

USING ARGUMENTATION IN TEXT GENERATION

Michael Elhadad
Department of Computer Science
Ben Gurion University of the Negev
Beer Sheva, 84105 Israel
elhadad@bengus.bgu.ac.il

1. Introduction

Text generation is a field of artificial intelligence aiming at modelling the process of natural language production. Text generation is best characterized as the process of making choices between alternate linguistic realizations under the constraints specified in the input to a text generator. Depending on the practical application, the input can take different forms - streams of numbers in report generation, traces of rule activations in expert system explanation, knowledge-based descriptions in data-base interfaces or in text translation. In a strong sense, the application determines what knowledge is available to drive the generation process. In the opposite direction, however, the analysis of quasi-paraphrases of a linguistic constituent can also determine what input must be present to explain the distinction between alternate linguistic realizations. In that sense, the linguistic system determines what knowledge must be made available to drive the generation process.

The problem faced by the designer of a text generation system is, therefore, to match the information provided by an application with the requests of a linguistic component. In this paper, we show how the linguistic theory of argumentation of Anscombes and Ducrot [1] can help to bridge this gap, in identifying a middle ground between the knowledge necessary to distinguish a whole class of linguistic phenomena and the type of knowledge used by applications. This bridge takes the form of an abstract descriptive tool - the *topos*. Topoi are gradual inference rules that capture “common sense” relations. For example, *the more an activity is interesting, the more people want to do it*. Topoi are used in D&A’s theory of argumentation to explain the semantics of connectives like “*but*” or of argumentative operators like “*however*” or “*at least*” [1, 9]. Recently, topoi have also been used in the description of lexical items like verbs and adjectives [4, 3]. In this paper, we also show that topoi are sufficiently abstract to serve as a convenient bridge between the type of knowledge representation used in knowledge-based computer applications and a linguistically motivated text generator.

Thus the main advantages of using the theory of argumentation in language for text generation can be summarized by two points:

- argumentation accounts for linguistic choices like gradual adjectives, contrastive connectives or judgment determiners. These phenomena pose problems to most semantic theories, and are often handled in an *ad-hoc* manner by generation systems.
- argumentation establishes a bridge between the knowledge provided by an application and the needs of linguistic realization.

The ADVISOR II text generation system illustrates how the handling of argumentative features throughout the generation process, from content determination to linguistic realization, helps control linguistic decisions that were not considered in previous work in text generation. Specifically, we give examples of the use of argumentative constraints to perform the following tasks:

- Content determination
- Content organization
- Lexical choice

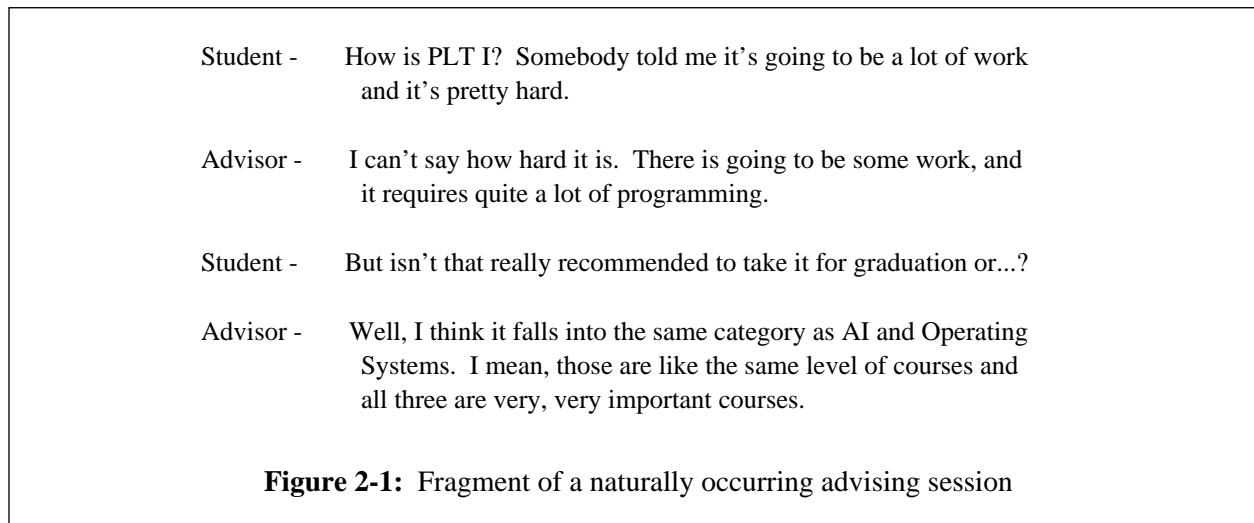
The main points of the paper are that argumentation is necessary to perform a wide range of lexical choice operations and that the argumentative information needed to perform these operations can be derived naturally from the generation process prior to lexicalization by using a single conceptual tool, the topoi, at *all the levels* of the generation process.

