment to demonstrate that the majority of judges preferred revised text produced by REVISOR over non-revised text. Similar to Text Structure in Meteer (1991b), Callaway and Lester specifically addressed the representation issues in revision by proposing an abstract representation which retained only essential information critical for making revisions. Their system REVISOR typically abstracted away 80% of the features in sentential specification, thus reducing the complexity of the revision operators and improving efficiency. Robin's Streak is one of the early, successful revision-based systems and his revision operators can formulate very complex sentences. Section 8.5.1 describes Robin's work in more detail. Early work in clause aggregation touched on many important issues, such as removing redundancies, efficiency of the operations, and underlying representations for aggregation. But none of them incorporated syntactic theories to guarantee that the complex aggregated results will be grammatical.

8.4 Specific Aggregation Operators

This section describes related work based on the four specific aggregation operators proposed in the current framework: interpretive, referential, syntactic, and lexical.

8.4.1 Interpretive Aggregation

Interpretive aggregation is divided into logical derivation and ad-hoc aggregation. In Section 8.3, a few logical derivation rules proposed in Mann and Moore (1981) are listed. Researchers working on proof presentation systems are particularly interested in logical derivations (Huang and Fiedler, 1996; Huang and Fiedler, 1997; Fehrer and Horacek, 1997; Holland-Minkley, Barzilay, and Constable, 1999). By removing obvious information for the readers by exploiting their inferential power, the generated text could be made more concise and natural.

Various researchers built systems that employed ad-hoc interpretation aggregations and described them either as *conceptual ellipsis* (Cook, Lehnert, and McDonald, 1984) or *conceptual subsumption* (Hovy, 1988; Hovy, 1990b). One of Hovy's papers which discussed such operations is appropriately named "Interpretation in Generation." Horacek (1990) used *terminological transformation* rules, based on ontology similar to KL-ONE (Brachman and Schmolze, 1985), in order to transform a structural description of multiple entities and relations into a more concise term "has-liquidity." In his later work, Horacek (1992) used the term *content-based grouping*, which combines "John hit Peter" and "Peter hit John back" into "John and Peter fight." In this process, information about who started the fight is lost. Due to our limited understanding of pragmatics and world knowledge, interpretive aggregation rules are application-specific and are not reusable in general. MAGIC content planner implemented a few domain specific interpretive aggregation rules to make generated text more interesting and concise. But since these rules are not portable, they are not discussed in the current work.

8.4.2 Referential Aggregation

In Chapter 3, referential aggregation operations are divided into two types: quantification over entities and identification of unique attributes for entities. Since various researchers have studied the process to identify unique attributes for entities in a text (Dale, 1992; Dale and Reiter, 1995; Horacek, 1997), this type of referential aggregation is not a focus in the current work.

Recent work on the generation of quantifiers (Gailly, 1988; Creaney, 1996; Creaney, 1999) follows the analysis viewpoint, discussing scope ambiguities extensively. Creaney discussed various imprecise quantifiers, such as '`some`', '`at least`', and '`at most`', and how to use scoping rules in the generation process. Although the present algorithm generates different sentences for different scope orderings, this was not achieved through the scoping operations used by others. Instead, the current study focused on a limited set of quantification operations in which unambiguous quantifier scope can be guaranteed by the system. The machine translation community also studied quantification in the generation process. But since the quantifiers are represented in the input to the generation modules (van Eijck and Alshawi, 1992; Copestake et al., 1999), machine translation systems do not have to derive quantifiers from the input representation as MAGIC does.

Generalization, replacing a set of entities with their common type in the ontology, is an essential process in quantification over entities. Many researchers used such a mechanism in their generation systems (i.e., conceptual subsumption proposed in Hovy (1990b)). Paasonneau et al. (1996) described a parameterized approach to perform generalization by balancing trade-offs between accuracy, specificity, and verbosity of description. CASPER also used generalization, but specifically for the synthesis of universal quantifiers.

8.4.3 Syntactic Aggregation

Since Meteer (1991a; 1993) pointed out the generation gap and proposed Text Structure to bridge this gap, many researchers have also looked into the problem of expressibility using linguistic resources. Many researchers have incorporated coordinating conjunctions in their systems. For example, Horacek (1992) proposed *structurally motivated propositional grouping*, and Mellish (1988) and Dale (1990; 1992) proposed *discourse-level optimization*. Dale used conjoined verbs in "Soak, drain and rinse the butterbeans" as an example (Dale, 1992, p. 86). Huang and Fiedler (1996; 1997) adopted Meteer's Text Structure as the underlying representation, with modification suggested by Panaget (1994). Their "semantic grouping" was performed in the content planner without any syntactic knowledge. Two examples of their rules are predicate grouping, which generated the sentence "F and G are sets," and semantic embedding, which combined two clauses "F is a set" and "F is a subset of G" into "The set F is a subset of G." After the sentence planner became a standard module in applied NLG systems (Reiter, 1994), many systems delegated the task of carrying out coordinating conjunctions to the sentence planner.

The proposed coordinating conjunction algorithm is compatible with Systemic Functional Grammar, the underlying linguistic theory of SURGE. By contrast, another major work on coordinating conjunctions in the generation community is not associated with any particular linguistic formalism. After citing Shaw (1998b) as a possible exception, Reape and Mellish (1999) stated that compared with available linguistic theories, treatments of coordination in implemented NLG systems to date have been relatively trivial. The only syntactic assumptions used in the coordinating algorithm in Dalianis (1996; 1999) are the specifications of subject, predicate (verb phrase), object, and direct object. Although multiple conjunction rules based on these syntactic constituents were proposed in Dalianis and Hovy (1996b) and Dalianis (1999), these rules only generated sentences with one conjoined constituent. They did not handle non-constituent coordination such as gapping, right-node-raising, or paratactic ellipsis. Dalianis discussed ordering between clauses and suggested a particular ordering for his domain. Despite his claim that ordering is based on the subjects of propositions, both predicates and entities were used in his ordering procedure. These included "state-change" and "supertype" which are relations, and animate object and inanimate object which are entities (Dalianis and Hovy, 1996a). It is not clear if this suggested ordering can be applied to a different domain. Dalianis (1993) suggested that cue markers such as "each" and "together" can be used to make aggregated sentences less ambiguous. As pointed out in Chapter 5, the use of such markers consistently is annoying to readers and should only be used selectively when disambiguation is necessary.

Hypotactic aggregation transforms the content of a proposition into a subconstituent of another proposition. The current study focuses on hypotactic operations related to ELABORATION when the two propositions share some entities (Shaw, 1998a). Early research in aggregation did not systematically transform clauses into modifiers, such as adjectives, PP, or relative clauses. In Shaw (1998a), CASPER ensures expressibility of the combined structure from hypotactic aggregation by accessing a lexicon to identify the syntactic properties of entities and predicate in a proposition before transforming it into a modifier in the nucleus clause. Scott and de Souza (1990) proposed heuristics for carrying out clause combining based on RST, and specifically identified which rhetorical relations were appropriate for "embedding," which corresponds to the present study's hypotactic operators. In Cheng, Mellish, and O'Donnell (1997), specific relations are proposed as candidates for hypotactic aggregation: 'Ownership' (e.g., a man <u>with black hair</u>), 'Identity' (e.g., <u>my friend</u> Jack), and 'Property-Ascription' (e.g., a <u>young</u> student). Similar hypotactic aggregation rules have been proposed in Shaw (1998a), in which '`is-an-instance`' can be mapped to Cheng's 'Identity' and '`has-attribute`' to 'Ownership' and 'Property-Ascription'. Scott and de Souza (1990) also proposed that when multiple constructions are possible to realize the same content, simpler constructions are preferred over more complex ones. Shaw (1998a) performed a limited corpus analysis to support the heuristics in which a simpler syntactic transformation was preferred over a more complex one, e.g., prepositional phrase transformations were used more often than relative clause transformations. Hypotactic operators are powerful because they are major linguistic constructions that introduce recursion into a system. Many linguistic constraints are associated with these phenomena. The adjective ordering issues discussed in this dissertation are only a small step in identifying and resolving these constraints. Malouf (2000) recently studied the ordering of premodifiers in a domain independent corpus. He used other statistic techniques and found similar results as those in the present study (Shaw and Hatzivassiloglou, 1999). However, many problems related to hypotactic aggregation remain open, such as incorporating mechanisms similar to c-command (Reinhart, 1983; Radford, 1988).

One important issue with aggregation is sentence delimitation, which is closely related to sentence complexity. Clearly, the type of sentences generated for a kindergarten audience is different from those for readers of *The New York Times*. The criteria for perform sentence delimitation should be parameterized and obtained from corpus analysis. The work in this area is very limited. For coordinating conjunctions, Shaw (1998b) proposed a heuristic which seems to produce quite complex sentences but is still easy to understand. According to Robin (1994), and McKeown, Robin, and Kukich (1995), a limit of 45 words and 10 levels of syntactic embedding is imposed based on the limits observed in the corpus of human-written game summaries.

8.4.4 Lexical Aggregation

Streak (Robin, 1995) performed lexical aggregation by combining aggregation and lexical choice to produce complex sentences which are also fluent. Streak was able to combine up to 12 facts into a sentence with 45 words, with 10 levels of embedding (Robin, 1995, Sec. 6.1.4). Performing lexical choice and aggregation simultaneously is costly because the best lexical decisions for n propositions might not be useful or correct for $n + 1$ propositions. This strategy generated impressive complex sentences, but for some complex sentences, Streak took more than half an hour. For real-time applications such as Magic, such inefficiency is unacceptable. Since Casper does not use detailed lexical information when it makes sentence boundary determinations, it traded some optimal aggregation for efficiency. Even though the lexicon is accessed twice in the present system, CASPER prunes the search space drastically by delaying expensive detailed lexical decisions after it knows how many concepts are involved in a sentence. The extensive use of unification to resolve constraints across multiple propositions in STREAK probably also contributed to its inefficiencies. Efficiency issues in generation were also addressed in McDonald, Meteer, and Pustejovsky (1987), and Elhadad, McKeown, and Robin (1997).

Lexical aggregation is closely related to lexical choice since after multiple clauses have been combined into the same linguistic structure, the concepts in the aggregated proposition can be further combined and expressed in fewer lexical items. Lexical choice is a major topic in natural language generation and deserves much more space than this dissertation can offer in its focus on referential and syntactic aggregations. Readers interested in the topic should refer to Robin (1990), Reiter (1990), Wanner (1994), Elhadad, McKeown, and Robin (1997), and Stede (1995). One critical issue in lexical choice is standardization of a sharable, large-scale lexicon for generation. Some efforts toward creating such a resource (Jing and McKeown, 1998; Knight and Luk, 1994) have been based on WordNet (Miller et al., 1990) and COMLEX (Grishman, Macleod, and Meyers, 1994). It is still to be seen how successful these efforts will be in minimizing development time.

8.5 Detailed Comparisons

Robin's (Robin, 1995) and Dalianis' (Dalianis, 1996) dissertations studied various issues related to clause aggregation. In this section, these works will be analyzed and compared to the framework proposed in this study.

8.5.1 Robin's Dissertation

Robin's system, STREAK, generated lead sentences for newspapers articles about basketball games. His research focused on how to opportunistically add historical and background information into a draft proposition to make the generated sentences more interesting. He proposed a set of revision operators which specified the structural transformation that a given draft proposition must undergo to incorporate a new piece of content. STREAK started out with a draft proposition containing four obligatory concepts (winner, game-result, loser, and score) and applied revision operators to opportunistically combine zero to eight additional concepts from the domain ontology into the draft proposition. These optional facts included historical information such as season high or streak[2] information. One of his revision operators, ABSORB, can transform the information in (108a) into the sentence (108b):

(108) a. `source`: Larry Bird scored 29 points Monday night including SEVEN 3 POINTERS.

b. `target`: Larry Bird scored 29 points Monday night including <u>matching his own club record</u> with SEVEN 3 POINTERS.

In (108b), the ABSORB operator replaced an constituent in the base structure of the source proposition ("SEVEN 3 POINTERS") with a new fact (the underlined expression in the target proposition) as the head and the original constituent as a modifier ("WITH SEVEN 3 POINTERS"). By limiting to a specific domain and using a corpus consisting only of lead sentences, Robin was able to model the domain using an ontology and provided an in-depth analysis of the aggregation phenomenon. Since Robin's revision operators involved adding more information into a clause, they are clause aggregation operators. An example of the added information, taken from Figure 4.7 of Robin's dissertation, is shown in Figure 8.1. Together with three other relations specified in a "game" concept, the following sentence can be generated in Streak: "Utah Jazz handed the Boston Celtics their six straight home defeats." In this representation, a relation does not correspond to a proposition as it did in Casper. In addition, the thematic roles used in Streak are domain-specific, such as "streak," "winner," "loser," "score," and "visitor." In Casper, the thematic roles used are not domain-specific and thus can be reused. Despite the overlap among the present study's operators, the focus of the current dissertation is to identify and resolve specific linguistic constraints related to adjective ordering, coordinated conjunction and quantification. In Streak, linguistic constraints related to revisions were encoded in a lexical chooser, but they were not the focus in Robin's dissertation. In that work, after a set of revision operators were identified based on corpus analysis, Robin (1995; 1996) focused on evaluation and demonstrated that by incorporating revision operators, Streak improved portability, robustness, and scalability.

Robin considered his revision operators as a type of summarization. He divided summarization operations into two categories: *conceptual summarization*, which selects the essential facts, and *linguistic summarization*, which expresses

[2]An unbroken series, as of wins or losses

```
((concept game)
 (... )
 (rels ((... )
        (... )
        (... )
        (streak1-ext ((deepsemcat relation)
                      (role streak-extension)
                      (token bos-streak-vs-uta-ext)
                      (args ((extension top-level)
                             (streak ((deepsemcat entity)
                                      (concept streak)
                                      (token bos-streak-vs-uta)
                                      (attrs ((card 6)))))))))))
```

Figure 8.1: This relation conveys the information "6 streaks", which is not a proposition.

the selected facts in compact surface form. In his dissertation, Robin focused on linguistic summarization and characterized his revision operators further into *monotonic* and *non-monotonic* revisions. Monotonic revision conserves both the argument structure and the lexical head of the base, and does not involve moving base constituents around. The base phrase content is not affected by the revision. Non-monotonic revision is less versatile than monotonic revision and is applicable only to specific types of base structure. These operators are based on the thematic roles in Systemic Functional Grammar (Halliday, 1994). In Table 8.1, the monotonic operators are closely related to the various aggregation operations analyzed in this dissertation. ADJOIN operations correspond to hypotactic aggregation operations while CONJOIN and APPEND operations correspond to paratactic aggregation operations.

Non-monotonic revision operations are meaning-preserving transformations that correspond to lexical aggregation. They include recast, adjunctization, nominalization, demotion, and promotion. For example, the "Recast in clause from location argument to instrument adjunct" revision operation can transform the source sentence (109a) into the target sentence (109b).

(109) a. `source`: to lead the New York Knicks to A 97-79 VICTORY OVER THE CHARLOTTE HORNETS

STREAK	CASPER	Examples in STREAK
Adjoin	hypotactic	
Classifier	apposition	Armon Gilliam scored **a franchise record** 39 points.
Describer	adjective	... the **hot-shooting** Boston Celtics ...
Origin PP	PP	Dana Barros contributed 21 points **off the bench**.
RCl	RCl	... a 106-103 win **that gave the Pistons their third straight defeat**.
Cl. Del. Ref.	reduced RCl	... the Golden State Warriors triumphed 110-105 **snapping the Boston Celtics 18-game home winning streak**.
Non-finite Cl.	reduced RCl	... leading the Los Angeles Clippers past the Indiana Pacers 122 107 **to snap a seven-game losing streak**.
Absorb		
Nominal	lexical	Ricky Pierce scored **a personal season high of** 33 points.
Cl.	lexical	Larry Bird scored 29 points Monday night including **matching his own club record with** seven 3 pointers.
Mean Adjunct	lexical	... to help the Charlotte Hornets **break a five-game losing streak by** holding off the Cleveland Cavaliers 115-107.
Conjoin		
Appositive	hypotactic	... leading the Denver Nuggets to **their first win of the season**, a 121-108 victory over the Minnesota Timberwolves.
Cl.	paratactic	to lead the Cleveland Cavaliers to a 94-78 victory over the Miami Heat **and break a three-game losing streak**.
Append		
Cl.	paratactic	Akeem Olajuwon score 27 points, grabbed 20 rebounds, **and blocked four shots**.
Ellipsis	paratactic	Willie Anderson scored 25 points, Terry Cummings 24, **and David Robinson 23**.
Nominal	paratactic	Benoit Benjamin contributed 19 points, 16 rebounds, **and six blocked shoots**.

Table 8.1: Mapping between Robin's monotonic revision operators and CASPER's aggregation operators

b. `target`: leading the Chicago Bulls to **their 23rd straight home court victory** WITH A 131-99 BLOWOUT OF THE MINNESOTA TIMBERWOLVES

To add the information that the victory is "the 23rd straight home court victory," the original location argument, "a 97-79 victory over the Charlotte Hornets," is moved to the instrument adjunct position. To compare such operators to the clause aggregation operators in the current work, the two propositions being combined in sentence (109b) need to be first identified:

(110) a. ...to lead the Chicago Bulls to A 131-99 BLOWOUT OF THE MINNESOTA TIMBERWOLVES.

b. The BLOWOUT is their 23rd straight home court victory.

Using relative clause attachment operators, the clauses are combined into (111).

(111) ... to lead the Chicago Bulls to A 131-99 BLOWOUT OF THE MINNESOTA TIMBERWOLVES, which is their 23rd straight home court victory.

Sentence (111) might not be as fluent as sentence (109b) which is produced by STREAK, but both sentences are semantic equivalents in the sense that readers would provide the same answers for all questions about the game result, whether they read sentence (110b) or sentence (111). Their main difference is that streak information (their 23rd straight home court victory) is highlighted in sentence (109b) since it occupies a head position. From this analysis, non-monotonic revisions are similar to lexical aggregation in which the aggregated proposition undergoes lexical choice to further optimize conciseness and fluency. The details of these non-monotonic revisions can be found in Robin (1994; 1995). Robin also described various side transformations which correct whatever redundancies, ambiguities or invalid lexical collocations that the revision may have introduced. One such transformation is *reference adjustment*, which is equivalent to the referring expression generation process after clause aggregation in CASPER. Another side transformation, *scope marking*, inserts distributive determiners to prevent collective interpretation (e.g., "Magic Johnson and Byron Scott scored 21 points apiece").

In terms of system architecture, the main difference between Robin's revision operators and the present aggregation operators is that CASPER differentiates between clause aggregation and lexical choice while STREAK combines the two processes into one. STREAK uses backtracking extensively to perform revision operations and lexical choice concurrently. Although clause aggregation and lexical choice are related, performing these operations together might be inefficient. Many clause-combining decisions, such as coordinating conjunctions, involve global constraints, and are thus very expensive if wrong branches are taken during the backtracking process. STREAK's problem with efficiency provided an incentive to separating the clause aggregation and lexical choice processes. The problem with such an approach was also discussed in Section 8.4.4. Except for my extensive work on conjunction and quantification and different viewpoints on lexical aggregation and lexical choice, many aspects of the revision operators in STREAK and CASPER are similar. Revision operators and clause aggregation operators are applied after content planners and before surface realization. Additional similarities include correspondence between the representations used in different generation modules. In STREAK, Deep Semantic Specification (DSS) corresponds to CASPER's unaggregated propositions; Surface Semantic Specification (SSS) corresponds to CASPER's partially lexicalized aggregated proposition. Since both STREAK and CASPER use SURGE as the surface realizer, Deep Grammatical Specification (DGS) and CASPER's lexicalized aggregation propositions use the same representation.

8.5.2 Dalianis' Dissertation

Dalianis (1999) categorized aggregation into four major types: syntactic aggregation, elision, lexical aggregation, and referential aggregation. Lexical aggregation is further divided into *bounded lexical aggregation*, which is meaning-preserving, and *unbounded lexical aggregation*, which is not meaning preserving. It is not clear how Dalianis arrived at these four major categories. In the present analysis, *bounded lexical aggregation* overlaps with referential aggregation, as elision overlaps syntactic aggregation. Instead of using terms such as *bounded lexical aggregation* and *unbounded lexical aggregation*, the taxonomy of clause aggregation operators in this dissertation uses terms that are more familiar to linguists and computational linguists. Dalianis (1999) focused on syntactic and lexical aggregation but he did not provide much information on elision or referential aggregation. Table 8.2 provides a mapping of the aggregation operators proposed between the current study and Dalianis' work.

Dalianis did not mentioned hypotactic aggregation in his work. The only hypotactic operation mentioned in his 1999 study appeared in his description of subject-predicate aggregation. The hypotactic operation, described as "additional sentence planning," transformed "John is a boy and tall" into "John is a tall boy."

This example is problematic because the first conjunct, "a boy," is a noun phrase, while the second conjunct, "tall," is an adjective. This is a non-constituent coordination which violates the premise that coordination construction involves constituents of equal syntactic status.

Dalianis	CASPER	Examples in (Dalianis, 1999)
Syntactic		
subject	paratactic	John **is a boy and is tall**.
subject & predicate	paratactic	John is **a boy and tall**.
predicate	paratactic	"John has a pen" and "Mary has a book" ⇒ **John and Mary** have **a book and a pen**.
PDO†	paratactic	"John wrote an article" and "Mary wrote an article" ⇒ **John and Mary** wrote an article.
Symmetric Relation	referential	**John and Mary** love **each other**.
Elision	paratactic	I would really like to have you guys over for dinner, so let me know whether for you it is better before [you] [leave for] [Florida] or [it is better] after [you] [come back from] Florida.
Lexical		
Bounded	referential	John uses his mobile phone on Monday, Tuesday, Wednesday, and Thursday. ⇒ John uses his mobile phone on weekdays except Friday.
Unbounded	interpretive	John is both a stationary subscriber and a mobile subscriber. ⇒ John is a subscriber.
Referential	referential	**They** are idle.

Table 8.2: Mapping between Dalianis' aggregation operators and CASPER's. Constituents in [ ] are deleted at the surface level. PDO† stands for Predicate and Direct Object.

With regard to coordinating conjunction, the main difference between Dalianis' work and the current study is that the algorithms proposed in the latter are more general — they can handle non-trivial coordinate constructions systematically, such as non-constituent coordination conjunctions. Dalianis and Hovy (1993), and Dalianis (1999) proposed various grouping rules to handle coordinating conjunctions. These are listed in Table 8.2 based on the definition given by Dalianis (1999). These aggregation rules do not handle conjunctions in prepositional phrases (i.e., "Mary shopped at Macy's and Saks"), or conjunctions of verbs (i.e., "John ate and enjoyed the dinner"). These rules generate sentences with only one conjoined constituent. Non-constituent coordination, such as gapping and paratactic ellipsis, are handled in a systematic manner in CASPER, but are not addressed by these operators.

Dalianis described lexical aggregation as the process by which a set of items is replaced with a single new lexeme that encompasses the same meaning. In the case of bounded lexical aggregation, the set of items is a closed set; thus, the aggregated information is recoverable. The bounded lexical aggregation is basically a quantification operation, a type of the referential aggregation under the current characterization, as shown in Table 8.2. The term "lexical" to describe quantification is a strange choice. Though Dalianis imposed no constraints on bounded lexical aggregation (universal quantification), Chapter 6 has shown that the differences among various universal quantifiers '`each`', '`every`', '`all`', and '`any`' matter when choosing a quantifier. Unbounded lexical aggregation is similar to interpretive aggregation. CASPER can ensure that a sentence with two quantifiers can be generated unambiguously. Dalianis (1999) noted that universal quantifiers can be used as a linguistic device for clause aggregation, but did not specify further constraints.

8.6 Other Work on Categorization of Aggregation

This section first describes research by Cheng (1997) who is also working on aggregation. Her work deals with hypotactic operators, referring expressions, and interactions between aggregation and text structuring. In their survey paper, Reape and Mellish (1999) discussed the definition of aggregation. They pointed out the need for generation systems to incorporate linguistic theories, such as GPSG, LFG, or CCG, to handle aggregation operations systematically. This dissertation attempts to accomplish the same by incorporating advanced linguistic constructions, such as coordinating conjunctions and quantification, into a generation system.

Wilkinson (1995) and Joanis (1999) recently wrote survey papers on aggregation. Because they did not propose new categories, these authors are not discussed here.

8.6.1 Hua Cheng

Cheng categorized domain-independent aggregation into four types: lexical, embedding, hypotactic, and paratactic. In addition to lacking interpretive and referential, the categorization of aggregation in the current work does not differentiate between embedding and hypotactic aggregation. According to Cheng, their main difference is that "hypotaxis deals with relations between two clauses, while embedding has a phrase as the bridge." (Cheng, Mellish, and O'Donnell, 1997). Because there are many instances where a proposition can be transformed into both phrase or clause (e.g., relative clause) during aggregation process, there is a significant overlap between the embedding and hypotactic ones. Current work does not distinguish between the two.

In Cheng, Mellish, and O'Donnell (1997), specific relations are proposed as candidates for hypotactic aggregation: 'Ownership' (e.g., a man <u>with black hair</u>), 'Identity' (e.g., <u>my friend</u> Jack), and 'Property-Ascription' (e.g., a <u>young</u> student). Similar hypotactic aggregation rules have been proposed in Shaw (1998a), in which '`is-an-instance`' can be mapped to Cheng's 'Identity' and '`has-attribute`' to 'Ownership' and 'Property-Ascription'. These relations provide further constraints for aggregation in addition to the rhetorical relations proposed in Scott and de Souza (1990).

Cheng (1998) discussed how to embed new information into a referring expression. Various embedding rules were proposed to minimize interactions with other types of aggregation and to ensure that the combined sentences did not mislead readers. Cheng and Mellish (2000) conducted experiments to validate the hypothesis that non-restrictive NP relative clause modifiers can be used to express rhetorical relations normally signaled by "`because`" and "`then`." Their conclusion was that if the inferrability between the two facts linked by a causal relation is strong, then non-restrictive clause can be used to express the causal relation. But, for temporal relation, cue phrases like "`then`" should be used in non-restrictive clauses. Before Cheng's work, non-restrictive NP modifiers were only used to express ELABORATION (Scott and de Souza, 1990; Cheng, 1998; Shaw, 1998a). Their work provides NLG systems with alternative realizations for a realization which is implicit rather than explicit.

8.6.2 Reape and Mellish

Reape and Mellish (1999) studied the following five questions related to aggregation:

- Why is aggregation done? (in Chapter 1, Introduction, and Chapter 3, Clause Aggregation)
- When is it done? (in Chapter 2, System Architecture)
- Where is it done? (in Chapter 2, System Architecture)
- What is it done on? (in Chapter 2, System Architecture and Representation)
- In what order are its subparts done? (in Chapter 2, System Architecture)

These questions have been addressed throughout the dissertation. Similar to this approach, Reape and Mellish (1999) also categorized aggregation operations based on the type of linguistic resources being used. They proposed six categories: conceptual aggregation, discourse aggregation, semantic aggregation, syntactic aggregation, lexical aggregation, and referential aggregation. They distinguished conceptual from semantic aggregation by assigning operations based on language-independent representations such as conceptual, and operations based on language-dependent representations such as semantic. The categorization is further complicated by lexical aggregation, which can also be viewed as a semantic aggregation because it is language-dependent. In this dissertation, such classification problems are avoided by assigning unconstrained aggregation operations as interpretive. Those aggregation operations based on restricted operations on the ontology are referential. Those aggregation operations that use the lexicon instead of the ontology are lexical. In addition, because all aggregation operations involve rhetorical relations, the current framework does not have a separate category for discourse or rhetorical aggregation.

This chapter describes previous work in clause aggregation. Table 8.3 provides a comparison between the categorization proposed in the current study and the ones proposed by other recent researchers in this area.

Shaw	Robin	Dalianis	Reape	Cheng
1. interpretive ad-hoc		1. UBLex	1. conceptual	
logical derivation		2. BLex	2. semantic	
2. referential identify attr				
quantification		1. UBLex 2. BLex 3. referential	1. conceptual 3. referential	
3. syntactic paratactic	1. conjoin 2. append	4. syntactic 5. elision	4. syntactic: paratactic	1. paratactic
hypotactic	3. adjoin 4. apposition		4. syntactic: hypotactic	2. hypotactic 3. embedding
4. lexical	5. absorb 6. nominalization 7. recast 8. others		5. lexical	4. lexical
Everywhere			6. discourse	

Table 8.3: CASPER's categories comparing with Robin, Dalianis, Reape and Mellish, and Cheng. UBLex=unbounded lexical, BLex=bounded lexical.

Chapter 9

Conclusion

This thesis addresses research issues in building practical natural language applications. It studies how to systematically combine clauses to produce complex and yet concise sentences in computational systems. By producing complex sentences like the ones humans write or speak, an automatic text generation system can create more realistic, natural sounding interactions with humans and improve human-computer interaction. Both of our applications, MAGIC and PLANDOC, use structured data stored in real-world applications as a basis to create a text/spoken output. Since original structured data were not designed with natural language generation in mind, straightforward translation of the structured data into English sentences results in verbose and repetitive output. To reduce redundancy and make the generated sentences more concise and similar to the ones humans write, techniques and algorithms were developed to ensure the grammaticality and fluency of the generated sentences. This thesis represents a milestone in natural language generation research where maturing technology and research are integrated to create practical applications that address users' needs. With a growing amount of structured data available online, such as XML (Prescod and Goldfarb, 1999), we will see more opportunities to use NLG technologies to transform the stored knowledge into comprehensible language for humans to read or to listen to.

9.1 Main Contributions

Much of the research described in this thesis started out by studying what linguists have done to address the problems encountered when building MAGIC and PLANDOC. Using linguists' observations as a basis, techniques were developed and incorporated into CASPER. The following are the dissertation's major contributions:

- **Incorporation of a wider range of domain independent clause aggregation operators into natural language generation systems than previously possible.** Many aggregation operators were analyzed and used to combine propositions into one complex sentence, including quantification, conjunction, adjective, prepositional phrase, and relative clause operators.

- **Conjunction, gapping, and related ellipsis constructions are unified into a paradigm which is compatible with Systemic Functional Grammar.** Before my current conjunction algorithm was developed, it was not clear how non-constituent coordinations, such as "John flew to Maryland on Monday and California on Tuesday," can be incorporated into Systemic Functional Grammar, one of the major formalisms used by text generation community. In the above example, the sentence contains non-constituent conjunction because the underlined conjuncts are not basic constituents. By using predicate argument structure and *directional constraints* proposed by Ross(1970) and Tai(1969), our algorithm treats simple conjunction, gapping, and related ellipsis constructions uniformly. As a result of developing this algorithm, CASPER can systematically determine which repeated constituents are redundant and delete them from the surface expression to make the generated sentence more concise.

- **A corpus-based approach is used to resolve adjective ordering decisions in sentence generation**. Ordering aggregated constituents is a task which must be addressed in order to produce complex but not awkward sounding sentences. This is an interesting point because awkward sounding sentences are grammatically correct, but humans can easily detect such disfluencies. Generating grammatically correct sentences is not good enough; a new dimension, fluency, is shown to be important in NLG and was addressed.

- **Discourse and contextual information are utilized to select universal quantifiers to make text more concise.** In most NLG systems, quantifiers are specified in the input representation. In the current work, universal quantifiers are derived from input representation, ontology, and discourse history. The proposed quantification algorithm incorporated findings from the linguistic literature to ensure correct distributive reading or collective reading is conveyed in the aggregated sentence. In addition, ambiguity related to quantification operations is addressed.

- **A corpus-based approach is used to analyze and study the sequential ordering of aggregation operators.** This thesis provides evidence from a corpus to demonstrate the general applicability of the proposed sequential ordering for aggregation operators. NLG researchers can be confident that NLG systems that employ the aggregation operators in the same ordering as CASPER will result in grammatical, fluent, and concise sentences.

9.2 Revisiting Issues in Clause Aggregation

In addition to investigate conjunction constructions in depth, the author also explored other types of clause aggregation operations to provide a broader view of issues that must be addressed to develop a robust, working system. The following are important issues in clause aggregation, each with references to chapters that address the issue.

- **Ensured grammaticality of the resulting sentences.** Simply putting propositions to be aggregated together into sentences by linking them with the conjunctor "and" does not result in a more concise or pleasant text to read or listen to. Clause aggregation is not an unconstrained process. Deletions of recurring constituents need to be performed correctly and the process is not straightforward. We have shown that the process at least involves the concepts of *constituent*, *identity*, and *directional constraints*. Chapter 5 specifically addresses this issue.

- **Ensured fluency.** Since clause aggregation creates sentences which might not exist before, there is no guarantee that they will sound natural. Chapter 4 specifically studies the ordering of adjectives in a domain specific corpus to emulate the way humans order them, which results in a more fluent text.

- **Minimized ambiguities resulting from clause aggregation.** By combining multiple propositions, surface constituents are shared in the final surface form in order to reduce verbosity. As a result of this sharing, ambiguities might arise. For paratactic operators, the scope of the conjunctor "and" can be a problem. For hypotactic operators, scope ambiguity of the modifiers, such as PP-attachment, is a well-known issue. One way to avoid ambiguity is not to aggregate, but since this solution would result in longer and repetitive text, it is not really acceptable. The approach taken in this thesis is to minimize ambiguity. Chapter 6 specifically addresses ambiguity related to universal quantifiers. Section 5.6.3 describes solutions in which conjoined constituents are reordered to make the scope of the modifiers unambiguous.

- **Analyzed interactions of clause aggregation with other modules in a text generation system.** The clause aggregation module is only one component of a sentence planner. In addition to clause aggregation, the sentence planner also handles lexical choice and referring expression decisions. The current approach is a pipelined one, where the referring expression module is followed by the clause aggregation module, the same referring expression module again, and finally lexical chooser. The rationale for such an ordering is discussed in Chapter 2. A more flexible ordering may help produce more expressive and varied sentences. Currently, the complex interactions between these modules are still insufficiently well understood for a definitive map to be drawn. The referential aggregation operator in Chapter 6 integrates both referring expression generation and clause aggregation operations to synthesize quantified expressions. Such combination of operations between different modules provide opportunities to understand their interactions.

- **Analyzed interactions among clause aggregation operators.** Inside the clause aggregation module, there is also an issue of how multiple aggregation operators should be ordered. Because clause aggregation operations involve many propositions and the search space grows exponentially as the number of propositions increases, efficiency is a critical issue in incorporating clause aggregation operators into applied natural language generation systems. Minimizing interactions between the operators can reduce the amount of backtracking needed. Chapter 7 specifically addresses these issues.

- **Identified features in the input representation which the content planner should provide to a clause aggregation module.** It was insufficient to provide Casper with only a set of propositions and let it combine them under the sole constraint of producing more concise text. The clause aggregation module requires certain information from the content planner in order to produce fluent and concise sentences while minimizing undesirable conversational implicatures. Both the operators in Chapters 4 and 5 require that the input representation contain not just surface strings, but also identifiers for the entities in the predicate argument structure. Chapter 7 presents a minimal interface between the content planner and the clause aggregation module in the annotation. The information includes rhetorical relations among the propositions being aggregated and a way for the content planner to specify the scope of satellite (or modifying) propositions. Using such an interface as a bridge, different text generation systems can share the input and more fruitful comparison of text generation system will be a possibility.

9.3 Where to Go from Here

Today, it is still a rather difficult and time-consuming process to develop an automated system converting structured data into cohesive text that users can read or listen to help them perform their tasks better. There are still many research and technical issues to be addressed before such a process is fully automated. Below is a list of fruitful areas which are natural extensions of the completed work.

- **Producing a more complex input representation from content planner.** Clause aggregation modules address syntactic issues in the generation process and provide an abstraction layer for the content planner, but there are other factors which affect the readability of a text. For example, the structured data might be too detailed for the task that users are trying to perform. In such cases, pragmatic or discourse processes inside the content planner should delete non-essential information or generalize from the structured information before giving the information to clause aggregation module. The goal of making the content planner more intelligent is becoming more pressing as lower level tasks, such as clause aggregation, are addressed. Improvements in content planner will have a direct impact on the quality of generated text.

- **Making the NLG system architecture more flexible in order to produce more expressive text.** Due to concerns for efficiency and simplicity of implementation, a pipelined architecture has been popular in applied text generation systems (Rambow and Korelsky, 1992; Reiter, 1994; Reiter and Dale, 2000). It is expected that an interleaved architecture will provide a boost to the quality of the generated text. Since clause aggregation is another module in the text generation process, interactions between clause aggregation and other modules should be captured and analyzed.

- **Studying clause aggregation of other rhetorical relations in addition to** Elaboration**,** Addition**, and** Sequence**.** Researchers (e.g., (Moser and Moore, 1995; Rösner and Stede, 1992)) have studied the positional ordering of the nucleus and satellite propositions in RST, and used corpus-based approaches to identify cue phrases for expressing rhetorical relations. An example of the relationship between RST and conciseness is the use of ellipsis in conjunction with the Contrast relation in "John likes orange but Mary doesn't." More fruitful research will come from this area.

- **Studying other interesting aggregation operations.** The linguistic constructions analyzed in this dissertation are some of the most common ones encountered in our target corpus. There are other significant and useful aggregation operators not cover in the current work, such as conjunction constructions with auxiliary verb which was extensively discussed in (Sag, 1976). One particular interesting and challenging topic is aggregation operations which involve extraction.

- **Employing text generation in practical applications which help users to be more productive.** Both Magic and PLANDoc produce text which help users to make more informed decisions. It is expected that with the widespread used of XML and shorter turnaround cycle in business environments, there will be more opportunities to incorporate NLG into systems that provide business value for users. Of course, the task of transforming structured data into fluent sentences is not easy, but that is what makes this research challenging and fun.

Appendix A

Guidelines for De-aggregation

This section describes the process in which a given sentence is de-aggregated into multiple propositions and annotated with useful information for aggregation operators. The process of de-aggregation is basically recovering deleted constituents – in other words, ellipsis identification and expansion. Due to our limited knowledge in common sense representation and parsing, there is no automated system which can perform the task as well as an educated human. The following guidelines were used by the author to annotate the corpus for the purpose of identify a sequential ordering of aggregation operators and for evaluating a proposed ordering.

To facilitate the annotation process, several assumptions were made. They are briefly described here.

- Each sentence is analyzed independent of other sentences in the corpus. As a result, the rhetorical relations among the propositions is not as varied as the set of rhetorical relations proposed by (Mann and Thompson, 1988). Inside each sentence, RST relations are used to describe the relations among de-aggregated propositions. The most critical aspect of RST employed in our analysis is that there are two major types of rhetorical relations among propositions – subordination (hypotaxis) and coordination (parataxis). Even without agreeing on the name of the rhetorical relations, the current analysis only requires that subordinate relations and coordinate relations are distinguishable, and there is a way to decide if two relations are identical or distinct. Once these two conditions are assumed, the results from the current analysis regarding ordering of aggregation operators should be valid.

- In Section 5.5.2 on constituent identity, we have shown that the knowledge of identifiers for entities are essential for aggregation algorithm. In the current annotation effort, the identifiers are implicit rather than explicitly marked. There are two main reasons for not inserting identifiers in the annotation. First, for the analysis of aggregation operator ordering, such information probably will not have an impact. Second, removing the identifiers also minimizes the annotation effort and makes the annotated corpus easier to read. As a result of this decision, identifiers for entities and relations are implicit.

- The current analysis focuses on the operator ordering among different types of operators. When the same operator is applied multiple times to obtain aggregated constituents, such as a sequence of adjectives, the ordering decision is outside the scope of the current annotation effort. Chapter 4 addresses such linearization issue using a different corpus-based approach. Among different type of operators, the current assumption is that the operators which are applied earlier should be closer to the head than the constituent results from operators applied later. For example, if the reduced relative clause operator is applied before the relative clause operator, at the surface level, the reduced relative clause will appear closer to the head than the relative clause.

- Ambiguities in the original sentence do exist and are resolved subjectively without consulting the original writer or domain expert. In a more general domain, such as a layman's financial domain, the attachments of modifiers are less of a problem. In contrast in medical domains, such ambiguities are much more common. Consulting domain experts, although possible, would have added much more effort. In addition, depending on the particular situation, even a domain expert might not be able to resolve the ambiguity unless he/she was present in the particular situation. Since we are interested mainly in the ordering of the operators, a few instances of incorrectly disambiguated attachments should not have a major impact on the result of the analysis.

The guidelines are divided into four sections. The first two are related to the annotation of propositions and propset. The third section concerns the specification of rhetorical relations among the propositions. The fourth section concerns the transformations which are applied to the de-aggregated propositions to formulate the original sentence.

A.1 Guideline for the Delimitation of Propositions

1. **The first proposition in a prop-set is always the nucleus proposition.** If the proposition is connected with other propositions with ADDITION or SEQUENCE relation, the designation of a proposition as being the nucleus proposition is ignored because no proposition is more prominent than others in such configuration.

2. As the basic unit for the current analysis, **a proposition must contains a verb and its argument(s)**. Even if a constituent in the original sentence is missing, as "blood pressure 100/60", the implied verb "is" is inserted into the de-aggregated proposition. Nominalization in the original sentence is another example, i.e. the circumstantial adjunct "After his complaint of chest pain..." is transformed into a proposition "the patient complained chest pain..." with the SEQUENCE relation attached.

3. **The propositions should be flat, with as few embedded constituent as possible**. Adjunct constituents which cannot be transformed into propositions easily are also included in the proposition.

4. **Propositions should be normalized.** This facilitates systematic and efficient operations for aggregation. Further, a uniform representation allows more manageable and complex inference operations.

A.2 Guideline for Specifying Propsets

1. **Use propset only if necessary.** In our algorithm, the propositions inside a propset are first aggregated before aggregation operators are applied to the proposition or propset at the higher level. Since propset can affect the ordering of the operators being applied, they should not be used unless there is a good reason to use them. Section 7.3.2 describes several reasons for using propset during our annotation process.

2. **Using propset for annotation keeps the number of rhetorical relations always one less than the number of de-aggregated proposition.**

This assumption simplifies the aggregation operators because the operators are closely related to rhetorical relations. The fewer of them, the simpler the process.

A.3 Guidelines for Specifying Rhetorical Relations

1. **Rhetorical relations must be used to connect all the de-aggregated propositions into a tree.** Rhetorical relations are a dominating factor of why propositions can be aggregated.

2. **There is only one rhetorical relation between any two propositions.** Together with the assumption that all the de-aggregated propositions must be connected, this means that the number of rhetorical relations is always $n-1$, n is the number of de-aggregated propositions. This is also made possible by the incorporation of the concept of propset into the analysis, which was discussed in Section 7.3.2.

3. **There is no predetermined set of rhetorical relations for annotation.** In the current analysis, 21 rhetorical relations were identified and they are shown in Figure 7.2.

4. **The determination of the rhetorical relation is strongly based on the syntactic construction and lexical cues employed in the original sentence.**

A.4 Guidelines for Specifying Transformations

1. **There is no predetermined set of transformations.** Although many syntactic and lexical transformations have been identified and analyzed in linguistics, there is no such thing as a comprehensive set of transformations.

2. **The number of transformations is always $n-1$, where n is the number of de-aggregated propositions**. The number of transformation is the

same as the number of rhetorical relations. This equality is no accident since each transformation corresponds to a rhetorical relation directly or indirectly.

3. **The operators are not selective.** When all the preconditions of an operator are satisfied, it must be applied. This prevents operators from behaving "intelligently" to formulate a fluent sentence.