In the rest of the paper, we proceed as follows: we first enumerate the different subtasks a text gener-

ator must fulfill, and after briefly reviewing the theory of argumentation, we discuss how argumentation can be combined with and complement existing generation techniques for each subtask up to lexicalization. Finally, we explain how the argumentative information included in the input to the lexical choice module is used to select classes of words that were not considered in generation prior to this work.

2. The Domain Problem: Generating Advice-Giving Paragraphs

Consider the task of advising a university student about the courses he should follow. A system providing this type of advice has to interact with a user and provide appropriate evaluations of the courses about which the student may inquire. A typical interaction between a student and a real academic advisor is shown in Figure 2-1.¹



We have developed an explanation component for a system called ADVISOR II which models this task. We focus on one of the subtasks of such an advising system: how to provide an evaluation of a course and communicate it to the hearer. The paragraph shown in Fig.2-2 illustrates the capabilities of our system. The answer is tailored to a particular student, assuming that the student has expressed interest in Natural Language Processing (NLP) and programming. Note that the answer uses judgment determiners (*most* and *many*) and scalar adjectives (*interesting*, *difficult*). These lexical decisions contribute to the natural and fluent character of the answer.

In order to produce such a paragraph, many operations need to be performed: relevant information about the course and the student must be gathered, an evaluation of the course for the particular student must be computed, the information supporting that evaluation must be organized and finally the evaluation must be converted into words and linguistic constructions. The architecture of our generator is shown in Fig.2-3. In this figure, active processes are shown in rectangle boxes and knowledge sources in shaded boxes. Edges indicate data flow. The figure highlights the central role of topoi in the system, as a bridge between conceptual and linguistic decisions.

Each module performs the following tasks:

- **Goal and content determination:** The system reasons about the interaction situation and determines what goals must be achieved by the answer. In the advising domain, the goal can be to inform the student, to recommend taking a course or against taking it. In ADVISOR II the goal of the answer is given as an input to the system. To support this goal, the system extracts relevant information from a knowledge base. In our example, the system decides to include the fact that AI covers a certain set of topics and requires a certain type of assignments. Information on the topics and assignments of classes is stored in the knowledge base. The

¹This example is an excerpt from transcripts of advising sessions between human academic advisors and students recorded at Columbia University in 1984.

Q: Should I take AI?

Answer 1:

AI has many programming assignments and it covers a lot of interesting topics, such as NLP, Vision and KR.

So it should be quite interesting.

I would recommend it.

Answer 2:

AI deals with many interesting topics, such as NLP, Vision and K. But it has many assignments which consist of writing papers.

You have little experience writing papers.

So it could be difficult.

I would not recommend it.

Figure 2-2: A paragraph generated by ADVISOR II

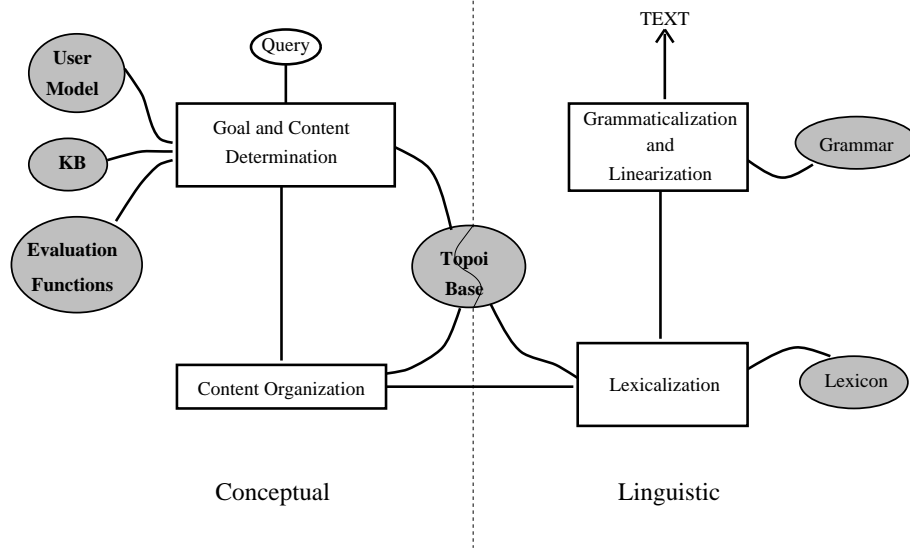


Figure 2-3: Architecture of a text generator

system must also perform some inferencing to conclude, for example, that NLP-related topics and programming assignments make up for an interesting class. The output of this stage is a collection of conceptual propositions, in a predicate-argument notation. In addition, in ADVISOR II, all propositions are annotated by their argumentative function, as explained below.

- **Content organization:** The system structures the information extracted at the previous stage into a network of rhetorical and/or logical relations. For example, the paragraph in Fig.2-2 contains three propositions structured as (premise conclusion), where premise itself is a conjunction of two propositions. The information extracted from the knowledge base is also filtered at this stage: for example, the answer does not mention the fact that AI covers many mathematical topics. This is also the time when an ordering is imposed on the information to be conveyed in the answer.

- **Lexical choice:** The conceptual structures produced by the content determination and organization modules are mapped onto linguistic structures, with heads and arguments. For example, the system chooses among structures like *AI assignments involve programming*, *programming homework in the AI domain* (a noun phrase) and *AI involves programming assignments*. In addition, lexical items (words) are selected. For example, the system chooses between *assignments involve programming* and *assignments require programming*, or between *homework* and *assignments*. The system also decides which determiners to use, *many* or *most* or unmarked ones (*the*). The output of this stage is a network of grammatical relations with lexical items appearing at each node.
- **Grammaticalization and linearization:** The system determines paragraph and sentence boundaries, the syntactic structure of the propositions (subordination and embedding), pronominalization of selected references and scoping relations between arguments, grammatical tense and modality selection, agreement and ordering constraints. The syntactic structure is linearized into a string of words. Morphology is handled at this stage.

To perform these tasks, the system accesses various knowledge sources. A knowledge base written in CLASSIC, a KL-ONE like knowledge representation system [40], describes courses offered in the semester, the user model is also described in CLASSIC and contains information such as which courses the student has already taken, what are his interests and experience. Specific to our approach are a set of topoi and a set of evaluation functions which are used to map the content of the knowledge base to situation specific utterances. These are described in detail in Sect.4.1. Finally, the linguistic components use a lexicon and a grammar described in the FUF formalism [11, 13 Chap.3-4]. The lexicon is described in Sect.6.

In the rest of the paper, we focus on the task of expressing an evaluation of a class: we first briefly review the linguistic theory of argumentation of [1]. We then present our model of evaluation, distinguishing between three types of evaluations and highlighting the role of topoi in the definition of the most complex type. We then describe how topoi are used to perform content determination and organization, and lexicalization, emphasizing their role as a bridge between conceptual and linguistic decisions.