Appendix B

Transformations in the Annotated Corpus

Excluding "arg" transformations, there are 523 transformations encountered in the annotated corpus. Below is the list of all the transformations. Those not handled by CASPER are marked by "*".

```
 10  adj
  1 *alternate
 16  apposition
 40  arg
  1 *colon-del-include
  1 *colon-sent
  1 *comparison-are-even
  4 *conj-complex-anaphor
 32  conj-mult
  1 *conj-mult-2nd-del-wh-be
  6 *conj-mult-del-wh-be
  1 *conj-mult-del-wh-be-pp
  1  conj-mult-gap
  1 *conj-mult-implied-subj-del
  1 *conj-mult-inf-vp
  1  conj-mult-passive
  1 *conj-mult-progress-vp
  1 *conj-mult-semicolon
 19  conj-mult-sent
 29  conj-mult-vp
  2 *conj-mult-vp-but
203  conj-simp
```

```
2 *conj-simp-as-well-as
3  conj-simp-neg
6  conj-simp-nested
1  conj-simp-quant-both
1 *conj-simp-reverse
6 *conj-with
3 *cue-after-sent
2 *cue-although-sent
1 *cue-apart-from
5 *cue-as-sent
1 *cue-at-which-point-sent
1 *cue-because-nominal
1 *cue-because-of
3 *cue-because-sent
1 *cue-before-sent
1 *cue-but-adj
1 *cue-but-conj-vp
1 *cue-but-otherwise-conj
1 *cue-but-sent
3 *cue-but-vp
1 *cue-despite-sent
1 *cue-even-though
1 *cue-except-np
2 *cue-except-pp
1 *cue-following-np
1 *cue-for-instance
1 *cue-for-vp-ing
2 *cue-if
1 *cue-in-addition-nominalization
1 *cue-in-expectation-that
3 *cue-since-nominal
1 *cue-such-as
1 *cue-therefore-conj-vp
1 *cue-thus-conj-vp
1 *cue-to
2 *cue-when
3 *cue-when-sent
1 *cue-while-del-many
1 *cue-while-sent
4 *disj-simp-neg
1 *hyphen-np
1 *hyphen-vp
1 *parenthesis-apposition
1 *parenthesis-contains
2  pp-as-np
1  pp-between
2  pp-for
1  pp-from
4  pp-in
1  pp-like
5  pp-of
1  pp-on
3  pp-to
9  pp-with
1  pp-without
2  prenominal
7  prenominal-title
```

```
 1 *ref-transposition
 1 *rel-inf
 1  rel-reduced-del-wh
24  rel-reduced-del-wh-be
 1 *rel-reduced-del-wh-be-adj
 1 *rel-reduced-del-wh-extraction
 1 *rel-reduced-del-wh-verb
11  rel-reduced-ing
 1 *rel-reduced-ing-colon
 1 *rel-reduced-ing-cue-for-example
 5  rel-that
 1 *rel-that-extraction
16  rel-wh
 1 *rel-wh-extraction
 1 *rel-when
 1 *rel-which-del-extract
 5 *rel-whose
```

Appendix C

wsj.xml


```
<document>
<sentence id=s1>
  WASHINGTON -- As a candidate in 1992, Bill Clinton spoke
  boldly about addressing America's "investment deficit" with
  $50 billion a year in new spending on high-tech highways,
  technology, education and training aimed at improving life in
  the future.
    <propset id=pset1-1>
      <prop id=p1-1>
        Bill Clinton spoke boldly about addressing America's
        "investment deficit" with $50 billion a year in new
        spending. </prop>
      <prop id=p1-2>
        The spending is on high-tech highways, technology, education and
        training.   </prop>
      <prop id=p1-3>
        Bill Clinton was a candidate in 1992.    </prop>
      <prop id=p1-4>
        New spending is aimed at improving life in the future.
        </prop>
    </propset>
    <focus entity='Bill Clinton'/>
    <rst-rel id=r1-1 name=elab
             nuc=p1-1   sat=p1-2   ref=no  />
    <rst-rel id=r1-1 name=elab
             nuc=p1-1   sat=p1-3   ref=no  />
    <rst-rel id=r1-2 name=elab
             nuc=p1-1   sat=p1-4   ref=no  />
    <trans id=tx1-1 name=pp-on      nuc=p-1 sat=p1-2 />
    <trans id=tx1-2 name=pp-as-np   nuc=tx1-1 sat=p1-3 />
    <trans id=tx1-3 name=rel-reduced-del-wh-be nuc=tx1-2 sat=p1-4 />
    <seqorder valid=true />
    <conj id=c1-1 type=coll />
</sentence>
<sentence id=s2>
```

```
  He promised to pay for these measures, and cut the deficit, by
  reducing government programs that subsidize current
  "consumption."
    <propset id=pset2-1>
      <propset id=pset2-2>
      <prop id=p2-1>
        He promised to pay for these measures by reducing
        government programs.    </prop>
      <prop id=p2-2>
        He promised to cut the deficit by reducing government
        programs.    </prop>
      </propset>
      <prop id=p2-3>
        The government programs subsidize current "consumption."
   </prop>
    </propset>
    <focus entity='he'/>
    <rst-rel id=r2-1 name=join
             nuc=p2-1   sat=p2-2   ref=no  />
    <rst-rel id=r2-2 name=elab
             nuc=pset2-2   sat=p2-3   ref=no  />
    <trans id=tx2-1 name=conj-simp   nuc=p2-1 sat=p2-2 />
    <trans id=tx2-2 name=rel-that   nuc=tx2-1 sat=p2-3 />
    <seqorder valid=true />
    <conj id=c2-1 type=dist />
</sentence>
<sentence id=s3>
  Both sides of the budget debate "have cut investments in
  things that would strengthen the economy -- the Republicans
  deeply, the president substantially," says Robert Shapiro of
  the Progressive Policy Institute, and an architect of Mr.
  Clinton's 1992 campaign manifesto.
    <propset id=pset3-1>
      <prop id=p3-1>
        Robert Shapiro says -X-        </prop>
      <propset id=pset3-2>
      <prop id=p3-2>
        The Republicans "have cut investments in things that would
        strengthen the economy" deeply.  </prop>
      <prop id=p3-3>
        The president "has cut investments in things
        that would strengthen the economy" substantially.  <prop>
      <prop id=p3-4>
        Both sides of the budget debate are the Republicans and
        the president. </prop>
      </propset>
      <prop id=p3-5>
        Robert Shapiro is a member of the Progressive Policy
        Institute.    </prop>
```

```
      <prop id=p3-6>
        Robert Shapiro is an architect of Mr. Clinton's 1992
        campaign manifesto.    </prop>
    </propset>
    <focus entity='Robert Shapiro'/>
    <rst-rel id=r3-1 name=arg
             nuc=p3-1   sat=pset3-2   ref=no  />
    <rst-rel id=r3-2 name=join
             nuc=p3-2   sat=p3-3   ref=no  />
    <rst-rel id=r3-3 name=elab
             nuc=p3-2   sat=p3-4   ref=no  />
    <rst-rel id=r3-4 name=elab
             nuc=p3-1   sat=p3-5   ref=no  />
    <rst-rel id=r3-5 name=elab
             nuc=p3-1   sat=p3-6   ref=no  />
    <trans id=tx3-1 name=conj-simp   nuc=p3-2 sat=p3-3 />
    <trans id=tx3-2 name=ref-transposition   nuc=p3-4 sat=tx3-1 />
    <trans id=tx3-3 name=arg   nuc=p3-1 sat=tx3-2 />
    <trans id=tx3-4 name=pp-of   nuc=tx3-3 sat=p3-5 />
    <trans id=tx3-5 name=conj-mult-del-wh-be nuc=tx3-4 sat=p3-6 />
    <seqorder valid=false note="tx3-1 tx3-2 are of wrong order." />
    <conj id=c3-1 type=dist />
    <comment text="- the transformations required for this is not handled."
    />
</sentence>
<sentence id=s4>
  For that year, both Mr. Clinton and the Republicans proposed
  defense and non-defense spending of about $515 billion,
  between 25% and 30% below current levels after adjusting for
  expected inflation of 3% a year.
    <propset id=pset4-1>
      <prop id=p4-1>
        For that year, Mr. Clinton and the Republicans proposed defense
        and non-defense spending of about $515 billion.  </prop>
      <prop id=p4-2>
        The spending is between 25% and 30% below current levels
        after adjusting for expected inflation of 3% a year.
        </prop>
    </propset>
    <focus entity='Mr. Clinton'/>
    <rst-rel id=r4-1 name=elab
             nuc=p4-1  sat=p4-2   ref=no  />
    <trans id=tx4-1 name=pp-between  nuc=p4-1 sat=p4-2 />
    <seqorder valid=true />
    <conj id=c4-1 type=coll />
    <conj id=c4-2 type=coll />
```

```
    <conj id=c4-3 type=betw />
</sentence>
<sentence id=s5>
  But nearly all investment -- education, training, highways,
  airports, research and development, health care and nutrition
  for pregnant women and children -- is funded through these
  bills.
    <propset id=pset5-1>
      <prop id=p5-1>
        But nearly all investment is funded through these bills.  </prop>
      <prop id=p5-2>
        The investment include education.    </prop>
      <prop id=p5-3>
        The investment include training.    </prop>
      <prop id=p5-4>
        The investment include highways.    </prop>
      <prop id=p5-5>
        The investment include airports.    </prop>
      <prop id=p5-6>
        The investment include research and development.    </prop>
      <prop id=p5-7>
        The investment include health care.    </prop>
      <prop id=p5-8>
        The investment include nutrition for pregnant women and children.
      </prop>
    </propset>
    <focus entity='investment'/>
    <rst-rel id=r5-1 name=elab
             nuc=p5-1   sat=p5-2   ref=no  />
    <rst-rel id=r5-2 name=elab
             nuc=p5-1   sat=p5-3   ref=no  />
    <rst-rel id=r5-3 name=elab
             nuc=p5-1   sat=p5-4   ref=no  />
    <rst-rel id=r5-4 name=elab
             nuc=p5-1   sat=p5-5   ref=no  />
    <rst-rel id=r5-5 name=elab
             nuc=p5-1   sat=p5-6   ref=no  />
    <rst-rel id=r5-6 name=elab
             nuc=p5-1   sat=p5-7   ref=no  />
    <rst-rel id=r5-7 name=elab
             nuc=p5-1   sat=p5-8   ref=no  />
    <trans id=tx5-1 name=conj-simp   nuc=p5-2 sat=p5-3 />
    <trans id=tx5-2 name=conj-simp   nuc=tx5-1 sat=p5-4 />
    <trans id=tx5-3 name=conj-simp   nuc=tx5-2 sat=p5-5 />
    <trans id=tx5-4 name=conj-simp   nuc=tx5-3 sat=p5-6 />
    <trans id=tx5-5 name=conj-simp   nuc=tx5-4 sat=p5-7 />
    <trans id=tx5-6 name=conj-simp   nuc=tx5-5 sat=p5-8 />
    <trans id=tx5-7 name=hyphen-np   nuc=p5-1 sat=tx5-6 />
    <seqorder valid=true />
    <conj id=c5-1 type=coll />
```

```
    <conj id=c5-2 type=coll />
    <conj id=c5-3 type=coll />

    <comment text="- all investment is a quantification" />
</sentence>

<sentence id=s6>
  The remainder of government spending includes benefit and
  retirement programs for the elderly, the poor, farmers and
  veterans -- which generally subsidize consumption, not
  investment -- and interest on government debt.

    <propset id=pset6-1>
      <propset id=pset6-2>
      <prop id=p6-1>
        The remainder of government spending includes benefit and
        retirement programs for the elderly.    </prop>
      <prop id=p6-2>
        The remainder of government spending includes benefit and
        retirement programs for the poor.    </prop>
      <prop id=p6-3>
        The remainder of government spending includes benefit and
        retirement programs for farmers.    </prop>
      <prop id=p6-4>
        The remainder of government spending includes benefit and
        retirement programs for veterans.    </prop>
      </propset>
      <prop id=p6-5>
        the benefit and retirement programs generally subsidize consumption.
      <prop id=p6-6>
        the benefit and retirement programs generally does not subsidize
        investment.    </prop>
      <prop id=p6-7>
        the remainder of government spending include interest on
        government debt.    </prop>
    </propset>

    <focus entity='remainder'/>
    <rst-rel id=r6-1 name=join
             nuc=p6-1   sat=p6-2   ref=no  />
    <rst-rel id=r6-2 name=join
             nuc=p6-2   sat=p6-3   ref=no  />
    <rst-rel id=r6-3 name=join
             nuc=p6-3   sat=p6-4   ref=no  />
    <rst-rel id=r6-4 name=elab
             nuc=pset6-2   sat=p6-5   ref=no  />
    <rst-rel id=r6-5 name=elab
             nuc=pset6-2   sat=p6-6   ref=no  />
    <rst-rel id=r6-6 name=join
             nuc=pset6-2   sat=p6-7   ref=no  />

    <trans id=tx6-1 name=conj-simp   nuc=p6-1 sat=p6-2 />
    <trans id=tx6-2 name=conj-simp   nuc=tx6-1 sat=p6-3 />
    <trans id=tx6-3 name=conj-simp   nuc=tx6-2 sat=p6-4 />
```

```
    <trans id=tx6-4 name=conj-simp-neg   nuc=p6-5 sat=p6-6 />
    <trans id=tx6-5 name=rel-wh   nuc=tx6-3 sat=tx6-4 />
    <trans id=tx6-6 name=conj-simp   nuc=tx6-5 sat=p6-7 />

    <seqorder valid=true />

    <conj id=c6-1 type=coll />
    <conj id=c6-2 type=dist />
    <conj id=c6-3 type=dist />

    <comment text="- conjunction of 'does not' does not use 'and' in
                   tx6-5.  This is an interesting case" />
</sentence>

<sentence id=s7>
  "To the extent the roads deteriorate, you spend a lot more on
  maintaining cars and trucks," says Mr. Penner, a Republican and
  self-described skeptic about the value of much of what is labeled
  "public investment."

    <propset id=pset7-1>
      <prop id=p7-1>
        Mr. Penner says -X-  </prop>
      <propset id=pset7-2>
      <prop id=p7-2>
       "To the extent the roads deteriorate, you
        spend a lot more on maintaining cars."   </prop>
      <prop id=p7-3>
       "To the extent the roads deteriorate, you
        spend a lot more on maintaining trucks."   </prop>
      </propset>
      <prop id=p7-4>
        Mr. Penner is a Republican.    </prop>
      <prop id=p7-5>
        Mr. Penner is a self-described skeptics about value of
        much of what is labeled "public investment."     </prop>
    </propset>

    <focus entity='Mr. Penner'/>
    <rst-rel id=r7-1 name=arg
             nuc=p7-1   sat=pset7-2   ref=no  />
    <rst-rel id=r7-2 name=join
             nuc=p7-2   sat=p7-3   ref=no  />
    <rst-rel id=r7-3 name=elab
             nuc=p7-1   sat=p7-4   ref=no  />
    <rst-rel id=r7-4 name=elab
             nuc=p7-1   sat=p7-5   ref=no  />

    <trans id=tx7-1 name=conj-simp   nuc=p7-2 sat=p7-3 />

    <trans id=tx7-2 name=arg   nuc=p7-1 sat=tx7-1 />
    <trans id=tx7-3 name=conj-simp   nuc=p7-4 sat=p7-5 />
    <trans id=tx7-4 name=apposition   nuc=tx7-3 sat=tx7-2 />

    <seqorder valid=true />

    <conj id=c7-1 type=dist />
```

```
    <conj id=c7-2 type=dist />
</sentence>
<sentence id=s8>
  The spending cuts Republicans and Mr. Clinton talk about for
  next year and the year after aren't nearly so deep.
    <propset id=pset8-1>
      <propset id=pset8-2>
      <prop id=p8-1>
        The spending cuts for next year isn't nearly so deep. </prop>
      <prop id=p8-2>
        The spending cuts for the year after isn't nearly so
        deep. </prop>
      </propset>
      <prop id=p8-3>
        Republicans and Mr. Clinton talk about the spending cuts.
    </propset>
    <focus entity='The spending cuts'/>
    <rst-rel id=r8-1 name=join
             nuc=p8-1   sat=p8-2   ref=no  />
    <rst-rel id=r8-2 name=elab
             nuc=p8-1   sat=p8-3   ref=no  />
    <trans id=tx8-1 name=conj-simp   nuc=p8-1 sat=p8-2 />
    <trans id=tx8-2 name=rel-reduced-del-wh-extraction  nuc=tx8-1
    sat=p8-3 />
    <seqorder valid=true />
    <conj id=c8-1 type=coll />
    <conj id=c8-2 type=dist />
    <comment text="- there is a passivization before transform p8-3
    into a deleted relative clause." />
</sentence>
<sentence id=s9>
  And Mr. Penner, now at Barents Group, a KPMG consulting unit,
  doubts the spending ceilings that the Republicans -- and
  lately Mr. Clinton -- proposed for six and seven years in the
  future will hold.
    <propset id=pset9-1>
      <prop id=p9-1>
        And Mr. Penners doubts the spending ceilings will hold. </prop>
      <prop id=p9-2>
        Mr. Penner is now at Barents Group.  </prop>
      <prop id=p9-3>
        Barents Group is a KPMG consulting unit.  </prop>
      <prop id=p9-4>
        The Republicans proposed the spending ceiling for six and seven
        years in the future. </prop>
      <prop id=p9-5>
        Mr. Clinton lately proposed the spending ceiling for six and seven
        years in the future. </prop>
    </propset>
```

```
    <focus entity='Mr. Penner'/>
    <rst-rel id=r9-1 name=elab
             nuc=p9-1   sat=p9-2   ref=no  />
    <rst-rel id=r9-2 name=elab
             nuc=p9-2   sat=p9-3   ref=no  />
    <rst-rel id=r9-3 name=elab
             nuc=p9-1   sat=p9-4   ref=no  />
    <rst-rel id=r9-4 name=elab
             nuc=p9-1   sat=p9-5   ref=no  />
    <trans id=tx9-1 name=conj-simp   nuc=p9-4 sat=p9-5 />
    <trans id=tx9-2 name=apposition   nuc=p9-2 sat=p9-3 />
    <trans id=tx9-3 name=rel-reduced-del-wh-be   nuc=p9-1 sat=tx9-2 />
    <trans id=tx9-4 name=rel-that-extraction   nuc=tx9-3 sat=tx9-1 />
    <seqorder valid=true />
    <conj id=c9-1 type=dist />
    <conj id=c9-2 type=dist />
    <conj id=c9-3 type=dist-or />
</sentence>
<sentence id=s10>
  "Cuts of that magnitude would require changing quite
  dramatically whole very large areas of government," and not
  just unpopular bureaucracy, he says.
    <propset id=pset10-1>
      <prop id=p10-1>
        He says -X-.  </prop>
      <propset id=pset10-2>
      <prop id=p10-2>
        "Cuts of that magnitude would require changing quite dramatically
        whole very large area of government".  </prop>
      <prop id=p10-3>
        Cuts of that magnitude would require changing not just unpopular
        bureaucracy.  </prop>
      </propset>
    </propset>
    <focus entity='he'/>
    <rst-rel id=r10-1 name=arg
             nuc=p10-1   sat=p10-2   ref=no  />
    <rst-rel id=r10-2 name=join
             nuc=p10-2   sat=p10-3   ref=no  />
    <trans id=tx10-1 name=conj-simp-neg   nuc=p10-2 sat=p10-3 />
    <trans id=tx10-2 name=arg  nuc=p10-1 sat=tx10-1 />
    <seqorder valid=true />
    <conj id=c10-1 type=dist />
</sentence>
<sentence id=s11>
  Nevertheless, except for frequent references to the urgent
```

```
  need for federal spending on education and the environment,
  even Mr. Clinton has stopped making the case for public
  investment.
    <propset id=pset11-1>
      <prop id=p11-1>
        Mr. Clinton has stopped making the case for public investment.
        </prop>
      <prop id=p11-2>
        Nevertheless, except Mr. Clinton made frequent references to the
        urgent need for federal spending on education.  </prop>
      <prop id=p11-3>
        Nevertheless, except Mr. Clinton made frequent references to the
        urgent need for federal spending on the environment.   </prop>
    </propset>

    <focus entity='Mr. Clinton'/>
    <rst-rel id=r11-1 name=condition-except
             nuc=p11-1   sat=p11-2   ref=no  />
    <rst-rel id=r11-2 name=condition-except
             nuc=p11-1   sat=p11-3   ref=no  />

    <trans id=tx11-1 name=conj-simp   nuc=p11-2 sat=p11-3 />
    <trans id=tx11-2 name=cue-except-pp   nuc=p11-1 sat=tx11-1 />

    <seqorder valid=true />

    <conj id=c11-1 type=dist />

</sentence>

<sentence id=s12>
  Mr. Clinton's ill-fated economic stimulus proposals early in
  his presidency soured the public and Congress on his "public
  investment" proposals.
    <propset id=pset12-1>
      <propset id=pset12-2>
      <prop id=p12-1>
        Mr. Clinton's economic stimulus proposals soured the
        public on his "public investment" proposals.    </prop>
      <prop id=p12-2>
        Mr. Clinton's economic stimulus proposals soured Congress
        on his "public investment" proposals.    </prop>
      </propset>
      <prop id=p12-3>
        The economic stimulus proposals are ill-fated.    </prop>
      <prop id=p12-4>
        The economic stimulus proposals are early in his
        presidency.    </prop>
    </propset>

    <focus entity='stimulus proposals'/>
    <rst-rel id=r12-1 name=join
             nuc=p12-1   sat=p12-2   ref=no  />
    <rst-rel id=r12-2 name=elab
             nuc=p12-1   sat=p12-3   ref=no  />
```

```
    <rst-rel id=r12-3 name=elab
             nuc=p12-1   sat=p12-4   ref=no  />

    <trans id=tx12-1 name=conj-simp   nuc=p12-1 sat=p12-2 />
    <trans id=tx12-2 name=adj   nuc=tx12-1 sat=p12-3 />
    <trans id=tx12-3 name=rel-reduced-del-wh-be   nuc=tx12-2 sat=p12-4 />

    <seqorder valid=true />

    <conj id=c12-1 type=dist />
</sentence>

<sentence id=s13>
  And Mr. Clinton's aides note that he has been forced to tailor
  his ambitions to the political reality of a Republican
  Congress that is insisting on balancing the budget over seven
  years while cutting taxes and resisting further cuts in
  defense spending.

    <propset id=pset13-1>
      <prop id=p13-1>
        Mr. Clinton's aides note -X-.  </prop>
    <propset id=pset13-2>
      <prop id=p13-2>
        Mr. Clinton has been forced to tailor his ambitions to the
        political reality of a Republic Congress.  </prop>
      <propset id=pset13-3>
      <prop id=p13-3>
        The Republican Congress is insisting on balancing the
        budget over seven  years.    </prop>
      <prop id=p13-4>
        The Republican Congress is insisting on cutting taxes.
        </prop>
      </propset>
      <prop id=p13-5>
        The Republican Congress is resisting further cuts in
        defense spending.    </prop>
    </propset>
    </propset>

    <focus entity='aides'/>

    <rst-rel id=r13-1 name=arg
             nuc=p13-1   sat=p13-2   ref=no  />
    <rst-rel id=r13-2 name=elab
             nuc=p13-2   sat=p13-3   ref=no  />
    <rst-rel id=r13-3 name=concurrent
             nuc=p13-3   sat=p13-4   ref=no  />
    <rst-rel id=r13-4 name=elab
             nuc=p13-2   sat=p13-5   ref=no  />

    <trans id=tx13-1 name=cue-while-del-many   nuc=p13-3 sat=p13-4 />

    <trans id=tx13-2 name=conj-mult-progress-vp   nuc=tx13-1 sat=p13-5 />
    <trans id=tx13-3 name=rel-that   nuc=p13-2 sat=tx13-2 />

    <trans id=tx13-4 name=arg   nuc=p13-1 sat=tx13-3 />

    <seqorder valid=true />
```

```
    <conj id=c13-1 type=dist />
    <conj id=c13-2 type=dist />
</sentence>
<sentence id=s14>
  But Mr. Clinton also has made a calculated political decision
  to style himself as the protector of Medicare and Social
  Security, the costly programs that benefit senior citizens of
  all incomes, as well as the Medicaid health-care program for
  the poor and disabled.
    <propset id=pset14-1>
      <propset id=pset14-2>
      <prop id=p14-1>
        Mr. Clinton also has made a political decision to style
        himself as the protector of Medicare. </prop>.  </prop>
      <prop id=p14-2>
        Mr. Clinton also has made a political decision to style
himself as the protector of Social Security.  </prop>
      <prop id=p14-3>
        Mr. Clinton also has made a political decision to style
        himself as the protector of Medicaid heal-care program for the
        poor and disabled.  </prop>
      <prop id=p14-4>
        Medicare is a costly program. </prop>
      <prop id=p14-5>
        Social Security is a costly program.  </prop>
      <prop id=p14-6>
        Medicare benefits senior citizens of all incomes. </prop>
      <prop id=p14-7>
        Social Security benefits senior citizens of all incomes.
      </propset>
      <prop id=p14-8>
        The political decision is calculated.  </prop>
    </propset>
    <focus entity='Mr. Clinton'/>
    <rst-rel id=r14-1 name=join
             nuc=p14-1   sat=p14-2   ref=no  />
    <rst-rel id=r14-2 name=join
             nuc=p14-2   sat=p14-3   ref=no  />
    <rst-rel id=r14-3 name=elab
             nuc=p14-1   sat=p14-4   ref=no  />
    <rst-rel id=r14-4 name=elab
             nuc=p14-2   sat=p14-5   ref=no  />
    <rst-rel id=r14-5 name=elab
             nuc=p14-1   sat=p14-6   ref=no  />
    <rst-rel id=r14-6 name=elab
             nuc=p14-2   sat=p14-7   ref=no  />
    <rst-rel id=r14-7 name=elab
             nuc=pset14-1   sat=p14-8   ref=no  />
    <trans id=tx14-1 name=rel-that nuc=p14-4 sat=p14-6 />
    <trans id=tx14-2 name=rel-that nuc=p14-5 sat=p14-7 />
    <trans id=tx14-3 name=apposition nuc=p14-1 sat=tx14-1 />
```

```
    <trans id=tx14-4 name=apposition nuc=p14-2 sat=tx14-2 />
    <trans id=tx14-5 name=conj-simp  nuc=tx14-3 sat=tx14-4 />
    <trans id=tx14-6 name=conj-simp-as-well-as   nuc=tx14-5 sat=p14-3 />
    <trans id=tx14-7 name=adj   nuc=tx14-6 sat=p14-7 />

    <seqorder valid=true />

    <conj id=c14-1 type=dist />
    <conj id=c14-2 type=dist />

    <comment text="complicated, but still follows the rule.  The adj
                   application is interesting." />

</sentence>

<sentence id=s15>
  To protect those benefit programs and still balance the budget
  without significant tax increases, he hasn't much choice but
  to pare investment spending plans.

    <propset id=pset15-1>
      <prop id=p15-1>
        Mr. Clinton hasn't much choice.  </prop>
      <prop id=p15-2>
        (but) The choice is to pare investment spending plans.  </prop>
      <prop id=p15-2>
        To protect those benefits programs.    </prop>
      <prop id=p15-3>
        To (still) balance the budget without significant tax
        increase.    </prop>
    </propset>

    <focus entity='Mr. Clinton'/>
    <rst-rel id=r15-1 name=contrast
             nuc=p15-1   sat=p15-2   ref=no  />
    <rst-rel id=r15-2 name=purpose
             nuc=p15-1   sat=p15-3   ref=no  />
    <rst-rel id=r15-3 name=purpose
             nuc=p15-1   sat=p15-4   ref=no  />

    <trans id=tx15-1 name=conj-mult-inf-vp   nuc=p15-3 sat=p15-4 />
    <trans id=tx15-2 name=pp-to   nuc=p15-1 sat=tx15-1 />
    <trans id=tx15-3 name=cue-but-vp   nuc=tx15-2 sat=p15-2 />

    <comment text="when represent r15-3 as purpose too,
                   the collective relation is not really specified.
                   There could be a propset for prop1 and prop2." />

    <seqorder valid=true />

    <conj id=c15-1 type=coll />

</sentence>

<sentence id=s16>
  "Perhaps the single biggest disappointment of the budget
  process thus far, apart from the basic mindlessness of the
  Republican economic and social policy paradigms, is that there
  will be -- at most -- significant restraint in Medicare
  spending with no structural reform," says Mr. Shapiro of the
```

```
  Progressive Policy Institute.
    <propset id=pset16-1>
      <prop id=p16-1>
        Mr. Shapiro says -X-.  </prop>
      <propset id=pset16-2>
        <prop id=p16-2>
          Perhaps the single biggest disappoint of the budget
          process is that there will be -- at most -- significant
          restraint in Medicare spending with no structural reform.
          </prop>
        <prop id=p16-3>
          (apart from) The biggest disappoint of the budget process
           is the basic mindlessness of the Republican economic and
           social policy paradigm.  </prop>
      </propset>
      <prop id=p16-4>
        Mr. Shapiro is of the Progressive Policy Institute. </prop>
    </propset>

    <focus entity='Mr. Shapiro'/>
    <rst-rel id=r16-1 name=arg
             nuc=p16-1   sat=p16-2   ref=no  />
    <rst-rel id=r16-2 name=evaluation
             nuc=p16-2   sat=p16-3   ref=no  />
    <rst-rel id=r16-3 name=elab
             nuc=p16-1   sat=p16-4   ref=no  />

    <trans id=tx16-1 name=cue-apart-from nominal=yes nuc=p16-2 sat=p16-3 />
    <trans id=tx16-2 name=arg   nuc=p16-1 sat=tx16-1 />
    <trans id=tx16-3 name=pp-of   nuc=tx16-2 sat=p16-4 />

    <seqorder valid=true />

    <conj id=c16-1 type=dist />
</sentence>
<sentence id=s17>
  The centrist Democratic think tank has been promoting a
  "cut-and-invest" strategy that would take on both corporate
  subsidies and venerable benefit programs to make room for more
  investment spending.
    <propset id=pset17-1>
      <prop id=p17-1>
        The centrist Democratic think thank has been promoting a
        "cut-and-invest" strategy.    </prop>
      <prop id=p17-2>
        The strategy would take on  corporate subsidies to
        make room for more investment spending.  </prop>
      <prop id=p17-3>
        The strategy would take on venerable benefit programs
        to make room for more investment spending.    </prop>
    </propset>

    <focus entity='think tank'/>
    <rst-rel id=r17-1 name=elab
```

```
          nuc=p17-1   sat=p17-2   ref=no  />
    <rst-rel id=r17-2 name=elab
          nuc=p17-2   sat=p17-3   ref=no  />

    <trans id=tx17-1 name=conj-simp   nuc=p17-2 sat=p17-3 />
    <trans id=tx17-2 name=rel-that   nuc=p17-1 sat=tx17-1 />

    <seqorder valid=true />

    <conj id=c17-1 type=dist />

</sentence>

<sentence id=s18>
  There were no public rallies or petitions to public officials
  when the Los Angeles Raiders headed back to Oakland, Calif.,
  and the Rams went to St. Louis.

    <propset id=pset18-1>
      <propset id=pset18-2>
      <prop id=p18-1>
        There were no public rallies.    </prop>
      <prop id=p18-2>
        There were no petitions to public officials.    </prop>
      </propset>
      <prop id=p18-3>
        (When) the Los Angeles Raiders head back to Oakland,
        Calif. </prop>
      <prop id=p18-4>
        (When) The Rams went to St. Louis.    </prop>
    </propset>

    <focus entity='-X-'/>
    <rst-rel id=r18-1 name=alternate
          nuc=p18-1   sat=p18-2   ref=no  />
    <rst-rel id=r18-2 name=circum-time
          nuc=pset18-2   sat=p18-3   ref=no  />
    <rst-rel id=r18-3 name=circum-time
          nuc=pset18-2   sat=p18-4   ref=no  />

    <trans id=tx18-1 name=disj-simp-neg   nuc=p18-1 sat=p18-2 />
    <trans id=tx18-2 name=conj-mult-sent   nuc=p18-3 sat=p18-4 />
    <trans id=tx18-3 name=cue-when   nuc=tx18-1 sat=tx18-2 />

    <comment text="- simple disjunction, with 2nd 'no' deleted" />

    <seqorder valid=true />

    <conj id=c18-1 type=dist />

</sentence>

<sentence id=s19>
  But the potential owners -- including Walt Disney Co.,
  race-track operator Hollywood Park and Los Angeles Dodgers
  baseball-team owner Peter O'Malley -- know they'll have to go
  to new lengths to attract and entertain jaded Los Angeles
  residents.

    <propset id=pset19-1>
      <propset id=pset19-2>
```

```
      <prop id=p19-1>
        The potential owners known they will have to go to new
        lengths to attract Los Angeles residents.    </prop>
      <prop id=p19-2>
        The potential owners known they will have to go to new
        lengths to entertain Los Angeles residents.    </prop>
      </propset>
      <prop id=p19-3>
        The potential owners include Walt Disney Co.    </prop>
      <prop id=p19-4>
        The potential owners include Hollywood Park.    </prop>
      <prop id=p19-5>
        The potential owners include Peter O'Malley.    </prop>
      <prop id=p19-6>
        Hollywood Park is a race-track operator. </prop>
      <prop id=p19-7>
        Peter O'Malley is Los Angeles Dodgers baseball-team owner.
      </prop>
      <prop id=p19-8>
        Los Angeles residents are jaded.   </prop>
    </propset>
    <focus entity='owners'/>
    <rst-rel id=r19-1 name=join
             nuc=p19-1   sat=p19-2   ref=no  />
    <rst-rel id=r19-2 name=elab
             nuc=pset19-2   sat=p19-3   ref=no  />
    <rst-rel id=r19-3 name=elab
             nuc=pset19-2   sat=p19-4   ref=no  />
    <rst-rel id=r19-4 name=elab
             nuc=pset19-2   sat=p19-5   ref=no  />
    <rst-rel id=r19-5 name=elab
             nuc=p19-4   sat=p19-6   ref=no  />
    <rst-rel id=r19-6 name=elab
             nuc=p19-5   sat=p19-7   ref=no  />
    <rst-rel id=r19-7 name=elab
             nuc=pset19-2   sat=p19-8   ref=no  />
    <trans id=tx19-1 name=conj-simp   nuc=p19-1 sat=p19-2 />
    <trans id=tx19-2 name=conj-simp   nuc=p19-3 sat=p19-4 />
    <trans id=tx19-3 name=conj-simp   nuc=tx19-2 sat=p19-5 />
    <trans id=tx19-4 name=prenominal   nuc=tx19-3 sat=p19-6 />
    <trans id=tx19-5 name=prenominal   nuc=tx19-4 sat=p19-7 />
    <trans id=tx19-6 name=adj   nuc=tx19-5 sat=p19-8 />
    <trans id=tx19-7 name=hyphen-vp   nuc=tx19-1 sat=tx19-6 />
    <seqorder valid=true />
    <conj id=c19-1 type=dist />
    <conj id=c19-2 type=dist />
</sentence>
<sentence id=s20>
  Just as important is the owner's ability to sell luxury seats
  and corporate suites to the local business community.
```

```
    <propset id=pset20-1>
      <prop id=p20-1>
        The owner's ability to sell luxury seats to the local
        business community is just as important.   </prop>
      <prop id=p20-2>
        The owner's ability to sell corporate suites to the local
        business community is just as important.   </prop>
    </propset>
    <focus entity='-x-'/>
    <rst-rel id=r20-1 name=join
             nuc=p20-1   sat=p20-2   ref=no  />
    <trans id=tx20-1 name=conj-simp   nuc=p20-1 sat=p20-2 />
    <seqorder valid=true />
    <conj id=c20-1 type=dist />
</sentence>
<sentence id=s21>
  The NFL and local business leaders insist they can create a
  sports frenzy in the city -- under the right conditions.
    <propset id=pset21-1>
      <prop id=p21-1>
        Under the right conditions, the NFL insist it can create a
        sports frenzy in the city.    </prop>
      <prop id=p21-2>
        Under the right conditions, the local business leaders
        insist they can create a sports frenzy in the city.
        </prop>
    </propset>
    <focus entity='NFL leaders'/>
    <rst-rel id=r21-1 name=join
             nuc=p21-1   sat=p21-2   ref=no  />
    <trans id=tx21-1 name=conj-simp   nuc=p21-1 sat=p21-2 />
    <seqorder valid=true />
    <conj id=c21-1 type=dist />
</sentence>
<sentence id=s22>
  They see many reasons for fan apathy, including the weather,
  an abundance of other recreational opportunities and the fact
  that many residents are transplanted.
    <propset id=pset22-1>
      <prop id=p22-1>
        They see many reasons for fan apathy    </prop>
      <prop id=p22-2>
        The reasons include the weather.   </prop>
      <prop id=p22-3>
        The reasons include an abundance of other recreational
        opportunities. </prop>
```

```
      <prop id=p22-4>
        The reasons include the fact that many residents are
        transplanted. </prop>
    </propset>
    <focus entity='they'/>
    <rst-rel id=r23-1 name=elab
             nuc=p23-1   sat=p23-2   ref=no  />
    <rst-rel id=r23-2 name=elab
             nuc=p23-1   sat=p23-3   ref=no  />
    <rst-rel id=r23-3 name=elab
             nuc=p23-1   sat=p23-4   ref=no  />
    <trans id=tx23-1 name=conj-simp   nuc=p23-2 sat=p23-3 />
    <trans id=tx23-2 name=conj-simp   nuc=tx23-1 sat=p23-4 />
    <trans id=tx23-3 name=rel-reduced-ing  nuc=p23-1 sat=tx23-2 />
    <seqorder valid=true />
    <conj id=c22-1 type=dist />
</sentence>
<sentence id=s23>
  That will mean building a state-of-the-art stadium brimming
  with luxury seating, say stadium planners and sports-team
  owners.
    <propset id=pset23-1>
      <prop id=p23-1>
        Stadium planners and sports-team owners say -X-.  </prop>
      <propset id=pset23-2>
      <prop id=p23-2>
        That will mean building a state-of-the-art stadium.    </prop>
      <prop id=p23-3>
        The stadium is brimming with luxury seating.    </prop>
    </propset>
    </propset>
    <focus entity='Stadium planners and sports-team owners'/>
    <rst-rel id=r23-1 name=arg
             nuc=p23-1   sat=p23-2   ref=no  />
    <rst-rel id=r23-2 name=elab
             nuc=p23-2   sat=p23-3   ref=no  />
    <trans id=tx23-1 name=rel-reduced-ing nuc=p23-2 sat=p23-3 />
    <trans id=tx23-2 name=arg   nuc=p23-1 sat=tx23-1 />
    <seqorder valid=true />
    <conj id=c23-1 type=coll />
    <comment text='- The conjunction in p23-1 is collective' />
</sentence>
<sentence id=s24>
  Hoping for a fresh start in Los Angeles, the league has
  rejected existing venues such as the Los Angeles Coliseum and
  the aging Rose Bowl and is examining four possible sites for a
  new stadium.
```

```
    <propset id=pset24-1>
    <propset id=pset24-2>
      <prop id=p24-1>
        The league has rejected existing venues.   </prop>
      <prop id=p24-2>
        The league is examining four possible sites for a new
        stadium.   </prop>
      <prop id=p24-3>
        The existing venues include the Los Angeles Coliseum.
        </prop>
      <prop id=p24-4>
        The existing venues include the Rose Bowl.    </prop>
      <prop id=p24-5>
        The Rose Bowl which is aging.    </prop>
    </propset>
      <prop id=p24-6>
        The league is hoping for a fresh start in Los Angeles.
        </prop>
    </propset>

    <focus entity='the league'/>
    <rst-rel id=r24-1 name=circum-purpose
             nuc=pset24-2   sat=p24-6   ref=no  />
    <rst-rel id=r24-2 name=evidence
             nuc=p24-1   sat=p24-3   ref=no  />
    <rst-rel id=r24-3 name=evidence
             nuc=p24-1   sat=p24-4   ref=no  />
    <rst-rel id=r24-5 name=join
             nuc=p24-1   sat=p24-2   ref=no  />
    <rst-rel id=r24-1 name=elab
             nuc=p24-4   sat=p24-5   ref=no  />

    <trans id=tx24-1 name=adj   nuc=p24-4 sat=p24-5 />
    <trans id=tx24-2 name=conj-simp   nuc=p24-3 sat=tx24-1 />
    <trans id=tx24-3 name=cue-such-as   nuc=p24-3 sat=tx24-2 />
    <trans id=tx24-4 name=conj-mult-vp   nuc=tx24-3 sat=p24-2 />
    <trans id=tx24-5 name=rel-reduced-ing   nuc=tx24-4 sat=p24-6 />

    <seqorder valid=true />

    <conj id=c24-1 type=dist />
    <conj id=c24-2 type=dist />

</sentence>

<sentence id=s25>
  If "you build a new stadium and put a decent team in there
  every week, this town will support it," says Hollywood Park
  Chairman R.D. Hubbard, who has announced plans to build a
  stadium near the company's thoroughbred racetrack, even though
  no team has committed to play there.

    <propset id=pset25-1>
      <prop id=p25-1>
        R. D. Hubbard said -X-.    </prop>
      <propset id=pset25-2>
```

```
      <prop id=p25-2>
        If you build a new stadium.    </prop>
      <prop id=p25-3>
        If you put a decent team in there every week.    </prop>
      <prop id=p25-4>
        This town will support it.    </prop>
      </propset>
      <prop id=p25-5>
        R. D. Hubbard is the chairman of Hollywood Park.  </prop>
      <prop id=p25-6>
        R. D. Hubbard has announced plans to build a stadium near
        the company's thoroughbred racetrack.    </prop>
      <prop id=p25-7>
        (even though) No team has committed to play there.    </prop>
    </propset>

    <focus entity='R. D. Hubbard'/>
    <rst-rel id=r25-1 name=arg
             nuc=p25-1   sat=p25-2   ref=no  />
    <rst-rel id=r25-2 name=join
             nuc=p25-2   sat=p25-3   ref=no  />
    <rst-rel id=r25-3 name=condition-if
             nuc=p25-4   sat=p25-3   ref=no  />
    <rst-rel id=r25-4 name=elab
             nuc=p25-1   sat=p25-5   ref=no  />
    <rst-rel id=r25-5 name=elab
             nuc=p25-1   sat=p25-6   ref=no  />
    <rst-rel id=r25-6 name=concession
             nuc=p25-6   sat=p25-7   ref=no  />

    <trans id=tx25-1 name=conj-mult-vp   nuc=p25-2 sat=p25-3 />
    <trans id=tx25-2 name=cue-if   nuc=p25-4 sat=tx25-1 />

    <trans id=tx25-3 name=arg   nuc=p25-1 sat=tx25-2 />
    <trans id=tx25-4 name=prenominal-title   nuc=tx25-3 sat=p25-5 />
    <trans id=tx25-5 name=rel-wh   nuc=tx25-4 sat=p25-6 />
    <trans id=tx25-6 name=cue-even-though   nuc=tx25-5 sat=p25-7 />

    <seqorder valid=true />

    <conj id=c25-1 type=coll />
</sentence>
<sentence id=s26>
  The extra hype may be necessary, because recent history
  suggests a flagging interest here in both the NFL and pro
  sports generally.

    <propset id=pset26-1>
      <prop id=p26-1>
        The extra hype may be necessary.   </prop>
      <prop id=p26-2>
        Recent history suggests a flagging interest here in the
        NFL.    </prop>
      <prop id=p26-3>
```

```
        Recent history suggests a flagging interest here in pro
        sports generally.     </prop>
    </propset>

    <focus entity='hype'/>
    <rst-rel id=r26-1 name=non-volitional-cause
             nuc=p26-1   sat=p26-2   ref=no  />
    <rst-rel id=r26-2 name=non-volitional-cause
             nuc=p26-1   sat=p26-3   ref=no  />

    <trans id=tx26-1 name=conj-simp   nuc=p26-2 sat=p26-3 />
    <trans id=tx26-2 name=cue-because-sent   nuc=p26-1 sat=tx26-1 />

    <seqorder valid=true />

    <conj id=c26-1 type=dist />

</sentence>

<sentence id=s27>
  Attendance at both Rams and Raiders games was in a deep slide
  when the teams decided to move.

    <propset id=pset27-1>
      <prop id=p27-1>
        Attendance at Rams games was in a deep slide.    </prop>
      <prop id=p27-2>
        (when) The Rams decided to move.     </prop>
      <prop id=p27-3>
        Attendance at Raiders games was in a deep slide.
        </prop>
      <prop id=p27-4>
        (when) the Raiders decided to move.     </prop>
    </propset>

    <focus entity='attendance'/>
    <rst-rel id=r27-1 name=circum-time
             nuc=p27-1   sat=p27-2   ref=no  />
    <rst-rel id=r27-2 name=join
             nuc=p27-1   sat=p27-3   ref=no  />
    <rst-rel id=r27-3 name=circum-time
             nuc=p27-3   sat=p27-4   ref=no  />

    <trans id=tx27-1 name=cue-when-sent   nuc=p27-1 sat=tx27-2 />
    <trans id=tx27-2 name=cue-when-sent   nuc=p27-3 sat=tx27-4 />
    <trans id=tx27-3 name=conj-complex-anaphor   nuc=tx27-1 sat=tx27-2 />

    <seqorder valid=true />

    <conj id=c27-1 type=dist />

</sentence>

<sentence id=s28>
  Sales of goods branded with Rams and Raiders logos, whose
  revenue is shared by the league as a whole, were also below
  par.

    <propset id=pset28-1>
      <prop id=p28-1>
        Sales of goods branded with Rams logos was below par.
```

```
        </prop>
      <prop id=p28-2>
        Sales of goods branded with Raiders logos was below par.
   </prop>
      <prop id=p28-3>
         Rams's revenue is shared by the league as a whole.
        </prop>
      <prop id=p28-4>
         Raiders' revenue is shared by the league as a whole.
        </prop>
    </propset>

    <focus entity='sales'/>
    <rst-rel id=r28-1 name=join
             nuc=p28-1   sat=p28-2   ref=no  />
    <rst-rel id=r28-2 name=elab
             nuc=p28-1   sat=p28-3   ref=no  />
    <rst-rel id=r28-2 name=elab
             nuc=p28-2   sat=p28-4   ref=no  />

    <trans id=tx28-1 name=rel-whose   nuc=p28-1 sat=p28-3 />
    <trans id=tx28-1 name=rel-whose   nuc=p28-2 sat=p28-4 />
    <trans id=tx28-2 name=conj-complex-anaphor   nuc=tx28-1 sat=tx28-2 />

    <seqorder valid=true />

    <conj id=c28-1 type=dist />

</sentence>

<sentence id=s29>
  But that was because of its popularity in the rest of the
  country, according to NFL Properties Inc., the league's
  licensing and merchandising arm.
    <propset id=pset29-1>
      <prop id=p29-1>
        But that was because of its popularity in the rest of the
        country, according to NFL Properties Inc.    </prop>
      <prop id=p29-2>
        NFL Properties Inc. is the league's licensing arm. </prop>
      <prop id=p29-3>
        NFL Properties Inc. is the league's merchandising arm.     </prop>
    </propset>

    <focus entity='that'/>
    <rst-rel id=r29-1 name=elab
             nuc=p29-1   sat=p29-2   ref=no  />
    <rst-rel id=r29-2 name=elab
             nuc=p29-2   sat=p29-3   ref=no  />

    <trans id=tx29-1 name=conj-simp   nuc=p29-2 sat=p29-3 />
    <trans id=tx29-2 name=apposition   nuc=p29-1 sat=tx29-2 />

    <seqorder valid=true />

    <conj id=c29-1 type=dist />

</sentence>
```

```
<sentence id=s30>
  "It seemed in Los Angeles that the link between team
  allegiance and immediate location did not match up," says
  Brian McCarthy, a spokesman for NFL Properties.
    <propset id=pset30-1>
      <prop id=p30-1>
        Brian McCarthy says -X-.  </prop>
      <prop id=p30-2>
        "It seemed in Los Angeles that the
        link between team allegiance and immediate location did
        not match up". </prop>
      <prop id=p30-3>
        Brian McCarthy is a spokesman for NFL Properties.   </prop>
    </propset>
    <focus entity='Brian McCarthy'/>
    <rst-rel id=r30-1 name=arg
             nuc=p30-1   sat=p30-2   ref=no  />
    <rst-rel id=r30-2 name=elab
             nuc=p30-1   sat=p30-3   ref=no  />
    <trans id=tx30-1 name=arg   nuc=p30-1 sat=p30-2 />
    <trans id=tx30-2 name=apposition   nuc=tx30-1 sat=p30-3 />
    <seqorder valid=true />
    <conj id=c30-1 type=coll />
</sentence>
<sentence id=s31>
  More games were broadcast, boosting total viewership, but far
  fewer fans tuned in: Football ratings on both Fox and NBC
  affiliates in Los Angeles were down about 30% from the
  previous season.
    <propset id=pset31-1>
      <prop id=p31-1>
        More games were broadcasted.    </prop>
      <prop id=p31-2>
        More games were broadcasted boosted total
        viewership.    </prop>
      <prop id=p31-3>
        Far fewer fans tuned in.    </prop>
      <prop id=p31-4>
        Football rating on Fox affiliates in Los Angeles were down
        about 30% from the previous season.    </prop>
      <prop id=p31-5>
        Football rating on NBC affiliates in Los Angeles were down
        about 30% from the previous season.    </prop>
    </propset>
    <focus entity='more games'/>
    <rst-rel id=r31-1 name=elab
             nuc=p31-1   sat=p31-2   ref=no  />
    <rst-rel id=r31-2 name=contrast
```

```
          nuc=p31-2   sat=p31-3   ref=no  />
    <rst-rel id=r31-3 name=evidence
          nuc=p31-3   sat=p31-4   ref=no  />
    <rst-rel id=r31-4 name=evidence
          nuc=p31-3   sat=p31-5   ref=no  />

    <trans id=tx31-1 name=rel-reduced-ing   nuc=p31-1 sat=p31-2 />
    <trans id=tx31-2 name=cue-but-sent   nuc=tx31-1 sat=p31-3 />
    <trans id=tx31-3 name=conj-simp   nuc=p31-4 sat=p31-5 />
    <trans id=tx31-4 name=colon-sent   nuc=tx31-2 sat=tx31-3 />

    <seqorder valid=true />

    <conj id=c31-1 type=dist />

</sentence>

<sentence id=s32>
  Local sports fans themselves, long known for their passive
  demeanor at games and propensity to leave early, don't resist
  the image.

    <propset id=pset32-1>
      <prop id=p32-1>
        Local sports fans don't resist the image.    </prop>
      <prop id=p32-2>
        Local sports fans are long known for their passive
        demeanor at games.    </prop>
      <prop id=p32-3>
        Local sports fans are long known for their propensity to
        leave early.    </prop>
    </propset>

    <focus entity='local sports fans'/>
    <rst-rel id=r32-1 name=elab
          nuc=p32-1   sat=p32-2   ref=no  />
    <rst-rel id=r32-2 name=elab
          nuc=p32-1   sat=p32-3   ref=no  />

    <trans id=tx32-1 name=conj-simp   nuc=p32-2 sat=p32-3 />