3. Previous Work: A Brief Review of the Theory of Argumentation

This work makes heavy use of the notion of topoi, introduced in the linguistic theory of argumentation of [1]. It also uses a theory of lexical semantics where quasi-synonyms differ by their connotation. We first review Ducrot's theory of argumentation and then discuss how Osgood's work [35] on scalar lexical semantics can be related to this theory.

3.1. Argumentation in Language

The theory of argumentation in language itself is introduced in [1]. We also rely on developments presented in [38, 3]. The goal of this section is primarily to introduce the terminology that is used in the rest of the paper.

The linguistic theory of argumentation relies first on a distinction between *sentence* and *utterance*: a sentence is an abstract linguistic object while an utterance is the concrete event occurring when a speaker says a sentence in a certain situation. The theory of argumentation is concerned with sentences and not utterances. The point of the theory is that the semantics of a sentence includes indications about how it can be presented, in all situations, to support a set of conclusions, and that therefore, argumentation is not a completely pragmatic phenomenon, but also concerns the semantics of a language. We call *argumentative orientation* the specification in the semantics of a sentence of what conclusions can be supported by its enunciation.

In addition, the theory of argumentation assumes that the argumentative orientation of a sentence can be described in a restricted formalism, based on semantic rules called *topoi*.² Topoi are gradual inference rules of the form: "*the more/less X is P, the more/less Y is Q*," where *X* and *Y* are semantic elements present in the sentence, and *P* and *Q* are called *topic scales*. For example, *the more a class is difficult, the less a student wants to take it* is a topos used in our domain. In this paper, we view a topos as the combination of four primitive relations:

²Topos singular. Topoi plural.

- An evaluation of the entity X on the scale P (the class on the scale of difficulty).
- An evaluation of the entity Y on the scale Q (the student on the scale *desire to take the class*).
- The expression of a gradual relation between P and Q (the harder the class, the less desire).
- The expression of a topical relation between X and Y (the class on the left side is the class rejected on the right hand side).

The argumentative orientation of a sentence does not determine completely which topoi can be triggered by its enunciation. It only constrains some aspects of the description of a topos that can felicitously be applied to the utterance in a given situation. Consider for example the following sequences:

- *AI is hard. You should not take it.*
- *AI is hard. You should take it.*
- *AI is hard. You would enjoy the challenge.*

The first sentence sounds natural while the second one sounds at first surprising. One might be inclined to conclude that the clause “*AI is hard*” therefore triggers the activation of a topos of the form “*the more a class is difficult, the less a student wants to take it.*” This is not, however, a valid conclusion, because even if the second sentence sounds surprising, one might imagine situations where it makes sense. For example, situations where the student has expressed a taste for challenge, as illustrated by the third example; that is, situations where the topos “*the more something is difficult, the more person X wants to do it.*” While such situations may be rare, the fact is that the topoi that are triggered by the single sentence “*AI is hard*” are not completely determined by its linguistic form. But when more linguistic markers are available in a sentence, more constraints can be derived and the set of topoi which can be triggered becomes smaller. For example, if instead of considering just the first clause “*AI is hard,*” one considers the whole sequence “*AI is hard, you should not take it*” then only the topos “*the more a course is difficult, the less a student wants to take it*” (formally, $/+ \text{Difficult}(\text{Action}), - \text{Want}(\text{Agent}(\text{Action}), \text{Do}(\text{Action})) /$) can be used in all situations.

As for the single clause “*AI is hard,*” we can assume that any topos triggered by this clause must include an evaluation of an entity of type “*class*” on the scale of difficulty. That is, of the four primitive relations that constitute a topos, only one is constrained by the linguistic form of the clause. In general, the theory predicts that lexical items and syntactic constructions constrain which topoi can be triggered from the utterance of a sentence in a given situation. For example, the constraint above would be created by the selection of the adjective “*hard*” and its use in a predicative construction [12]. Other classes of words impose different types of constraints: for example, connectives like “*but*” or “*because*” impose constraints not on which scales are activated but on the orientation of the evaluation on the scales [9].