    <trans id=tx32-2 name=rel-reduced-del-wh-be   nuc=p32-1 sat=tx32-1 />

    <seqorder valid=true />

    <conj id=c32-1 type=dist />

</sentence>

<sentence id=s33>
  Four games were played in 60,000-seat Busch Stadium and four
  in new 65,700-seat Trans World Dome.

    <propset id=pset33-1>
      <prop id=p33-1>
        Four games were played in 60,000-seat Busch Stadium.
        </prop>
      <prop id=p33-2>
        Four games in new 65,700-seat Trans World Dome.
        </prop>
    </propset>
```

```
    <focus entity='four games'/>
    <rst-rel id=r33-1 name=join
             nuc=p33-1   sat=p33-2   ref=no  />

    <trans id=tx33-1 name=conj-mult-gap   nuc=p33-1 sat=p33-2 />

    <seqorder valid=true />

    <conj id=c33-1 type=dist />

</sentence>

<sentence id=s34>
  NEW YORK -- Insurance companies will turn in decent, but not
  outstanding, fourth-quarter results, as storm losses took
  their toll on property-casualty writers, and competitive
  pressures damped earnings at life and health insurers.

    <propset id=pset34-1>
      <prop id=p34-1>
        Insurance companies will turn in decent fourth-quarter
        results.   </prop>
      <prop id=p34-2>
        Insurance companies will turn in not outstanding
        fourth-quarter results.  </prop>
      <prop id=p34-3>
        Storm losses took their toll on property-casualty writers.
    </prop>
      <prop id=p34-4>
        Competitive pressures damped earnings at life insurers.
        </prop>
      <prop id=p34-5>
        Competitive pressures damped earnings at health insurers.
   </prop>
    </propset>

    <focus entity='insurance companies'/>
    <rst-rel id=r34-1 name=contrast
             nuc=p34-1   sat=p34-2   ref=no  />
    <rst-rel id=r34-2 name=non-volitional-cause
             nuc=p34-2   sat=p34-3   ref=no  />
    <rst-rel id=r34-3 name=non-volitional-cause
             nuc=p34-2   sat=p34-4   ref=no  />
    <rst-rel id=r34- name=non-volitional-cause
             nuc=p34-2   sat=p34-5   ref=no  />

    <trans id=tx34-1 name=conj-simp   nuc=p34-4 sat=p34-5 />
    <trans id=tx34-2 name=conj-mult-sent   nuc=p34-3 sat=tx34-1 />
    <trans id=tx34-3 name=cue-as-sent   nuc=p34-2 sat=tx34-2 />
    <trans id=tx34-4 name=cue-but-adj   nuc=p34-1 sat=tx34-3 />

    <seqorder valid=true />

    <conj id=c34-1 type=dist />
    <conj id=c34-2 type=dist />

</sentence>

<sentence id=s35>
  Hurricane Opal, which struck the Florida Panhandle in October,
```

```
  cost insurers $2.1 billion and ranks as the nation's
  third-costliest hurricane.
    <propset id=pset35-1>
      <propset id=pset35-2>
      <prop id=p35-1>
        Hurricane Opal cost insurers $2.1 billion.    </prop>
      <prop id=p35-2>
        Hurricane Opal ranks as the nation's third-costliest
        hurricane.     </prop>
      </propset>
      <prop id=p35-3>
        Hurricane Opal struck the Florida Panhandle in
        October.   </prop>
    </propset>
    <focus entity='Hurricane Opal'/>
    <rst-rel id=r35-1 name=join
             nuc=p35-1   sat=p35-2   ref=no  />
    <rst-rel id=r35-2 name=elab
             nuc=pset35-1   sat=p35-3   ref=no  />
    <trans id=tx35-1 name=conj-mult   nuc=p35-1 sat=p35-3 />
    <trans id=tx35-2 name=rel-wh   nuc=tx35-1 sat=p35-2 />
    <seqorder valid=true />
    <conj id=c35-1 type=dist />
</sentence>
<sentence id=s36>
  Allstate Corp., the nation's largest publicly traded insurer
  of homes and autos, said it has paid claims totaling $120
  million as a result of Opal.
    <propset id=pset36-1>
      <prop id=p36-1>
        Allstate Corp. said -X-.  </prop>
      <prop id=p36-2>
        It has paid claims totaling $120 million as a result of
        Opal. </prop>
      <prop id=p36-3>
        Allstate Corp. is the nation's largest publicly traded
        insurer of homes.    </prop>
      <prop id=p36-4>
        Allstate Corp. is the nation's largest publicly traded
        insurer of autos.    </prop>
    </propset>
    <focus entity='Allstate Corp'/>
    <rst-rel id=r36-1 name=arg
             nuc=p36-1   sat=p36-2   ref=no  />
    <rst-rel id=r36-2 name=elab
             nuc=p36-1   sat=p36-3   ref=no  />
    <rst-rel id=r36-3 name=elab
             nuc=p36-1   sat=p36-4   ref=no  />
```

```
    <trans id=tx36-1 name=arg   nuc=p36-1 sat=p36-2 />
    <trans id=tx36-2 name=conj-simp   nuc=p36-3 sat=p36-4 />
    <trans id=tx36-3 name=apposition   nuc=tx36-1 sat=tx36-2 />

    <seqorder valid=true />

    <conj id=c36-1 type=dist />
</sentence>
<sentence id=s37>
  A consensus estimate of analysts on First Call Inc. looks for
  Allstate to post operating earnings, which exclude realized
  investment gains and losses, of 79 cents a share, compared
  with 35 cents a share in the year-ago period.
    <propset id=pset37-1>
      <prop id=p37-1>
        A consensus estimate of analysts on First Call Inc. looks
        for Allstate to post operating earnings of 79 cents a
        share    </prop>
      <prop id=p37-2>
        The operating earning exclude realized investment gains.
         </prop>
      <prop id=p37-3>
        The operating earning exclude realized investment losses.
   </prop>
      <prop id=p37-4>
        79 cents a share is compared with 35 cents a share in the
        year-ago period.    </prop>
    </propset>

    <focus entity='a consensus estimate'/>
    <rst-rel id=r37-1 name=elab
             nuc=p37-1   sat=p37-2   ref=no  />
    <rst-rel id=r37-2 name=elab
             nuc=p37-1   sat=p37-3   ref=no  />
    <rst-rel id=r37-3 name=elab
             nuc=p37-1   sat=p37-4   ref=no  />

    <trans id=tx37-1 name=rel-reduced-del-wh-be   nuc=p37-1 sat=p37-4 />
    <trans id=tx37-2 name=conj-simp   nuc=p37-2 sat=p37-3 />
    <trans id=tx37-3 name=rel-wh   nuc=tx37-1 sat=tx37-2 />

    <seqorder valid=true />

    <conj id=c37-1 type=dist />

    <comment text="interesting because conj is specific for each
                   hypotactic operator, not all the hypotactic" />

</sentence>

<sentence id=s38>
  The 1994 period included charges from the Northridge, Calif.,
  earthquake of January 1994 and an early retirement program.
    <propset id=pset38-1>
      <prop id=p38-1>
        The 1994 period included charges from the Northridge,
```

```
      Calif., earthquake of January 1994.    </prop>
    <prop id=p38-2>
      The 1994 period included charges from an early retirement
      program.    </prop>
  </propset>
  <focus entity='the 1994 period'/>
  <rst-rel id=r38-1 name=join
          nuc=p38-1   sat=p38-2   ref=no  />
  <trans id=tx38-1 name=conj-simp   nuc=p38-1 sat=p38-2 />
  <seqorder valid=true />
  <conj id=c38-1 type=dist />
</sentence>
<sentence id=s39>
  As has become the norm, better results will come from the
  "select insurers," companies able to offer specialized
  products and services at a profitable price, said Steven A.
  Gavios, an analyst with Bear Stearns & Co.
  <propset id=pset39-1>
    <prop id=p39-1>
      Steven A Gavios said -X-.  </prop>
    <propset id=pset39-2>
    <prop id=p39-2>
      As has become the norm, better results will come from the
      "select insurers".    </prop>
    <prop id=p39-3>
      "Select insurers" are companies able to offer specialized
      products at a profitable price.  </prop>
    <prop id=p39-4>
      "Select insurers" are companies able to offer specialized
      services at a profitable price.  </prop>
    </propset>
    <prop id=p39-5>
      Steven A Gavios is an analyst with Bear Stearns & Co.
      </prop>
  </propset>
  <focus entity='Gavios'/>
  <rst-rel id=r39-1 name=arg
          nuc=p39-1   sat=p39-2   ref=no  />
  <rst-rel id=r39-2 name=elab
          nuc=p39-2   sat=p39-3   ref=no  />
  <rst-rel id=r39-3 name=elab
          nuc=p39-2   sat=p39-4   ref=no  />
  <rst-rel id=r39-4 name=elab
          nuc=p39-1   sat=p39-5   ref=no  />
  <trans id=tx39-1 name=conj-simp   nuc=p39-3 sat=p39-4 />
  <trans id=tx39-2 name=apposition   nuc=p39-2 sat=tx39-1 />
  <trans id=tx39-3 name=arg   nuc=p39-1 sat=tx39-2 />
  <trans id=tx39-4 name=apposition   nuc=tx39-3 sat=p39-5 />
```

```
    <seqorder valid=true />
    <conj id=c39-1 type=dist />
</sentence>
<sentence id=s40>
  Mr. Gavios looks for Chubb to post operating earnings between
  $1.75 and $1.80 a share, compared with operating earnings of
  $1.63 a share in the fourth-quarter of 1994.
    <propset id=pset40-1>
      <prop id=p40-1>
        Mr. Gavios looks for Chubb to post operating earnings
        between $1.75 and $1.80 a share.   </prop>
      <prop id=p40-2>
        Operating earnings between $1.75 and $1.80 a share is
        compared with operating earnings of $1.63 a share in the
        fourth-quarter of 1994.   </prop>
    </propset>
    <focus entity='Mr. Gavios'/>
    <rst-rel id=r40-1 name=elab
             nuc=p40-1   sat=p40-2   ref=no  />
    <trans id=tx40-1 name=rel-reduced-del-wh-be   nuc=p40-1 sat=p40-2 />
    <seqorder valid=true />
    <conj id=c40-1 type=between />
</sentence>
<sentence id=s41>
  Analysts also caution that the fourth-quarter results could
  include some unannounced additions to environmental and
  asbestos reserves, perhaps from smaller property-casualty
  insurers.
    <propset id=pset41-1>
      <prop id=p41-1>
        Analysts also caution -X-.  </prop>
      <propset id=pset41-2>
      <prop id=p41-2>
        The fourth-quarter results could include some unannounced
        additions to environmental reserves.  </prop>
      <prop id=p41-3>
        The fourth-quarter results could include some unannounced
        additions to asbestos reserves.    </prop>
      <prop id=p41-4>
        Environmental reserves is perhaps from smaller property-casualty
        insurers.  </prop>
      <prop id=p41-5>
        Asbestos reserves is perhaps from smaller property-casualty
        insurers.  </prop>
      </propset>
    </propset>
    <focus entity='analysts'/>
```

```
    <rst-rel id=r41-1 name=arg
             nuc=p41-1   sat=p41-2   ref=no  />
    <rst-rel id=r41-2 name=join
             nuc=p41-2   sat=p41-3   ref=no  />
    <rst-rel id=r41-3 name=elab
             nuc=p41-2   sat=p41-4   ref=no  />
    <rst-rel id=r41-4 name=elab
             nuc=p41-3   sat=p41-5   ref=no  />

    <trans id=tx41-1 name=rel-reduced-del-wh-be nuc=p41-2 sat=p41-4 />
    <trans id=tx41-2 name=rel-reduced-del-wh-be nuc=p41-2 sat=p41-4 />
    <trans id=tx41-3 name=conj-simp   nuc=tx41-1 sat=tx41-2 />
    <trans id=tx41-4 name=arg   nuc=p41-1 sat=tx41-3 />

    <seqorder valid=true />

    <conj id=c41-1 type=dist />
</sentence>

<sentence id=s42>
  Multiline insurers, whose business includes property-casualty,
  group health and life insurance, will also put forth moderate
  results for the fourth quarter, said Ira Zuckerman, an analyst
  with Nutmeg Securities in Westport, Conn.
    <propset id=pset42-1>
      <prop id=p42-1>
        Ira Zuckerman said -X-.  </prop>
      <propset id=pset42-2>
      <prop id=p42-2>
        Multiline insurers will also put forth moderate results for the
        fourth quarter.   </prop>
      <prop id=p42-3>
        Multiline insurers's business includes property-casualty.
   </prop>
      <prop id=p42-4>
        Multiline insurers's business includes group health.
        </prop>
      <prop id=p42-5>
        Multiline insurers's business includes life insurance.
        </prop>
      </propset>
      <prop id=p42-6>
        Ira Zuckerman is an analyst with Nutmeg Securities.
        </prop>
      <prop id=p42-7>
        Nutmeg Securities is in Westport, Conn.   </prop>
    </propset>

    <focus entity='Ira Zuckerman'/>
    <rst-rel id=r42-1 name=arg
             nuc=p42-1   sat=p42-2   ref=no  />
    <rst-rel id=r42-2 name=elab
             nuc=p42-2   sat=p42-3   ref=no  />
    <rst-rel id=r42-3 name=elab
             nuc=p42-2   sat=p42-4   ref=no  />
```

```
    <rst-rel id=r42-4 name=elab
             nuc=p42-2   sat=p42-5   ref=no  />
    <rst-rel id=r42-5 name=elab
             nuc=p42-1   sat=p42-6   ref=no  />
    <rst-rel id=r42-6 name=elab
             nuc=p42-6   sat=p42-7   ref=no  />

    <trans id=tx42-1 name=conj-simp   nuc=p42-3 sat=p42-4 />
    <trans id=tx42-2 name=conj-simp   nuc=tx42-1 sat=p42-5 />
    <trans id=tx42-3 name=rel-whose   nuc=p42-2 sat=tx42-2 />

    <trans id=tx42-4 name=arg   nuc=p42-1 sat=tx42-3 />
    <trans id=tx42-5 name=pp-in   nuc=p42-6 sat=p42-7 />
    <trans id=tx42-6 name=apposition   nuc=tx42-4 sat=tx42-5 />

    <seqorder valid=true />

    <conj id=c42-1 type=dist />
</sentence>

<sentence id=s43>
  In addition to their underperforming property-casualty
  segments, the multilines' pension business is not growing, and
  their group health margins are under competitive pressures
  from health maintenance organizations, Mr. Zuckerman said.

    <propset id=pset43-1>
      <prop id=p43-1>
        Mr. Zuckerman said -X-.  </prop>
      <propset id=pset43-2>
      <prop id=p43-2>
        Their property-casualty segments is underperforming.  </prop>
      <prop id=p43-3>
        the multilines' pension business is not growing.   </prop>
      <prop id=p43-4>
        Their group health margins are under competitive pressures from
        health maintenance organizations. </prop>
      </propset>
    </propset>

    <focus entity='Zuckerman'/>
    <rst-rel id=r43-1 name=arg
             nuc=p43-1   sat=p43-2   ref=no  />
    <rst-rel id=r43-2 name=join
             nuc=p43-2   sat=p43-3   ref=no  />
    <rst-rel id=r43-3 name=join
             nuc=p43-3   sat=p43-4   ref=no  />

    <trans id=tx43-1 name=conj-mult-sent   nuc=p43-2 sat=p43-3 />
    <trans id=tx43-2 name=cue-in-addition-nominalization nuc=tx43-1 sat=p43-4 />

    <trans id=tx43-3 name=arg   nuc=p43-1 sat=tx43-2 />

    <seqorder valid=false />

    <conj id=c43-1 type=dist />

    <comment text="The issue probably can be resolved by using another
                   propset on p43-2 and p43-3.  Now conj applied
```

```
                   before cue." />
</sentence>
<sentence id=s44>
  But Gloria Vogel, an analyst with Ladenburg Thalmann & Co., is
  more optimistic on earnings from the two multiline giants,
  Aetna Life & Casualty Corp. and Cigna Corp.
    <propset id=pset44-1>
      <propset id=pset44-2>
      <prop id=p44-1>
        Gloria Vogel is more optimistic on earnings from Aetna
        Life & Casualty Corp.    </prop>
      <prop id=p44-2>
        Gloria Vogel is more optimistic on earnings from Cigna
        Corp. </prop>
      <prop id=p44-3>
        Aetna Life & Casualty Corp. is a multiline giant.
        </prop>
      <prop id=p44-4>
        Cigna Corp. is a multiline giant.    </prop>
      </propset>
      <prop id=p44-5>
        Gloria Vogel is an analyst with Ladenburg Thalmann & Co.
      </prop>
    </propset>
    <focus entity='Vogel'/>
    <rst-rel id=r44-1 name=join
             nuc=p44-1   sat=p44-2   ref=no  />
    <rst-rel id=r44-2 name=elab
             nuc=p44-1   sat=p44-3   ref=no  />
    <rst-rel id=r44-3 name=elab
             nuc=p44-2   sat=p44-4   ref=no  />
    <rst-rel id=r44-4 name=elab
             nuc=pset44-2   sat=p44-5   ref=no  />
    <trans id=tx44-1 name=apposition   nuc=p44-1 sat=p44-3 />
    <trans id=tx44-2 name=apposition   nuc=p44-2 sat=p44-4 />
    <trans id=tx44-3 name=conj-simp-reverse nuc=tx44-1 sat=tx44-2 />
    <trans id=tx44-4 name=apposition   nuc=tx44-3 sat=p44-5 />
    <seqorder valid=true />
    <conj id=c44-1 type=dist />
</sentence>
<sentence id=s45>
  And, said Ms. Vogel, "Cigna's group health operations tend to
  be strongest in the fourth quarter and I see no reason to
  believe this year will be any different."
    <propset id=pset45-1>
      <prop id=p45-1>
        Ms. Vogel said -X-.  </prop>
      <propset id=pset45-2>
      <prop id=p45-2>
```

```
        "Cigna's group health operations tend to be strongest in the
        fourth quarter"   </prop>
      <prop id=p45-3>
        "I see no reason to believe this year will be any different."
        </prop>
    </propset>
    </propset>
    <focus entity='Vogel'/>
    <rst-rel id=r45-1 name=arg
             nuc=p45-1   sat=p45-2   ref=no  />
    <rst-rel id=r45-2 name=join
             nuc=p45-2   sat=p45-3   ref=no  />
    <trans id=tx45-1 name=conj-mult-sent   nuc=p45-2 sat=p45-3 />
    <trans id=tx45-2 name=arg   nuc=p45-1 sat=tx45-1 />
    <seqorder valid=true />
    <conj id=c45-1 type=dist />
    <conj id=c45-2 type=dist />
</sentence>
<sentence id=s46>
  "Revenues have generally been weak, and nobody's crowing about
  year-end renewals," he said.
    <propset id=pset46-1>
      <prop id=p46-1>
        He said -X-.  </prop>
      <propset id=pset46-2>
      <prop id=p46-2>
         "Revenues have generally been weak."    </prop>
      <prop id=p46-2>
        "nobody's crowing about year-end renewals"
        </prop>
    </propset>
    </propset>
    <focus entity='he'/>
    <rst-rel id=r46-1 name=arg
             nuc=p46-1   sat=p46-2   ref=no  />
    <rst-rel id=r46-2 name=join
             nuc=p46-2   sat=p46-3   ref=no  />
    <trans id=tx46-1 name=conj-mult-sent   nuc=p46-2 sat=p46-3 />
    <trans id=tx46-2 name=arg   nuc=p46-1 sat=tx46-1 />
    <seqorder valid=true />
    <conj id=c46-1 type=dist />
</sentence>
<sentence id=s47>
  A consensus estimate on First Call shows analysts looking for
  operating earnings of 31 cents a share in the fourth quarter,
```

```
  compared with a loss equivalent to $2.61 a year ago, due to
  restructuring, settlement and reserve charges.
    <propset id=pset47-1>
      <prop id=p47-1>
        A consensus estimate on First Call shows analysts looking
        for operating earnings of 31 cents a share in the fourth
        quarter.   </prop>
      <prop id=p47-2>
        operating earning of 31 cents a share is compared with a
        loss equivalent to $2.61 a year ago.    </prop>
      <prop id=p47-3>
        Operating earnings of 31 cents a share is due to restructuring,
        settlement and reserve charges.
        </prop>
    </propset>
    <focus entity='a consensus estimate'/>
    <rst-rel id=r47-1 name=elab
             nuc=p47-1   sat=p47-2   ref=no  />
    <rst-rel id=r47-2 name=elab
             nuc=p47-1   sat=p47-3   ref=no  />
    <trans id=tx47-1 name=rel-reduced-del-wh-be   nuc=p47-1 sat=p47-2 />
    <trans id=tx47-2 name=rel-reduced-del-wh-be   nuc=tx47-1 sat=p47-3 />
    <seqorder valid=true />
    <conj id=c47-1 type=dist />
</sentence>
<sentence id=s48>
  MADRID -- The global offer of an 11% stake in Repsol SA of
  Spain is expected to begin on Wednesday, in a move that will
  raise as much as 130 billion pesetas ($1.08 billion) and
  reduce the government's shareholding to 10% on the oil,
  natural gas and chemical group.
    <propset id=pset48-1>
      <prop id=p48-1>
        The global offer of an 11% stake in Repsol SA of Spain is
        expected to begin on Wednesday.    </prop>
      <prop id=p48-2>
        It is a move that will raise as much as 130 billion pesetas
        ($1.08 billion).   </prop>
      <prop id=p48-3>
        It is a move that will reduce the government's shareholding to
        10% on the oil, natural gas and chemical group.  </prop>
    </propset>
    <focus entity='global offer'/>
    <rst-rel id=r48-1 name=circum-location
             nuc=p48-1   sat=p48-2   ref=no  />
    <rst-rel id=r48-2 name=circum-location
             nuc=p48-1   sat=p48-3   ref=no  />
    <trans id=tx48-1 name=conj-mult-vp   nuc=p48-2 sat=p48-3 />
```

```
    <trans id=tx48-2 name=pp-in   nuc=p48-1 sat=tx48-1 />

    <seqorder valid=true />

    <conj id=c48-1 type=dist />
    <conj id=c48-2 type=coll />
</sentence>

<sentence id=s49>
  Moreover, the Spanish stock exchange has rallied over the past
  month, fueled by an inflow of foreign investment and
  expectations of lower interest rates.
    <propset id=pset49-1>
      <prop id=p49-1>
        (Moreover) the Spanish stock exchange has rallied over the
        past month.    </prop>
      <prop id=p49-2>
        The rally is fueled by an inflow of foreign investment.  </prop>
      <prop id=p49-3>
        The rally is fueled by expectations of lower interest
        rates. </prop>
    </propset>

    <focus entity='stock exchange'/>
    <rst-rel id=r49-1 name=elab
             nuc=p49-1   sat=p49-2   ref=no  />
    <rst-rel id=r49-2 name=elab
             nuc=p49-1   sat=p49-3   ref=no  />

    <trans id=tx49-1 name=conj-simp   nuc=p49-2 sat=p49-3 />
    <trans id=tx49-2 name=rel-reduced-del-wh-be   nuc=p49-1 sat=tx49-1 />

    <seqorder valid=true />

    <conj id=c49-1 type=dist />

</sentence>

<sentence id=s50>
  Analysts said the flotation, to be comanaged by Banco Bilbao
  Vizcaya of Spain and Goldman Sachs, had all the makings of a
  success.
    <propset id=pset50-1>
      <prop id=p50-1>
        Analysts said -X-.  </prop>
      <propset id=pset50-2>
      <prop id=p50-2>
        the flotation had all the makings of a success.  </prop>
      <prop id=p50-3>
        The flotation will be comanaged by Banco Bilbao Vizcaya of
        Spain and Goldman Sachs.     </prop>
      </propset>
    </propset>

    <focus entity='Analysts'/>
    <rst-rel id=r50-1 name=arg
             nuc=p50-1   sat=p50-2   ref=no  />
```

```
    <rst-rel id=r50-2 name=elab
             nuc=p50-2   sat=p50-3   ref=no  />

    <trans id=tx50-1 name=rel-inf   nuc=p50-2 sat=p50-3 />

    <trans id=tx50-2 name=arg   nuc=p50-1 sat=tx50-1 />

    <seqorder valid=true />

    <conj id=c50-1 type=coll />

</sentence>

<sentence id=s51>
  Mr. Lepetit, head of the bank since 1988, helped Suez Chairman
  Gerard Mestrallet draw up the restructuring, and his "views
  converged completely" with Mr. Mestrallet's, Suez said in a
  statement.

    <propset id=pset51-1>
      <prop id=p51-1>
        Suez said in a statement.  </prop>
      <propset id=pset51-2>
      <prop id=p51-2>
        Mr. Lepetit helped Gerard Mestrallet draw up the restructuring.
        </prop>
      <prop id=p51-3>
        his "views converged completely" with Mr. Mestrallet's.  </prop>
      <prop id=p51-4>
        Mr. Lepetit is head of the bank since 1988.    </prop>
      <prop id=p51-5>
       Gerard Mestrallet is Suez Chairman.  </prop>
      </propset>
    </propset>

    <focus entity='Suez'/>
    <rst-rel id=r51-1 name=arg
             nuc=p51-1   sat=p51-2   ref=no  />
    <rst-rel id=r51-2 name=join
             nuc=p51-2   sat=p51-3   ref=no  />
    <rst-rel id=r51-3 name=elab
             nuc=p51-2   sat=p51-4   ref=no  />
    <rst-rel id=r51-4 name=elab
             nuc=p51-2   sat=p51-5   ref=no  />

    <trans id=tx51-1 name=prenominal-title   nuc=p51-2 sat=p51-5 />
    <trans id=tx51-2 name=apposition   nuc=tx51-1 sat=p51-4 />
    <trans id=tx51-3 name=conj-mult-sent   nuc=tx51-2 sat=p51-3 />

    <trans id=tx51-4 name=apposition   nuc=p51-1 sat=tx51-3 />

    <seqorder valid=true />

    <conj id=c51-1 type=dist />

</sentence>

<sentence id=s52>
  Nevertheless, "it seemed preferable that the new strategy be
  implemented by a new manager and Lepetit, from his
  perspective, preferred to develop his career independently,"
```

```
  the statement also said.
    <propset id=pset52-1>
      <prop id=p52-1>
        the statement also said -X-  </prop>
      <propset id=pset52-2>
      <prop id=p52-2>
        (Nevertheless) it seemed preferable that the new strategy
        be implemented by a new manager.   </prop>
      <prop id=p52-3>
        Lepetit, from his perspective, preferred to develop his
        career independently,</prop>
      </propset>
    </propset>
    <focus entity='the statement'/>
    <rst-rel id=r52-1 name=arg
             nuc=p52-1   sat=p52-2   ref=no  />
    <rst-rel id=r52-2 name=join
             nuc=p52-2   sat=p52-3   ref=no  />
    <trans id=tx52-1 name=conj-mult-sent   nuc=p52-2 sat=p52-3 />
    <trans id=tx52-2 name=arg   nuc=p52-1 sat=tx52-1 />
    <seqorder valid=true />
    <conj id=c52-1 type=dist />
</sentence>
<sentence id=s53>
  Mr. Mestrallet last July was placed at the top of Suez after
  institutional shareholders ousted Gerard Worms for failing to
  stop large losses at the industrial and financial holding
  company.
    <propset id=pset53-1>
      <prop id=p53-1>
        Mr. Mestrallet last July was placed at the top of Suez
        </prop>
      <prop id=p53-2>
        (after) institutional shareholders ousted Gerard Worms.
        </prop>
      <prop id=p53-3>
        (for) Gerard Worms failed to stop large losses at the
        industrial and financial holding company.      </prop>
    </propset>
    <focus entity='Mr. Mestrallet'/>
    <rst-rel id=r53-1 name=sequence
             nuc=p53-1   sat=p53-2   ref=no  />
    <rst-rel id=r53-2 name=non-volitional-cause
             nuc=p53-2   sat=p53-3   ref=no  />
    <trans id=tx53-1 name=cue-for-vp-ing  nuc=p53-2 sat=p53-3 />
    <trans id=tx53-2 name=cue-after-sent   nuc=p53-1 sat=tx53-1 />
    <seqorder valid=true />
```

```
    <conj id=c53-1 type=coll />
</sentence>
<sentence id=s54>
  Mr. Mestrallet quickly promised to restructure Banque
  Indosuez, whose exposure to the disastrous French real-estate
  market contributed to massive losses at Suez in 1994 and the
  1995 first half.
    <propset id=pset54-1>
      <prop id=p54-1>
        Mr. Mestrallet quickly promised to restructure Banque
        Indosuez.    </prop>
      <prop id=p54-2>
        Banque Indosueq's exposure to the disastrous French
        real-estate market contributed to massive losses at Suez
        in 1994.   </prop>
      <prop id=p54-3>
        Banque Indosueq's exposure to the disastrous French
        real-estate market contributed to massive losses at Suez
        in the 1995 first half.   </prop>
    </propset>
    <focus entity='Mr. Mestrallet'/>
    <rst-rel id=r54-1 name=elab
             nuc=p54-1   sat=p54-2   ref=no  />
    <rst-rel id=r54-2 name=elab
             nuc=p54-1   sat=p54-3   ref=no  />
    <trans id=tx54-1 name=conj-simp   nuc=p54-2 sat=p54-3 />
    <trans id=tx54-2 name=rel-whose   nuc=p54-1 sat=tx54-1 />
    <seqorder valid=true />
    <conj id=c54-1 type=dist />
</sentence>
<sentence id=s55>
  Banque Indosuez had a 1994 net loss of 1.1 billion francs
  ($222.4 million) following write-offs for bad real-estate
  loans and investments of 2.4 billion francs.
    <propset id=pset55-1>
      <prop id=p55-1>
        Banque Indosuez had a 1994 net loss of 1.1 billion francs
        ($222.4 million).    </prop>
      <prop id=p55-2>
        (following) Banque Indosuez write-offed bad real-estate
        loans and investments of 2.4 billion francs. </prop>
    </propset>
    <focus entity='Banque Indosuez'/>
    <rst-rel id=r55-1 name=sequence
             nuc=p55-1   sat=p55-2   ref=no  />
    <trans id=tx55-1 name=cue-following-np nominal=yes  nuc=p55-1 sat=p55-2 />
    <seqorder valid=true />
```

```
    <conj id=c55-1 type=coll />
</sentence>
<sentence id=s56>
  Under the three-year contract, Source Media will provide
  national news, weather, sports and other programming available
  through a phone number in the Yellow Pages directory.
    <propset id=pset56-1>
      <prop id=p56-1>
        Under the three-year contract, Source Media will provide
        national news available through a phone number in the
        Yellow Pages directory.   </prop>
      <prop id=p56-2>
        Under the three-year contract, Source Media will provide
        weather available through a phone number in the Yellow
        Pages directory.    </prop>
      <prop id=p56-3>
        Under the three-year contract, Source Media will provide
        sports available through a phone number in the Yellow
        Pages directory.    </prop>
      <prop id=p56-4>
        Under the three-year contract, Source Media will provide
        other programming available through a phone number in the
        Yellow Pages directory.   </prop>
    </propset>

    <focus entity='Source Media'/>
    <rst-rel id=r56-1 name=join
             nuc=p56-1   sat=p56-2   ref=no  />
    <rst-rel id=r56-2 name=join
             nuc=p56-2   sat=p56-3   ref=no  />
    <rst-rel id=r56-3 name=join
             nuc=p56-3   sat=p56-4   ref=no  />

    <trans id=tx56-1 name=conj-simp   nuc=p56-1 sat=p56-2 />
    <trans id=tx56-2 name=conj-simp   nuc=tx56-1 sat=p56-3 />
    <trans id=tx56-3 name=conj-simp   nuc=tx56-2 sat=p56-4 />

    <comment text="- existential quantification, 'a phone number'" />

    <seqorder valid=true />

    <conj id=c56-1 type=dist />
</sentence>
<sentence id=s57>
  HONG KONG -- If money talks, then property developers and
  property-related firms are kicking up quite a din as they tap
  the Hong Kong stock market for cash.
    <propset id=pset57-1>
      <prop id=p57-1>
       (If) Money talks.    </prop>
      <propset id=pset57-2>
```

```
      <prop id=p57-2>
        Property developers are kick up quite a din    </prop>
      <prop id=p57-3>
        property-related firms are kick up quite a din </prop>
      <prop id=p57-4>
        Property developers tap the Hong Kong stock market for
        cash.    </prop>
      <prop id=p57-5>
        property-related firms tap the Hong Kong stock market for cash.
        </prop>
      </propset>
    </propset>
    <focus entity='property developers'/>
    <rst-rel id=r57-1 name=condition-if
             nuc=p57-1   sat=p57-2   ref=no  />
    <rst-rel id=r57-2 name=join
             nuc=p57-2   sat=p57-3   ref=no  />
    <rst-rel id=r57-3 name=non-volitional-cause
             nuc=p57-2   sat=p57-4   ref=no  />
    <rst-rel id=r57-4 name=non-volitional-cause
             nuc=p57-3   sat=p57-5   ref=no  />
    <trans id=tx57-1 name=cue-as-sent   nuc=p57-2 sat=p57-4 />
    <trans id=tx57-2 name=cue-as-sent   nuc=p57-3 sat=p57-5 />
    <trans id=tx57-3 name=conj-complex-anaphor   nuc=tx57-1 sat=tx57-2 />
    <trans id=tx57-4 name=cue-if   nuc=tx57-3 sat=p57-1 />
    <seqorder valid=true />
    <conj id=c57-1 type=dist />
</sentence>
<sentence id=s58>
  Property development and investment group Sun Hung Kai
  Properties raised 4.04 billion Hong Kong dollars (US$522
  million) in a share placement last week.
    <propset id=pset58-1>
      <prop id=p58-1>
        Sun Hung Kai Properties raised 4.04 billion Hong Kong
        dollars in a share placement last week.   </prop>
      <prop id=p58-2>
        4.04 billion Hong Kong dollar is US$522 million.
        </prop>
      <prop id=p58-3>
        Sun Hung Kai Properties is a property development and
        investment group.    </prop>
    </propset>
    <focus entity='Sun Hung Kai'/>
    <rst-rel id=r58-1 name=elab
             nuc=p58-1   sat=p58-2   ref=no  />
    <rst-rel id=r58-2 name=elab
             nuc=p58-1   sat=p58-3   ref=no  />
```

```
    <trans id=tx58-1 name=prenominal-title   nuc=p58-1 sat=p58-3 />
    <trans id=tx58-2 name=parenthesis-apposition   nuc=tx58-1 sat=p58-2 />

    <seqorder valid=true />

    <conj id=c58-1 type=coll />

</sentence>

<sentence id=s59>
  That deal came hot on the heels of a HK$3.24 billion share
  placement by China-backed Citic Pacific, which has interests
  in property development, airlines and telecommunications.

    <propset id=pset59-1>
      <prop id=p59-1>
        That deal came hot on the heels of a HK$3.24 billion share
        placement by China-backed Citic Pacific.   </prop>
      <prop id=p59-2>
        Citic Pacific has interests in property development.
        </prop>
      <prop id=p59-3>
        Citic Pacific has interests in airlines.   </prop>
      <prop id=p59-4>
        Citic Pacific has interests in telecommunications.
        </prop>
    </propset>

    <focus entity='that deal'/>
    <rst-rel id=r59-1 name=elab
             nuc=p59-1   sat=p59-2   ref=no  />
    <rst-rel id=r59-2 name=elab
             nuc=p59-1   sat=p59-3   ref=no  />
    <rst-rel id=r59-3 name=elab
             nuc=p59-1   sat=p59-4   ref=no  />

    <trans id=tx59-1 name=conj-simp   nuc=p59-2 sat=p59-3 />
    <trans id=tx59-2 name=conj-simp   nuc=tx59-3 sat=p59-4 />
    <trans id=tx59-3 name=rel-wh   nuc=p59-1 sat=tx59-2 />

    <seqorder valid=true />

    <conj id=c59-1 type=dist />

</sentence>

<sentence id=s60>
  "Given that the Hong Kong market has been buoyant and more
  red-blooded than for quite some time."

    <propset id=pset60-1>
      <prop id=p60-1>
        Given -X-.    </prop>
      <propset id=pset60-2>
      <prop id=p60-2>
        The Hong Kong market has been buoyant than for quite some
        time.    </prop>
      <prop id=p60-3>
        The Hong Kong market has been red-blooded than for quite
        some time.    </prop>
```

```
      </propset>
    </propset>
    <focus entity='-X-'/>
    <rst-rel id=r60-1 name=arg
             nuc=p60-1   sat=p60-2   ref=no  />
    <rst-rel id=r60-2 name=join
             nuc=p60-2   sat=p60-3   ref=no  />
    <trans id=tx60-1 name=conj-simp   nuc=p60-2 sat=p60-3 />
    <trans id=tx60-2 name=arg   nuc=p60-1 sat=tx60-1 />
    <seqorder valid=true />
    <conj id=c60-1 type=dist />
</sentence>
<sentence id=s61>
  "People are upbeat about property developer stocks, in
  particular Sun Hung Kai and Cheung Kong (Holdings), because of
  the anticipation that as interest rates fall, property prices
  and the stock price of property developers will go up," Ms.
  Ting said.
    <propset id=pset61-1>
      <prop id=p61-1>
        Ms. Ting said -X-.    </prop>
      <propset id=pset61-2>
      <prop id=p61-2>
        People are upbeat about property developer stocks.
        </prop>
      <prop id=p61-3>
        in particular people are upbeat about Sun Hung Kai.
        </prop>
      <prop id=p61-4>
        in particular people are upbeat about Cheung Kong
        (Holdings).    </prop>
      <propset id=pset61-3>
      <prop id=p61-5>
        People anticipate -X-. </prop>
      <propset id=pset61-4>
      <prop id=p61-6>
        Interest rates fall.    </prop>
      <prop id=p61-7>
        Property prices will go up.    </prop>
      <prop id=p61-8>
        the stock price of property developers will go up.
        </prop>
      </propset>
      </propset>
    </propset>
    </propset>
    <focus entity='Ms. Ting'/>
    <rst-rel id=r61-1 name=arg
```

```
          nuc=p61-1   sat=p61-2   ref=no  />
<rst-rel id=r61-2 name=elab
          nuc=p61-2   sat=p61-3   ref=no  />
<rst-rel id=r61-3 name=elab
          nuc=p61-2   sat=p61-4   ref=no  />
<rst-rel id=r61-4 name=non-volitional-cause
          nuc=p61-4   sat=p61-5   ref=no  />
<rst-rel id=r61-5 name=arg
          nuc=p61-5   sat=p61-6   ref=no  />
<rst-rel id=r61-6 name=sequence
          nuc=p61-6   sat=p61-7   ref=no  />
<rst-rel id=r61-7 name=sequence
          nuc=p61-6   sat=p61-8   ref=no  />

<trans id=tx61-1 name=conj-simp   nuc=p61-7 sat=p61-8 />
<trans id=tx61-2 name=pp-as-np   nuc=p61-6 sat=tx61-1 />
<trans id=tx61-3 name=arg   nuc=p61-5 sat=tx61-2 />

<trans id=tx61-4 name=conj-simp   nuc=p61-3 sat=p61-4 />
<trans id=tx61-5 name=pp-in   nuc=p61-2 sat=tx61-4 />
<trans id=tx61-6 name=arg   nuc=p61-1 sat=tx61-5 />

<trans id=tx61-7 name=cue-because-of   nuc=tx61-6 sat=tx61-3 />

<seqorder valid=true />

<conj id=c61-1 type=dist />
<conj id=c61-2 type=dist />
</sentence>
<sentence id=s62>
  Yet, like many analysts and investors, she believes the recent
  moves to raise money look opportunistic.
    <propset id=pset62-1>
      <prop id=p62-1>
        she believes the recent moves to raise money look
        opportunistic.  </prop>
      <prop id=p62-2>
        She is like many analysts.    </prop>
      <prop id=p62-3>
        She is like many investors.    </prop>
    </propset>

    <focus entity='she'/>

    <rst-rel id=r62-1 name=elab
              nuc=p62-1   sat=p62-2   ref=no  />
    <rst-rel id=r62-2 name=elab
              nuc=p62-1   sat=p62-3   ref=no  />

    <trans id=tx62-1 name=conj-simp   nuc=p62-2 sat=p62-3 />
    <trans id=tx62-2 name=pp-like   nuc=p62-1 sat=tx62-1 />

    <seqorder valid=true />

    <conj id=c62-1 type=dist />
</sentence>
```

```
<sentence id=s63>
  "Sun Hung Kai Properties is one of the most conservatively
  managed property companies, they always have had a healthy
  cash flow, and have good planning," with a relatively low net
  debt-to-equity ratio, she said.
    <propset id=pset63-1>
      <prop id=p63-1>
        She said -X-    </prop>
      <propset id=pset63-2>
      <prop id=p63-2>
        Sun Hung Kai Properties is one of the most conservatively
        managed property companies.    </prop>
      <prop id=p63-3>
        Sun Hung Kai Properties always have had a healthy cash
        flow.    </prop>
      <prop id=p63-4>
        Sun Hung Kai Properties have good planning.    </prop>
      <prop id=p63-5>
        Sun Hung Kai Properties have a relatively low net
        debt-to-equity ratio.  </prop>
      </propset>
    </propset>
    <focus entity='she'/>
    <rst-rel id=r63-1 name=arg
             nuc=p63-1   sat=p63-2   ref=no  />
    <rst-rel id=r63-2 name=join
             nuc=p63-2   sat=p63-3   ref=no  />
    <rst-rel id=r63-3 name=join
             nuc=p63-3   sat=p63-4   ref=no  />
    <rst-rel id=r63-4 name=join
             nuc=p63-4   sat=p63-5   ref=no  />
    <trans id=tx63-1 name=conj-mult-sent   nuc=p63-2 sat=p63-3 />
    <trans id=tx63-2 name=conj-mult-vp   nuc=tx63-1 sat=p63-4 />
    <trans id=tx63-3 name=conj-with   nuc=tx63-2 sat=p63-5 />
    <trans id=tx63-4 name=arg   nuc=p63-1 sat=tx63-3 />
    <seqorder valid=false explain="'with' should be applied first" />
    <conj id=c63-1 type=dist />
    <comment text="- join 'with', maybe an elaboration relation " />
</sentence>
<sentence id=s64>
  But analysts pointed out that property developers are eager to
  build up their war chests in the expectation that some
  attractive sites and projects will come up over the next few
  months.
    <propset id=pset64-1>
      <prop id=p64-1>
        Analysts pointed out -X-  </prop>
    <propset id=pset64-2>
```

```
      <prop id=p64-2>
        Property developers are eager to build up their war
        chests.  </prop>
      <prop id=p64-3>
        (in the expectation) Some attractive sites will come up
        over the next few months.     </prop>
      <prop id=p64-4>
        (in the expectation) Some attractive projects will come up
        over the next few months.   </prop>
    </propset>
    </propset>
    <focus entity='analysts'/>
    <rst-rel id=r64-1 name=arg
             nuc=p64-1   sat=p64-2   ref=no  />
    <rst-rel id=r64-2 name=purpose
             nuc=p64-2   sat=p64-3   ref=no  />
    <rst-rel id=r64-3 name=purpose
             nuc=p64-2   sat=p64-4   ref=no  />
    <trans id=tx64-1 name=conj-simp   nuc=p64-3 sat=p64-4 />
    <trans id=tx64-2 name=cue-in-expectation-that nuc=p64-2 sat=tx64-1 />
    <trans id=tx64-3 name=arg   nuc=p64-1 sat=tx64-2 />
    <seqorder valid=true />
    <conj id=c64-1 type=dist />
    <comment text="deletion of 2nd adjective" />

</sentence>
<sentence id=s65>
  The Dow Jones World Stock Index was at 132.63, down 0.18,
  reflecting lower Americas and Asia/Pacific markets.
    <propset id=pset65-1>
      <prop id=p65-1>
        The Dow Jones World Stock Index was at 132.63,</prop>
      <prop id=p65-2>
        The Index is down 0.18.  </prop>
      <prop id=p65-3>
        The Index reflects lower Americas and Asia/Pacific
        markets.    </prop>
    </propset>
    <focus entity='The Dow Jones World Stock Index'/>
    <rst-rel id=r65-1 name=elab
             nuc=p65-1   sat=p65-2   ref=no  />
    <rst-rel id=r65-2 name=elab
             nuc=p65-1   sat=p65-3   ref=no  />
    <trans id=tx65-1 name=rel-reduced-del-wh-be   nuc=p65-1 sat=p65-2 />
    <trans id=tx65-2 name=rel-reduced-ing   nuc=tx65-1 sat=p65-3 />
    <seqorder valid=true />
    <conj id=c65-1 type=coll />
```

```
    <comment text="This examples shows that relative-reduced-del-wh-be
                   appears before relative-reduced-ing, more concise first" />
</sentence>
<sentence id=s66>
  The Tokyo market opened higher after U.S. stocks rose Thursday
  but descended as investors, including institutions, took
  profits amid some anxiety about the new coalition government
  and its ability to form policy.
    <propset id=pset66-1>
      <propset id=pset66-2>
      <prop id=p66-1>
        The Tokyo market opened higher.  </prop>
      <prop id=p66-2>
        (after) U.S. stocks rose Thursday.    </prop>
      <prop id=p66-3>
        (but) Tokyo stocks descended.    </prop>
      <prop id=p66-4>
        (as) investors took profits amid some anxiety about the new
        coalition government.    </prop>
      <prop id=p66-5>
        (as) investors took profits amid some anxiety about the new
        coalition government' ability to form policy. </prop>
      </propset>
      <prop id=p66-6>
        investors includes institutions.  </prop>
    </propset>
    <focus entity='The Tokyo market'/>
    <rst-rel id=r66-1 name=sequence
             nuc=p66-1   sat=p66-2   ref=no  />
    <rst-rel id=r66-2 name=contrast
             nuc=p66-1   sat=p66-3   ref=no  />
    <rst-rel id=r66-3 name=non-volitional-cause
             nuc=p66-3   sat=p66-4   ref=no  />
    <rst-rel id=r66-4 name=non-volitional-cause
             nuc=p66-3   sat=p66-5   ref=no  />
    <rst-rel id=r66-5 name=elab
             nuc=pset66-2   sat=p66-6   ref=no  />
    <trans id=tx66-1 name=conj-simp   nuc=p66-4 sat=p66-5 />
    <trans id=tx66-2 name=cue-as-sent   nuc=p66-3 sat=tx66-1 />
    <trans id=tx66-3 name=cue-after-sent   nuc=p66-1 sat=p66-2 />
    <trans id=tx66-4 name=cue-but-vp   nuc=tx66-3 sat=p66-3 />
    <trans id=tx66-5 name=rel-reduced-ing   nuc=tx66-4 sat=p66-6 />
    <seqorder valid=true />
    <conj id=c66-1 type=dist />
    <comment text="interesting of using elaboration 'include', instead
                   of conj" />
```

```
</sentence>
<sentence id=s67>
  The market found early support from Wall Street's recovery
  Thursday, higher government bond prices in Germany and
  Britain, and data showing strong British retail sales last
  month.
    <propset id=pset67-1>
      <prop id=p67-1>
        The market found early support from Wall Street's recovery
        Thursday,    </prop>
      <prop id=p67-2>
        early support includes higher government bond prices in
        Germany.   </prop>
      <prop id=p67-3>
        early support includes higher government bond prices in
        Britain.   </prop>
      <prop id=p67-4>
        early support includes data shows strong British retail sales last month.
        </prop>
    </propset>
    <focus entity='The market'/>
    <rst-rel id=r67-1 name=elab
             nuc=p67-1   sat=p67-2   ref=no  />
    <rst-rel id=r67-2 name=elab
             nuc=p67-1   sat=p67-3   ref=no  />
    <rst-rel id=r67-3 name=elab
             nuc=p67-1   sat=p67-4   ref=no  />
    <trans id=tx67-1 name=conj-simp   nuc=p67-2 sat=p67-3 />
    <trans id=tx67-2 name=conj-simp   nuc=tx67-1 sat=p67-4 />
    <trans id=tx67-3 name=apposition   nuc=p67-1 sat=tx67-2 />
    <seqorder valid=true />
    <conj id=c67-1 type=dist />
    <conj id=c67-2 type=dist />
</sentence>
<sentence id=s68>
  In Paris, stocks strengthened, helped by technical factors and
  continued hopes for lower interest rates in France and
  elsewhere in Europe.
    <propset id=pset68-1>
      <prop id=p68-1>
        In Paris, stocks strengthened, </prop>
      <prop id=p68-2>
        The strengthening is helped by technical factors.  </prop>
      <prop id=p68-3>
        The strengthening is helped by continued hopes for lower
        interest rates in France.    </prop>
      <prop id=p68-4>
        The strengthening is helped by continued hopes for lower
        interest rates elsewhere in Europe.      </prop>
```

```
    </propset>

    <focus entity='stocks'/>
    <rst-rel id=r68-1 name=elab
             nuc=p68-1   sat=p68-2   ref=no  />
    <rst-rel id=r68-2 name=elab
             nuc=p68-1   sat=p68-3   ref=no  />
    <rst-rel id=r68-3 name=elab
             nuc=p68-1   sat=p68-4   ref=no  />

    <trans id=tx68-1 name=conj-simp   nuc=p68-3 sat=p68-4 />
    <trans id=tx68-2 name=conj-simp   nuc=p68-2 sat=tx68-2 />
    <trans id=tx68-3 name=rel-reduced-del-wh-be   nuc=p68-1 sat=tx68-2 />

    <seqorder valid=true />

    <conj id=c68-1 type=dist />
    <conj id=c68-2 type=dist />

    <comment text="- interesting use of 'elsewhere', need to refer to some
                   particular thing" />
</sentence>
<sentence id=s69>
  In Madrid, the market rose handily after Spain's central bank
  cut an interest rate and the bond market rallied.
    <propset id=pset69-1>
      <prop id=p69-1>
        In Madrid, the market rose handily.    </prop>
      <prop id=p69-2>
        (after) Spain's central bank cut an interest rate.
        </prop>
      <prop id=p69-3>
        (after) the bond market rallied.    </prop>
    </propset>

    <focus entity='the market'/>
    <rst-rel id=r69-1 name=sequence
             nuc=p69-1   sat=p69-2   ref=no  />
    <rst-rel id=r69-2 name=sequence
             nuc=p69-1   sat=p69-3   ref=no  />

    <trans id=tx69-1 name=conj-mult-sent   nuc=p69-2 sat=p69-3 />
    <trans id=tx69-2 name=cue-after-sent   nuc=p69-1 sat=tx69-1 />

    <seqorder valid=true />

    <conj id=c69-1 type=dist />

</sentence>
<sentence id=s70>
  In Mexico City, stocks slid 2.1%, pushed down by the telephone
  sector and by profit-taking.
    <propset id=pset70-1>
      <prop id=p70-1>
        In Mexico City, stocks slid 2.1%.    </prop>
      <prop id=p70-2>
        The stock is pushed down by the telephone sector.
```

```
      </prop>
    <prop id=p70-3>
      The stock is pushed down by profit-taking. </prop>
  </propset>

  <focus entity='stocks'/>
  <rst-rel id=r70-1 name=elab
           nuc=p70-1   sat=p70-2   ref=no  />
  <rst-rel id=r70-2 name=elab
           nuc=p70-1   sat=p70-3   ref=no  />

  <trans id=tx70-1 name=conj-simp   nuc=p70-2 sat=p70-3 />
  <trans id=tx70-2 name=rel-reduced-del-wh-be   nuc=p70-1 sat=tx70-2 />

  <seqorder valid=true />

  <conj id=c70-1 type=dist />

</sentence>
<sentence id=s71>
  In Buenos Aires, the market climbed 1.5%, as foreign and
  Argentine investors continued to be optimistic about an
  economic recovery.
    <propset id=pset71-1>
      <prop id=p71-1>
        In Buenos Aires, the market climbed 1.5%.    </prop>
      <prop id=p71-2>
        (as) foreign investors continued to be optimistic about an
        economic recovery.     </prop>
      <prop id=p71-3>
        (as) Argentine investors continued to be optimistic about
        an economic recovery. </prop>
    </propset>

    <focus entity='the market'/>
    <rst-rel id=r71-1 name=non-volitional-cause
             nuc=p71-1   sat=p71-2   ref=no  />
    <rst-rel id=r71-2 name=non-volitional-cause
             nuc=p71-1   sat=p71-3   ref=no  />

    <trans id=tx71-1 name=conj-simp   nuc=p71-2 sat=p71-3 />
    <trans id=tx71-2 name=cue-as-sent   nuc=p71-1 sat=tx71-1 />

    <seqorder valid=true />

    <conj id=c71-1 type=dist />

</sentence>
<sentence id=s72>
  In Manila, the market surged 3.1%, led by commercial and
  industrial issues.
    <propset id=pset72-1>
      <prop id=p72-1>
        In Manila, the market surged 3.1%.    </prop>
      <prop id=p72-2>
        The surge is led by commercial.  </prop>
      <prop id=p72-3>
```

```
        The surge is led by industrial issues.
        </prop>
    </propset>

    <focus entity='the market'/>
    <rst-rel id=r72-1 name=elab
             nuc=p72-1   sat=p72-2   ref=no  />
    <rst-rel id=r72-2 name=elab
             nuc=p72-1   sat=p72-3   ref=no  />

    <trans id=tx72-1 name=conj-simp   nuc=p72-2 sat=p72-3 />
    <trans id=tx72-2 name=rel-reduced-del-wh-be   nuc=p72-1 sat=tx72-1 />

    <seqorder valid=true />

    <conj id=c72-1 type=dist />

</sentence>

<sentence id=s73>
  WASHINGTON -- The Supreme Court agreed to decide whether
  prosecutors violate the Constitution when they give drug
  defendants a one-two punch: prosecuting them criminally and
  suing to take away crime-related property.

    <propset id=pset73-1>
      <prop id=p73-1>
        The Supreme Court agreed to decide whether prosecutors
        violate the Constitution.    </prop>
      <prop id=p73-2>
        (when) They give drug defendants a one-two
        punch. </prop>
      <prop id=p73-3>
        Prosecutors prosecuted them criminally.  </prop>
      <prop id=p73-4>
        Prosecutors sued to take away crime-related property.
        </prop>
    </propset>

    <focus entity='the Supreme Court'/>
    <rst-rel id=r73-1 name=circum-time
             nuc=p73-1   sat=p73-2   ref=no  />
    <rst-rel id=r73-2 name=elab
             nuc=p73-2   sat=p73-3   ref=no  />
    <rst-rel id=r73-3 name=elab
             nuc=p73-3   sat=p73-4   ref=no  />

    <trans id=tx73-1 name=conj-mult-vp   nuc=p73-3 sat=p73-4 />
    <trans id=tx73-2 name=rel-reduced-ing-colon  nuc=p73-2 sat=tx73-1 />
    <trans id=tx73-3 name=cue-when-sent   nuc=p73-1 sat=tx73-2 />

    <seqorder valid=true />

    <conj id=c73-1 type=dist />
</sentence>

<sentence id=s74>
  The high court for several years has been cautiously reining
  in federal and state authorities who have tried to hit drug
  traffickers in the pocketbook.
```

```
    <propset id=pset74-1>
      <prop id=p74-1>
        The high court for several years has been cautiously
        reining in federal authorities.  </prop>
      <prop id=p74-2>
        The high court for several years has been cautiously
        reining in state authorities.    </prop>
      <prop id=p74-3>
        Federal authorities has tried to hit drug traffickers in
        the pocketbook.   </prop>
      <prop id=p74-4>
        State authorities has tried to hit drug traffickers in the
        pocketbook.     </prop>
    </propset>

    <focus entity='The high court'/>
    <rst-rel id=r74-1 name=join
             nuc=p74-1   sat=p74-2   ref=no  />
    <rst-rel id=r74-2 name=elab
             nuc=p74-1   sat=p74-3   ref=no  />
    <rst-rel id=r74-3 name=elab
             nuc=p74-2   sat=p74-4   ref=no  />

    <trans id=tx74-1 name=rel-wh   nuc=p74-1 sat=p74-3 />
    <trans id=tx74-2 name=rel-wh   nuc=p74-2 sat=p74-4 />
    <trans id=tx74-3 name=conj-complex-anaphor   nuc=tx74-1 sat=tx74-2 />

    <seqorder valid=true />

    <conj id=c74-1 type=dist />
</sentence>
<sentence id=s75>
  Skeptics of forfeiture actions contend that since the
  escalation of the "war on drugs" in the late 1980s,
  authorities have abused their power to grab defendants'
  property and money -- for example, targeting alleged criminals
  based on their bank accounts or real estate.

    <propset id=pset75-1>
      <prop id=p75-1>
        Skeptics of forfeiture actions contend that -X-.
        </prop>
      <propset id=pset75-2>
      <propset id=pset75-3>
      <prop id=p75-2>
        (since) the "war on drugs" escalated in the late
        1980s. </prop>
      <prop id=p75-3>
        Authorities have abused their power to grab defendants'
        property.    </prop>
      <prop id=p75-4>
        Authorities have abused their power to grab defendants'
        money. </prop>
      </propset>
```

```
      <prop id=p75-5>
        For example, targeting alleged criminals based on their
        bank accounts.   </prop>
      <prop id=p75-6>
        (or) For example, targeting alleged criminals based on
        their real estate.    </prop>
      </propset>
    </propset>

    <focus entity='Skeptics'/>
    <rst-rel id=r75-1 name=arg
             nuc=p75-1   sat=p75-3   ref=no  />
    <rst-rel id=r75-2 name=circum-time
             nuc=p75-3   sat=p75-2   ref=no  />
    <rst-rel id=r75-3 name=join
             nuc=p75-3   sat=p75-4   ref=no  />
    <rst-rel id=r75-4 name=evidence
             nuc=pset75-3   sat=p75-5   ref=no  />
    <rst-rel id=r75-5 name=evidence
             nuc=pset75-3   sat=p75-6   ref=no  />

    <trans id=tx75-1 name=conj-simp   nuc=p75-3 sat=p75-4 />
    <trans id=tx75-2 name=cue-since-nominal nominal=yes nuc=tx75-1 sat=p75-2 />
    <trans id=tx75-3 name=conj-simp   nuc=p75-5 sat=p75-6 />
    <trans id=tx75-4 name=rel-reduced-ing-cue-for-example nuc=tx75-2 sat=tx75-3 /
    <trans id=tx75-5 name=arg   nuc=p75-1 sat=tx75-4 />

    <seqorder valid=true />

    <conj id=c75-1 type=dist />
</sentence>

<sentence id=s76>
  Although civil forfeiture most frequently has been used -- and
  criticized -- in connection with drug prosecutions, the
  Supreme Court's ruling could also affect some white-collar
  cases.

    <propset id=pset76-1>
      <prop id=p76-1>
        The Supreme Court's ruling could also affect some
        white-collar cases.     </prop>
      <prop id=p76-2>
        Although civil forfeiture most frequently has been used in
        connection with drug prosecutions    </prop>
      <prop id=p76-3>
        Although civil forfeiture most frequently has been
        criticized in connection with drug prosecutions  </prop>
    </propset>

    <focus entity='the Supreme Court's ruling'/>
    <rst-rel id=r76-1 name=concession
             nuc=p76-1   sat=p76-2   ref=no  />
    <rst-rel id=r76-2 name=concession
             nuc=p76-1   sat=p76-3   ref=no  />

    <trans id=tx76-1 name=conj-simp   nuc=p76-2 sat=p76-3 />
```

```
    <trans id=tx76-2 name=cue-although-sent   nuc=p76-1 sat=tx76-1 />
    <seqorder valid=true />
    <conj id=c76-1 type=dist />
</sentence>
<sentence id=s77>
  The cases before the justices involve federal drug
  prosecutions from California and Michigan.
    <propset id=pset77-1>
      <prop id=p77-1>
        The cases before the justices involve federal drug
        prosecutions from California.    </prop>
      <prop id=p77-2>
        The cases before the justices involve federal drug
        prosecutions from Michigan.     </prop>
    </propset>
    <focus entity='the cases'/>
    <rst-rel id=r77-1 name=join
             nuc=p77-1   sat=p77-2   ref=no  />
    <trans id=tx77-1 name=conj-simp   nuc=p77-1 sat=p77-2 />
    <seqorder valid=true />
    <conj id=c77-1 type=dist />
</sentence>
<sentence id=s78>
  The California case concerns two men convicted in 1992 of
  making methamphetamine and laundering the proceeds through
  front corporations.
    <propset id=pset78-1>
      <prop id=p78-1>
        The California case concerns two men.
      <prop id=p78-2>
        The men are convicted in 1992.</prop>
      <prop id=p78-3>
        The men made methamphetamine.  </prop>
      <prop id=p78-4>
        The men laundered the proceeds through front corporations.
        </prop>
    </propset>
    <focus entity='case'/>
    <rst-rel id=r78-1 name=elab
             nuc=p78-1   sat=p78-2   ref=no  />
    <rst-rel id=r78-2 name=elab
             nuc=p78-2   sat=p78-3   ref=no  />
    <rst-rel id=r78-3 name=elab
             nuc=p78-2   sat=p78-4   ref=no  />
    <trans id=tx78-1 name=conj-mult-vp   nuc=p78-3 sat=p78-4 />
    <trans id=tx78-2 name=pp-of   nuc=p78-2 sat=tx78-1 />
    <trans id=tx78-3 name=rel-reduced-del-wh-be   nuc=p78-1 sat=tx78-2 />
```

```
    <seqorder valid=true />

    <conj id=c78-1 type=dist />

    <comment text="- 'front corporate' is a concept" />

</sentence>

<sentence id=s79>
  Other federal appeals courts have concluded that civil
  forfeiture suits can be viewed as intertwined with criminal
  prosecutions, and therefore don't necessarily violate the
  double-jeopardy clause.

    <propset id=pset79-1>
      <prop id=p79-1>
        Other federal appeals courts have concluded that -X-.
        </prop>
      <propset id=pset79-2>
      <prop id=p79-2>
        civil forfeiture suits can be viewed as intertwined with
        criminal prosecutions. </prop>
      <prop id=p79-3>
        (therefore) civil forfeiture don't necessarily violate
        the double-jeopardy clause.     </prop>
      </propset>
    </propset>

    <focus entity='appeals courts'/>
    <rst-rel id=r79-1 name=arg
             nuc=p79-1   sat=p79-2   ref=no  />
    <rst-rel id=r79-2 name=non-volitional-result
             nuc=p79-2   sat=p79-3   ref=no  />

    <trans id=tx79-1 name=cue-therefore-conj-vp   nuc=p79-2 sat=p79-3 />
    <trans id=tx79-2 name=arg   nuc=p79-1 sat=tx79-1 />

    <seqorder valid=true />

    <conj id=c79-1 type=dist />

</sentence>

<sentence id=s80>
  This approach makes the prosecutor's job harder, though,
  because the burden of proof and other procedural rules are
  more stringent in criminal proceedings.

    <propset id=pset80-1>
      <prop id=p80-1>
        This approach makes the prosecutor's job harder (though).
   </prop>
      <prop id=p80-2>
        (because) the burden of proof is more stringent in
        criminal proceedings. </prop>
      <prop id=p80-3>
        (because) Other procedural rules are more stringent in criminal
        proceedings.  </prop>
    </propset>
```

```
    <focus entity='approach'/>
    <rst-rel id=r80-1 name=non-volitional-cause
             nuc=p80-1   sat=p80-2   ref=no  />
    <rst-rel id=r80-2 name=non-volitional-cause
             nuc=p80-1   sat=p80-3   ref=no  />

    <trans id=tx80-1 name=conj-simp   nuc=p80-2 sat=p80-3 />
    <trans id=tx80-2 name=cue-because-sent   nuc=p80-1 sat=tx80-1 />

    <seqorder valid=true />

    <conj id=c80-1 type=dist />

    <comment text="- interesting use of 'other'" />
</sentence>

<sentence id=s81>
  Although some prosecutors have already begun wrapping
  forfeiture into criminal indictments, the Justice Department
  asserted in its appeal to the Supreme Court that without the
  flexibility to use civil suits, state and federal authorities
  will lose a vital weapon against criminals.

    <propset id=pset81-1>
      <prop id=p81-1>
        The Justice Department asserted in its appeal to the
        Supreme Court that -X-.  </prop>
      <propset id=pset81-2>
      <prop id=p81-2>
        without the flexibility to use civil suits, state
        authorities will lose a vital weapon against criminals.
         </prop>
      <prop id=p81-3>
        without the flexibility to use civil suits, federal
        authorities will lose a vital weapon against criminals.
         </prop>
      </propset>
      <prop id=p81-4>
        (Although) some prosecutors have already begun wrapping
        forfeiture into criminal indictments.    </prop>
    </propset>

    <focus entity='The Justice Department'/>
    <rst-rel id=r81-1 name=arg
             nuc=p81-1   sat=p81-2   ref=no  />
    <rst-rel id=r81-2 name=join
             nuc=p81-2   sat=p81-3   ref=no  />
    <rst-rel id=r81-3 name=concession
             nuc=p81-1   sat=p81-4   ref=no  />

    <trans id=tx81-1 name=conj-simp   nuc=p81-2 sat=p81-3 />
    <trans id=tx81-2 name=arg   nuc=p81-1 sat=tx81-1 />
    <trans id=tx81-3 name=cue-although-sent   nuc=tx81-2 sat=p81-4 />

    <seqorder valid=true />

    <conj id=c81-1 type=dist />
```

```
</sentence>
<sentence id=s82>
  The department also argued that civil forfeiture serves a
  remedial, not punitive, purpose and thus isn't covered by the
  double jeopardy clause.
    <propset id=pset82-1>
      <prop id=p82-1>
        The department also argued -X-.  </prop>
      <propset id=pset82-2>
      <propset id=pset82-3>
      <prop id=p82-2>
        Civil forfeiture serves a remedial purpose.    </prop>
      <prop id=p82-3>
        Civil forfeiture does not serves a punitive purpose. </prop>
      </propset>
      <prop id=p82-3>
        (thus) civil forfeiture isn't covered by the
        double jeopardy clause.  </prop>
      </propset>
    </propset>
    <focus entity='The department'/>
    <rst-rel id=r82-1 name=arg
             nuc=p82-1   sat=p82-2   ref=no  />
    <rst-rel id=r82-2 name=contrast
             nuc=p82-2   sat=p82-3   ref=no  />
    <rst-rel id=r82-3 name=non-volitional-result
             nuc=p82-3   sat=p82-4   ref=no  />
    <trans id=tx82-1 name=conj-simp-neg   nuc=p82-2 sat=p82-3 />
    <trans id=tx82-2 name=cue-thus-conj-vp   nuc=tx82-1 sat=p82-3 />
    <trans id=tx82-3 name=arg   nuc=p82-1 sat=tx82-2 />
    <seqorder valid=true />
    <conj id=c82-1 type=dist />
</sentence>
<sentence id=s83>
  TRW Inc. said it formed a joint venture with Rane Ltd. of
  Madras, India, to make and sell automotive seat belts in
  India.
    <propset id=pset83-1>
      <prop id=p83-1>
        TRW Inc. said -X-.    </prop>
      <propset id=pset83-2>
      <propset id=pset83-3>
      <prop id=p83-2>
        it formed a joint venture with Rane Ltd to make automotive
        seat belts in India.    </prop>
      <prop id=p83-3>
        it formed a joint venture with Rane Ltd to sell automotive
        seat belts in India.    </prop>
      </propset>
```

```
      <prop id=p83-4>
        Rane Ltd. is in Madras, India. </prop>
    </propset>
    </propset>

    <focus entity='TRW'/>
    <rst-rel id=r83-1 name=arg
             nuc=p83-1   sat=pset83-2   ref=no  />
    <rst-rel id=r83-2 name=join
             nuc=p83-2   sat=p83-3   ref=no  />
    <rst-rel id=r83-3 name=elab
             nuc=pset83-3   sat=p83-4   ref=no  />

    <trans id=tx83-1 name=conj-simp   nuc=p83-2 sat=p83-3 />
    <trans id=tx83-2 name=pp-of  nuc=tx83-1 sat=p83-4 />
    <trans id=tx83-3 name=arg   nuc=p83-1 sat=tx83-2 /

    <seqorder valid=true />

    <conj id=c83-1 type=dist />

</sentence>

<sentence id=s84>
  Cleveland-based TRW sells high-technology products and
  services to the automotive, aerospace, defense and information
  markets.
    <propset id=pset84-1>
      <prop id=p84-1>
        Cleveland-based TRW sells high-technology products to the
        automotive market.    </prop>
      <prop id=p84-2>
        Cleveland-based TRW sells high-technology products to the
        aerospace market.    </prop>
      <prop id=p84-3>
        Cleveland-based TRW sells high-technology products to the
        defense market.    </prop>
      <prop id=p84-4>
        Cleveland-based TRW sells high-technology products to the
        information markets.     </prop>
      <prop id=p84-5>
        Cleveland-based TRW sells high-technology services to the
        automotive market.    </prop>
      <prop id=p84-6>
        Cleveland-based TRW sells high-technology services to the
        aerospace market.    </prop>
      <prop id=p84-7>
        Cleveland-based TRW sells high-technology services to the
        defense market.    </prop>
      <prop id=p84-8>
        Cleveland-based TRW sells high-technology services to the
        information markets.     </prop>
    </propset>
```

```
    <focus entity='TRW'/>
    <rst-rel id=r84-1 name=join
             nuc=p84-1   sat=p84-2   ref=no  />
    <rst-rel id=r84-2 name=join
             nuc=p84-2   sat=p84-3   ref=no  />
    <rst-rel id=r84-3 name=join
             nuc=p84-3   sat=p84-4   ref=no  />
    <rst-rel id=r84-4 name=join
             nuc=p84-4   sat=p84-5   ref=no  />
    <rst-rel id=r84-5 name=join
             nuc=p84-5   sat=p84-6   ref=no  />
    <rst-rel id=r84-6 name=join
             nuc=p84-6   sat=p84-7   ref=no  />
    <rst-rel id=r84-7 name=join
             nuc=p84-7   sat=p84-8   ref=no  />

    <trans id=tx84-1 name=conj-simp   nuc=p84-1 sat=p84-2 />
    <trans id=tx84-2 name=conj-simp   nuc=tx84-1 sat=p84-3 />
    <trans id=tx84-3 name=conj-simp   nuc=tx84-2 sat=p84-4 />
    <trans id=tx84-4 name=conj-simp   nuc=p84-5 sat=p84-6 />
    <trans id=tx84-5 name=conj-simp   nuc=tx84-4 sat=p84-7 />
    <trans id=tx84-6 name=conj-simp   nuc=tx84-5 sat=p84-8 />
    <trans id=tx84-7 name=conj-simp   nuc=tx84-3 sat=tx84-6 />

    <seqorder valid=true />

    <conj id=c84-1 type=dist />
    <conj id=c84-2 type=dist />

    <comment text="- This is an interesting case to deciding the order
                  of conjunction for inter-conj operators.
                  - We can also apply propset to simplify input, but
                  this is a more reasonable DB input representation." />
</sentence>
<sentence id=s85>
  Italian leisure-wear maker Benetton SpA said it signed a
  three-year agreement, through United Optical, with Brazilian
  eyeglass company Tecnol for the manufacture under license of
  eyeglasses and sunglasses bearing the brand names Benetton and
  Benetton Formula in Brazil, Argentina, Paraguay and Uruguay.

    <propset id=pset85-1>
      <prop id=p85-1>
        Benetton SpA said -X-. </prop>

      <propset id=pset85-2>
      <propset id=pset85-3>
      <prop id=p85-2>
        Benetton signed a three-year agreement with Tecnol through
        United Optical. </prop>
      <prop id=p85-3>
        Tecnol is a Brazilian eyeglass company.  </prop>
      <prop id=p85-4>
        The three-year agreement is for the manufacture under
```

```
  license of eyeglasses.  </prop>
<prop id=p85-5>
  The three-year agreement is for the manufacture under
  license of sunglasses.  </prop>
</propset>
<prop id=p85-6>
  The eyeglasses bear the brand names Benetton in
  Brazil.  </prop>
<prop id=p85-7>
  The eyeglasses bear the brand names Benetton in
  Argentina.      </prop>
<prop id=p85-8>
  The eyeglasses bear the brand names Benetton in
  Paraguay.    </prop>
<prop id=p85-9>
  The eyeglasses bear the brand names Benetton in
  Uruguay.    </prop>
<prop id=p85-10>
  The eyeglasses bear the brand names Benetton Formula in
  Brazil.  </prop>
<prop id=p85-11>
  The eyeglasses bear the brand names Benetton Formula in
  Argentina.    </prop>
<prop id=p85-12>
  The eyeglasses bear the brand names Benetton Formula in
  Paraguay.    </prop>
<prop id=p85-13>
  The eyeglasses bear the brand names Benetton Formula in
  Uruguay.    </prop>
<prop id=p85-14>
  The sunglasses bear the brand names Benetton in
  Brazil.  </prop>
<prop id=p85-15>
  The sunglasses bear the brand names Benetton in
  Argentina.      </prop>
<prop id=p85-16>
  The sunglasses bear the brand names Benetton in
  Paraguay.    </prop>
<prop id=p85-17>
  The sunglasses bear the brand names Benetton in
  Uruguay.    </prop>
<prop id=p85-18>
  The sunglasses bear the brand names Benetton Formula in
  Brazil.  </prop>
<prop id=p85-19>
  The sunglasses bear the brand names Benetton Formula in
  Argentina.    </prop>
<prop id=p85-20>
  The sunglasses bear the brand names Benetton Formula in
  Paraguay.    </prop>
```

```
  <prop id=p85-21>
    The sunglasses bear the brand names Benetton Formula in
    Uruguay.    </prop>
  </propset>
  <prop id=p85-22>
    Benetton SpA is an Italian leisure-wear maker.  </prop>
</propset>
<focus entity='Benetton SpA'/>
<rst-rel id=r85-1 name=arg
         nuc=p85-1   sat=pset85-2   ref=no  />
<rst-rel id=r85-2 name=elab
         nuc=p85-2   sat=p85-3   ref=no  />
<rst-rel id=r85-3 name=elab
         nuc=p85-2   sat=p85-4   ref=no  />
<rst-rel id=r85-4 name=elab
         nuc=p85-2   sat=p85-5   ref=no  />
<rst-rel id=r85-5 name=elab
         nuc=p85-4   sat=p85-6   ref=no  />
<rst-rel id=r85-6 name=elab
         nuc=p85-4   sat=p85-7   ref=no  />
<rst-rel id=r85-7 name=elab
         nuc=p85-4   sat=p85-8   ref=no  />
<rst-rel id=r85-8 name=elab
         nuc=p85-4   sat=p85-9   ref=no  />
<rst-rel id=r85-9 name=elab
         nuc=p85-4   sat=p85-10   ref=no  />
<rst-rel id=r85-10 name=elab
         nuc=p85-4   sat=p85-11   ref=no  />
<rst-rel id=r85-11 name=elab
         nuc=p85-4   sat=p85-12   ref=no  />
<rst-rel id=r85-12 name=elab
         nuc=p85-4   sat=p85-13   ref=no  />
<rst-rel id=r85-13 name=elab
         nuc=p85-3   sat=p85-14   ref=no  />
<rst-rel id=r85-14 name=elab
         nuc=p85-3   sat=p85-15   ref=no  />
<rst-rel id=r85-15 name=elab
         nuc=p85-3   sat=p85-16   ref=no  />
<rst-rel id=r85-16 name=elab
         nuc=p85-3   sat=p85-17   ref=no  />
<rst-rel id=r85-17 name=elab
         nuc=p85-3   sat=p85-18   ref=no  />
<rst-rel id=r85-18 name=elab
         nuc=p85-3   sat=p85-19   ref=no  />
<rst-rel id=r85-19 name=elab
         nuc=p85-3   sat=p85-20   ref=no  />
<rst-rel id=r85-20 name=elab
         nuc=p85-3   sat=p85-21   ref=no  />
<rst-rel id=r85-21 name=elab
         nuc=p85-1   sat=p85-22   ref=no  />

<trans id=tx85-1 name=prenominal-title nuc=p85-2 sat=p85-3 />
```

```
    <trans id=tx85-2 name=conj-simp   nuc=p85-4 sat=p85-5 />
    <trans id=tx85-3 name=pp-for   nuc=tx85-1 sat=tx85-2 />

    <trans id=tx85-4 name=conj-simp   nuc=p85-6 sat=p85-7 />
    <trans id=tx85-5 name=conj-simp   nuc=tx85-4 sat=p85-8 />
    <trans id=tx85-6 name=conj-simp   nuc=tx85-5 sat=p85-9 />

    <trans id=tx85-7 name=conj-simp   nuc=p85-10 sat=p85-11 />
    <trans id=tx85-8 name=conj-simp   nuc=tx85-7 sat=p85-12 />
    <trans id=tx85-9 name=conj-simp   nuc=tx85-8 sat=p85-13 />

    <trans id=tx85-10 name=conj-simp-nested   nuc=tx85-6 sat=tx85-9 />

    <trans id=tx85-11 name=conj-simp   nuc=p85-14 sat=p85-15 />
    <trans id=tx85-12 name=conj-simp   nuc=tx85-12 sat=p85-16 />
    <trans id=tx85-13 name=conj-simp   nuc=tx85-13 sat=p85-17 />

    <trans id=tx85-14 name=conj-simp   nuc=p85-18 sat=p85-19 />
    <trans id=tx85-15 name=conj-simp   nuc=tx85-14 sat=p85-20 />
    <trans id=tx85-16 name=conj-simp   nuc=tx85-15 sat=p85-21 />

    <trans id=tx85-17 name=conj-simp-nested   nuc=tx85-13 sat=tx85-16 />

    <trans id=tx85-18 name=conj-simp-nested   nuc=tx85-10 sat=tx85-17 />

    <trans id=tx85-19 name=rel-reduced-ing   nuc=tx85-3 sat=tx85-18 />

    <trans id=tx85-20 name=arg   nuc=p85-1 sat=tx85-19 />
    <trans id=tx85-21 name=prenominal-title   nuc=tx85-20 sat=p85-22 />

    <seqorder valid=true />

    <conj id=c85-1 type=dist />
    <conj id=c85-2 type=dist />
    <conj id=c85-3 type=dist />

</sentence>

<sentence id=s86>
  Under the pact, Tecnol will produce some of the frames locally
  and will import sunglasses.
    <propset id=pset86-1>
      <prop id=p86-1>
        Under the pact, Tecnol will produce some of the frames
        locally.   </prop>
      <prop id=p86-2>
        Under the pact, Tecnol will import sunglasses.  </prop>
    </propset>

    <focus entity='Tecnol'/>

    <rst-rel id=r86-1 name=join
             nuc=p86-1   sat=p86-2   ref=no  />

    <trans id=tx86-1 name=conj-mult-vp   nuc=p86-1 sat=p86-2 />

    <seqorder valid=true />

    <conj id=c86-1 type=dist />

</sentence>
```

```
<sentence id=s87>
  Tecnol, whose sales amounted to $40 million in 1995, holds a
  20% market share in Brazil and also has distribution networks
  in Argentina, Paraguay and Uruguay.
    <propset id=pset87-1>
      <prop id=p87-1>
        Tecnol  holds a 20% market share in Brazil.    </prop>
      <prop id=p87-2>
        Tecnol also has distribution networks in Argentina.  </prop>
      <prop id=p87-3>
        Tecnol also has distribution networks in Paraguay.   </prop>
      <prop id=p87-4>
        Tecnol also has distribution networks in Uruguay.    </prop>
      <prop id=p87-5>
        Tecnol's sales amounted to $40 million in 1995, </prop>
    </propset>
    <focus entity='Tecnol'/>
    <rst-rel id=r87-1 name=join
             nuc=p87-1   sat=p87-2   ref=no  />
    <rst-rel id=r87-2 name=join
             nuc=p87-1   sat=p87-3   ref=no  />
    <rst-rel id=r87-3 name=join
             nuc=p87-1   sat=p87-4   ref=no  />
    <rst-rel id=r87-4 name=elab
             nuc=p87-1   sat=p87-5   ref=no  />
    <trans id=tx87-1 name=rel-whose   nuc=p87-1 sat=p87-5 />
    <trans id=tx87-2 name=conj-simp   nuc=p87-2 sat=p87-3 />
    <trans id=tx87-3 name=conj-simp   nuc=tx87-2 sat=p87-4 />
    <trans id=tx87-4 name=conj-mult-vp   nuc=tx87-1 sat=tx87-3 />
    <seqorder valid=true />
    <conj id=c87-1 type=dist />
    <conj id=c87-2 type=dist />
</sentence>
<sentence id=s88>
  United Optical produces and distributes Benetton brand-name
  frames and sunglasses.
    <propset id=pset88-1>
      <prop id=p88-1>
        United Optical produces Benetton brand-name frames.
        </prop>
      <prop id=p88-2>
        United Optical produces Benetton brand-name sunglasses.
         </prop>
      <prop id=p88-3>
        United Optical distributes Benetton brand-name frames.
        </prop>
      <prop id=p88-4>
        United Optical distributes Benetton brand-name sunglasses.
     </prop>
```

```
    </propset>
    <focus entity='United Optical'/>
    <rst-rel id=r88-1 name=join
             nuc=p88-1   sat=p88-2   ref=no  />
    <rst-rel id=r88-2 name=join
             nuc=p88-2   sat=p88-3   ref=no  />
    <rst-rel id=r88-3 name=join
             nuc=p88-3   sat=p88-4   ref=no  />
    <trans id=tx88-1 name=conj-simp   nuc=p88-1 sat=p88-2 />
    <trans id=tx88-2 name=conj-simp   nuc=p88-3 sat=p88-4 />
    <trans id=tx88-3 name=conj-simp   nuc=tx88-1 sat=tx88-2 />
    <seqorder valid=true />
    <conj id=c88-1 type=dist />
    <conj id=c88-2 type=dist />
</sentence>
<sentence id=s89>
  "Tomkins and Gates are working together to resolve these
  matters before contracts are fully executed," Tomkins said in
  a statement.
    <propset id=pset89-1>
      <prop id=p89-1>
        Tomkins said in a statement:     </prop>
      <propset id=pset89-2>
      <prop id=p89-2>
        "Tomkins and Gates are working together to resolve these
        matters."   </prop>
      <prop id=p89-3>
        "(before) contracts are fully executed."  </prop>
    </propset>
    </propset>
    <focus entity='Tomkins'/>
    <rst-rel id=r89-1 name=arg
             nuc=p89-1   sat=p89-2   ref=no  />
    <rst-rel id=r89-2 name=circum-time
             nuc=p89-2   sat=p89-3   ref=no  />
    <trans id=tx89-1 name=cue-before-sent   nuc=p89-2 sat=p89-3 />
    <trans id=tx89-2 name=arg   nuc=p89-1 sat=tx89-1 />
    <seqorder valid=true />
    <conj id=c89-1 type=coll />
</sentence>
<sentence id=s90>
  The departure of Mr. Griffin, 32 years old, leaves Tiger's
  founder and chief executive officer, Julian Robertson, without
  an heir apparent at the $7 billion investment partnership.
    <propset id=pset90-1>
      <prop id=p90-1>
```

```
      The departure of Mr. Griffin leaves Julian Robertson
      without an heir apparent at the $7 billion investment
      partnership.    </prop>
    <prop id=p90-2>
      Mr. Griffin is a 32 years old. </prop>
    <prop id=p90-3>
      Julian Robertson is Tiger's founder.    </prop>
    <prop id=p90-4>
      Julian Robertson is Tiger's chief executive officer.
      </prop>
  </propset>

  <focus entity='The departure'/>
  <rst-rel id=r90-1 name=elab
           nuc=p90-1   sat=p90-2   ref=no  />
  <rst-rel id=r90-2 name=elab
           nuc=p90-1   sat=p90-3   ref=no  />
  <rst-rel id=r90-3 name=elab
           nuc=p90-1   sat=p90-4   ref=no  />

  <trans id=tx90-1 name=conj-simp   nuc=p90-2 sat=p90-3 />
  <trans id=tx90-2 name=prenominal-title   nuc=p90-1 sat=tx90-1 />
  <trans id=tx90-3 name=rel-reduced-del-wh-be nuc=tx90-2 sat=p90-2 />

  <seqorder valid=true />

  <conj id=c90-1 type=dist />

</sentence>

<sentence id=s91>
  Mr. Griffin's move follows a second disappointing year in a
  row for Tiger and other large hedge funds.

  <propset id=pset91-1>
    <prop id=p91-1>
      Mr. Griffin's move follows a second disappointing year in
      a row for Tiger.   </prop>
    <prop id=p91-2>
      Mr. Griffin's move follows a second disappointing year in
      a row for other large hedge funds.     </prop>
  </propset>

  <focus entity='Mr. Griffin'/>
  <rst-rel id=r91-1 name=join
           nuc=p91-1   sat=p91-2   ref=no  />

  <trans id=tx91-1 name=conj-simp   nuc=p91-1 sat=p91-2 />

  <seqorder valid=true />

  <conj id=c91-1 type=dist />

</sentence>

<sentence id=s92>
  Mr. Robertson will assume Mr. Griffin's responsibilities,
  which have included hiring investment personnel and making
  many investment decisions.

  <propset id=pset92-1>
```

```
      <prop id=p92-1>
        Mr. Robertson will assume Mr. Griffin's responsibilities.
  </prop>
      <prop id=p92-2>
        Mr. Griffin's responsibilities have included hiring
        investment personnel.    </prop>
      <prop id=p92-3>
        Mr. Griffin's responsibilities have included making many
        investment decisions. </prop>
    </propset>

    <focus entity='Mr. Robertson'/>
    <rst-rel id=r92-1 name=elab
             nuc=p92-1   sat=p92-2   ref=no  />
    <rst-rel id=r92-2 name=elab
             nuc=p92-1   sat=p92-3   ref=no  />

    <trans id=tx92-1 name=conj-simp   nuc=p92-2 sat=p92-3 />
    <trans id=tx92-2 name=rel-wh   nuc=p92-1 sat=tx92-1 />

    <seqorder valid=true />

    <conj id=c92-1 type=dist />

</sentence>

<sentence id=s93>
  "I made a personal decision to go out on my own and manage my
  own and friends' money," Mr. Griffin said.
    <propset id=pset93-1>
      <prop id=p93-1>
        Mr. Griffin said -X-.    </prop>
      <propset id=pset93-2>
      <prop id=p93-2>
        I made a personal decision to go out on my own. </prop>
      <prop id=p93-3>
        I made a personal decision to manage my own money.</prop>
      <prop id=p93-4>
        I made a personal decision to manage my friends'
        money. </prop>
      </propset>
    </propset>

    <focus entity='Mr. Griffin'/>
    <rst-rel id=r93-1 name=arg
             nuc=p93-1   sat=p93-2   ref=no  />
    <rst-rel id=r93-2 name=join
             nuc=p93-2   sat=p93-3   ref=no  />
    <rst-rel id=r93-3 name=join
             nuc=p93-3   sat=p93-4   ref=no  />

    <trans id=tx93-1 name=conj-simp   nuc=p93-3 sat=p93-4 />
    <trans id=tx93-2 name=conj-mult-vp   nuc=p93-2 sat=tx93-1 />
    <trans id=tx93-3 name=arg   nuc=p93-1 sat=tx93-2 />

    <seqorder valid=true />
```

```
    <conj id=c93-1 type=dist />
    <conj id=c93-2 type=dist />
    <comment text="- deletion of 'my' in the second clause, deletion
              of possessor." />
</sentence>
<sentence id=s94>
  "He leaves with his imprint on our portfolio and our people."
    <propset id=pset94-1>
      <prop id=p94-1>
        He leaves with his imprint on our portfolio.    </prop>
      <prop id=p94-2>
        He leaves with his imprint on our people.     </prop>
    </propset>
    <focus entity='he'/>
    <rst-rel id=r94-1 name=join
          nuc=p94-1   sat=p94-2   ref=no  />
    <trans id=tx94-1 name=conj-simp   nuc=p94-1 sat=p94-2 />
    <seqorder valid=true />
    <conj id=c94-1 type=dist />
</sentence>
<sentence id=s95>
  TOKYO -- Sony Corp. President Nobuyuki Idei said he has no
  intention of selling part or all of Sony's entertainment
  operations, and added he will consider a public offering of
  the company's U.S. unit only in the relatively distant future.
    <propset id=pset95-1>
      <prop id=p95-1>
        Nobuyuki Idei said -X-.  </prop>
      <propset id=pset95-2>
      <propset id=pset95-3>
      <prop id=p95-2>
        Nobuyuki Idei has no intention of selling part of Sony's
        entertainment operations.    </prop>
      <prop id=p95-3>
        (or) Nobuyuki Idei has no intention of selling all of
        Sony's entertainment operations.   </prop>
      </propset>
      <prop id=p95-4>
        Nobuyuki Idei added -X-.   </prop>
      <prop id=p95-5>
        Nobuyuki will consider a public offering of the company's
        U.S. unit only in the relatively distant future.
        </prop>
      </propset>
      <prop id=p95-6>
        Nobuyuki Idei is Sony Corp. President.  </prop>
    </propset>
```

```
    <focus entity='Idei'/>
    <rst-rel id=r95-1 name=arg
             nuc=p95-1   sat=p95-2   ref=no  />
    <rst-rel id=r95-2 name=alternate
             nuc=p95-2   sat=p95-3   ref=no  />
    <rst-rel id=r95-3 name=join
             nuc=pset95-3   sat=p95-4   ref=no  />
    <rst-rel id=r95-4 name=arg
             nuc=p95-4   sat=p95-5   ref=no  />
    <rst-rel id=r95-5 name=elab
             nuc=p95-1   sat=p95-6   ref=no  />
    <trans id=tx95-1 name=alternate   nuc=p95-2 sat=p95-3 />
    <trans id=tx95-2 name=arg   nuc=p95-4 sat=p95-5 />
    <trans id=tx95-3 name=conj-mult-vp   nuc=tx95-1 sat=tx95-2 />
    <trans id=tx95-4 name=arg   nuc=p95-1 sat=tx95-3 />
    <trans id=tx95-5 name=prenominal-title   nuc=tx95-4 sat=p95-6 />
    <seqorder valid=true />
    <conj id=c95-1 type=dist />
</sentence>
<sentence id=s96>
  In his first major meeting with the press since the abrupt
  departure of Sony Corp. of America President Michael Schulhof
  last month, Mr. Idei sought to quash a variety of rumors
  involving Sony and its long-troubled Hollywood studio.
    <propset id=pset96-1>
      <propset id=pset96-2>
      <prop id=p96-1>
        In his first major meeting with the press,
        Mr.  Idei sought to quash a variety of rumors involving
        Sony.    </prop>
      <prop id=p96-2>
        In his first major meeting with the press,
        Mr.  Idei sought to quash a variety of rumors involving
        Sony's long-troubled Hollywood studio.  </prop>
      </propset>
      <prop id=p96-3>
        (since) President Michael Schulhof abruptly departed Sony
        Corp. of America last month, </prop>
    </propset>
    <focus entity='Mr. Idei'/>
    <rst-rel id=r96-1 name=join
             nuc=p96-1   sat=p96-2   ref=no  />
    <rst-rel id=r96-2 name=circum-time
             nuc=pset96-2   sat=p96-3   ref=no  />
    <trans id=tx96-1 name=conj-simp   nuc=p96-1 sat=p96-2 />
    <trans id=tx96-2 name=cue-since-nominal nominal=yes  nuc=tx96-1 sat=p96-3 />
    <seqorder valid=true />
```

```
    <conj id=c96-1 type=dist />
    <comment text="- p96-3 is modifying the 'first meeting'" />
</sentence>
<sentence id=s97>
  In particular, he insisted that Sony still sees long term
  strategic value in owning motion-picture and music units and
  expressed his confidence in Sony's current U.S. executives.
    <propset id=pset97-1>
      <prop id=p97-1>
        In particular, he insisted that -X-.    </prop>
      <propset id=pset97-2>
      <prop id=p97-2>
        Sony still sees long term strategic value in owning
        motion-picture unit.    </prop>
      <prop id=p97-3>
        Sony still sees long term strategic value in owning music
        unit.    </prop>
      <prop id=p97-4>
        he expressed his confidence in Sony's current U.S.
        executives.     </prop>
      </propset>
    </propset>
    <focus entity='he'/>
    <rst-rel id=r97-1 name=arg
             nuc=p97-1   sat=pset97-2   ref=no  />
    <rst-rel id=r97-2 name=join
             nuc=p97-2   sat=p97-3   ref=no  />
    <rst-rel id=r97-3 name=join
             nuc=p97-3   sat=p97-4   ref=no  />
    <trans id=tx97-1 name=conj-simp   nuc=p97-2 sat=p97-3 />
    <trans id=tx97-2 name=conj-mult-vp   nuc=tx97-1 sat=p97-4 />
    <trans id=tx97-3 name=arg   nuc=p97-1 sat=tx97-2 />
    <seqorder valid=true />
    <conj id=c97-1 type=dist />
    <conj id=c97-2 type=dist />
</sentence>
<sentence id=s98>
  Since Mr. Schulhof's resignation, Mr. Idei has been much more
  involved in the day-to-day operations of Sony's U.S.
  businesses, commuting every month to New York and Los Angeles
  in a tiring schedule he said he's unlikely to keep up.
    <propset id=pset98-1>
      <propset id=pset98-2>
      <propset id=pset98-3>
      <prop id=p98-1>
        Mr. Idei has been much more involved in the day-to-day
        operations of Sony's U.S. businesses.    </prop>
```

```
      <prop id=p98-2>
        Mr Idei commutes every month to New York in a tiring
        schedule.   </prop>
      <prop id=p98-3>
        Mr Idei commutes every month to Los Angeles in a tiring
        schedule.   </prop>
      </propset>
      <prop id=p98-4>
        He said he's unlikely to keep up the schedule.
        </prop>
      </propset>
      <prop id=p98-5>
        (Since) Mr. Schulhof resigned.   </prop>
    </propset>

    <focus entity='Mr. Idei'/>
    <rst-rel id=r98-1 name=evidence
             nuc=p98-1   sat=p98-2   ref=no  />
    <rst-rel id=r98-2 name=evidence
             nuc=p98-1   sat=p98-3   ref=no  />
    <rst-rel id=r98-3 name=elab
             nuc=pset98-3   sat=p98-4   ref=no  />
    <rst-rel id=r98-4 name=circum-time
             nuc=pset98-2   sat=p98-5   ref=no  />

    <trans id=tx98-1 name=conj-simp   nuc=p98-2 sat=p98-3 />
    <trans id=tx98-2 name=rel-reduced-ing  nuc=p98-1 sat=tx98-1 />
    <trans id=tx98-3 name=rel-which-del-extract nuc=tx98-2 sat=p98-4 />
    <trans id=tx98-4 name=cue-since-nominal nominal=yes  nuc=tx98-3 sat=p98-5 />

    <seqorder valid=true />

    <conj id=c98-1 type=dist />

    <comment text="- Interesting!  For dates use of "since" or time
      use, it's usually a noun." />
</sentence>

<sentence id=s99>
  While Mr. Idei didn't say what Mr. Kawai's responsibilities
  will be, he and his U.S. executives insisted that executives
  in Tokyo don't intend to micromanage the U.S. business.

    <propset id=pset99-1>
      <prop id=p99-1>
        Mr. Idei insisted that -X-  </prop>
      <prop id=p99-2>
        Mr. Idei's U.S. executives insisted -X- </prop>
      <prop id=p99-3>
        Executives in Tokyo don't intend to micromanage the
        U.S. business.  </prop>
      <prop id=p99-4>
        (While) Mr. Idei didn't say what Mr. Kawai's
        responsibilities will be.   </prop>
    </propset>

    <focus entity='Mr. Idei'/>
```

```
    <rst-rel id=r99-1 name=join
             nuc=p99-1   sat=p99-2   ref=no  />
    <rst-rel id=r99-2 name=arg
             nuc=p99-1   sat=p99-3   ref=no  />
    <rst-rel id=r99-3 name=circum-time
             nuc=p99-1   sat=p99-4   ref=no  />

    <trans id=tx99-1 name=conj-simp   nuc=p99-1 sat=p99-2 />
    <trans id=tx99-2 name=arg   nuc=tx99-1 sat=p99-3 />
    <trans id=tx99-3 name=cue-while-sent   nuc=tx99-2 sat=p99-4 />

    <seqorder valid=true />

    <conj id=c99-1 type=dist />

</sentence>

<sentence id=s100>
  For instance, when Sony Pictures Entertainment President Alan
  Levine boasted of Sony Pictures' increased box-office share,
  Mr. Idei snapped, "I don't care about market share," and added
  that his main concern is for the movie operations to be
  profitable.

    <propset id=pset100-1>
      <prop id=p100-1>
        Mr. Idei snapped -X-.  </prop>
      <prop id=p100-2>
        "I don't care about market share,"  </prop>
      <prop id=p100-3>
        Mr Idei added -X-.  </prop>
      <prop id=p100-4>
        his main concern is for the movie operations to be profitable.
        </prop>
      <prop id=p100-3>
        (For instance), when Sony Pictures Entertainment
        President Alan Levine boasted of Sony Pictures' increased
        box-office share.    </prop>
    </propset>

    <focus entity='Mr. Idei'/>
    <rst-rel id=r100-1 name=arg
             nuc=p100-1   sat=p100-2   ref=no  />
    <rst-rel id=r100-2 name=join
             nuc=p100-1   sat=p100-3   ref=no  />
    <rst-rel id=r100-3 name=arg
             nuc=p100-3   sat=p100-4   ref=no  />
    <rst-rel id=r100-4 name=evidence
             nuc=p100-1   sat=p100-5   ref=no  />

    <trans id=tx100-1 name=arg   nuc=p100-1 sat=p100-2 />
    <trans id=tx100-2 name=arg   nuc=p100-3 sat=p100-4 />
    <trans id=tx100-3 name=conj-mult-vp   nuc=tx100-1 sat=tx100-2 />
    <trans id=tx100-4 name=cue-for-instance  nuc=tx100-3 sat=p100-5 />

    <seqorder valid=true />

    <conj id=c100-1 type=dist />
```

```
</sentence>
</document>
```