The claim of the theory of argumentation in language is that the set of conclusions that can be supported from the enunciation of the sentence “*AI is hard*” is constrained by the semantics of its components - namely the noun “*AI*”, the adjective “*hard*” and the predicative relation between them. From a generation perspective, the same claim can be phrased as: the argumentative orientation of a sentence partially determines what lexical items and syntactic relations can be used in the sentence.

We close that review by highlighting the function of the topoi in the theory: on one hand, topoi are part of the domain knowledge of speakers, on the other hand, topoi-related features are also included in the lexicon and in the grammar, for each lexical item and syntactic construction. In that sense, and that is our main claim in this paper, topoi can serve as a bridge between the conceptual representations used in an application and the linguistic demands of a text generator.

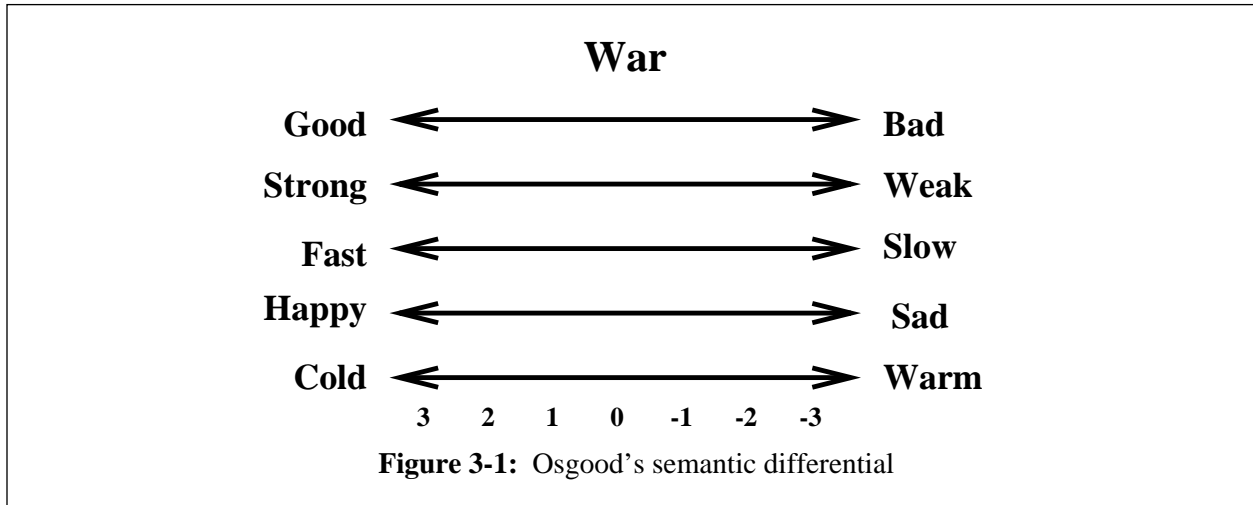
3.2. Scalar Lexical Semantics and Scalar Connotations

One of the claims of the linguistic theory of argumentation is that the semantics of certain lexical items is scalar. In a different perspective and with different goals, Osgood has also developed a theory of lexical semantics relying on the notion of scalar dimensions to meaning.

Consider for example the verbs *require*, *necessitate* and *demand* in a context like *AI requires many assignments*. All three verbs have a scalar connotation that projects an evaluation of their subject on the scale of difficulty. In contrast, the verbs *have*, *include* and *involve* lack this connotation. The notion of connotation that we use here is extensively reviewed in [27]. The underlying intuition is, when considering a group of words like $\{require, necessitate, demand, have, include\}$, to distinguish between a “core

meaning” which is shared by all elements, and “connotations” which capture the “added meaning” of each element. There is no logical relation between the core meaning of a word and its connotations. For example, the evaluation of AI as a difficult class when the verb “*require*” is used is independent of the denotation of “*require*” (which is the conceptual relation `assignments-of`) and not motivated by it (the relation `assignments-of` does not imply the evaluation of the class on the scale of difficulty).