Appendix D

cardio.xml

```
<document>
<sentence id=s1>
  The patient reports that she was in her usual state of health
  until approximately two weeks ago when she began to have
  worsening dyspnea on exertion, shortness of breath and
  orthopnea.
    <propset id=pset1-1>
      <prop id=p1-1>
        The patient reports -X-.  </prop>
      <propset id=pset1-2>
      <prop id=p1-2>
        The patient was in her usual state of health until
        approximately two weeks ago.  </prop>
      <prop id=p1-3>
        Two weeks ago is when she began to have worsening dyspnea on
        exertion.  </prop>
      <prop id=p1-4>
        Two weeks ago is when she began to have shortness of breath.
        </prop>
      <prop id=p1-5>
        Two weeks ago is when she began to have orthopnea.  </prop>
    </propset>
    </propset>
    <focus entity='the patient'/>
    <rst-rel id=r1-1 name=arg
             nuc=p1-1    sat=p1-2 />
    <rst-rel id=r1-2 name=elab
             nuc=p1-2    sat=p1-3    ref=no  />
    <rst-rel id=r1-3 name=elab
             nuc=p1-2    sat=p1-4    ref=no  />
```

```
    <rst-rel id=r1-4 name=elab
             nuc=p1-2    sat=p1-5    ref=no  />

    <trans id=tx1-1 name=conj-simp nuc=p1-3 sat=p1-4/>
    <trans id=tx1-2 name=conj-simp nuc=tx1-1 sat=p1-5/>
    <trans id=tx1-3 name=cue-when nuc=p1-2 sat=tx1-2/>
    <trans id=tx1-4 name=arg       nuc=p1-1 sat=tx1-3/>

    <seqorder valid=true />

    <conj id=c1-1 type=dist />

</sentence>
<sentence id=s2>
  The patient had a breast cyst removed in 1970, also had
  cataract surgery and retinal detachment with decreased vision
  in one eye.
    <propset id=pset2-1>
      <prop id=p2-1>
        The patient had a breast cyst removed in 1970. </prop>
      <prop id=p2-2>
        The patient had cataract surgery.   </prop>
      <prop id=p2-3>
        The patient had retinal detainment.   </prop>
      <prop id=p2-4>
        The patient had decrease vision in one eye.  </prop>
    </propset>

    <focus entity='the patient'/>
    <rst-rel id=r2-1 name=join
             nuc=p2-1   sat=p2-2   ref=no  />
    <rst-rel id=r2-2 name=join
             nuc=p2-2   sat=p2-3   ref=no  />
    <rst-rel id=r2-3 name=join
             nuc=p2-3   sat=p2-4   ref=no  />

    <trans id=tx2-1 name=conj-with      nuc=p2-3 sat=p2-4/>
    <trans id=tx2-2 name=conj-simp nuc=p2-2 sat=tx2-1/>
    <trans id=tx2-3 name=conj-mult-vp   nuc=p2-1 sat=tx2-2/>

    <seqorder valid=true />

    <conj id=c2-1 type=dist />
</sentence>
<sentence id=s3>
  On admission, the patient had a blood pressure of 140/80,
  pulse 130 to 150, irregular, temperature 99,
  respirations 18-20 and in no distress.
    <propset id=pset3-1>
      <prop id=p3-1>
        On admission, the patient had a blood pressure of 140/80.  </prop>
      <prop id=p3-2>
        On admission, the patient had pulse 130 to 150.  </prop>
      <prop id=p3-3>
        The pulse is irregular. </prop>
```

```
      <prop id=p3-4>
        On admission, the patient had temperature 99.      </prop>
      <prop id=p3-5>
        On admission, the patient had respirations 18-20.  </prop>
      <prop id=p3-6>
        On admission, the patient is in no distress. </prop>
    </propset>
    <focus entity='the patient'/>
    <rst-rel id=r3-1 name=join
             nuc=p3-1   sat=p3-2   ref=no  />
    <rst-rel id=r3-2 name=elab
             nuc=p3-2   sat=p3-3   ref=no  />
    <rst-rel id=r3-3 name=join
             nuc=p3-2   sat=p3-4   ref=no  />
    <rst-rel id=r3-4 name=join
             nuc=p3-4   sat=p3-5   ref=no  />
    <rst-rel id=r3-4 name=join
             nuc=p3-5   sat=p3-6   ref=no  />
    <trans id=tx3-1 name=rel-reduced-del-wh-be-adj   nuc=p3-2 sat=p3-3 />
    <trans id=tx3-2 name=conj-simp      nuc=p3-1 sat=tx3-1 />
    <trans id=tx3-3 name=conj-simp      nuc=tx3-2 sat=p3-4 />
    <trans id=tx3-4 name=conj-simp      nuc=tx3-3 sat=p3-5 />
    <trans id=tx3-5 name=conj-mult-del-wh-be-pp   nuc=tx3-4 sat=p3-6 />
    <seqorder valid=true />
    <conj id=c3-1 type=dist />
</sentence>
<sentence id=s4>
  Lung examination revealed bibasilar crackles 2/3rds of the
  lung fields with areas of A to E egophony and dullness to
  percussion immediately the crackles.
    <propset id=pset4-1>
      <prop id=p4-1>
        Lung examination revealed bibasilar crackles 2/3rds of the
        lung fields.    </prop>
      <prop id=p4-2>
        Lung examination revealed areas of A to E egophony.
        </prop>
      <prop id=p4-3>
        Lung examination revealed dullness to percussion immediately
        the cracles.  </prop>
    </propset>
    <focus entity='lung examination'/>
    <rst-rel id=r4-1 name=join
             nuc=p4-1   sat=p4-2   ref=no  />
    <rst-rel id=r4-2 name=join
             nuc=p4-2   sat=p4-3   ref=no  />
    <trans id=tx1 name=conj-with        nuc=p4-1 sat=p4-2 />
```

```
    <trans id=tx2 name=conj-simp   nuc=tx4-1 sat=p4-3 />
    <seqorder valid=true />
    <conj id=c4-1 type=dist />
</sentence>
<sentence id=s5>
  The patient had good pedal pulses and was noncyanotic.
    <propset id=pset5-1>
      <prop id=p5-1>
        The patient had good pedal pulses.  </prop>
      <prop id=p5-2>
        The patient was noncyanotic.  </prop>
    </propset>
    <focus entity='the patient'/>
    <rst-rel id=r5-1 name=join
             nuc=p5-1   sat=p5-2   ref=no  />
    <trans id=tx5-1 name=conj-mult-vp   nuc=p5-1 sat=p5-2 />
    <seqorder valid=true />
    <conj id=c5-1 type=dist />
</sentence>
<sentence id=s6>
  Neurological examination showed the patient to be alert and
  oriented to person, place.
    <propset id=pset6-1>
      <prop id=p6-1>
        Neurological examination showed -X-. </prop>
      <propset id=pset6-2>
      <prop id=p6-2>
        The patient is alert.      </prop>
      <prop id=p6-3>
        The patient is oriented to person.    </prop>
      <prop id=p6-4>
        The patient is oriented to place.    </prop>
      </propset>
    </propset>
    <focus entity='Neurological examination'/>
    <rst-rel id=r6-1 name=arg
             nuc=p6-1   sat=pset6-2   ref=no  />
    <rst-rel id=r6-2 name=join
             nuc=p6-2   sat=p6-3   ref=no  />
    <rst-rel id=r6-3 name=join
             nuc=p6-3   sat=p6-4   ref=no  />
    <trans id=tx6-1 name=conj-simp   nuc=p6-3 sat=p6-4 />
    <trans id=tx6-2 name=conj-mult   nuc=p6-2 sat=tx6-1 />
    <trans id=tx6-3 name=arg   nuc=p6-1 sat=tx6-2 />
    <seqorder valid=true />
```

```
    <conj id=c6-1 type=dist />
</sentence>
<sentence id=s7>
  Good strength and sensation, poor recall for recent events but
  otherwise relatively intact mentation.
    <propset id=pset7-1>
      <prop id=p7-1>
        The patient has good strength. </prop>
      <prop id=p7-2>
        The patient has good sensation. </prop>
      <prop id=p7-3>
        The patient has poor recall for recent events.
        </prop>
      <prop id=p7-4>
        (but otherwise) The patient has relatively intact mentation.
        </prop>
    </propset>
    <focus entity='-X-'/>
    <rst-rel id=r7-1 name=join
             nuc=p7-1   sat=p7-2   ref=no  />
    <rst-rel id=r7-2 name=join
             nuc=p7-2   sat=p7-3   ref=no  />
    <rst-rel id=r7-3 name=contrast
             nuc=p7-1   sat=p7-4   ref=no  />
    <trans id=tx7-1 name=conj-simp   nuc=p7-1 sat=p7-2 />
    <trans id=tx7-2 name=conj-simp   nuc=tx7-1 sat=p7-3 />
    <trans id=tx7-3 name=cue-but-otherwise-conj   nuc=tx7-2 sat=p7-4 />
    <seqorder valid=true />
    <conj id=c7-1 type=dist />
    <comment text="- deletion of same adjective." />
</sentence>
<sentence id=s8>
  Electrocardiogram showed the patient in rapid atrial
  fibrillation at a rate of 150 with left ventricular
  hypertrophy and normal axis.
    <propset id=pset8-1>
      <prop id=p8-1>
        Electrocardiogram showed -X-.  </prop>
      <propset id=pset8-2>
      <prop id=p8-2>
        The patient is in rapid atrial fibrillation at a rate
        of 150.  </prop>
      <prop id=p8-3>
        The patient has left ventricular hypertrophy.   </prop>
      <prop id=p8-4>
        The patient has normal axis.    </prop>
      </propset>
    </propset>
```

```
    <focus entity='electrocardiogram'/>
    <rst-rel id=r8-1 name=arg
             nuc=p8-1   sat=p8-2   ref=no  />
    <rst-rel id=r8-2 name=join
             nuc=p8-2   sat=p8-3   ref=no  />
    <rst-rel id=r8-3 name=join
             nuc=p8-3   sat=p8-4   ref=no  />

    <trans id=tx8-1 name=conj-simp   nuc=p8-3 sat=p8-4 />
    <trans id=tx8-2 name=conj-with   nuc=p8-2 sat=tx8-1 />
    <trans id=tx8-3 name=arg   nuc=p8-1 sat=tx8-2 />

    <seqorder valid=true />

    <conj id=c8-1 type=dist />

    <comment text="- There might be an issue with ordering inside
    conjunction operators." />
</sentence>

<sentence id=s9>
  The patient was loaded on Digoxin on admission and on 5/15/93,
  the patient was found to have a Digoxin level of 4.6, at which
  point, the patient was in bradycardic sinus rhythm with a rate
  of 51.
    <propset id=pset9-1>
      <prop id=p9-1>
        The patient was loaded on Digoxin on admission.  </prop>
    <propset id=pset9-2>
      <prop id=p9-2>
        On 5/15/93, the patient was found to have a Digoxin level
        of 4.6.  </prop>
      <prop id=p9-3>
        (at which point) the patient was in bradycardic sinus
        rhythm with a rate of 51.    </prop>
    </propset>
    </propset>

    <focus entity='the patient'/>

    <rst-rel id=r9-1 name=join
             nuc=p9-1   sat=p9-2   ref=no  />
    <rst-rel id=r9-2 name=circum-time
             nuc=p9-2   sat=p9-3   ref=no  />

    <trans id=tx9-1 name=cue-at-which-point-sent   nuc=p9-2 sat=p9-3 />
    <trans id=tx9-2 name=conj-mult-sent   nuc=p9-1 sat=tx9-1 />

    <seqorder valid=true />

    <conj id=c9-1 type=dist />

    <comment text="- There is no deletion despite same subject." />
</sentence>

<sentence id=s10>
  The patient ruled out for myocardial infarction by serial
  CPK's and electrocardiograms.
```

```
    <propset id=pset10-1>
      <prop id=p10-1>
        The patient rules out for myocardial infarction by serial
        CPK's. </prop>
      <prop id=p10-2>
        The patient rules out for myocardial infarction by
        electrocardiograms.    </prop>
    </propset>

    <focus entity='the patient'/>

    <rst-rel id=r10-1 name=join
             nuc=p10-1   sat=p10-2   ref=no  />

    <trans id=tx10-1 name=conj-simp   nuc=p10-1 sat=p10-2 />

    <seqorder valid=true />

    <conj id=c10-1 type=dist />

    <comment test="- this could be either collective or distributive.
    Require domain knowledge." />
</sentence>

<sentence id=s11>
  The patient's Digoxin was held and the rate was attempted to
  be controlled with lower levels of Digoxin and Diltiazem.
    <propset id=pset11-1>
      <prop id=p11-1>
        The patient's Digoxin was held.  </prop>
      <prop id=p11-2>
        The rate was attempted to be controlled with lower levels
        of Digoxin.    </prop>
      <prop id=p11-3>
        The rate was attempted to be controlled with
        lower levels of Diltizem.    </prop>
    </propset>

    <focus entity='the patient'/>
    <rst-rel id=r11-1 name=join
             nuc=p11-1   sat=p11-2   ref=no  />
    <rst-rel id=r11-2 name=join-collective
             nuc=p11-2   sat=p11-3   ref=no  />

    <trans id=tx11-1 name=conj-simp   nuc=p11-2 sat=p11-3 />
    <trans id=tx11-2 name=conj-mult   nuc=p11-1 sat=tx11-2 />

    <seqorder valid=true />

    <conj id=c11-1 type=dist />
    <conj id=c11-2 type=coll />

</sentence>

<sentence id=s12>
  On 5/20/93, the patient was deemed to be stable for discharge
  home and the patient was discharged home with the following
  diagnosis; rapid atrial fibrillation with congestive heart
  failure, ruled out for myocardial infarction.
```

```
    <propset id=pset12-1>
      <prop id=p12-1>
        On 5/20/93, the patient was deemed to be stable for
        discharge home.  </prop>
      <prop id=p12-2>
        The patient was discharged home with the following
        diagnosis.    </prop>
      <prop id=p12-3>
        The patient's diagnosis include rapid atrial fibrillation.
   </prop>
      <prop id=p12-4>
         The rapid atrial fibrillation is associated with congestive heart
         failure.    </prop>
      <prop id=p12-5>
        The patient's diagnosis is ruled out for myocardial
        infarction.    </prop>
    </propset>
    <focus entity='the patient'/>
    <rst-rel id=r12-1 name=join
             nuc=p12-1   sat=p12-2   ref=no  />
    <rst-rel id=r12-2 name=elab
             nuc=p12-2   sat=p12-3   ref=no  />
    <rst-rel id=r12-3 name=elab
             nuc=p12-3   sat=p12-4   ref=no  />
    <rst-rel id=r12-4 name=elab
             nuc=p12-2   sat=p12-5   ref=no  />
    <trans id=tx12-1 name=pp-with   nuc=p12-3 sat=p12-4 />
    <trans id=tx12-2 name=conj-mult-2nd-del-wh-be   nuc=tx12-1 sat=p12-5 />
    <trans id=tx12-3 name=colon-del-include   nuc=p12-2 sat=tx12-2 />
    <trans id=tx12-4 name=conj-mult   nuc=p12-1 sat=tx12-3 />
    <seqorder valid=true />
    <conj id=c12-1 type=dist />
</sentence>
<sentence id=s13>
  The patient is a forty-three year old woman who was stabbed
  twice, once to the left upper quadrant and once to the left
  hip, who was admitted to Area A.
    <propset id=pset13-1>
      <prop id=p13-1>
        The patient is a woman.  </prop>
      <prop id=p13-2>
        The patient is a forty-three year old. </prop>
      <prop id=p13-3>
        The patient was stabbed twice.
        </prop>
      <prop id=p13-4>
        The patient was stabbed once to the left upper quadrant.
        </prop>
      <prop id=p13-5>
```

```
        The patient was stabbed once to the left hip.   </prop>
      <prop id=p13-6>
        The patient was admitted to Area A.    </prop>
    </propset>

    <focus entity='the patient'/>
    <rst-rel id=r13-1 name=elab
             nuc=p13-1   sat=p13-2   ref=no  />
    <rst-rel id=r13-2 name=elab
             nuc=p13-1   sat=p13-3   ref=no  />
    <rst-rel id=r13-3 name=elab
             nuc=p13-1   sat=p13-4   ref=no  />
    <rst-rel id=r13-4 name=elab
             nuc=p13-1   sat=p13-5   ref=no  />
    <rst-rel id=r13-5 name=elab
             nuc=p13-1   sat=p13-6   ref=no  />

    <trans id=tx13-1 name=adj   nuc=p13-1 sat=p13-2 />
    <trans id=tx13-2 name=conj-simp   nuc=p13-4 sat=p13-5 />
    <trans id=tx13-3 name=pp-to   nuc=p13-3 sat=tx13-2 />
    <trans id=tx13-4 name=rel-wh   nuc=tx13-1 sat=tx13-3 />
    <trans id=tx13-5 name=rel-wh   nuc=tx13-4 sat=p13-6 />

    <seqorder valid=true />

    <conj id=c13-1 type=dist />

    <comment text="- a complex example with elab with conj
                   - interesting that 'who was admit.' was last." />
</sentence>
<sentence id=s14>
  Her postoperative course has been uncomplicated and she is
  being discharged today with no medications.
    <propset id=pset14-1>
      <prop id=p14-1>
        The patient's postoperative course has been
        uncomplicated.  </prop>
      <prop id=p14-2>
        The patient is being discharged today. </prop>
      <prop id=p14-3>
        The discharge is without medication.    </prop>
    </propset>

    <focus entity='her postoperative course'/>

    <rst-rel id=r14-1 name=join
             nuc=p14-1   sat=p14-2   ref=no  />
    <rst-rel id=r14-2 name=elab
             nuc=p14-2   sat=p14-3   ref=no  />

    <trans id=tx14-1 name=pp-without   nuc=p14-2 sat=p14-3 />
    <trans id=tx14-2 name=conj-mult   nuc=p14-1 sat=tx14-2 />

    <seqorder valid=true />

    <conj id=c14-1 type=dist />

</sentence>
```

```
<sentence id=s15>
  Her status is stable and doing well.
    <propset id=pset15-1>
      <prop id=p15-1>
        Her status is stable.    </prop>
      <prop id=p15-2>
        She is doing well.    </prop>
    </propset>
    <focus entity='her status'/>
    <rst-rel id=r15-1 name=join
             nuc=p15-1   sat=p15-2   ref=no  />
    <trans id=tx15-1 name=conj-mult   nuc=p15-1 sat=p15-2 />
    <seqorder valid=true />
    <conj id=c15-1 type=dist />
</sentence>
<sentence id=s16>
  STATUS POST STAB WOUNDS TO LEFT UPPER QUADRANT CHEST AREA AND
  LEFT HIP.
    <propset id=pset16-1>
      <prop id=p16-1>
        The patient is status post stab wounds to left upper
        quadrant chest area.    </prop>
      <prop id=p16-2>
        The patient is status post stab wounds to left hip.
        </prop>
    </propset>
    <focus entity='-X-'/>
    <rst-rel id=r16-1 name=join
             nuc=p16-1   sat=p16-2   ref=no  />
    <trans id=tx16-1 name=conj-simp   nuc=p16-1 sat=p16-2 />
    <seqorder valid=true />
    <conj id=c16-1 type=dist />
</sentence>
<sentence id=s17>
  CHEST TUBE PLACEMENT AND REMOVAL.
    <propset id=pset17-1>
      <prop id=p17-1>
        chest tube placement.    </prop>
      <prop id=p17-2>
        chest tube removal.    </prop>
    </propset>
    <focus entity='-X-'/>
    <rst-rel id=r17-1 name=join
             nuc=p17-1   sat=p17-2   ref=no  />
```

```
    <trans id=tx17-1 name=conj-simp   nuc=p17-1 sat=p17-2 />
    <seqorder valid=true />
    <conj id=c17-1 type=dist />
    <comment text="- clearly using directional deletion rule" />
</sentence>
<sentence id=s18>
  Because of a abnormal PAP smear she was colposcoped by Dr.
  Smith and cervical curettage was positive for squamous
  dysplasia.
    <propset id=pset18-1>
    <propset id=pset18-2>
      <prop id=p18-1>
        The patient was colposcoped by Dr. Smith.    </prop>
      <prop id=p18-2>
        (Because of) There is a abnormal PAP smear.   </prop>
    </propset>
      <prop id=p18-3>
        Cervical curettage was positive for squamous dysplasia.
        </prop>
    </propset>
    <focus entity='she'/>
    <rst-rel id=r18-1 name=non-volitional-cause
             nuc=p18-1   sat=p18-2   ref=no  />
    <rst-rel id=r18-2 name=join
             nuc=p18-1   sat=p18-3   ref=no  />
    <trans id=tx18-1 name=cue-because-sent   nuc=p18-1 sat=p18-2 />
    <trans id=tx18-2 name=conj-mult   nuc=tx18-1 sat=p18-3 />
    <seqorder valid=true />
    <conj id=c18-1 type=dist />
    <comment text="- a good example of why scope is needed." />
</sentence>
<sentence id=s19>
  CAT scan of the abdomen and pelvis was normal except for left
  ovarian cyst.
    <propset id=pset19-1>
      <prop id=p19-1>
        Cat scan of the abdomen and pelvis was normal.    </prop>
      <prop id=p19-2>
        Cat scan showed left ovarian cyst.    </prop>
    </propset>
    <focus entity='CAT scan'/>
    <rst-rel id=r19-1 name=condition-except
             nuc=p19-1   sat=p19-2   ref=no  />
    <trans id=tx19-1 name=cue-except-pp    nuc=p19-1 sat=p19-2 />
    <seqorder valid=true />
```

```
    <conj id=c19-1 type=coll />
    <comment text="- 'abdomen and pelvis' is a collective term here
                   - not clear if 'except' should be an aggregation" />
</sentence>
<sentence id=s20>
  Alternatives were discussed with the patient and she agreed to
  proceed with radical surgery.
    <propset id=pset20-1>
      <prop id=p20-1>
        Alternatives were discussed with the patient.    </prop>
      <prop id=p20-2>
        The patient agreed to proceed with radical surgery.
        </prop>
    </propset>
    <focus entity='alternatives'/>
    <rst-rel id=r20-1 name=sequence
             nuc=p20-1   sat=p20-2   ref=no  />
    <trans id=tx20-1 name=conj-mult   nuc=p20-1 sat=p20-2 />
    <seqorder valid=true />
    <conj id=c20-1 type=dist />
    <comment text="- if aggregated using elaboration 'who', then there
                   is a lost of sequence information."
</sentence>
<sentence id=s21>
  She has had tonsillectomy and adenoidectomy as a child.
    <propset id=pset21-1>
      <prop id=p21-1>
        She has had tonsillectomy as a child.    </prop>
      <prop id=p21-2>
        She has had adenoidectomy as a child.   </prop>
    </propset>
    <focus entity='she'/>
    <rst-rel id=r21-1 name=join
             nuc=p21-1   sat=p21-2   ref=no  />
    <trans id=tx21-1 name=conj-simp   nuc=p21-1 sat=p21-2 />
    <seqorder valid=true />
    <conj id=c21-1 type=dist />
</sentence>
<sentence id=s22>
  Chest: Clear to auscultation and percussion.
    <propset id=pset22-1>
      <prop id=p22-1>
        Chest is clear to auscultation. </prop>
      <prop id=p22-2>
```

```
        Chest is clear to percussion.    </prop>
    </propset>
    <focus entity='-X-'/>
    <rst-rel id=r22-1 name=join
             nuc=p22-1   sat=p22-2   ref=no  />
    <trans id=tx22-1 name=conj-simp   nuc=p22-1 sat=p22-2 />
    <seqorder valid=true />
    <conj id=c22-1 type=dist />
</sentence>
<sentence id=s23>
  HCT 30, Chest x-ray, EKG and chem 7 were otherwise normal.
    <propset id=pset23-1>
      <prop id=p23-1>
        HCT 30 was otherwise normal.    </prop>
      <prop id=p23-2>
        Chest x-ray was otherwise normal.  </prop>
      <prop id=p23-3>
        EKG was otherwise normal.  </prop>
      <prop id=p23-4>
        Chem 7 was otherwise normal.    </prop>
    </propset>
    <focus entity='HCT 30'/>
    <rst-rel id=r23-1 name=join
             nuc=p23-1   sat=p23-2   ref=no  />
    <rst-rel id=r23-2 name=join
             nuc=p23-2   sat=p23-3   ref=no  />
    <rst-rel id=r23-3 name=join
             nuc=p23-3   sat=p23-4   ref=no  />
    <trans id=tx23-1 name=conj-simp   nuc=p23-1 sat=p23-2 />
    <trans id=tx23-2 name=conj-simp   nuc=tx23-1 sat=p23-3 />
    <trans id=tx23-3 name=conj-simp   nuc=tx23-2 sat=p23-4 />
    <seqorder valid=true />
    <conj id=c23-1 type=dist />
</sentence>
<sentence id=s24>
  She received one unit of her own blood during the operation
  and ultimately postop her HCT was 30.
    <propset id=pset24-1>
      <prop id=p24-1>
        She received one unit of her own blood during the
        operation.    </prop>
      <prop id=p24-2>
        Ultimately postop her HCT was 30.    </prop>
    </propset>
    <focus entity='she'/>
```

```
    <rst-rel id=r24-1 name=sequence
             nuc=p24-1   sat=p24-2   ref=no  />
    <trans id=tx24-1 name=conj-mult   nuc=p24-1 sat=p24-2 />
    <seqorder valid=true />
    <conj id=c24-1 type=dist />
</sentence>
<sentence id=s25>
  Complications and problems were discussed with the patient for
  her home care.
    <propset id=pset25-1>
      <prop id=p25-1>
        Complications were discussed with the patient for her home
        care. </prop>
      <prop id=p25-2>
        Problems were discussed with the patient for her home
        care. </prop>
    </propset>
    <focus entity='complications'/>
    <rst-rel id=r25-1 name=join
             nuc=p25-1   sat=p25-2   ref=no  />
    <trans id=tx25-1 name=conj-simp   nuc=p25-1 sat=p25-2 />
    <seqorder valid=true />
    <conj id=c25-1 type=dist />
</sentence>
<sentence id=s26>
  She seemed quite satisfied and was discharged to be followed
  as an outpatient.
    <propset id=pset26-1>
      <prop id=p26-1>
        She seemed quite satisfied.    </prop>
      <prop id=p26-2>
        She was discharge to be followed as an outpatient.
        </prop>
    </propset>
    <focus entity='she'/>
    <rst-rel id=r26-1 name=join
             nuc=p26-1   sat=p26-2   ref=no  />
    <trans id=tx26-1 name=conj-mult-passive   nuc=p26-1 sat=p26-2 />
    <seqorder valid=true />
    <conj id=c26-1 type=dist />
    <comment text="- there is a passive-voice transformation" />
</sentence>
<sentence id=s27>
  The patient is a 64-year-old gentleman with rheumatic heart
```

```
disease, status post Starr-Edward's mitral valve in 1972,
aortograft in 1985 and Porcine tricuspid valve in 1993;
presents for evaluation of six to eight months of progressive
right sided heart failure despite increase dose of diuretics
with recent hospitalization x 2 at Columbia
Hospital for RV failure.
```

```
  <propset id=pset27-1>
    <propset id=pset27-2>
    <prop id=p27-1>
      The patient is a gentleman.    </prop>
    <prop id=p27-2>
      The patient is 64-year old.    </prop>
    <prop id=p27-3>
      The patient has rheumatic heart disease.  </prop>
    <prop id=p27-4>
      The patient is status post Starr-Edward's mitral value in
      1972.    </prop>
    <prop id=p27-5>
      The patient is status post aortograft in 1985.   </prop>
    <prop id=p27-6>
      The patient is status post Porcine tricuspid valve in
      1993.    </prop>
    </propset>
    <propset id=pset27-3>
    <prop id=p27-7>
      The patient presents for evaluation of six to eight months
      of progressive right sided heart failure.    </prop>
    <prop id=p27-8>
      (despite) The patient has increased dose of diuretics with
      recent hospitalization x 2 at Columbia Hospital
      for RV failure.  </prop>
    </propset>
  </propset>

  <focus entity='the patient'/>
  <rst-rel id=r27-1 name=elab
           nuc=p27-1   sat=p27-2   ref=no  />
  <rst-rel id=r27-2 name=elab
           nuc=p27-1   sat=p27-3   ref=no  />
  <rst-rel id=r27-3 name=elab
           nuc=p27-1   sat=p27-4   ref=no  />
  <rst-rel id=r27-4 name=elab
           nuc=p27-1   sat=p27-5   ref=no  />
  <rst-rel id=r27-5 name=elab
           nuc=p27-1   sat=p27-6   ref=no  />
  <rst-rel id=r27-6 name=join
           nuc=p27-1   sat=p27-7   ref=no  />
  <rst-rel id=r27-7 name=concession
           nuc=p27-7   sat=p27-8   ref=no  />

  <trans id=tx27-1 name=adj   nuc=p27-1 sat=p27-2 />
  <trans id=tx27-2 name=pp-with   nuc=tx27-1 sat=p27-3 />
  <trans id=tx27-3 name=conj-simp   nuc=p27-4 sat=p27-5 />
  <trans id=tx27-4 name=conj-simp   nuc=tx27-3 sat=p27-6 />
```

```
    <trans id=tx27-5 name=conj-simp   nuc=tx27-2 sat=tx27-4 />
    <trans id=tx27-6 name=cue-despite-sent   nuc=p27-7 sat=p27-8 />
    <trans id=tx27-7 name=conj-mult-semicolon   nuc=tx27-5 sat=tx27-6 />
    <seqorder valid=true />
    <conj id=c27-1 type=dist />
</sentence>
<sentence id=s28>
  However, two weeks after discharge, he was readmitted with
  worsening edema and ascites, September 6, 1996.
    <propset id=pset28-1>
      <prop id=p28-1>
        However, two weeks after discharge, he was readmitted, September
        6, 1996.   </prop>
      <prop id=p28-2>
        He had worsening edema.    </prop>
      <prop id=p28-3>
        He had worsening ascites.    </prop>
    </propset>
    <focus entity='he'/>
    <rst-rel id=r28-1 name=elab
             nuc=p28-1   sat=p28-2   ref=no  />
    <rst-rel id=r28-2 name=elab
             nuc=p28-1   sat=p28-3   ref=no  />
    <trans id=tx28-1 name=conj-simp   nuc=p28-2 sat=p28-3 />
    <trans id=tx28-2 name=pp-with   nuc=p28-1 sat=tx28-1 />
    <seqorder valid=true />
    <conj id=c28-1 type=dist />
</sentence>
<sentence id=s29>
  He was transferred for right heart catheterization and
  evaluation for possible transplant and or mitral valve
  surgery.
    <propset id=pset29-1>
      <prop id=p29-1>
        He was transferred for right heart catheterization.
        </prop>
      <prop id=p29-2>
        He was transferred for evaluation for possible transplant.
   </prop>
      <prop id=p29-3>
        (and or) He was transferred for evaluation for mitral valve
        surgery.   </prop>
    </propset>
    <focus entity='he'/>
    <rst-rel id=r29-1 name=join
             nuc=p29-1   sat=p29-2   ref=no  />
```

```
    <rst-rel id=r29-2 name=join
             nuc=p29-2   sat=p29-3   ref=no  />
    <trans id=tx29-1 name=conj-simp   nuc=p29-2 sat=p29-3 />
    <trans id=tx29-2 name=conj-simp   nuc=p29-1 sat=tx29-1 />
    <seqorder valid=true />
    <conj id=c29-1 type=dist />
    <conj id=c29-2 type=dist />
</sentence>
<sentence id=s30>
  in 1993 and underwent tricuspid valve replacement.
    <propset id=pset30-1>
      <prop id=p30-1>
        ... in 1993  </prop>
      <prop id=p30-2>
        he underwent tricuspid valve replacement.    </prop>
    </propset>
    <focus entity='-X-'/>
    <rst-rel id=r30-1 name=join
             nuc=p30-1   sat=p30-2   ref=no  />
    <trans id=tx30-1 name=conj-mult   nuc=p30-1 sat=p30-2 />
    <seqorder valid=true />
    <conj id=c30-1 type=dist />
</sentence>
<sentence id=s31>
  Recent previous medical history is noted for GI bleeds for
  gastritis, diagnosed EGD and colonoscopy, managed as an
  outpatient without interruption of Coumadin.
    <propset id=pset31-1>
    <propset id=pset31-2>
      <prop id=p31-1>
        Recent previous medical history is noted for GI bleeds for
        gastritis.    </prop>
      <prop id=p31-2>
        Recent previous medical history include diagnosed EGD.
        </prop>
      <prop id=p31-3>
        Recent previous medical history include colonscopy.
        </prop>
      </propset>
      <prop id=p31-4>
        He is managed as an outpatient without interruption of
        Coumadin.  </prop>
    </propset>
    <focus entity='recent previous medical history'/>
    <rst-rel id=r31-1 name=join
             nuc=p31-1   sat=p31-2   ref=no  />
```

```
    <rst-rel id=r31-2 name=join
             nuc=p31-2   sat=p31-3   ref=no  />
    <rst-rel id=r31-3 name=join
             nuc=pset31-2   sat=p31-4   ref=no  />

    <trans id=tx31-1 name=conj-simp   nuc=p31-1 sat=p31-2 />
    <trans id=tx31-2 name=conj-simp   nuc=tx31-1 sat=p31-3 />
    <trans id=tx31-3 name=rel-reduced-del-wh-be  nuc=tx31-2 sat=p31-4 />

    <seqorder valid=true />

    <conj id=c31-1 type=dist />
</sentence>
<sentence id=s32>
  He has a history of schizophrenia since 1960's, but has been
  without hallucinations and well compensated.
    <propset id=pset32-1>
      <prop id=p32-1>
        He has a history of schizophrenia since 1960.    </prop>
      <prop id=p32-2>
        (but) He has been without hallucination.   </prop>
      <prop id=p32-3>
        (but) He is well compensated.  </prop>
    </propset>

    <focus entity='he'/>
    <rst-rel id=r32-1 name=contrast
             nuc=p32-1   sat=p32-2   ref=no  />
    <rst-rel id=r32-2 name=contrast
             nuc=p32-1   sat=p32-3   ref=no  />

    <trans id=tx32-1 name=conj-mult   nuc=p32-2 sat=p32-3 />
    <trans id=tx32-2 name=cue-but-vp   nuc=p32-1 sat=tx32-1 />

    <seqorder valid=true />

    <conj id=c32-1 type=dist />

</sentence>

<sentence id=s33>
  Blood pressure 100/60, pulse 60 and regular, respiratory rate
  16.
    <propset id=pset33-1>
      <prop id=p33-1>
        His blood pressure is 100/60.    </prop>
      <prop id=p33-2>
        His pulse is 60.   </prop>
      <prop id=p33-3>
        His pulse is regular.    </prop>
      <prop id=p33-4>
        His respiratory rate is 16.    </prop>
    </propset>

    <focus entity='-X-'/>
    <rst-rel id=r33-1 name=join
             nuc=p33-1   sat=p33-2   ref=no  />
```

```
    <rst-rel id=r33-2 name=join
             nuc=p33-2   sat=p33-3   ref=no  />
    <rst-rel id=r33-3 name=join
             nuc=p33-3   sat=p33-4   ref=no  />

    <trans id=tx33-1 name=conj-simp   nuc=p33-2 sat=p33-3 />
    <trans id=tx33-2 name=conj-mult   nuc=p33-1 sat=tx33-1 />
    <trans id=tx33-3 name=conj-mult   nuc=tx33-2 sat=p33-4 />

    <seqorder valid=true />

    <conj id=c33-1 type=dist />

</sentence>

<sentence id=s34>
  Bowels sounds are positive, liver is 16 cm palpable with
  ascites and dulled distended abdomen.

    <propset id=pset34-1>
      <prop id=p34-1>
        Bowels sounds are positive,    </prop>
      <prop id=p34-2>
        Liver is 16 cm palpable with ascites.    </prop>
      <prop id=p34-3>
        he has dulled distended abdomen.    </prop>
    </propset>

    <focus entity='bowels sounds'/>
    <rst-rel id=r34-1 name=join
             nuc=p34-1   sat=p34-2   ref=no  />
    <rst-rel id=r34-2 name=join
             nuc=p34-2   sat=p34-3   ref=no  />

    <trans id=tx34-1 name=conj-mult   nuc=p34-1 sat=p34-2 />
    <trans id=tx34-2 name=conj-mult-del-wh-be   nuc=tx34-1 sat=p34-3 />

    <seqorder valid=true />

    <conj id=c34-1 type=dist />
</sentence>

<sentence id=s35>
  Mitral valve area 1.9, no apparent valvular leaks, mild LVE,
  increased RA and RV with organic TR and more significant
  ostial PDA lesion.

    <propset id=pset35-1>
      <prop id=p35-1>
        he has mitral valve area 1.9.    </prop>
      <prop id=p35-2>
        he has no apparent valvular leaks.    </prop>
      <prop id=p35-3>
        he has mild LVE.   </prop>
      <prop id=p35-4>
        He has increased RA.    </prop>
      <prop id=p35-5>
        He has RV with organic TR.    </prop>
      <prop id=p35-6>
```

```
        He has more significant ostial PDA lesion.    </prop>
    </propset>
    <focus entity='-X-'/>
    <rst-rel id=r35-1 name=join
             nuc=p35-1   sat=p35-2   ref=no  />
    <rst-rel id=r35-2 name=join
             nuc=p35-2   sat=p35-3   ref=no  />
    <rst-rel id=r35-3 name=join
             nuc=p35-3   sat=p35-4   ref=no  />
    <rst-rel id=r35-4 name=join
             nuc=p35-4   sat=p35-5   ref=no  />
    <rst-rel id=r35-5 name=join
             nuc=p35-5   sat=p35-6   ref=no  />
    <trans id=tx35-1 name=conj-simp   nuc=p35-1 sat=p35-2 />
    <trans id=tx35-2 name=conj-simp   nuc=tx35-1 sat=p35-3 />
    <trans id=tx35-3 name=conj-simp   nuc=tx35-2 sat=p35-4 />
    <trans id=tx35-4 name=conj-simp   nuc=tx35-3 sat=p35-5 />
    <trans id=tx35-5 name=conj-simp   nuc=tx35-4 sat=p35-6 />
    <seqorder valid=true />
    <conj id=c35-1 type=dist />
    <conj id=c35-2 type=dist />
</sentence>
<sentence id=s36>
  On the day of the IgG and monoclonal gammopathy and
  neuropathy.
    <propset id=pset36-1>
      <prop id=p36-1>
        On the day of IGG.    </prop>
      <prop id=p36-2>
        On the day of monoclonal gammopathy.    </prop>
      <prop id=p36-3>
        On the day of neuropathy.    </prop>
    </propset>
    <focus entity='-X-'/>
    <rst-rel id=r36-1 name=join
             nuc=p36-1   sat=p36-2   ref=no  />
    <rst-rel id=r36-2 name=join
             nuc=p36-2   sat=p36-3   ref=no  />
    <trans id=tx36-1 name=conj-simp   nuc=p36-1 sat=p36-2 />
    <trans id=tx36-2 name=conj-simp   nuc=tx36-1 sat=p36-3 />
    <seqorder valid=true />
    <conj id=c36-1 type=dist />
    <conj id=c36-2 type=dist />
</sentence>
<sentence id=s37>
  In the second instance, deny monoclonal and compensated
```

```
  microangiopathic hemolytic anemia and will include restriction
  and transfer to the CCU for administration of Dopamine.
    <propset id=pset37-1>
      <prop id=p37-1>
        In the second instance, deny monoclonal.   </prop>
      <prop id=p37-2>
        In the second instance, the patient has compensated
        microangiopathic hemolytic anemia.    </prop>
      <prop id=p37-3>
        ... will include restriction    </prop>
      <prop id=p37-4>
        ... transfer to the CCU for administration of Dopamine.
        </prop>
    </propset>
    <focus entity='-X-'/>
    <rst-rel id=r37-1 name=join
             nuc=p37-1   sat=p37-2   ref=no  />
    <rst-rel id=r37-2 name=join
             nuc=p37-2   sat=p37-3   ref=no  />
    <rst-rel id=r37-3 name=join
             nuc=p37-3   sat=p37-4   ref=no  />
    <trans id=tx37-1 name=conj-mult-del-wh-be   nuc=p37-1 sat=p37-2 />
    <trans id=tx37-2 name=conj-mult   nuc=tx37-1 sat=p37-3 />
    <trans id=tx37-3 name=conj-mult   nuc=tx37-2 sat=p37-4 />
    <seqorder valid=true />
    <conj id=c37-1 type=dist />
    <conj id=c37-2 type=dist />
</sentence>
<sentence id=s38>
  He continued with slow diuresis on combined Dopamine and
  Dobutamine.
    <propset id=pset38-1>
      <prop id=p38-1>
        He continues with slow diuresis on Dopamin. </prop>
      <prop id=p38-2>
        He continues with slow diuresis on Dobutamine. </prop>
    </propset>
    <focus entity='he'/>
    <rst-rel id=r38-1 name=join-collective
             nuc=p38-1   sat=p38-2   ref=no  />
    <trans id=tx38-1 name=conj-simp   nuc=p38-1 sat=p38-2 />
    <seqorder valid=true />
    <conj id=c38-1 type=coll />
</sentence>
<sentence id=s39>
  He was seen in consultation by Dr. Smith on October 1,
```

```
  1996, during the course of his diuresis and arrangements were
  made for the patient to go home and return for surgery.
    <propset id=pset39-1>
      <prop id=p39-1>
        He was seen in consultation by Dr. Smith on October 1,
        1996 during the course of his diuresis.
        </prop>
      <prop id=p39-2>
        Arrangement was made for the patient to go home.   </prop>
      <prop id=p39-3>
        Arrangement was made for the patient to return for surgery.  </prop>
    </propset>
    <focus entity='he'/>
    <rst-rel id=r39-1 name=join
             nuc=p39-1   sat=p39-2   ref=no  />
    <rst-rel id=r39-2 name=sequence
             nuc=p39-2   sat=p39-3   ref=no  />
    <trans id=tx39-1 name=conj-mult   nuc=p39-2 sat=p39-3 />
    <trans id=tx39-2 name=conj-mult   nuc=p39-1 sat=tx39-1 />
    <seqorder valid=true />
    <conj id=c39-1 type=dist />
    <conj id=c39-2 type=dist />
</sentence>
<sentence id=s40>
  Discharge BUN and creatinine 48/1.8.
    <propset id=pset40-1>
      <prop id=p40-1>
        Discharge BUN. </prop>
      <prop id=p40-2>
        Creatinine is 48/1.8.    </prop>
    </propset>
    <focus entity='-X-'/>
    <rst-rel id=r40-1 name=join
             nuc=p40-1   sat=p40-2   ref=no  />
    <trans id=tx40-1 name=conj-mult   nuc=p40-1 sat=p40-2 />
    <seqorder valid=true />
    <conj id=c40-1 type=dist />
</sentence>
<sentence id=s41>
  Plans for readmission for MVR and coronary artery bypass
  graft, tentatively for October 16, 1996, with readmission
  three days prior.
    <propset id=pset41-1>
      <propset id=pset41-2>
      <prop id=p41-1>
        Plans for readmission for MVR is tentatively for
```

```
      October 16, 1996.    </prop>
    <prop id=p41-2>
      Plans for readmission for coronary artery bypass graft
      is tentatively for October 16, 1996.    </prop>
      </prop>
   </propset>
    <prop id=p41-3>
      The readmission is three days prior.    </prop>
  </propset>
  <focus entity='Plan for readmission'/>
  <rst-rel id=r41-1 name=join
           nuc=p41-1   sat=p41-2   ref=no  />
  <rst-rel id=r41-2 name=elab
           nuc=pset41-2   sat=p41-3   ref=no  />
  <trans id=tx41-1 name=conj-simp   nuc=p41-1 sat=p41-2 />
  <trans id=tx41-2 name=pp-with nominal=other   nuc=tx41-1 sat=p41-3 />
  <seqorder valid=true />
  <conj id=c41-1 type=dist />
</sentence>
<sentence id=s42>
  The patient was a 38-year-old woman from the Dominican
  Republic who presented to the Cardiology Clinic in 11/90
  complaining of dyspnea on exertion and palpitations.
  <propset id=pset42-1>
    <prop id=p42-1>
      The patient was a woman.   </prop>
    <prop id=p42-2>
      The patient was a 38-year-old. </prop>
    <prop id=p42-3>
      The patient is from Dominican Republic.  </prop>
    <prop id=p42-4>
      The patient presented to the Cardiology Clinic in 11/90.
       </prop>
    <prop id=p42-5>
      The patient complained of dyspnea on exertion.  </prop>
    <prop id=p42-6>
      The patient complained of palpitation.
      </prop>
  </propset>
  <focus entity='the patient'/>
  <rst-rel id=r42-1 name=elab
           nuc=p42-1   sat=p42-2   ref=no  />
  <rst-rel id=r42-2 name=elab
           nuc=p42-1   sat=p42-3   ref=no  />
  <rst-rel id=r42-3 name=elab
           nuc=p42-1   sat=p42-4   ref=no  />
  <rst-rel id=r42-4 name=elab
           nuc=p42-1   sat=p42-5   ref=no  />
  <rst-rel id=r42-5 name=elab
```

```
             nuc=p42-1   sat=p42-6   ref=no  />
    <trans id=tx42-1 name=adj   nuc=p42-1 sat=p42-2 />
    <trans id=tx42-2 name=pp-from   nuc=tx42-1 sat=p42-3 />
    <trans id=tx42-3 name=rel-wh   nuc=tx42-2 sat=p42-4 />
    <trans id=tx42-4 name=conj-simp   nuc=p42-5 sat=p42-6 />
    <trans id=tx42-5 name=rel-reduced-ing   nuc=tx42-3 sat=tx42-4 />

    <seqorder valid=false />

    <conj id=c42-1 type=dist />

    <comment text="- rel-wh is applied before rel-reduced-ing, if
                  the ordering between same class of construction
                  does not matter, such as adj-order, then this is
                  not a violation." />

</sentence>
<sentence id=s43>
  She has had two uneventful pregnancies, all delivered
  vaginally, and a murmur was noted in 1980 during a pregnancy
  in Argentina and again while she as pregnant again in 1985 at
  Columbia Hospital.
    <propset id=pset43-1>
      <prop id=p43-1>
        She has had two uneventful pregnancies.  </prop>
      <prop id=p43-2>
        These two pregnancies are all delivered vaginally.
        </prop>
      <prop id=p43-3>
        A murmur was noted in 1980 during a pregnancy in
        Argentina.   </prop>
      <prop id=p43-4>
        A murmur was noted (again) while she was pregnant again in
        1985 at Columbia Hospital.    </prop>
    </propset>

    <focus entity='she'/>
    <rst-rel id=r43-1 name=elab
             nuc=p43-1   sat=p43-2   ref=no  />
    <rst-rel id=r43-2 name=join
             nuc=p43-1   sat=p43-3   ref=no  />
    <rst-rel id=r43-3 name=join
             nuc=p43-3   sat=p43-4   ref=no  />

    <trans id=tx43-1 name=rel-reduced-del-wh-be   nuc=p43-1 sat=p43-2 />
    <trans id=tx43-2 name=conj-simp   nuc=p43-3 sat=p43-4 />
    <trans id=tx43-3 name=conj-mult   nuc=tx43-1 sat=tx43-2 />

    <seqorder valid=true />

    <conj id=c43-1 type=dist />
    <conj id=c43-2 type=dist />
</sentence>
<sentence id=s44>
```

```
  She has had no further investigation until 1985 when she
  immigrated to the US and a murmur was heard during a pregnancy
  at that time.
    <propset id=pset44-1>
      <prop id=p44-1>
        She has had no further investigation until 1985.
        </prop>
      <prop id=p44-2>
        In 1985, she immigrated to the US.    </prop>
      <prop id=p44-3>
        In 1985, a murmur was heard during a pregnancy at that
        time.    </prop>
    </propset>
    <focus entity='she'/>
    <rst-rel id=r44-1 name=elab
             nuc=p44-1   sat=p44-2   ref=no  />
    <rst-rel id=r44-2 name=elab
             nuc=p44-1   sat=p44-3   ref=no  />
    <trans id=tx44-1 name=conj-mult   nuc=p44-2 sat=p44-3 />
    <trans id=tx44-2 name=rel-when   nuc=p44-1 sat=tx44-1 />
    <seqorder valid=true />
    <conj id=c44-1 type=dist />
</sentence>
<sentence id=s45>
  Holter monitor and electrocardiogram were done at Columbia
  Hospital.
    <propset id=pset45-1>
      <prop id=p45-1>
        Holter monitor was done at Columbia Hospital.    </prop>
      <prop id=p45-2>
        Electrocardiogram was done at Columbia Hospital.  </prop>
    </propset>
    <focus entity='Holter monitor'/>
    <rst-rel id=r45-1 name=join
             nuc=p45-1   sat=p45-2   ref=no  />
    <trans id=tx45-1 name=conj-simp   nuc=p45-1 sat=p45-2 />
    <seqorder valid=true />
    <conj id=c45-1 type=dist />
</sentence>
<sentence id=s46>
  There was significant narrowing at the level of the pulmonic
  valve and the main pulmonary artery appeared normal.
    <propset id=pset46-1>
      <prop id=p46-1>
        There was significant narrowing at the level of the
        pulmonic valve.  </prop>
      <prop id=p46-2>
```

```
        The main pulmonary artery appeared normal.    </prop>
    </propset>

    <focus entity='narrowing'/>
    <rst-rel id=r46-1 name=join
             nuc=p46-1   sat=p46-2   ref=no  />

    <trans id=tx46-1 name=conj-mult   nuc=p46-1 sat=p46-2 />

    <seqorder valid=true />

    <conj id=c46-1 type=dist />

</sentence>

<sentence id=s47>
  Left heart catheterization revealed LV of 110/5, aorta 110/65,
  and arterial O2 sat of 98%.

    <propset id=pset47-1>
      <prop id=p47-1>
        Left heart catheterization revealed LV of 110/5.
        </prop>
      <prop id=p47-2>
        Left heart catheterization revealed aorta of 110/65.  </prop>
      <prop id=p47-3>
        Left heart catheterization revealed arterial O2 Sat of 98%.
        </prop>
    </propset>

    <focus entity='catheterization'/>

    <rst-rel id=r47-1 name=join
             nuc=p47-1   sat=p47-2   ref=no  />
    <rst-rel id=r47-2 name=join
             nuc=p47-2   sat=p47-3   ref=no  />

    <trans id=tx47-1 name=conj-simp   nuc=p47-1 sat=p47-2 />
    <trans id=tx47-2 name=conj-simp   nuc=tx47-1 sat=p47-3 />

    <seqorder valid=true />

    <conj id=c47-1 type=dist />
</sentence>

<sentence id=s48>
  Echocardiogram showed normal LV size and function, right
  ventricular hypertrophy, no dilatation, no aortic valve
  pathology and no tricuspid regurgitation.

    <propset id=pset48-1>
      <prop id=p48-1>
        Echocardiogram showed normal LV size.    </prop>
      <prop id=p48-2>
        Echocardiogram showed normal function. </prop>
      <prop id=p48-3>
        Echocardiogram showed right ventricular hypertrophy. </prop>
      <prop id=p48-4>
        Echocardiogram showed no dilatation,</prop>
      <prop id=p48-5>
```

```
      Echocardiogram showed no aortic valve pathology.    </prop>
    <prop id=p48-6>
      Echocardiogram showed no tricuspid regurgitation.    </prop>
  </propset>
  <focus entity='x'/>
  <rst-rel id=r48-1 name=join
           nuc=p48-1   sat=p48-2   ref=no  />
  <rst-rel id=r48-2 name=join
           nuc=p48-2   sat=p48-3   ref=no  />
  <rst-rel id=r48-3 name=join
           nuc=p48-3   sat=p48-4   ref=no  />
  <rst-rel id=r48-4 name=join
           nuc=p48-4   sat=p48-5   ref=no  />
  <rst-rel id=r48-5 name=join
           nuc=p48-5   sat=p48-6   ref=no  />
  <trans id=tx48-1 name=conj-simp   nuc=p48-1 sat=p48-2 />
  <trans id=tx48-2 name=conj-simp   nuc=tx48-1 sat=p48-3 />
  <trans id=tx48-3 name=conj-simp   nuc=tx48-2 sat=p48-4 />
  <trans id=tx48-4 name=conj-simp   nuc=tx48-3 sat=p48-5 />
  <trans id=tx48-5 name=conj-simp   nuc=tx48-4 sat=p48-6 />
  <seqorder valid=true />
  <conj id=c48-1 type=dist />
  <conj id=c48-2 type=dist />
  <comment text="- conjunction of negation is not deleted" />
</sentence>
<sentence id=s49>
  The patient exercised four minutes and 51 seconds of the BRUCE
  protocol to a maximum heart rate of 74, blood pressure 186/72
  and stopped because of shortness of breath, leg fatigue and
  noncompliance.
  <propset id=pset49-1>
    <prop id=p49-1>
      The patient exercised four minutes and 51 seconds of the
      BRUCE protocol.  </prop>
    <prop id=p49-2>
      The patient exercised to a maximum heart rate of 74.
      </prop>
    <prop id=p49-3>
      The patient exercised to blood pressure 186/72. </prop>
    <prop id=p49-4>
      The patient stopped.
      </prop>
    <prop id=p49-5>
     (because) The patient has shortness of breath.
     <prop id=p49-6>
     (because) The patient has leg fatigue.    </prop>
     <prop id=p49-6>
     (because) The patient has leg noncompliance.
```

```
        </prop>
    </propset>
    <focus entity='the patient'/>
    <rst-rel id=r49-1 name=elab
             nuc=p49-1   sat=p49-2   ref=no  />
    <rst-rel id=r49-2 name=elab
             nuc=p49-1   sat=p49-3   ref=no  />
    <rst-rel id=r49-3 name=sequence
             nuc=p49-1   sat=p49-4   ref=no  />
    <rst-rel id=r49-4 name=non-volitional-cause
             nuc=p49-4   sat=p49-5   ref=no  />
    <rst-rel id=r49-5 name=non-volitional-cause
             nuc=p49-4   sat=p49-6   ref=no  />
    <rst-rel id=r49-6 name=non-volitional-cause
             nuc=p49-4   sat=p49-7   ref=no  />
    <trans id=tx49-1 name=conj-simp   nuc=p49-2 sat=p49-3 />
    <trans id=tx49-2 name=pp-to   nuc=p49-1 sat=tx49-1 />
    <trans id=tx49-3 name=conj-simp   nuc=p49-4 sat=p49-5 />
    <trans id=tx49-4 name=conj-simp   nuc=tx49-3 sat=p49-6 />
    <trans id=tx49-5 name=cue-because-nominal nominal=yes nuc=p49-4 sat=tx49-4 />
    <trans id=tx49-6 name=conj-mult   nuc=tx49-2 sat=tx49-5 />
    <seqorder valid=true />
    <conj id=c49-1 type=coll />
    <conj id=c49-2 type=dist />
    <conj id=c49-3 type=dist />
</sentence>
<sentence id=s50>
  She was admitted on 12/91 for repeat catheterization to
  excluse infundibular stenosis and undergo pulmonic
  valvuloplasty.
    <propset id=pset50-1>
      <prop id=p50-1>
        She was admitted on 12/91 for repeated catheterization.
        </prop>
      <prop id=p50-2>
        She was admitted to excluse infundibular stenosis.
        </prop>
      <prop id=p50-3>
        She was admitted to undergo pulmonic valvuloplasty.
        </prop>
    </propset>
    <focus entity='x'/>
    <rst-rel id=r50-1 name=purpose
             nuc=p50-1   sat=p50-2   ref=no  />
    <rst-rel id=r50-2 name=purpose
             nuc=p50-1   sat=p50-3   ref=no  />
    <trans id=tx50-1 name=conj-simp   nuc=p50-2 sat=p50-3 />
    <trans id=tx50-2 name=cue-to   nuc=p50-1 sat=tx50-1 />
    <seqorder valid=true />
```

```
    <conj id=c50-1 type=dist />
</sentence>
<sentence id=s51>
  The patient refused surgery at that time and was not followed
  in clinic.
    <propset id=pset51-1>
      <prop id=p51-1>
        The patient refused surgery at that time.    </prop>
      <prop id=p51-2>
        The patient was not followed in clinic.  </prop>
    </propset>
    <focus entity='the patient'/>
    <rst-rel id=r51-1 name=join
             nuc=p51-1   sat=p51-2   ref=no  />
    <trans id=tx51-1 name=conj-mult-vp   nuc=p51-1 sat=p51-2 />
    <seqorder valid=true />
    <conj id=c51-1 type=dist />
</sentence>
<sentence id=s52>
  She finally presented in April of 1992 to clinic complaining
  of shortness of breath, with ordinary housework and now
  desires surgical intervention.
    <propset id=pset52-1>
      <prop id=p52-1>
        She finally presented in April of 1992 to clinic.
        </prop>
      <prop id=p52-2>
        She complained of shortness of breadth with ordinary
        housework.    </prop>
      <prop id=p52-3>
        She now desires surgical intervention. </prop>
    </propset>
    <focus entity='she'/>
    <rst-rel id=r52-1 name=elab
             nuc=p52-1   sat=p52-2   ref=no  />
    <rst-rel id=r52-2 name=join
             nuc=p52-2   sat=p52-3   ref=no  />
    <trans id=tx52-1 name=rel-reduced-ing   nuc=p52-1 sat=p52-2 />
    <trans id=tx52-2 name=conj-mult   nuc=tx52-1 sat=p52-3 />
    <seqorder valid=true />
    <conj id=c52-1 type=dist />
    <comment text="- it might be difficult to figure the scope of relative
               clause attachment to which entity in tx52-1.  An issue with
               tagging mechanism.  But specifying pt2-1 and p52-2 first
               is not the proposed ordering.  Need scope mechanism?" />
```

```
</sentence>
<sentence id=s53>
  Cardiac: S1/S2, regular with a II/VI systolic murmur heard
  best at the right and left upper sternal borders.
    <propset id=pset53-1>
      <prop id=p53-1>
        Cardiac S1/S2 is regular with a II/VI systolic murmur.
        </prop>
      <prop id=p53-2>
        The murmur is heard best at the right upper
        sternal borders.   </prop>
      <prop id=p53-3>
        The murmur is heard best at the left upper
        sternal borders.   </prop>
    </propset>
    <focus entity='cardiac'/>
    <rst-rel id=r53-1 name=elab
             nuc=p53-1   sat=p53-2   ref=no  />
    <rst-rel id=r53-2 name=elab
             nuc=p53-1   sat=p53-3   ref=no  />
    <trans id=tx53-1 name=conj-simp   nuc=p53-2 sat=p53-3 />
    <trans id=tx53-2 name=rel-reduced-del-wh-be   nuc=p53-1 sat=tx53-1 />
    <seqorder valid=true />
    <conj id=c53-1 type=dist />
</sentence>
<sentence id=s54>
  On 5/1/92, she underwent suture closure of her propatent
  foramen ovale with excision of the pulmonary valve and
  pericardial patch reconstruction of the pulmonary artery.
    <propset id=pset54-1>
      <prop id=p54-1>
        On 5/1/92, she underwent suture closure of her propatent
        foramen ovale. </prop>
      <prop id=p54-2>
        On 5/1/92, she underwent excision of the pulmonary valve. </prop>
      <prop id=p54-3>
        On 5/1/92, She underwent pericardial path reconstruction of the
        pulmonary artery.    </prop>
    </propset>
    <focus entity='she'/>
    <rst-rel id=r54-1 name=join
             nuc=p54-1   sat=p54-2   ref=no  />
    <rst-rel id=r54-2 name=join
             nuc=p54-2   sat=p54-3   ref=no  />
    <trans id=tx54-1 name=conj-simp   nuc=p54-2 sat=p54-3 />
    <trans id=tx54-2 name=conj-with   nuc=p54-1 sat=tx54-1 />
    <seqorder valid=true />
```

```
    <conj id=c54-1 type=dist />
    <comment text="- The using of 'with' is less common" />
</sentence>
<sentence id=s55>
  There were no complications and she received on exogenous
  blood intraoperatively.
    <propset id=pset55-1>
      <prop id=p55-1>
        There were no complications.    </prop>
      <prop id=p55-2>
        She received on exogenous blood intraoperatively.
        </prop>
    </propset>
    <focus entity='x'/>
    <rst-rel id=r55-1 name=join
             nuc=p55-1   sat=p55-2   ref=no  />
    <trans id=tx55-1 name=conj-mult   nuc=p55-1 sat=p55-2 />
    <seqorder valid=true />
    <conj id=c55-1 type=dist />
</sentence>
<sentence id=s56>
  On postoperative day one, she was extubated, weaned off the
  Nipride and a 10 millimeter systolic gradient was noted across
  the pulmonary valve during removal of the Swan-Gans catheter
  (PA 26, RV 36).
    <propset id=pset56-1>
      <prop id=p56-1>
        On postoperative day one, she was extubated.    </prop>
      <prop id=p56-2>
        She weaned off the Nipride.    </prop>
      <prop id=p56-3>
        A 10 millimeter systolic gradient was noted across the
        pulmonary valve during removal of the Swan-Gans catheter.
   </prop>
      <prop id=p56-4>
        Swan-Gans showed PA 26.    </prop>
      <prop id=p56-5>
        Swan-Gans showed RV 36.    </prop>
    </propset>
    <focus entity='she'/>
    <rst-rel id=r56-1 name=sequence
             nuc=p56-1   sat=p56-2   ref=no  />
    <rst-rel id=r56-2 name=sequence
             nuc=p56-2   sat=p56-3   ref=no  />
    <rst-rel id=r56-3 name=elab
             nuc=p56-3   sat=p56-4   ref=no  />
    <rst-rel id=r56-4 name=elab
```

```
          nuc=p56-3   sat=p56-5   ref=no  />
    <trans id=tx56-1 name=conj-simp   nuc=p56-4 sat=p56-5 />
    <trans id=tx56-2 name=parenthesis-contains   nuc=p56-3 sat=tx56-1 />
    <trans id=tx56-3 name=conj-mult-vp   nuc=p56-1 sat=p56-2 />
    <trans id=tx56-4 name=conj-mult   nuc=tx56-3 sat=tx56-2 />

    <seqorder valid=true />

    <conj id=c56-1 type=dist />
</sentence>

<sentence id=s57>
  A mammogram revealed a left breast 12 millimeter lesion medial
  to and below the left nipple.

    <propset id=pset57-1>
      <prop id=p57-1>
        A mammogram revealed a left breast 12 millimeter lesion.
         </prop>
      <prop id=p57-2>
        The lesion is medial to and below the left nipple.
        </prop>
    </propset>

    <focus entity='a mammogram'/>
    <rst-rel id=r57-1 name=elab
          nuc=p57-1   sat=p57-2   ref=no  />

    <trans id=tx57-1 name=rel-reduced-del-wh-be   nuc=p57-1 sat=p57-2 />

    <seqorder valid=true />

    <conj id=c57-1 type=coll />
</sentence>

<sentence id=s58>
  Discharge medications included Digoxin 0.125 milligrams each
  day and Ferrous sulfate 325 milligrams three times a day.

    <propset id=pset58-1>
      <prop id=p58-1>
        Discharge medications include Digoxin 0.125 milligrams
        each day.    </prop>
      <prop id=p58-2>
        Discharge medications include Ferrous sulfate 325
        milligrams three times a day.    </prop>
    </propset>

    <focus entity='discharge medications'/>

    <rst-rel id=r58-1 name=join
          nuc=p58-1   sat=p58-2   ref=no  />

    <trans id=tx58-1 name=conj-mult   nuc=p58-1 sat=p58-2 />

    <seqorder valid=true />

    <conj id=c58-1 type=dist />

    <comment text="- an example of np-pp conjunction" />
</sentence>
```

```
<sentence id=s59>
  She was instructed to follow-up with Dr. Smith in Cardiology
  Clinic on 5/29/92 and call for a follow-up appointment with
  Dr. Barney in four weeks and also in Breast Clinic.
    <propset id=pset59-1>
      <prop id=p59-1>
        She was instructed to follow-up with Dr. Smith on 5/29/92.
    </prop>
      <prop id=p59-2>
        Dr. Gogal is in Cardiology Clinic.    </prop>
      <prop id=p59-3>
        She was instructed to call for a follow-up appointment
        with Dr. Barney in four weeks.  </prop>
      <prop id=p59-4>
        She was instructed to call for a follow-up appointment also
        in Breast Clinic.  </prop>
    </propset>
    <focus entity='she'/>
    <rst-rel id=r59-1 name=elab
             nuc=p59-1   sat=p59-2   ref=no  />
    <rst-rel id=r59-2 name=join
             nuc=p59-2   sat=p59-3   ref=no  />
    <rst-rel id=r59-3 name=join
             nuc=p59-3   sat=p59-4   ref=no  />
    <trans id=tx59-1 name=pp-in   nuc=p59-2 sat=p59-2 />
    <trans id=tx59-2 name=conj-simp   nuc=p59-3 sat=p59-4 />
    <trans id=tx59-3 name=conj-simp   nuc=tx59-1 sat=tx59-2 />
    <seqorder valid=true />
    <conj id=c59-1 type=dist />
    <conj id=c59-2 type=dist />
    <comment text="- not clear why 'also' is used" />
</sentence>
<sentence id=s60>
  Mr. Jones is a seventy-two year old male with a past medical
  history of multiple vascular surgery reconstructive procedures
  as well as noninsulin dependent diabetes mellitus and
  hypertension, who presented to the Emergency Room complaining
  of frequency and urinary frequency and fever for two days.
    <propset id=pset60-1>
      <prop id=p60-1>
        Mr. Jones is a male.    </prop>
      <prop id=p60-2>
        Mr. Jones is a seventy-two year old.
      <prop id=p60-3>
        Mr. Jones has a past medical history of multiple vascular
        surgery reconstructive procedures.    </prop>
      <prop id=p60-4>
        Mr. Jones has a past medical history of noninsulin
```

```
      dependent diabetes mellitus.
      </prop>
    <prop id=p60-5>
      Mr. Jones has a past medical history of hypertension.    </prop>
    <prop id=p60-6>
      Mr. Jones presented to the Emergency Room.    </prop>
    <prop id=p60-7>
      He complained of urinary frequency for two days.
      </prop>
    <prop id=p60-8>
      He complained of fever for two days.    </prop>
  </propset>

  <focus entity='Mr. Jones'/>
  <rst-rel id=r60-1 name=elab
           nuc=p60-1   sat=p60-2  ref=no  />
  <rst-rel id=r60-2 name=elab
           nuc=p60-1   sat=p60-3  ref=no  />
  <rst-rel id=r60-3 name=elab
           nuc=p60-1   sat=p60-4  ref=no  />
  <rst-rel id=r60-4 name=elab
           nuc=p60-1   sat=p60-5  ref=no  />
  <rst-rel id=r60-5 name=elab
           nuc=p60-1   sat=p60-6   ref=no  />
  <rst-rel id=r60-6 name=elab
           nuc=p60-1   sat=p60-7  ref=no  />
  <rst-rel id=r60-7 name=elab
           nuc=p60-1   sat=p60-8  ref=no  />

  <trans id=tx60-1 name=adj   nuc=p60-1 sat=p60-2 />
  <trans id=tx60-2 name=conj-simp-as-well-as   nuc=p60-3 sat=p60-4 />
  <trans id=tx60-3 name=conj-simp   nuc=tx60-2 sat=p60-5 />
  <trans id=tx60-4 name=pp-with   nuc=tx60-1 sat=tx60-3 />
  <trans id=tx60-5 name=rel-wh   nuc=tx60-4 sat=p60-6 />
  <trans id=tx60-6 name=conj-simp   nuc=p60-7 sat=p60-8 />
  <trans id=tx60-7 name=rel-reduced-ing   nuc=tx60-4 sat=tx60-5 />

  <seqorder valid=false />

  <conj id=c60-1 type=dist />
  <conj id=c60-2 type=dist />
  <conj id=c60-3 type=dist />

  <comment text="- if the application of operator determines the
  order, then this violates the proposed order, rel-who is applied
  before rel-reduced-ing" />

</sentence>

<sentence id=s61>
 He was initially evaluated by his private medical doctor who
 gave him Tylenol and Keflex for temperature of 100 degrees and
 sent him to the Columbia Emergency Room.

  <propset id=pset61-1>
    <prop id=p61-1>
      He was initially evaluated by his private medical doctor.
```

```
  </prop>
     <prop id=p61-2>
       his private medical doctor gave him Tylenol and Keflex for
       temperature of 100 degrees.    </prop>
     <prop id=p61-3>
       his private doctor sent him to the Columbia
       Emergency Room.  </prop>
   </propset>
   <focus entity='He'/>
   <rst-rel id=r61-1 name=elab
            nuc=p61-1   sat=p61-2   ref=no  />
   <rst-rel id=r61-2 name=elab
            nuc=p61-1   sat=p61-3   ref=no  />
   <trans id=tx61-1 name=conj-mult-vp   nuc=p61-2 sat=p61-3 />
   <trans id=tx61-2 name=rel-wh   nuc=p61-1 sat=tx61-1 />
   <seqorder valid=true />
   <conj id=c61-1 type=coll />
   <conj id=c61-2 type=dist />
</sentence>
<sentence id=s62>
  Past surgical history is significant for a right axillofemoral
  profunda artery bypass and revision and repair of
  pseudoaneurysm on 04/93.
   <propset id=pset62-1>
     <prop id=p62-1>
       Past surgical history is significant for a right
       axillofemoral profunda artery bypass.    </prop>
     <prop id=p62-2>
       Past surgical history is significant for revision and repair of
       pseudoaneurysm on 04/93.   </prop>
   <focus entity='past medical history'/>
   <rst-rel id=r62-1 name=join
            nuc=p62-1   sat=p62-2   ref=no  />
   <trans id=tx62-1 name=conj-simp   nuc=p62-2 sat=p62-3 />
   <seqorder valid=true />
   <conj id=c62-1 type=dist />
   <conj id=c62-2 type=coll />
</sentence>
<sentence id=s63>
  A left carotid endarterectomy, a thrombectomy of an axillary
  femoral bypass graft on 05/93, removal of his infected bypass
  graft on 08/93 and a right above the knee amputation.
   <propset id=pset63-1>
     <prop id=p63-1>
       A left carotid endarterectomy. </prop>
     <prop id=p63-2>
```

```
      A thrombectomy of an axillary femoral bypass graft on
      05/93    </prop>
    <prop id=p63-3>
      Removal of his infected bypass graft on 08/93    </prop>
    <prop id=p63-4>
      A right above the knee amputation.     </prop>
  </propset>

  <focus entity='-X-'/>

  <rst-rel id=r63-1 name=join
           nuc=p63-1   sat=p63-2   ref=no  />
  <rst-rel id=r63-2 name=join
           nuc=p63-2   sat=p63-3   ref=no  />
  <rst-rel id=r63-3 name=join
           nuc=p63-3   sat=p63-4   ref=no  />

  <trans id=tx63-1 name=conj-simp   nuc=p63-1 sat=p63-2 />
  <trans id=tx63-2 name=conj-simp   nuc=tx63-1 sat=p63-3 />
  <trans id=tx63-3 name=conj-simp   nuc=tx63-2 sat=p63-4 />

  <seqorder valid=true />

  <conj id=c63-1 type=dist />

</sentence>

<sentence id=s64>
  Past medical history significant for hypertension, noninsulin
  dependent diabetes, cerebrovascular accident in 1988 and a
  past medical history of syphilis.

  <propset id=pset64-1>
    <prop id=p64-1>
      Past medical  history is significant for hypertension.
      </prop>
    <prop id=p64-2>
      Past medical  history is significant for noninsulin
      dependent diabetes.    </prop>
    <prop id=p64-3>
      Past medical  history is significant for cerebrovascular
      accident in 1988.    </prop>
    <prop id=p64-4>
      He has a past medical history of syphilis.    </prop>
  </propset>

  <focus entity='Past medical history'/>

  <rst-rel id=r64-1 name=join
           nuc=p64-1   sat=p64-2   ref=no  />
  <rst-rel id=r64-2 name=join
           nuc=p64-2   sat=p64-3   ref=no  />
  <rst-rel id=r64-3 name=join
           nuc=p64-3   sat=p64-4   ref=no  />

  <trans id=tx64-1 name=conj-simp   nuc=p64-1 sat=p64-2 />
  <trans id=tx64-2 name=conj-simp   nuc=tx64-1 sat=p64-3 />
  <trans id=tx64-3 name=conj-mult-del-wh-be   nuc=tx64-2 sat=p64-4 />
```

```
    <seqorder valid=true />
    <conj id=c64-1 type=dist />
</sentence>
<sentence id=s65>
  His medications include Micronase 10 milligrams by mouth twice
  a day, Procardia XL 30 milligrams by mouth each day and
  Multivitamins.
    <propset id=pset65-1>
      <prop id=p65-1>
        His medications include Micronase 10 milligrams by mouth
        twice a day.    </prop>
      <prop id=p65-2>
        His medications include Procardia XL 30 milligrams by
        mouth each day.  </prop>
      <prop id=p65-3>
        His medications include Multivitamins.  </prop>
    </propset>
    <focus entity='his medication'/>
    <rst-rel id=r65-1 name=join
             nuc=p65-1   sat=p65-2   ref=no  />
    <rst-rel id=r65-2 name=join
             nuc=p65-2   sat=p65-3   ref=no  />
    <trans id=tx65-1 name=conj-simp   nuc=p65-1 sat=p65-2 />
    <trans id=tx65-2 name=conj-simp   nuc=tx65-1 sat=p65-3 />
    <seqorder valid=true />
    <conj id=c65-1 type=dist />
</sentence>
<sentence id=s66>
  His lungs had good breath sounds bilaterally and were clear to
  auscultation.
    <propset id=pset66-1>
      <prop id=p66-1>
        His lungs had good breath sounds bilaterally.    </prop>
      <prop id=p66-2>
        The sounds were clear to auscultation. </prop>
    </propset>
    <focus entity='his lungs'/>
    <rst-rel id=r66-1 name=join
             nuc=p66-1   sat=p66-2   ref=no  />
    <trans id=tx66-1 name=conj-mult-vp   nuc=p66-1 sat=p66-2 />
    <seqorder valid=true />
    <conj id=c66-1 type=dist />
</sentence>
<sentence id=s67>
  His abdomen was firm and was tender diffusely to deep
```

```
  palpation.
    <propset id=pset67-1>
      <prop id=p67-1>
        His abdomen was firm     </prop>
      <prop id=p67-2>
        His abdomen was tender diffusely to deep palpation.
        </prop>
    </propset>

    <focus entity='his abdomen'/>
    <rst-rel id=r67-1 name=join
             nuc=p67-1   sat=p67-2   ref=no  />

    <trans id=tx67-1 name=conj-mult-vp   nuc=p67-1 sat=p67-2 />

    <seqorder valid=true />

    <conj id=c67-1 type=dist />

</sentence>

<sentence id=s68>
  He had no percussion or tap tenderness but did have focal
  guarding in the right and left lower quadrants.

    <propset id=pset68-1>
      <propset id=pset68-2>
      <prop id=p68-1>
        He had no percussion.    </prop>
      <prop id=p68-2>
        He had no tap tenderness.  </prop>
      </propset>
      <prop id=p68-3>
        He did had focal guarding in the right quadrants.
        </prop>
      <prop id=p68-4>
        He did had focal guarding in the left lower quadrants.
        </prop>
    </propset>

    <focus entity='he'/>
    <rst-rel id=r68-1 name=disj-simp-neg
             nuc=p68-1   sat=p68-2   ref=no  />
    <rst-rel id=r68-2 name=contrast
             nuc=pset68-2   sat=p68-3   ref=no  />
    <rst-rel id=r68-3 name=contrast
             nuc=pset68-2   sat=p68-4   ref=no  />

    <trans id=tx68-1 name=disj-simp-neg   nuc=p68-1 sat=p68-2 />
    <trans id=tx68-2 name=conj-simp   nuc=p68-3 sat=p68-4 />
    <trans id=tx68-3 name=cue-but-conj-vp   nuc=tx68-1 sat=tx68-2 />

    <seqorder valid=true />

    <conj id=c68-1 type=dist />

    <comment text="- interesting, negation over 'or'"    </prop>

</sentence>
```

```
<sentence id=s69>
  His left groin had a well-healed scar as well and his feet
  were warm.
    <propset id=pset69-1>
      <prop id=p69-1>
        His left groin had a well-healed scar as well. </prop>
      <prop id=p69-2>
        His feet were warm.    </prop>
    </propset>
    <focus entity='his left groin'/>
    <rst-rel id=r69-1 name=join
             nuc=p69-1   sat=p69-2   ref=no  />
    <trans id=tx69-1 name=conj-mult-sent   nuc=p69-1 sat=p69-2 />
    <seqorder valid=true />
    <conj id=c69-1 type=dist />
</sentence>
<sentence id=s70>
  The patient was emergently taken to the Operating Room with
  the presumptive diagnosis of perforated viscus and possible
  perforated diverticulitis.
    <propset id=pset70-1>
      <prop id=p70-1>
        The patient was emergently taken to the OR.    </prop>
      <prop id=p70-2>
        The patient has the presumptive diagnosis of perforated
        viscus.  </prop>
      <prop id=p70-3>
        The patient has the presumptive diagnosis of possible
        perforated diverticulitis.    </prop>
    </propset>
    <focus entity='the patient'/>
    <rst-rel id=r70-1 name=join
             nuc=p70-1   sat=p70-2   ref=no  />
    <rst-rel id=r70-2 name=join
             nuc=p70-2   sat=p70-3   ref=no  />
    <trans id=tx70-1 name=conj-simp   nuc=p70-2 sat=p70-3 />
    <trans id=tx70-2 name=pp-with   nuc=p70-1 sat=tx70-1 />
    <seqorder valid=true />
    <conj id=c70-1 type=dist />
</sentence>
<sentence id=s71>
  The sigmoid was noted to have multiple diverticuli and
  appeared to have been ruptured.
    <propset id=pset71-1>
      <prop id=p71-1>
        The sigmoid was noted to have multiple diverticuli.
```

```
      </prop>
    <prop id=p71-2>
      The sigmoid appeared to have been ruptured.    </prop>
  </propset>

  <focus entity='the sigmoid'/>
  <rst-rel id=r71-1 name=join
           nuc=p71-1   sat=p71-2   ref=no  />

  <trans id=tx71-1 name=conj-mult-vp   nuc=p71-1 sat=p71-2 />

  <seqorder valid=true />

  <conj id=c71-1 type=dist />

  <comment text="- first sentence was passive to allow deletion." />
</sentence>

<sentence id=s72>
  The patient was explored and underwent a left sigmoid
  resection with formation of a left rectal pouch and a left
  transverse colon colostomy.
  <propset id=pset72-1>
    <prop id=p72-1>
      The patient was explored.    </prop>
    <prop id=p72-2>
      The patient underwent a left sigmoid resection.  </prop>
    <prop id=p72-3>
      The sigmoid resection has formation of a left rectal
      pouch. </prop>
    <prop id=p72-4>
      The patient underwent a left transverse colon colostomy.    </prop>
  </propset>

  <focus entity='the patient'/>
  <rst-rel id=r72-1 name=join
           nuc=p72-1   sat=p72-2   ref=no  />
  <rst-rel id=r72-2 name=elab
           nuc=p72-2   sat=p72-3   ref=no  />
  <rst-rel id=r72-3 name=join
           nuc=p72-2   sat=p72-4   ref=no  />

  <trans id=tx72-1 name=pp-with   nuc=p72-2 sat=p72-3 />
  <trans id=tx72-2 name=conj-simp   nuc=tx72-1 sat=p72-4 />
  <trans id=tx72-3 name=conj-mult-vp   nuc=p72-1 sat=tx72-2 />

  <seqorder valid=true />

  <conj id=c72-1 type=dist />
  <conj id=c72-2 type=dist />
</sentence>

<sentence id=s73>
  The colostomy was left closed and was matured at the bedside
  with a Bovie electrocautery on the first postoperative day.
  <propset id=pset73-1>
    <prop id=p73-1>
```

```
      The colostomy was left closed. </prop>
    <prop id=p73-2>
      The colostomy was matured at the beside with a Bovie
      electrocautery on the first postoperative day. </prop>
  </propset>
  <focus entity='the colostomy'/>
  <rst-rel id=r73-1 name=join
           nuc=p73-1   sat=p73-2   ref=no  />
  <trans id=tx73-1 name=conj-mult-vp   nuc=p73-1 sat=p73-2 />
  <seqorder valid=true />
  <conj id=c73-1 type=dist />
</sentence>
<sentence id=s74>
  He remained intubated on the first postoperative day and his
  vital signs remained stable with the exception of his heart
  rate of 120-130.
  <propset id=pset74-1>
    <prop id=p74-1>
      He remained intubated ont eh first postoperative day.
      </prop>
    <prop id=p74-2>
      His vital signs remained stable.  </prop>
    <prop id=p74-3>
      His heart rate is 120-130.    </prop>
  </propset>
  <focus entity='he'/>
  <rst-rel id=r74-1 name=join
           nuc=p74-1   sat=p74-2   ref=no  />
  <rst-rel id=r74-2 name=condition-except
           nuc=p74-2   sat=p74-3   ref=no  />
  <trans id=tx74-1 name=cue-except-np nominal=yes   nuc=p74-2 sat=p74-3 />
  <trans id=tx74-2 name=conj-mult-sent   nuc=p74-1 sat=tx74-1 />
  <seqorder valid=true />
  <conj id=c74-1 type=dist />
</sentence>
<sentence id=s75>
  He remained without chest pain or shortness of breath
  throughout the first and second postoperative days but had
  been intubated during this period.
  <propset id=pset75-1>
    <propset id=pset75-2>
    <prop id=p75-1>
      Throughout the first postoperative day, he remained
      without chest pain.    </prop>
    <prop id=p75-2>
      Throughout the first postoperative day, he remained
      without shortness of breath.   </prop>
```

```
      <prop id=p75-3>
        Throughout the second postoperative day, he remained
        without chest pain.    </prop>
      <prop id=p75-4>
        Throughout the second postoperative day, he remained
        without shortness of breath. </prop>
      </propset>
      <prop id=p75-5>
        Throughout the first postoperative day, he had been
        intubated.     </prop>
      <prop id=p75-6>
        Throughout the second postoperative day, he had been
        intubated.     </prop>
    </propset>
    <focus entity='he'/>
    <rst-rel id=r75-1 name=join
             nuc=p75-1   sat=p75-2   ref=no  />
    <rst-rel id=r75-2 name=join
             nuc=p75-1   sat=p75-3   ref=no  />
    <rst-rel id=r75-3 name=join
             nuc=p75-3   sat=p75-4   ref=no  />
    <rst-rel id=r75-4 name=contrast
             nuc=pset75-2   sat=p75-5   ref=no  />
    <rst-rel id=r75-5 name=contrast
             nuc=pset75-2   sat=p75-6   ref=no  />
    <trans id=tx75-1 name=disj-simp-neg   nuc=p75-1 sat=p75-2 />
    <trans id=tx75-2 name=disj-simp-neg   nuc=p75-3 sat=p75-4 />
    <trans id=tx75-3 name=conj-simp-nested   nuc=tx75-1 sat=tx75-2 />
    <trans id=tx75-4 name=conj-simp   nuc=p75-5 sat=p75-6 />
    <trans id=tx75-5 name=conj-mult-vp-but   nuc=tx75-3 sat=tx75-4 />
    <seqorder valid=true />
    <conj id=c75-1 type=dist />
    <comment text="= negation 'without' scope over negation." />
</sentence>
<sentence id=s76>
  His first postoperative white count was 12.4 and hematocrit
  was 34.9.
    <propset id=pset76-1>
      <prop id=p76-1>
        His first postoperative white count was 12.4.    </prop>
      <prop id=p76-2>
        His first postoperative hematocrit was 34.9.    </prop>
    </propset>
    <focus entity='his first postoperative white count'/>
    <rst-rel id=r76-1 name=join
             nuc=p76-1   sat=p76-2   ref=no  />
    <trans id=tx76-1 name=conj-mult   nuc=p76-1 sat=p76-2 />
```

```
    <seqorder valid=true />
    <conj id=c76-1 type=dist />
    <comment text="- deletion of premodifiers across propositions.
                   surface ordering is in play here.  My algo. as it is
                   does not handle this case." />
</sentence>
<sentence id=s77>
  He made good urine and was felt to be adequately hydrated.
    <propset id=pset77-1>
      <prop id=p77-1>
        He made good urine.    </prop>
      <prop id=p77-2>
        He was felt to be adequately hydrated.  </prop>
    </propset>
    <focus entity='he'/>
    <rst-rel id=r77-1 name=join
             nuc=p77-1   sat=p77-2   ref=no  />
    <trans id=tx77-1 name=conj-mult-vp   nuc=p77-1 sat=p77-2 />
    <seqorder valid=true />
    <conj id=c77-1 type=dist />
    <comment text="- passive 2nd clause in order to delete." />

</sentence>
<sentence id=s78>
  The patient was extubated on the second postoperative day and
  did well.
    <propset id=pset78-1>
      <prop id=p78-1>
        The patient was extubated on the second postoperative day
        and    </prop>
      <prop id=p78-2>
        The patient did well.    </prop>
    </propset>
    <focus entity='the patient'/>
    <rst-rel id=r78-1 name=join
             nuc=p78-1   sat=p78-2   ref=no  />
    <trans id=tx78-1 name=conj-mult-vp   nuc=p78-1 sat=p78-2 />
    <seqorder valid=true />
    <conj id=c78-1 type=dist />
    <comment text="It is possible that 'on the second postoperative day'
                   might be attached to the 2nd proposition too.  That would
                   not be covered by deletion direction in my algorithm." />
</sentence>
```

```
<sentence id=s79>
  His T-max on the second postoperative day was 101 and he was
  continued on Cefoxitin and Flagyl.
    <propset id=pset79-1>
      <prop id=p79-1>
        His T-max on the second postoperative day was 101.
        </prop>
      <prop id=p79-2>
        He was continued on Cefoxitin on the second postoperative
        day.    </prop>
      <prop id=p79-3>
        He was continued on Flagyl on the second postoperative
        day.     </prop>
    </propset>
    <focus entity='his t-max'/>
    <rst-rel id=r79-1 name=join
             nuc=p79-1   sat=p79-2   ref=no  />
    <rst-rel id=r79-2 name=join
             nuc=p79-2   sat=p79-3   ref=no  />
    <trans id=tx79-1 name=conj-simp   nuc=p79-2 sat=p79-3 />
    <trans id=tx79-2 name=conj-mult-sent   nuc=p79-1 sat=tx79-1 />
    <seqorder valid=true />
    <conj id=c79-1 type=dist />
    <conj id=c79-2 type=dist />
</sentence>
<sentence id=s80>
  The patient was back in normal sinus rhythm by the end of the
  second postoperative day and continued to be digitalized.
    <propset id=pset80-1>
      <prop id=p80-1>
        The patient was back in normal sinus rhythm by the end of
        the second postoperative day.    </prop>
      <prop id=p80-2>
        The patient continued to be digitalized.  </prop>
    </propset>
    <focus entity='the patient'/>
    <rst-rel id=r80-1 name=sequence
             nuc=p80-1   sat=p80-2   ref=no  />
    <trans id=tx80-1 name=conj-mult-vp   nuc=p80-1 sat=p80-2 />
    <seqorder valid=true />
    <conj id=c80-1 type=dist />
</sentence>
<sentence id=s81>
  He was evaluated by Dr. Smith of Medicine who agreed that this
  episode was probably self-limited and was secondary to atrial
  distention and fluid overload on the second and third
```

postoperative day.


```
  <propset id=pset81-1>
    <prop id=p81-1>
      He was evaluated by Dr. Smith. </prop>
    <prop id=p81-2>
      Dr. Block is of Medicine department.    </prop>
    <prop id=p81-3>
      Dr. Block agreed that -X-  </prop>
    <propset id=pset81-2>
    <prop id=p81-4>
      This episode was probably self-limited.    </prop>
    <prop id=p81-5>
      This episode was secondary to atrial distention
       on the second postoperative day.    </prop>
    <prop id=p81-6>
      This episode was secondary to fluid overload
      on the second postoperative day.    </prop>
    <prop id=p81-7>
      This episode was secondary to atrial distention
      on the third postoperative day.    </prop>
    <prop id=p81-8>
      This episode was secondary to fluid overload on the
      third postoperative day.  </prop>
    </propset>
  </propset>

  <focus entity='he'/>
  <rst-rel id=r81-1 name=elab
           nuc=p81-1   sat=p81-2   ref=no  />
  <rst-rel id=r81-2 name=elab
           nuc=p81-1   sat=p81-3   ref=no  />
  <rst-rel id=r81-3 name=arg
           nuc=p81-3   sat=p81-4   ref=no  />
  <rst-rel id=r81-4 name=join
           nuc=p81-4   sat=p81-5   ref=no  />
  <rst-rel id=r81-5 name=join
           nuc=p81-5   sat=p81-6   ref=no  />
  <rst-rel id=r81-6 name=join
           nuc=p81-6   sat=p81-7   ref=no  />
  <rst-rel id=r81-7 name=join
           nuc=p81-7   sat=p81-8   ref=no  />

  <trans id=tx81-1 name=conj-simp   nuc=p81-5 sat=p81-6 />
  <trans id=tx81-2 name=conj-simp   nuc=p81-7 sat=p81-8 />
  <trans id=tx81-3 name=conj-simp-nested   nuc=tx81-1 sat=tx81-2 />
  <trans id=tx81-4 name=conj-mult-vp   nuc=p81-4 sat=tx81-3 />
  <trans id=tx81-5 name=arg   nuc=p81-3 sat=tx81-4 />
  <trans id=tx81-6 name=pp-of   nuc=p81-6 sat=p81-2 />
  <trans id=tx81-7 name=rel-wh   nuc=tx81-6 sat=tx81-5 />

  <seqorder valid=true />

  <conj id=c81-1 type=dist />
```

```
    <conj id=c81-2 type=dist />
    <conj id=c81-3 type=dist />
</sentence>

<sentence id=s82>
  He had recommended to continue the Digoxin and to discontinue
  the Verapami eventually and switched to an ACE inhibitor.

    <propset id=pset82-1>
      <prop id=p82-1>
        He had recommended to continue the Digoxin.    </prop>
      <prop id=p82-2>
        He had recommended to discontinue the Verapami eventually.
    </prop>
      <prop id=p82-3>
        He had recommended to switch to an ACE inhibitor.
        </prop>
    </propset>

    <focus entity='he'/>

    <rst-rel id=r82-1 name=join
             nuc=p82-1   sat=p82-2   ref=no  />
    <rst-rel id=r82-2 name=join
             nuc=p82-2   sat=p82-3   ref=no  />

    <trans id=tx82-1 name=conj-mult   nuc=p82-1 sat=p82-2 />
    <trans id=tx82-2 name=conj-mult   nuc=tx82-1 sat=p82-3 />

    <seqorder valid=true />

    <conj id=c82-1 type=dist />
    <conj id=c82-2 type=dist />

</sentence>

<sentence id=s83>
  His I's and O's after diuresis in the unit were roughly even.

    <propset id=pset83-1>
      <prop id=p83-1>
        His I's after diuresis in the unit were X.    </prop>
      <prop id=p83-2>
        His O's after diuresis in the unit were X'.   </prop>
    </propset>

    <focus entity='his I's'/>

    <rst-rel id=r83-1 name=comparative
             nuc=p83-1   sat=p83-2   ref=no  />

    <trans id=tx83-1 name=comparison-are-even   nuc=p83-1 sat=p83-2 />

    <seqorder valid=true />

    <conj id=c83-1 type=coll />
</sentence>

<sentence id=s84>
  The fifth postoperative day, he was tolerating clear liquid
  and after irrigation and digitalization of his colostomy,
  began producing an abundant amount of soft stool and air.
```

```
    <propset id=pset84-1>
      <prop id=p84-1>
        The fifth postoperative day, he was tolerating clear
        liquid.  </prop>
      <prop id=p84-2>
        After irrigation his colostomy, began producing an
        abundant amount of soft stool. </prop>
      <prop id=p84-3>
        After digitalization his colostomy, began producing an
        abundant amount of soft stool. </prop>
      <prop id=p84-4>
        After irrigation his colostomy, began producing an
        abundant amount of air.    </prop>
      <prop id=p84-5>
        After digitalization his colostomy, began producing an
        abundant amount of air.    </prop>
    </propset>

    <focus entity='he'/>

    <rst-rel id=r84-1 name=join
             nuc=p84-1   sat=p84-2   ref=no  />
    <rst-rel id=r84-2 name=join
             nuc=p84-2   sat=p84-3   ref=no  />
    <rst-rel id=r84-3 name=join
             nuc=p84-3   sat=p84-4   ref=no  />
    <rst-rel id=r84-4 name=join
             nuc=p84-4   sat=p84-5   ref=no  />

    <trans id=tx84-1 name=conj-simp   nuc=p84-2 sat=p84-3 />
    <trans id=tx84-2 name=conj-simp   nuc=p84-4 sat=p84-5 />
    <trans id=tx84-3 name=conj-simp-nested   nuc=tx84-1 sat=tx84-2 />
    <trans id=tx84-4 name=conj-mult-sent   nuc=p84-1 sat=tx84-3 />

    <seqorder valid=true />

    <conj id=c84-1 type=dist />
    <conj id=c84-2 type=dist />
    <conj id=c84-3 type=dist />

</sentence>

<sentence id=s85>
  His magnesium, potassium and other electrolytes were found to
  be normal and the follow up EKGs and SMACs were without change.

    <propset id=pset85-1>
      <prop id=p85-1>
        His magnesium was found to be normal.    </prop>
      <prop id=p85-2>
        His potassium was found to be normal.    </prop>
      <prop id=p85-3>
        His other electrolytes were found to be normal.  </prop>
      <prop id=p85-4>
        The followup EKGs was without change. </prop>
      <prop id=p85-5>
```

```
      The followup SMACs was without change.  </prop>
    </propset>

    <focus entity='his magnesium'/>
    <rst-rel id=r85-1 name=join
             nuc=p85-1   sat=p85-2   ref=no  />
    <rst-rel id=r85-2 name=join
             nuc=p85-2   sat=p85-3   ref=no  />
    <rst-rel id=r85-3 name=join
             nuc=p85-3   sat=p85-4   ref=no  />
    <rst-rel id=r85-4 name=join
             nuc=p85-4   sat=p85-5   ref=no  />

    <trans id=tx85-1 name=conj-simp   nuc=p85-1 sat=p85-2 />
    <trans id=tx85-2 name=conj-simp   nuc=tx85-1 sat=p85-3 />
    <trans id=tx85-3 name=conj-simp   nuc=p85-4 sat=p85-5 />
    <trans id=tx85-4 name=conj-mult-sent   nuc=tx85-2 sat=tx85-3 />

    <seqorder valid=true />

    <conj id=c85-1 type=dist />
    <conj id=c85-2 type=dist />
    <conj id=c85-3 type=dist />

    <comment text="- the use of 'other' is tricky." />
</sentence>
<sentence id=s86>
  The patient was taken off his Digoxin and his Verapamil
  secondary to this junctional rhythm and was begun on ACE
  inhibition, Vasotec 5 milligrams by mouth each day for blood
  pressure control.
    <propset id=pset86-1>
      <prop id=p86-1>
        The patient was taken off his Digoxin. </prop>
      <prop id=p86-2>
        The patient was taken off his Verapamil  </prop>
      <prop id=p86-3>
        his Verapamil was secondary to this junctional rhythm.
        </prop>
      <prop id=p86-4>
        The patient was begun on ACE inhibition.   </prop>
      <prop id=p86-5>
        The patient was begun on Vasotec 5 milligrams by mouth each
        day.   </prop>
      <prop id=p86-6>
        Vasotec is for blood pressure control.   </prop>
    </propset>

    <focus entity='the patient'/>
    <rst-rel id=r86-1 name=join
             nuc=p86-1   sat=p86-2   ref=no  />
    <rst-rel id=r86-2 name=elab
             nuc=p86-2   sat=p86-3   ref=no  />
    <rst-rel id=r86-3 name=join
```

```
          nuc=p86-3   sat=p86-4   ref=no  />
    <rst-rel id=r86-4 name=join
          nuc=p86-4   sat=p86-5   ref=no  />
    <rst-rel id=r86-5 name=elab
          nuc=p86-5   sat=p86-6   ref=no  />

    <trans id=tx86-1 name=pp-for   nuc=p86-5 sat=p86-6 />
    <trans id=tx86-2 name=rel-reduced-del-wh-be nuc=p86-2 sat=p86-3 />
    <trans id=tx86-3 name=conj-simp   nuc=p86-1 sat=tx86-2 />
    <trans id=tx86-4 name=conj-simp   nuc=p86-4 sat=tx86-1 />
    <trans id=tx86-5 name=conj-mult-vp   nuc=tx86-3 sat=tx86-4 />

    <seqorder valid=true />

    <conj id=c86-1 type=dist />
    <conj id=c86-2 type=dist />
</sentence>

<sentence id=s87>
  The patient continued to do well and his diet was advanced
  which he tolerated without any abdominal distention, nausea or
  vomiting.
    <propset id=pset87-1>
      <prop id=p87-1>
        The patient continued to do well.    </prop>
    <propset id=pset87-2>
      <prop id=p87-2>
        His diet was advanced. </prop>
      <prop id=p87-3>
        He tolerated diet without any abdominal distention.  </prop>
      <prop id=p87-4>
        He tolerated diet without any nausea.   </prop>
      <prop id=p87-5>
        He tolerated diet without any vomiting.    </prop>
    </propset>
    </propset>

    <focus entity='the patient'/>
    <rst-rel id=r87-1 name=join
          nuc=p87-1   sat=p87-2   ref=no  />
    <rst-rel id=r87-2 name=elab
          nuc=p87-2   sat=p87-3   ref=no  />
    <rst-rel id=r87-3 name=elab
          nuc=p87-2   sat=p87-4   ref=no  />
    <rst-rel id=r87-4 name=elab
          nuc=p87-2   sat=p87-5   ref=no  />

    <trans id=tx87-1 name=conj-simp   nuc=p87-3 sat=p87-4 />
    <trans id=tx87-2 name=conj-simp   nuc=tx87-1 sat=p87-5 />
    <trans id=tx87-3 name=rel-wh-extraction   nuc=p87-2 sat=tx87-2 />
    <trans id=tx87-4 name=conj-mult-sent   nuc=p87-1 sat=tx87-3 />

    <seqorder valid=true />

    <conj id=c87-1 type=dist />

</sentence>
```

```
<sentence id=s88>
  His family has been taught colostomy care and he will have
  both visiting nurse services and Home Health Aide to help him
  at home.
    <propset id=pset88-1>
      <prop id=p88-1>
        His family has been taught colostomy care.    </prop>
      <prop id=p88-2>
        He will have visiting nurse services to help him at home.
    </prop>
      <prop id=p88-3>
        He will have Home Health Aide to help him at home.
        </prop>
    </propset>
    <focus entity='his family'/>
    <rst-rel id=r88-1 name=join
             nuc=p88-1   sat=p88-2   ref=no  />
    <rst-rel id=r88-2 name=join
             nuc=p88-2   sat=p88-3   ref=no  />
    <trans id=tx88-1 name=conj-simp-quant-both   nuc=p88-2 sat=p88-3 />
    <trans id=tx88-2 name=conj-mult-sent   nuc=p88-1 sat=tx88-1 />
    <seqorder valid=true />
    <conj id=c88-1 type=dist />
    <conj id=c88-2 type=dist />
</sentence>
<sentence id=s89>
  Discharge medications include only Micronase 10 milligrams by
  mouth twice a day and Vasotec 50 milligrams by mouth each day.
    <propset id=pset89-1>
      <prop id=p89-1>
        Discharge medications include Micronase 10 milligrams by
        mouth twice a day.     </prop>
      <prop id=p89-2>
        Discharge medications include Vasotec 50 milligrams by
        mouth each day.  </prop>
    </propset>
    <focus entity='discharge medications'/>
    <rst-rel id=r89-1 name=join
             nuc=p89-1   sat=p89-2   ref=no  />
    <trans id=tx89-1 name=conj-simp   nuc=p89-1 sat=p89-2 />
    <seqorder valid=true />
    <conj id=c89-1 type=dist />
    <comment text="- 'only' is removed." />
</sentence>
<sentence id=s90>
  The patient is a sixty-five year old white male with
```

```
  hypertension and former smoker, admitted for coronary artery
  bypass grafting.
    <propset id=pset90-1>
    <propset id=pset90-2>
      <prop id=p90-1>
        The patient is male.    </prop>
      <prop id=p90-2>
        the patient is a sixty-five year old.    </prop>
      <prop id=p90-3>
        the patient is a white person.  </prop>
      <prop id=p90-4>
        the patient has hypertension.    </prop>
    </propset>
      <prop id=p90-5>
        the patient is former smoker.    </prop>
      <prop id=p90-6>
        the patient admitted for coronary artery bypass grafting.
    </prop>
    </propset>

    <focus entity='the patient'/>
    <rst-rel id=r90-1 name=elab
             nuc=p90-1   sat=p90-2   ref=no  />
    <rst-rel id=r90-2 name=elab
             nuc=p90-1   sat=p90-3   ref=no  />
    <rst-rel id=r90-3 name=elab
             nuc=p90-1   sat=p90-4   ref=no  />
    <rst-rel id=r90-4 name=elab
             nuc=p90-1   sat=p90-5   ref=no  />
    <rst-rel id=r90-5 name=elab
             nuc=p90-5   sat=p90-6   ref=no  />

    <trans id=tx90-1 name=adj   nuc=p90-1 sat=p90-2 />
    <trans id=tx90-2 name=adj   nuc=tx90-1 sat=p90-3 />
    <trans id=tx90-3 name=pp-with   nuc=tx90-2 sat=p90-4 />
    <trans id=tx90-4 name=conj-mult-del-wh-be   nuc=tx90-3 sat=p90-5 />
    <trans id=tx90-5 name=rel-reduced-del-wh-be  nuc=tx90-4 sat=p90-6 />

    <seqorder valid=false />

    <conj id=c90-1 type=dist />

    <comment text="- the transformation of an elaboration into 'and'
    messed up the order of transformation operators." />

</sentence>

<sentence id=s91>
  There, he ruled in for an acute non-Q wave myocardial
  infarction and was treated with Lopressor, IV Nitroglycerin
  and Heparin.
    <propset id=pset91-1>
      <prop id=p91-1>
        There, he ruled in for an acute non-Q wave myocardial
        infarction.    </prop>
      <prop id=p91-2>
```

```
      He was treated with Lopressor. </prop>
    <prop id=p91-3>
      He was treated with IV Nitroglycerin.    </prop>
    <prop id=p91-4>
      He was treated with IV Heparin.  </prop>
  </propset>
  <focus entity='he'/>
  <rst-rel id=r91-1 name=join
           nuc=p91-1   sat=p91-2   ref=no  />
  <rst-rel id=r91-2 name=join
           nuc=p91-2   sat=p91-3   ref=no  />
  <rst-rel id=r91-3 name=join
           nuc=p91-3   sat=p91-4   ref=no  />
  <trans id=tx91-1 name=conj-simp   nuc=p91-2 sat=p91-3 />
  <trans id=tx91-2 name=conj-simp   nuc=tx91-1 sat=p91-4 />
  <trans id=tx91-3 name=conj-mult   nuc=p91-1 sat=tx91-2 />
  <seqorder valid=true />
  <conj id=c91-1 type=dist />
  <conj id=c91-2 type=dist />
</sentence>
<sentence id=s92>
  On 08/31/93, he underwent cardiac catheterization which
  revealed left ventricular ejection fraction of 30% with global
  hypokinesis, PA 55/25, mean 35, wedge 20 with V wave to 27 and
  cardiac output 4.4.
  <propset id=pset92-1>
    <prop id=p92-1>
      On 08/31/93, he underwent cardiac catheterization.
      </prop>
    <prop id=p92-2>
      The cardiac catheterization revealed left ventricular
      eject fraction of 30%. </prop>
    <prop id=p92-3>
      The cardiac catheterization revealed global hypokinesis.
      </prop>
    <prop id=p92-4>
      The catheterization revealed PA 55/25. </prop>
    <prop id=p92-5>
      PA has mean of 35.    </prop>
    <prop id=p92-6>
      The catheterization revealed wedge 20 with V wave to 27.
       </prop>
    <prop id=p92-7>
      The catheterization revealed cardiac output 4.4.  </prop>
  </propset>
  <focus entity='he'/>
  <rst-rel id=r92-1 name=elab
           nuc=p92-1   sat=p92-2   ref=no  />
  <rst-rel id=r92-2 name=elab
```

```
             nuc=p92-1   sat=p92-3   ref=no  />
    <rst-rel id=r92-3 name=elab
             nuc=p92-1   sat=p92-4   ref=no  />
    <rst-rel id=r92-4 name=elab
             nuc=p92-4   sat=p92-5   ref=no  />
    <rst-rel id=r92-5 name=join
             nuc=p92-1   sat=p92-6   ref=no  />
    <rst-rel id=r92-6 name=join
             nuc=p92-1   sat=p92-7   ref=no  />

    <trans id=tx92-1 name=rel-reduced-del-wh-be   nuc=p92-4 sat=p92-5 />
    <trans id=tx92-2 name=conj-simp   nuc=p92-2 sat=p92-3 />
    <trans id=tx92-3 name=conj-simp   nuc=tx92-2 sat=tx92-1 />
    <trans id=tx92-4 name=conj-simp   nuc=tx92-3 sat=p92-6 />
    <trans id=tx92-5 name=conj-simp   nuc=tx92-4 sat=p92-7 />
    <trans id=tx92-6 name=rel-wh   nuc=p92-1 sat=tx92-5 />

    <seqorder valid=true />

    <conj id=c92-1 type=dist />

</sentence>

<sentence id=s93>
  Coronary angiogram revealed 95% left main stenosis, 90%
  proximal right coronary artery stenosis, 100% proximal
  circumflex occlusion and a small diffusely diseased left
  anterior descending.

    <propset id=pset93-1>
      <prop id=p93-1>
        Coronary angiogram revealed 95% left main stenosis.
        </prop>
      <prop id=p93-2>
        Coronary angiogram revealed 90% proximal right coronary
        artery stenosis,  </prop>
      <prop id=p93-3>
        Coronary angiogram revealed 100% proximal circumflex
        occlusion.    </prop>
      <prop id=p93-4>
        Coronary angiogram revealed a small diffusely diseased
        left anterior descending.    </prop>
    </propset>

    <focus entity='coronary angiogram'/>
    <rst-rel id=r93-1 name=join
             nuc=p93-1   sat=p93-2   ref=no  />
    <rst-rel id=r93-2 name=join
             nuc=p93-2   sat=p93-3   ref=no  />
    <rst-rel id=r93-3 name=join
             nuc=p93-3   sat=p93-4   ref=no  />

    <trans id=tx93-1 name=conj-simp   nuc=p93-1 sat=p93-2 />
    <trans id=tx93-2 name=conj-simp   nuc=tx93-1 sat=p93-3 />
    <trans id=tx93-3 name=conj-simp   nuc=tx93-2 sat=p93-4 />

    <seqorder valid=true />
```

```
    <conj id=c93-1 type=dist />
</sentence>
<sentence id=s94>
  Following the catheterization, he developed pulmonary edema
  and hypotension which did not respond to fluids but required
  Dobutamine.
    <propset id=pset94-1>
      <prop id=p94-1>
        Following the catheterization, he developed pulmonary
        edema. </prop>
      <prop id=p94-2>
        Following the catheterization, he developed hypotension.
         </prop>
      <prop id=p94-3>
        The hypotention did not respond to fluids    </prop>
      <prop id=p94-4>
        The hypotention required Dobutamine.
        </prop>
    </propset>
    <focus entity='he'/>
    <rst-rel id=r94-1 name=join
             nuc=p94-1   sat=p94-2   ref=no  />
    <rst-rel id=r94-2 name=elab
             nuc=p94-2   sat=p94-3   ref=no  />
    <rst-rel id=r94-3 name=elab
             nuc=p94-2   sat=p94-4   ref=no  />
    <trans id=tx94-1 name=conj-mult-vp-but   nuc=p94-3 sat=p94-4 />
    <trans id=tx94-2 name=rel-wh   nuc=p94-2 sat=tx94-1 />
    <trans id=tx94-3 name=conj-simp   nuc=p94-1 sat=tx94-2 />
    <seqorder valid=true />
    <conj id=c94-1 type=dist />
    <comment text="- probably should have multiple rhetorical
                 relations in order to address 'but'." />
</sentence>
<sentence id=s95>
  He was diuresed and the chest pain subsided with decreasing
  Dobutamine dosages.
    <propset id=pset95-1>
      <prop id=p95-1>
        He was diuresed.   </prop>
    <propset id=pset95-2>
      <prop id=p95-2>
        The chest pain subsided.  </prop>
      <prop id=p95-3>
        He received decreasing Dobutamine dosages.  </prop>
    </propset>
    </propset>
```

```
    <focus entity='he'/>
    <rst-rel id=r95-1 name=join
             nuc=p95-1   sat=p95-2   ref=no  />
    <rst-rel id=r95-2 name=non-volitional-cause
             nuc=p95-2   sat=p95-3   ref=no  />

    <trans id=tx95-1 name=conj-with   nuc=p95-2 sat=p95-3 />
    <trans id=tx95-2 name=conj-mult-sent   nuc=p95-1 sat=tx95-1 />

    <seqorder valid=true />

    <conj id=c95-1 type=dist />
</sentence>
<sentence id=s96>
  Of note, en route in the ambulance, he experienced four
  episodes of severe chest pain and upon arrival, had 10/10
  chest pain with tachycardia and diffuse ST segment depressions
  and hyperacute T waves in leads V1-V3.
    <propset id=pset96-1>
      <prop id=p96-1>
        Of note, en route in the ambulance, he experienced four
        episodes of severe chest pain. </prop>
      <prop id=p96-2>
        Upon arrival, he had 10/10 chest pain. </prop>
      <prop id=p96-3>
        Upon arrival, he had tachycardia.    </prop>
      <prop id=p96-4>
        Upon arrival, he had diffuse ST segment depressions.
        </prop>
      <prop id=p96-5>
        Upon arrival, he had hyperacute T waves in leads V1-V3.
         </prop>
    </propset>

    <focus entity='he'/>
    <rst-rel id=r96-1 name=join
             nuc=p96-1   sat=p96-2   ref=no  />
    <rst-rel id=r96-2 name=join
             nuc=p96-2   sat=p96-3   ref=no  />
    <rst-rel id=r96-3 name=join
             nuc=p96-3   sat=p96-4   ref=no  />
    <rst-rel id=r96-4 name=join
             nuc=p96-4   sat=p96-5   ref=no  />

    <trans id=tx96-1 name=conj-simp   nuc=p96-2 sat=p96-3 />
    <trans id=tx96-2 name=conj-simp   nuc=tx96-1 sat=p96-4 />
    <trans id=tx96-3 name=conj-simp   nuc=tx96-2 sat=p96-5 />
    <trans id=tx96-4 name=conj-mult-vp   nuc=p96-1 sat=tx96-3 />

    <seqorder valid=true />

    <conj id=c96-1 type=dist />
    <conj id=c96-2 type=dist />
    <conj id=c96-3 type=dist />
```

```
</sentence>
<sentence id=s97>
  The Dobutamine was stopped and was given IV Nitroglycerin and
  Morphine.
    <propset id=pset97-1>
      <prop id=p97-1>
        The Dobutamine was stopped.    </prop>
      <prop id=p97-2>
        He was given IV Nitroglycerin. </prop>
      <prop id=p97-3>
        He was given Morphine.  </prop>
    </propset>
    <focus entity='the Dobutamine'/>
    <rst-rel id=r97-1 name=join
             nuc=p97-1   sat=p97-2   ref=no  />
    <rst-rel id=r97-2 name=join
             nuc=p97-2   sat=p97-3   ref=no  />
    <trans id=tx97-1 name=conj-simp   nuc=p97-2 sat=p97-3 />
    <trans id=tx97-2 name=conj-mult-implied-subj-del   nuc=p97-1 sat=tx97-1 />
    <seqorder valid=true />
    <conj id=c97-1 type=dist />
    <conj id=c97-2 type=dist />
</sentence>
<sentence id=s98>
  Forty-five minutes later, the chest pain subsided and EKG
  showed normalized ST segments.
    <propset id=pset98-1>
      <prop id=p98-1>
        Forty-five minutes later, the chest pain subsided.
        </prop>
      <prop id=p98-2>
        Forty-five minutes later, EKG showed normalized ST
        segments.    </prop>
    </propset>
    <focus entity='the chest pain'/>
    <rst-rel id=r98-1 name=sequence
             nuc=p98-1   sat=p98-2   ref=no  />
    <trans id=tx98-1 name=conj-mult-sent   nuc=p98-1 sat=p98-2 />
    <seqorder valid=true />
    <conj id=c98-1 type=dist />
</sentence>
<sentence id=s99>
  As above, status post left hemispheric cerebrovascular
  accident fifteen years ago with residual right hemiparesis and
  expressive aphasia, seizure disorder following the
  cerebrovascular accident, former smoker quit ten years ago.
```

```
    <propset id=pset99-1>
      <prop id=p99-1>
        (As above) the patient has status post left hemispheric
        cerebrovascular accident fifteen years ago.    </prop>
      <prop id=p99-2>
        The accident caused residual right hemiparesis.    </prop>
      <prop id=p99-3>
        The accident caused expressive aphasia,    </prop>
      <prop id=p99-4>
        The patient had seizure disorder following the
        cerebrovascular accident.    </prop>
      <propset id=pset99-2>
      <prop id=p99-5>
        He is a former smoker. </prop>
      <prop id=p99-6>
        He quit ten years ago.  </prop>
      </propset>
    </propset>
    <focus entity='-X-'/>
    <rst-rel id=r99-1 name=elab
             nuc=p99-1   sat=p99-2   ref=no  />
    <rst-rel id=r99-2 name=elab
             nuc=p99-1   sat=p99-3   ref=no  />
    <rst-rel id=r99-3 name=elab
             nuc=p99-1   sat=p99-4   ref=no  />
    <rst-rel id=r99-4 name=elab
             nuc=p99-1   sat=p99-5   ref=no  />
    <rst-rel id=r99-5 name=elab
             nuc=p99-5   sat=p99-6   ref=no  />
    <trans id=tx99-1 name=rel-reduced-del-wh   nuc=p99-5 sat=p99-6 />
    <trans id=tx99-2 name=conj-simp   nuc=p99-2 sat=p99-3 />
    <trans id=tx99-3 name=pp-with     nuc=p99-1 sat=tx99-2 />
    <trans id=tx99-4 name=conj-mult   nuc=p99-4 sat=tx99-1 />
    <trans id=tx99-5 name=rel-reduced-del-wh-verb nuc=tx99-3 sat=tx99-4 />
    <seqorder valid=true />
    <conj id=c99-1 type=dist />
</sentence>
<sentence id=s100>
  Neurological: alert and oriented times two, could not recall
  the date and right hemiparesis.
    <propset id=pset100-1>
      <prop id=p100-1>
        He is alert.    </prop>
      <prop id=p100-2>
        His is oriented times two.    </prop>
      <prop id=p100-3>
        He could not recall the date.    </prop>
      <prop id=p100-4>
        He has right hemiparesis.    </prop>
    </propset>
```

```
    <focus entity='-X-'/>
    <rst-rel id=r100-1 name=join
             nuc=p100-1   sat=p100-2   ref=no  />
    <rst-rel id=r100-2 name=join
             nuc=p100-2   sat=p100-3   ref=no  />
    <rst-rel id=r100-3 name=join
             nuc=p100-3   sat=p100-4   ref=no  />

    <trans id=tx100-1 name=conj-simp   nuc=p100-1 sat=p100-2 />
    <trans id=tx100-2 name=conj-mult-vp   nuc=tx100-1 sat=p100-3 />
    <trans id=tx100-3 name=conj-mult-del-wh-be   nuc=tx100-2 sat=p100-4 />

    <seqorder valid=true />

    <conj id=c100-1 type=dist />
    <conj id=c100-2 type=dist />

</sentence>

</document>
```

References

Aho, Alfred V., John E. Hopcroft, and Jeffrey D. Ullman. 1974. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, Massachusetts.