Scalar connotations are the particular type of connotations that convey a scalar evaluation. In [35], a technique to recognize scalar connotations is described. The technique, known as the *semantic differential* relies on a statistical analysis of the answers of subjects to a test associating a lexical item with a set of bipolar scales. An example of such test is shown in Fig.3-1. The results of such an analysis provide a sort of “connotative profile” for lexical items.



One of the goals of Osgood's study was to identify the scales playing a role in explaining the “conceptual distance” between words, by using factor analysis to identify independent scales. The experiment started by identifying 50 scales by asking subjects to associate adjectives with a random sample of nouns. (In another version of the experiment, some 289 adjective pairs were identified from an analysis of Roget's thesaurus.) Then 100 subjects were asked to rank 20 nouns on these 50 scales, with a ranking from -3 to 3. A factor analysis of the results identified three major dimensions along which judgments were made: evaluative (good-bad), potency (strong-weak) and activity (fast-slow). A traditional criticism of the method is that the original set of scales is quite arbitrary and can therefore taint the final results.

In our case, we are interested in choosing lexical items which can be used to express a scalar evaluation. Lexical items carrying a scalar connotation are prime candidates for this task. For our purposes, however, we do not need to uncover a set of “universal scales” that could define the independent dimensions of some conceptual space. In contrast, we only need to identify the connotations of items on the domain-dependent argumentative scales that we have identified in our domain. So to uncover the argumentative connotations in our domain, we can use a much simplified form of the semantic differential to present grids of evaluation similar to the one in Fig.3-1, but only with the 7 scales of our domain presented in Sect.4.3 (applying the technique as advocated by its inventors themselves [35, p.76]).

Another difference of our use of the technique from the original Osgood test is that we test partial syntactic constructions as opposed to simple nouns. For example, our test question is: *when ‘X requires Y’, X is <scalar evaluations> and Y is <scalar evaluations>*. This notion of construction is related to the notion of “lexically open idioms” of [18]. The underlying principle is that argumentation is not only conveyed by single words but by constructions that are conventionally associated with some pragmatic effect.

In our system, scalar connotations are stored in the lexicon with each entry as discussed in Sect.6. Scalar connotations are expressed in terms of topoi, thus an important part of the lexical semantics in our system is expressed in argumentative terms. In our implementation, the connotations correspond to our intuition and have not been validated by a statistical test. The mechanisms for using the connotation information however would not be changed even if our intuition is not corroborated by statistical evidence. Our technique only relies on the assumption that some statistically significant scalar connotations can be identified.

4. The Role of Topoi in Content Determination

In the ADVISOR II domain, consider a student asking the system: “*Should I take AI this semester?*” The system must first determine what will be the goal of the answer. ADVISOR II takes as input one of the following goals:

- Recommend the course to the student
- Recommend against taking the course
- Inform the student about some properties of the course.

In this paper, we focus on the first two goals, which are argumentative in nature. The task of content determination is, given this argumentative intent as input, to gather relevant information for inclusion in the answer. The system has access to several sources of information: a knowledge base describing the courses offered in the university, information about the student, including which courses he or she has already taken, topics of interest, and experience. The eventual paragraphs produced by ADVISOR II, as shown for example in Fig.2-2, present only a subset of the information gathered at this stage. Content determination produces the larger set of propositions that are relevant to the evaluation of the class for a specific student.

The focus of this work is to study how evaluations can be expressed in language. In this section, we first present our model of what constitutes an evaluation in terms of topoi. We then show how the system produces evaluations by extracting information from the knowledge base and relating it to the argumentative intent of the answer it produces. The next sections discuss how these evaluations are mapped onto linguistic structures.

4.1. Three Types of Argumentative Evaluation

We have introduced topoi as simple gradual rules, and informally written their definition as in */+homework, +difficult/*. Such notation is an abbreviation for a more complex structure. Consider the topos: */the more a class has assignments, the more difficult the class/*. By definition, this topos is the composition of two gradual evaluations: on the left-hand side, an evaluation of the cardinal of the set of assignments, on the right-hand side, an evaluation of the class on the topical scale of difficulty.