Appelt, Douglas E. 1985. *Planning English Sentences*. Cambridge University Press, Cambridge, UK.

Bache, Carl. 1978. *The Order of Premodifying Adjectives in Present-Day English*. Odense University Press.

Ballard, D., R. Conrad, and Robert Longacre. 1971. The deep and surface grammar of interclausal relations. *Foundation of Language*, 4:70–118.

Barwise, Jon and Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4:159–219.

Bateman, John A., Thomas Kamps, Jorg Kleinz, and Klaus Reichenberger. 1998. Communicative goal-driven NL generation and data-driven graphics generation: an architectural synthesis for multimedia page generation. In *Proceedings of the 9th International Workshop on Natural Language Generation*, pages 8–17.

Borgida, Alexander, Ronald Brachman, Deborah McGuinness, and Lori Alperin Resnick. 1989. CLASSIC: A structural data model for objects. In *ACM SIGMOD International Conference on Management of Data*.

Bouma, Gosse, Robert Malouf, and Ivan A. Sag. in press. Satisfying constraints on extraction and adjunction. *Natural Language and Linguistic Theory*.

Brachman, Ronald J. and J. Schmolze. 1985. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, pages 171–216, August.

Brandow, Ronald, Karl Mitze, and Lisa F. Rau. 1995. Automatic condensation of electronic publications by sentence selection. *Information Processing and Management*, 31(5):675–685.

Brill, Eric. 1992. A simple rule-based part of speech tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing*, pages 152–155.

Brown, E. Keith. 1991. Transformational-generative grammar. In Kirsten Malmkjar, editor, *The Linguistics Encyclopedia*. Routledge, London, pages 482–497.

Brownston, Lee, Robert Farrell, Elaine Kant, and Nancy Martin. 1985. *Programming Expert Systems in OPS5: An Introduction to Rule-Based Programming.* Addison-Wesley, Reading, MA.

Callaway, Charles B. and James C. Lester. 1997. Dynamically improving explanations: A revision-based approach to explanation generation. In *Proceedings of the 15th IJCAI*, pages 952–958, Nagoya, Japan.

Carpenter, Bob. 1992. *The Logic of Typed Feature Structures.* Cambridge University Press, New York, NY.

Carpenter, Bob. 1997. *Type-Logical Semantics.* MIT Press, Cambridge, Massachusetts.

Cheng, Hua. 1998. Embedding new information into referring expressions. In *Proc. of COLING-ACL'98*, pages 1478–1480, Montreal, Canada.

Cheng, Hua and Chris Mellish. 2000. An empirical analysis of constructing non-restrictive np components to express semantic relations. In *Proc. of the International Natural Language Generation Conference*, Mitzpe Ramon, Israel.

Cheng, Hua, Chris Mellish, and Michael O'Donnell. 1997. Aggregation based on text structure for descriptive text generation. In *Proc. of the PhD Workshop on Natural Language Generation, 9th European Summer School in Logic, Language and Information*, France.

Chomsky, Noam. 1957. *Syntactic Structures.* Mouton, The Hague.

Chomsky, Noam. 1965. *Aspects of a Theory of Syntax.* MIT Press, Cambridge, Mass.

Cimino, James J., Paul D. Clayton, George Hripcsak, and Stephen B. Johnson. 1994. Knowledge-based approaches to the maintenance of a large controlled medical terminology. *The Journal of the American Medical Informatics Association*, 1(1):35–50.

Cook, Malcom E., Wendy G. Lehnert, and David D. McDonald. 1984. Conveying implicit content in narrative summaries. In *Proceedings of the Tenth International Conference on Computational Linguistics (COLING-84) and the 22nd Annual Meeting of the ACL*, pages 5–7, Stanford University, Stanford, CA.

Copestake, Ann, Dan Flickinger, Ivan A. Sag, and Carl J. Pollard. 1999. Minimal recursion semantics: An introduction. Manuscript available via http://lingo.stanford.edu/pubs.html.

Creaney, Norman. 1996. An algorithm for generating quantifiers. In *Proceedings of the 8th International Workshop on Natural Language Generation*, Sussex, UK.

Creaney, Norman. 1999. Generating quantified logical forms from raw data. In *Proceedings of the ESSLLI-99 Workshop on the Generation of Nominal Expressions*.

Crystal, David. 1997. *A Dictionary of Linguistics and Phonetics*. Blackwell Publishers Ltd, Oxford.

Dalal, M., S. Feiner, K. McKeown, D. Jordan, B. Allen, and Y. alSafadi. 1996. MAGIC: An experimental system for generating multimedia briefings about post-bypass patient status. In *Proceedings 1996 AMIA Annual Fall Symposium*, pages 684–688, Washington, DC, October 26–30.

Dale, Robert. 1990. Generating recipes: An overview of Epicure. In Dale et al. (Dale, Mellish, and Zock, 1990), pages 229–255. Also appears as EUCCS Tech Report RP-37, Edinburgh.

Dale, Robert. 1992. *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes*. MIT Press, Cambridge, MA.

Dale, Robert, Eduard H. Hovy, Dietmar Rösner, and Oliviero Stock. 1992. *Aspects of Automated Natural Language Generation*. Lecture Notes in Artificial Intelligence, 587. Springer-Verlag, Berlin, April.

Dale, Robert, Chris Mellish, and Michael Zock, editors. 1990. *Current Research in Natural Language Generation*. Academic Press, New York.

Dale, Robert and Ehud Reiter. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19:233–263.

Dalianis, Hercules. 1996. *Concise Natural Language Generation from Formal Specifications*. Ph.D. thesis, Royal Institute of Technology, April.

Dalianis, Hercules. 1999. Aggregation in natural language generation. *Computational Intelligence*, 15(4):384–414.

Dalianis, Hercules and Eduard Hovy. 1993. Aggregation in natural language generation. In *Proceedings of the 4th European Workshop on Natural Language Generation*, Pisa, Italy.

Dalianis, Hercules and Eduard Hovy. 1996a. Aggregation in natural language generation. In Giovanni Adorni and Michael Zock, editors, *Trends in Natural Language Generation: An Artificial Intelligence Perspective*, Lecture Notes in Artificial Intelligence, 1036, pages 88–105, Berlin. Springer-Verlag.

Dalianis, Hercules and Eduard Hovy. 1996b. On lexical aggregation and ordering. In *Proceedings of the 8th European Workshop on Natural Language Generation, Demonstrations and Posters*, pages 29–32, Sussex, UK.

Davey, Anthony C. 1979. *Discourse Production*. Edinburgh University Press, Edinburgh.

de Swart, Henriette. 1998. *Introduction to Natural Language Semantics*. CSLI Publications.

Derr, Marcia A. and Kathleen R. McKeown. 1984. Using focus to generate complex and simple sentences. In *Proceedings of the Tenth International Conference on Computational Linguistics (COLING-84) and the 22nd Annual Meeting of the ACL*, pages 319–326, Stanford University, Stanford, CA.

Dik, Simon C. 1968. *Coordination: Its Implications for the Theory of General Linguistics*. North-Holland, Amsterdam.

Dixon, R. M. W. 1982. *Where Have All the Adjectives Gone?* Mouton Publishers, New York.

Dougherty, Ray C. 1970. A grammar of coordinate conjoined structures, part i. *Language*, 46:850–898.

Elhadad, Michael. 1990. Constraint-based text generation: Using local constraints and argumentation to generate a turn in conversation. Technical Report CUCS-003-90, Columbia University.

Elhadad, Michael, Kathleen McKeown, and Jacques Robin. 1997. Floating constraints in lexical choice. *Computational Linguistics*, 23(2):195–239, June.

Elhadad, Michael and Jacques Robin. 1997. SURGE: A comprehensive plug-in syntactic realisation component for text generation. Technical report, Department of Computer Science, Ben-Gurion University, Beer Sheva, Israel.

Fehrer, Detlef and Helmut Horacek. 1997. Exploiting the addressee's inferential capabilities in presenting mathematical proofs. In *Proceedings of the 15th IJCAI*, pages 965–970, Nagoya, Japan.

Frawley, William. 1992. *Linguistic Semantics.* Lawrence Erlbaum Associates, Hillsdale, NJ.

Gabriel, Richard P. 1988. Deliberate writing. In *Natural Language Generation Systems.* Springer-Verlag, New York, NY, pages 1–46.

Gailly, Pierre-Joseph. 1988. Expressing quantifier scope in French generation. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING-88)*, volume 1, pages 182–184, Budapest, August 22-27,.

Gazdar, Gerald. 1981. Unbounded dependencies and coordinated structure. *Linguistic Inquiry*, 12:155–182.

Gazdar, Gerald, Ewan Klein, Geoffrey Pullum, and Ivan Sag. 1985. *Generalized Phrase Structure Grammar.* Harvard University Press.

Giarratano, Joseph and Gary Riley. 1998. *Expert Systems: Principles and Programming.* PWS Publishing.

Gleitman, Lila R. 1965. Coordinating conjunctions in English. *Language*, 41:260–293.

Goldman, Neil M. 1975. Conceptual generation. In Roger C. Schank and Christopher K. Riesbeck, editors, *Conceptual Information Processing.* American Elsevier, New York, NY.

Goodall, Grant. 1987. *Parallel Structures in Syntax.* Cambridge University Press, Cambridge.

Goyvaerts, D. L. 1968. An introductory study on the ordering of a string of adjectives in present-day English. *Philologica Pragensia*, 11:12–28.

Graham, Ronald L., Donald E. Knuth, and Oren Patashnik. 1991. *Concrete Mathematics.* Addison-Wesley Publishing Company, Reading, MA.

Grice, H. Paul. 1975. Logic and conversation. In P. Cole and J. L. Morgan, editors, *Syntax and Semantics*, volume 3: Speech Acts. Academic Press, New York, pages 41–58.

Grimes, Joseph Evans. 1975. *The Thread of Discourse*. The Hague: Mouton.

Grishman, Ralph, Catherine Macleod, and Adam Meyers. 1994. COMLEX syntax: Building a computational lexicon. In *Proceedings of COLING '94*, Kyoto, Japan.

Grosz, Barbara J., Douglas E. Appelt, Paul A. Martin, and Fernando C. N. Pereira. 1987. TEAM: An experiment in the design of transportable natural-language interfaces. *Artificial Intelligence*, 32(2):173–243, May.

Hacker, Diana. 1994. *The Bedford Handbook for Writers*. Bedford Books, Boston, 4th edition.

Halliday, Michael A. K. 1994. *An Introduction to Functional Grammar*. Edward Arnold, London, 2nd edition.

Halliday, Michael A. K. and R. Hasan. 1976. *Cohesion in English*. Longman, London.

Harvey, Terrence and Sandra Carberry. 1998. Integrating text plans for conciseness and coherence. In *Proceedings of the 17th COLING and the 36th Annual Meeting of the ACL.*, pages 512–518.

Hatzivassiloglou, Vasileios and Kathleen McKeown. 1993. Towards the automatic identification of adjectival scales: Clustering adjectives according to meaning. In *Proceedings of the 31st Annual Meeting of the ACL*, pages 172–182.

Hatzivassiloglou, Vasileios and Kathleen McKeown. 1995. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the ACL*, pages 174–181, Madrid, Spain, July. Association for Computational Linguistics.

Hobbs, Jerry and Stuart Shieber. 1987. An algorithm for generating quantifier scopings. *Computational Linguistics*, 13(1-2):47–63, January-June.

Holland-Minkley, Amanda M., Regina Barzilay, and Robert Constable. 1999. Verbalization of high-level formal proofs. In *Proceedings of AAAI*, Orlando, Florida.

Horacek, Helmut. 1990. The architecture of a generation component in a complete natural language dialog system. In Dale et al. (Dale, Mellish, and Zock, 1990), pages 193–227.

Horacek, Helmut. 1992. An integrated view of text planning. In *Aspects of Automated Natural Language Generation* (Dale et al., 1992), pages 29–44.

Horacek, Helmut. 1997. An algorithm for generating referential descriptions with flexible interfaces. In *Proceedings of the 35th ACL and 8th EACL*, pages 206–213.

Hovy, Eduard H. 1988. *Generating Natural Language under Pragmatic Constraints.* Lawrence Erlbaum Associates, Hillsdale, NJ.

Hovy, Eduard H. 1990a. Parsimonious and profligate approaches to the question of discourse structure relations. In *Proceedings of the Fifth International Natural Language Generation Workshop*, Dawson, PA.

Hovy, Eduard H. 1990b. Unresolved issues in paragraph planning. In Robert Dale, Chris Mellish, and Michael Zock, editors, *Current Research in Natural Language Generation.* Academic Press, New York, pages 17–45.

Hovy, Eduard H. 1993. Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63. Special Issue on NLP.

Huang, Xiaorong and Armin Fiedler. 1996. Paraphrasing and aggregating argumentative text using text structure. In *Proceedings of the 8th International Natural Language Generation Workshop*, pages 21–3, Sussex, UK.

Huang, Xiaorong and Armin Fiedler. 1997. Proof verbalization as an application of NLG. In *Proceedings of the 15th IJCAI*, pages 965–970, Nagoya, Japan.

Hudson, Richard A. 1976. *Arguments for a Non-Transformational Grammar.* University of Chicago Press.

Inui, Kentaro, Takenobu Tokunaga, and Hozumi Tanaka. 1992. Text revision: A model and its implementation. In *Aspects of Automated Natural Language Generation* (Dale et al., 1992), pages 215–230.

Jackendoff, Ray. 1985. *Semantics and Cognition.* MIT Press, Cambridge, MA.

Jackendoff, Ray. 1990. *Semantic Structures.* MIT Press, Cambridge, MA.

Jacobs, Paul S. and Lisa F. Rau. 1990. SCISOR: Extracting information from on-line news. *Communications of the ACM*, 33(11):88–97.

Jing, Hongyan and Kathleen McKeown. 1998. Combining multiple,large-scale resources in a reusable lexicon for natural language generation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, pages 607–613, Quebec, Canada.

Joanis, Eric J. S. 1999. Review of the literature on aggregation in natural language generation. Technical Report CSRG-398, Department of Computer Science, University of Toronto, Canada, September.

Johnson, Mark, editor. 1988. *Attribute-value Logic and the Theory of Grammar*. CSLI, Stanford, CA.

Jordan, Desmond A., Kathleen R. McKeown, Kristian J. Concepcion, Steven K. Feiner, and Vasileios Hatzivassiloglou. 2001. Generation and evaluation of intraoperative inferences for automated health care briefings on patient status after bypass surgery. *Journal of the American Medical Informatics Association*, 8:267–280.

Jorgensen, Hanne and Anne Abeille. 1992. Coordination of 'unlike' categories in TAG. *Proceedings of the 2nd TAG Workshop.*

Joshi, Aravind K. 1986. An introduction to tree adjoining grammars. Technical Report MS-CIS-86-64, Department of Computer and Information Science, University of Pennsylvania.

Joshi, Aravind K. and Yves Schabes. 1990. Fixed and flexible phrase structure: Coordination in tree adjoining grammars. *Language Research*, 26(4).

Justeson, John and Slava Katz. 1991. Co-occurrences of antonymous adjectives and their contexts. *Computational Linguistics*, 17(1).

Kamp, J.A.W. 1975. Two theories of adjectives. In E. L. Keenan, editor, *Formal Semantics of Natural Language*. Cambridge University Press, Cambridge.

Kaplan, Ronald and Joan Bresnan. 1983. Lexical-functional grammar. In Joan Bresnan, editor, *The mental representation of grammatical relations*. MIT Press, Cambridge, MA.

Kaplan, Ronald M. and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*. MIT Press, chapter 4.

Kaplan, Ronald M. and John T. Maxwell III. 1988. Constituent coordination in Lexical-Functional Grammar. In *Proceedings of COLING-88*, pages 303–305, Budapest.

Kasper, R. T. and W. C. Rounds. 1986. A logical semantics for feature structure. In *Proceedings of the 24th ACL*, pages 257–266, New York.

Kay, Martin. 1979. Functional grammar. In *Proceedings of the 5th Annual Meeting of the Berkeley Linguistic Society*, pages 142–158, Berkeley, CA, February 17-19,.

Kay, Martin. 1984. Functional Unification Grammar: A formalism for machine translation. In *Proceedings of the 10th COLING and 22nd ACL*, pages 75–78.

Kay, Martin, Jean Mark Gawron, and Peter Norvig. 1994. *Verbmobil: A Translation System for Face-to-Face Dialog.* CSLI.

Kehler, Andrew, Mary Dalrymple, John Lampring, and Vijay Saraswat. 1999. Resource sharing in glue language semantics. In Mary Dalrymple, editor, *Semantics and Syntax in Lexical Functional Grammar*. MIT Press, Massachusetts, pages 191–208.

Kempen, Gerard, editor. 1987. *Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics.* NATO ASI Series – 135. Martinus Nijhoff Publishers, Boston.

Kendall, Maurice G. 1938. A new measure of rank correlation. *Biometrika*, **30**(1–2):81–93, June.

Knight, Kevin and Vasileios Hatzivassiloglou. 1995. Two-level, many-paths generation. In *Proceedings of the 33th Annual Meeting of the ACL*, pages 252–260, Massachusetts Institute of Technology, Cambridge, MA.

Knight, Kevin and Steve K. Luk. 1994. Building a large-scale knowledge base for machine translation. In *Proceedings of the AAAI*.

Knott, Alistair. 1996. *A Data-Driven Methodology for Motivating a Set of Coherence Relations*. Ph.D. thesis, University of Edinburgh.

Knott, Alistair, Jon Oberlander, Michael O'Donnell, and Chris Mellish. 2000. Beyond elaboration: the interaction of relations and focus in coherent text. In T. Sanders, J. Schilperoord, and W. Spooren, editors, *Text representation: linguistic and psycholinguistic aspects*. Benjamins. in press.

Knuth, Donald E. 1973. *Fundamental Algorithms*, volume 1 of *The Art of Computer Programming*. Addison-Wesley, Reading, Massachusetts, 2nd edition.

Kukich, Karen. 1983. Design of a knowledge-based report generator. In *Proceedings of the 21st Annual Meeting of the ACL*, pages 145–150, Cambridge, MA, June 15-17,.

Kukich, Karen, Kathleen McKeown, James Shaw, Jacques Robin, J. Lim, N. Morgan, and J. Phillips. 1994. User-needs analysis and design methodology for an automated document generator. In A. Zampolli, N. Calzolari, and M. Palmer, editors, *Linguistica Computazionale, Vol. IX-X*. Kluwer Academic Publishers, Norwell, MA, pages 109–115.

Lakoff, George and Stanley Peters. 1969. Phrasal conjunction and symmetric predicates. In David Reibel and Sanford Schane, editors, *Modern Studies in English*. Prentice Hall, Englewood Cliffs, pages 113–142.

Lakoff, Robin. 1971. If's, And's and But's about conjunction. In C. J. Fillmore and D. T. Langendoen, editors, *Studies in Linguistic Semantics*. Hot, Rinehart and Winston, New York, pages 114–149.

Langkilde, Irene and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 17th COLING and the 36th Annual Meeting of the ACL.*, pages 704–710.

Lavoie, Benoit and Owen Rambow. 1997. A fast and portable realizer for text generation. In *Proceedings of the 5th ACL Conference on ANLP*, pages 265–268, Washington, D.C.

Lester, James and Bruce Porter. 1997. Developing and empirically evaluating robust explanation generators: The KNIGHT experiments. *Computational Linguistics*, 23(1):65–101.

Levin, Beth. 1993. *English Verb Classes and Alternations*. The University of Chicago Press, Chicago.

Longacre, Robert E. 1983. *The Grammar of Discourse: Notional and Surface Structures*. Plenum Press, New York.

Luhn, H. P. 1958. The automatic creation of literature abstracts. *IBM Journal of Research Development*, 2:159–165.

Malkiel, Yakov. 1959. Studies in irreversible binomials. *Lingua*, 8(2):113–160.

Malouf, Robert. 2000. The order of prenominal adjectives in natural language generation. In *Proceedings of the 38th ACL*, Hong Kong.

Mani, Inderjeet and Mark T. Maybury, editors. 1999. *Advances in Automatic Text Summarization*. MIT Press, Cambridge, MA.

Mann, William C. 1984. Discourse structures for text generation. In *Proceedings of the Tenth International Conference on Computational Linguistics (COLING-84) and the 22nd Annual Meeting of the ACL*, pages 367–375, Stanford University, Stanford, CA. Also appears as USC/Information Sciences Institute Technical Report RR-84-127.

Mann, William C., Madeleine Bates, Barbara J. Grosz, David D. McDonald, Kathleen R. McKeown, and William R. Swartout. 1982. Text generation. *American Journal of Computational Linguistics*, 8(2):62–69, April-June. Also appears as ISI Tech Report RR-81-101 "Text Generation: The State of the Art and the Literature" December 1981.

Mann, William C. and James A. Moore. 1980. Computer as author – results and prospects. Technical Report RR-79-82, USC Information Science Institute, Marina del Rey, CA.

Mann, William C. and James A. Moore. 1981. Computer generation of multi-paragraph English text. *American Journal of Computational Linguistics*, 7(1):17–29.

Mann, William C. and Sandra A. Thompson. 1987. Rhetorical structure theory: Description and construction of text structures. In Kempen (Kempen, 1987), pages 85–96. Also appears as USC/Information Sciences Institute Tech Report RS-86-174, October 1986.

Mann, William C. and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281. Also available as USC/Information Sciences Institute Research Report RR-87-190.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19:313–330.

Martin, J. E. 1970. Adjective order and juncture. *Journal of verbal learning and verbal behavior*, 9:379–384.

Matthiessen, Christian and John A. Bateman. 1991. *Text Generation and Systemic-Functional Linguistics: Experiences from English and Japanese.* Francis Pinter Publishers, London.

Matthiessen, Christian and Sandra A. Thompson. 1988. The structure of discourse and 'subordination'. In John Halman and Sandra A. Thompson, editors, *Clause Combining in Grammar and Discourse.* John Benjamins Publishing Co., Amsterdam, pages 275–329.

McCawley, James D. 1981. *Everything that linguists have always wanted to know about logic (but were ashamed to ask).* University of Chicago Press.

McCawley, James D. 1988. *The Syntactic Phenomena of English.* University of Chicago Press.

McDonald, David D. 1983a. Description directed control: Its implications for natural language generation. *Computers and Mathematics with Applications*, 9(1):111–129.

McDonald, David D. 1983b. Natural language generation as a computational problem: An introduction. In *Computational Models of Discourse.* MIT Press, Cambridge, MA, pages 209–266.

McDonald, David D. 1984. Description directed control: Its implications for natural language generation. In Nick Cercone, editor, *Computational Linguistics.* Pergamon Press, London, pages 111–130.

McDonald, David D. 1987. Natural language generation: Complexities and techniques. In Sergei Nirenburg, editor, *Machine Translation: Theoretical and*

Methodological Issues. Cambridge University Press, Cambridge, chapter 12, pages 192–224.

McDonald, David D. 1992. Natural language generation. In Stuart C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*. John Wiley and Sons, New York, 2nd edition, pages 983–997.

McDonald, David D., Marie M. Meteer, and James D. Pustejovsky. 1987. Factors contributing to efficiency in natural language generation. In Kempen (Kempen, 1987), pages 159–182.

McKeown, Kathleen, Karen Kukich, and James Shaw. 1994. Practical issues in automatic documentation generation. In *Proceedings of the 4th ACL Conference on Applied Natural Language Processing*, pages 7–14, Stuttgart.

McKeown, Kathleen, Shimei Pan, James Shaw, Desmond Jordan, and Barry Allen. 1997. Language generation for multimedia healthcare briefings. In *Proceedings of the Fifth ACL Conference on ANLP*, pages 277–282.

McKeown, Kathleen R. 1985. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press, Cambridge.

McKeown, Kathy, Jacques Robin, and Karen Kukich. 1995. Generating concise natural language summaries. *Information Processing and Management*, 31(5).

Mellish, Chris. 1988. Implementing systemic classification by unification. *Computational Linguistics*, 14(1):40–51, Winter.

Mel'čuk, Igor and Alain Polguère. 1987. A formal lexica in meaning-text theory (or how to do lexica with words). *Computational Linguistics*, 13:276–289.

Meteer, Marie. 1991a. Bridging the generation gap between text planning and linguistic realization. *Computational Intelligence*, 7(4):296–304, November.

Meteer, Marie. 1991b. The implications of revisions for natural language generation. In Cécile L. Paris, William R. Swartout, and William C. Mann, editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*. Kluwer Academic Publishers, Boston, pages 155–178.

Meteer, Marie. 1993. *Expressibility and the Problem of Efficient Text Planning.* Francis Pinter Publishers, London.

Meteer, Marie M. and David D. McDonald. 1986. The writing process as a model for natural language generation. In *Proceedings of the 24th Annual Meeting of the ACL*, pages 90–96, Columbia University, New York, June 10-13,.

Miller, George, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1990. Five papers on WordNet. CSL Report 43, Cognitive Science Laboratory, Princeton University.

Moore, Johanna D. and Cécile L. Paris. 1989. Planning text for advisory dialogues. In *Proceedings of the 27th Annual Meeting of the ACL*, pages 203–211, University of British Columbia, Vancouver, BC, June 26-29,.

Moran, Douglas B. and Fernando C. N. Pereira. 1992. Quantifier scoping. In Hiyan Alshawi, editor, *The Core Language Engine*. MIT Press, Cambridge, MA, pages 149–172.

Moser, Megan and Johanna D. Moore. 1995. Investigating cue selection and placement in tutorial discourse. In *Proceedings of the 33rd ACL*, pages 130–135, Boston, MA.

Neijt, Anneke H. 1979. *Gapping: a contribution to Sentence Grammar.* Dordrecht: Foris Publications.

Nogier, J. and Michael Zock. 1991. Lexical choice as pattern matching. In T. Nagle, J. Nagle, L. Gerholz, and P. Elklund, editors, *Current directions in conceptual structures research.* Springer-Verlag, New York, NY.

Oberlander, Jon and Johanna D. Moore. 1999. Cue phrases in discourse: further evidence for the core:contributor distinction. In *Proceedings of the Workshop on Levels of Representation in Discourse*, Edinburgh, Scotland.

Paice, C. D. 1990. Constructing literature abstracts by computer: Techniques and prospects. *Information Processing and Management*, 26:171–186.

Pan, Shimei. 2000. *Automatic Prosody Modeling in Concept-to-Speech Generation.* Ph.D. thesis, Columbia University. To appear.

Pan, Shimei and Julia Hirschberg. 2000. Modeling local context for pitch accent prediction. In *Proceedings of the 38th Annual Meeting of the Association of Computational Linguistics*, Hong Kong.

Pan, Shimei and Kathleen McKeown. 1997. Integrating language generation with speech synthesis in a Concept-to-Speech system. In *Proceedings of ACL/EACL'97 Concept to Speech Workshop*, Madrid, Spain.

Pan, Shimei and Kathleen McKeown. 1998. Learning intonation rules for concept to speech generation. In *Proceedings of COLING/ACL'98*, Montreal, Canada.

Pan, Shimei and Kathleen McKeown. 1999. Word informativeness and automatic pitch accent modeling. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 148–157.

Panaget, Frank. 1994. Using a textual representation level component in the context of discourse and dialogue generation. In *Proceedings of the Seventh International Workshop on Natural Language Generation*, pages 127–136, Nonantum Inn, Kennebunkport, Maine.

Park, Jong C. 1995. Quantifier scope and constituency. In *Proceedings of the 33rd ACL*, pages 205–212.

Partee, B. H. 1970. Negation, conjunction and quantifiers: Syntax vs. semantics. *Foundations of Language*, 6:153–165.

Passonneau, Rebecca, Karen Kukich, Vasileios Hatzivassiloglou, Larry Lefkowitz, and Hongyan Jing. 1996. Generating summaries of work flow diagrams. In *Proceedings of the International Conference on Natural Language Processing and Industrial Applications*, pages 204–210, New Brunswick, Canada. University of Moncton.

Penberthy, J. S. and D. Weld. 1992. UCPOP: A sound, complete, partial-order planner for ADL. In *Third International Conference on Knowledge Representation and Reasoning (KR-92)*, Cambridge.

Pereira, Fernando C. N. 1990. Categorial semantics and scoping. *Computational Linguistics*, 16(1):1–10.

Pereira, Fernando C. N. and Michael D. Riley. 1997. Speech recognition by composition of weighted finite automata. In Emmanuel Roche and Yves Schabes, editors, *Finite-State Language Processing*. MIT Press, Cambridge, MA, pages 431–453.

Pollard, Carl and Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

Prescod, Paul and Charles F. Goldfarb. 1999. *The XML Handbook*. Prentice Hall, Englewood Cliffs, 2nd edition.

Pustejovsky, James D. 1991. The generative lexicon. *Computational Linguistics*, 17(4):409–441, December.

Pustejovsky, James D. 1995. *The Generative Lexicon*. MIT Press, Cambridge, MA.

Quirk, Randolph and Sidney Greenbaum. 1973. *A Concise Grammar of Contemporary English*. Harcourt Brace Jovanovich, Inc., London.

Quirk, Randolph, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman Publishers, London.

Radford, Andrew. 1988. *Transformational Grammar*. Cambridge University Press, Cambridge.

Rambow, Owen and Tanya Korelsky. 1992. Applied text generation. In *Proceedings of the Third ACL Conference on Applied Natural Language Processing*, pages 40–47, Trento, Italy.

Rath, G. J., A. Resnick, and T. R. Savage. 1961. The formation of abstracts by the selection of sentences. *Information Processing and Management*, 26(1):139–141.

Reape, Mike and Chris Mellish. 1999. Just what is aggregation anyway? In *Proceedings of the 7th European Workshop on Natural Language Generation*, Toulouse, France.

Reinhart, Tanya. 1983. *Anaphora and Semantic Interpretation*. University of Chicago Press, Chicago.

Reiter, Ehud. 1990. A new model for lexical choice for open-class words. In *Proceedings of the Fifth International Natural Language Generation Workshop*, pages 23–30, Dawson, PA.

Reiter, Ehud. 1994. Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible? In *Proceedings of the Seventh International Workshop on Natural Language Generation*, pages 163–170, Nonantum Inn, Kennebunkport, Maine.

Reiter, Ehud and Robert Dale. 1992. A fast algorithm for the generation of referring expressions. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pages 232–238, Nantes, France.

Reiter, Ehud and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge.

Reynar, Jeffrey C. and Adwait Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the 5th Conference on Applied Natural Language Processing*.

Robin, Jacques. 1990. Lexical choice in natural language generation. Technical Report CUCS-040-90, Department of Computer Science, Columbia University, New York.

Robin, Jacques. 1994. Automatic generation and revision of natural language summaries providing historical background. In *Proceedings of the 11th Brazilian Symposium on Artificial Intelligence (SBIA'94)*, Fortaleza, CE, Brazil.

Robin, Jacques. 1995. *Revision-Based Generation of Natural Language Summaries Providing Historical Background*. Ph.D. thesis, Columbia University.

Robin, Jacques. 1996. Evaluating the portability of revision rules for incremental summary generation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, Santa Cruz, CA.

Rösner, Dietmar and Manfred Stede. 1992. Customizing RST for the automatic production of technical manuals. In *Aspects of Automated Natural Language Generation* (Dale et al., 1992), pages 199–214.

Ross, John Robert. 1967. *Constraints on variables in syntax*. Ph.D. thesis, MIT.

Ross, John Robert. 1970. Gapping and the order of constituents. In Manfred Bierwisch and Karl Heidolph, editors, *Progress in Linguistics*. Mouton, The Hague, pages 249–259.

Rubinoff, Robert. 1992. Integrating text planning and linguistic choice. In *Aspects of Automated Natural Language Generation* (Dale et al., 1992), pages 45–56.

Sag, Ivan A. 1976. *Deletion and Logical Form*. Ph.D. thesis, MIT.

Sag, Ivan A. and Janet D. Fodor. 1994. Extraction without traces. In *Proceedings of the 13th Annual Meeting of the West Coast Conference on Formal Linguistics*, Stanford. CSLI Publications.

Sag, Ivan A., Gerald Gazdar, Thomas Wasow, and Steven Weisler. 1985. Coordination and how to distinguish categories. *Natural language and linguistic theory*, 3:117–171.

Sag, Ivan A. and Thomas Wasow. 1999. *Syntactic Theory: A Formal Introduction*. CSLI.

Sarkar, Anoop. 1997. Separating dependency from constituency in a tree rewriting system. In *Proceedings of the Fifth Meeting on Mathematics of Language*, Saarbruecken.

Sarkar, Anoop and Aravind Joshi. 1996. Coordination in tag: Formalization and implementation. In *Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen.

Scott, Donia R. and Clarisse S. de Souza. 1990. Getting the message across in RST-based text generation. In Robert Dale, Chris Mellish, and Michael Zock, editors, *Current Research in Natural Language Generation*. Academic Press, New York, pages 47–73.

Shaw, James. 1995. Conciseness through aggregation in text generation. In *Proceedings of the 33rd ACL (Student Session)*, pages 329–331.

Shaw, James. 1998a. Clause aggregation using linguistic knowledge. In *Proceedings of the 9th International Workshop on Natural Language Generation.*, pages 138–147.

Shaw, James. 1998b. Segregatory coordination and ellipsis in text generation. In *Proceedings of the 17th COLING and the 36th Annual Meeting of the ACL.*, pages 1220–1226.

Shaw, James and Vasileios Hatzivassiloglou. 1999. Ordering among premodifiers. In *Proceedings of the 37th Annual Meeting of the Assoc. of Computational Linguistics*, pages 135–143.

Shaw, James and Kathleen McKeown. 2000. Generating referring quantified expressions. In *Proceedings of the International Natural Language Generation Conference*, pages 100–107, Mitzpe Ramon, Israel.

Shieber, Stuart M. 1986. *An Introduction to Unification-Based Approaches to Grammar*. CSLI.

Shieber, Stuart M., Gertjan van Noord, Fernando C. N. Pereira, and Robert C. Moore. 1990. Semantic-head-driven generation. *Computational Linguistics*, 16(1):30–42, March.

Shortliffe, E. H. and B. Buchanan. 1984. *Rule Based Expert Systems: the MYCIN experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, Reading, MA.

Smadja, Frank A. and Kathleen R. McKeown. 1991. Using collocations for language generation. *Computational Intelligence*, 7(4):229–239, November.

Smith, Carlota S. 1969. Ambiguous sentences with *And*. In David Reibel and Sanford Schane, editors, *Modern Studies in English*. Prentice Hall, Englewood Cliffs, pages 75–79.

Späth, Helmuth. 1985. *Cluster Dissection and Analysis: Theory, FORTRAN Programs, Examples*. Ellis Horwood, Chichester, West Sussex, England.

Stede, Manfred. 1995. Lexicalization in natural language generation: a survey. *Artificial Intelligence Review*, 8:309–336.

Stede, Manfred. 1996. Lexical options in multilingual generation from a knowledge base. In Giovanni Adorni and Michael Zock, editors, *Trends in Natural Language Generation: An Artificial Intelligence Perspective*. Springer, Berlin, pages 222–237.

Steedman, Mark. 1985. Dependency and coordination in the grammar of Dutch and English. *Language*, 61:523–568.

Steedman, Mark. 1990. Gapping as constituent coordination. *Linguistics and Philosophy*, 13:207–263, April.

Steedman, Mark. 2000. *The syntactic process*. MIT Press, Cambridge, MA.

Tai, James Hau-Y. 1969. *Coordination Reduction*. Ph.D. thesis, Indiana University.

Teyssier, J. 1968. Notes on the syntax of the adjective in modern English. *Behavioral Science*, 20:225–249.

Thompson, Henry S. 1977. Strategy and tactics: A model for language production. In W. A. Beach, S. E. Fox, and S. Philosoph, editors, *Papers from the 13th Regional Meeting of the Chicago Linguistics Society*, pages 651–668, Chicago, IL, April 14-16,.

van Eijck, Jan and Hiyan Alshawi. 1992. Logical forms. In Hiyan Alshawi, editor, *The Core Language Engine*. MIT Press, Cambridge, MA, pages 11–38.

van Oirsouw, Robert. 1987. *The Syntax of Coordination*. Croom Helm, Beckenham.

Vander Linden, Keith and James H. Martin. 1995. Expressing rhetorical relations in instructional text: A case study of the purpose relation. *Computational Linguistics*, 21(1):29–57.

Vendler, Zeno. 1967. Each and every, any and all. In *Linguistics in Philosophy*. Cornell University Press, Ithaca and London, pages 70–96.

Vendler, Zeno. 1968. *Adjectives and Nominalizations*. Mouton and Co., The Netherlands.

Wanner, Leo. 1994. Building another bridge over the generation gap. In *Proceedings of the Seventh International Workshop on Natural Language Generation*, pages 137–144, Nonantum Inn, Kennebunkport, Maine.

Wanner, Leo and Eduard Hovy. 1996. The HealthDoc sentence planner. In *Proceedings of the 8th International Workshop on Natural Language Generation*, pages 1–10, Sussex, UK.

Whorf, Benjamin Lee. 1956. *Language, thought, and reality; selected writings*. Technology Press of MIT, Cambridge.

Wierzbicka, Anna. 1980. *Lingua Mentalis: The Semantics of Natural Language*. Academic Press.

Wilkinson, John. 1995. Aggregation in natural language generation: Another look. Co-op work term report, Department of Computer Science, University of Waterloo, September.

Woods, William A. 1978. Semantics and quantification in natural language question answering. In *Advances in Computers*, volume 17. Academic Press, pages 1–87.

Yazdani, Masoud. 1987. Reviewing as a component of the text generation process. In Kempen (Kempen, 1987), pages 183–190.

Zwarts, Frans. 1983. Determiners: a relational perspective. In A. ter Meulen, editor, *Studies in model-theoretic semantics*. Dordrecht: Foris, pages 37–62.