An evaluation is an answer to a question of the form *how P is X?*. For example, the left-hand side evaluation is an answer to the question “*how many assignments are there in AI?*”. We distinguish between three possible types of answers to this question. Following [3], we distinguish between *simple* and *composite* topical evaluations. We also introduce a distinction between *simple absolute* and *simple relative* evaluations.

The first type of answer to the “*how many*” question is an exact answer: “*AI has 7 assignments*” which corresponds to a *simple absolute evaluation*. It consists in “measuring” an entity on a conceptual scale. For example, one can measure the number of assignments given in a class, or the number of topics a class covers. In practical terms, such measuring can be performed by looking up the knowledge base of the system. Using the same notation as [3], we note the result of such a measuring operation as: $\langle \text{CONCEPTUAL-SCALE, VALUE} \rangle$. For example: $\langle \text{cardinal}(\{\text{hw, assignments-of}(\text{AI, hw})\}), 7 \rangle$ stands for the simple absolute evaluation that AI has 7 assignments.

When the scale on which the evaluation is performed is not “measurable” in the knowledge base, then the value of a simple absolute evaluation is by convention either + or -. For example, the difficulty of a class is not a primitive notion in the knowledge base: it is not very meaningful to characterize a class as “difficulty level 5”. Instead, this evaluation must be derived by an argumentative inference in a specific situation (taking into account the goal of the speaker and information about the student). In such a case, we note the evaluation as: $\langle \text{difficulty}(\text{AI}), + \rangle$. Since such evaluations cannot be simply “measured”, they can be derived in two possible ways: they are either derived by argumentative inference or they correspond to an *a priori* bias of the answer. We discuss below how argumentative inference is modeled in our system.

Another type of answer to the question “*how many assignments are there in AI*” is to compare the number of AI assignments with some other value. We call such evaluations, not covered in the model of [3], *simple relative evaluations*, and note them as in: $\langle \text{cardinal}(\{\text{hw, assignments-of}(\text{AI, hw})\}), + \{ \text{comparison-class} \} \rangle$. This formula can be read as: the set of assignments in AI is large in relation to some comparison class. The comparison class is a pragmatic variable that can be instantiated in a given situation of enunciation, and can be constrained by certain elements in the sentence. For

example, the following reference sets can be used in different situations:

AI has more assignments than

- *Introduction to programming*
- *any other course in the University*
- *a typical class in computer science*
- *a typical class in the university*

The notion of comparison class is very general and has been used to provide a semantics to scalar adjectives (a comprehensive treatment of this notion is provided by Klein in [28]), to account for comparatives (cf. [39] for a practical application of Klein’s model) and for constructions like “*big for an elephant*” [29].

The third type of answer is given by evaluating the scale in terms of another evaluation. For example, the number of assignments would be qualified as being a number such that AI is difficult. Such *composite evaluations* measure a conceptual scale by using another evaluation. They are of the form: $\langle \text{conceptual-scale}, \langle \text{conceptual-scale}, \text{value} \rangle \rangle$. For example:

$$\langle \text{cardinal}(\{\text{hw}, \text{assignments-of}(\text{AI}, \text{hw})\}), \\ + \langle \text{difficult}(\text{AI}), + \rangle \rangle$$

This recursive definition captures the intuition that an evaluation is a way of looking at an entity. We are evaluating the number of assignments in a class as it relates to the difficulty of the class; that is, as a criterion to conclude how difficult AI is.

The notion of composite evaluation is useful to distinguish between *linear* and *non-linear* topical scales (a terminology used in [28]). A linear scale is a scale for which only one criterion can be used to compute an evaluation. In contrast, several criteria can be used to evaluate on a non-linear scale. For example, the scale of difficulty is non-linear: a class can be difficult because it is theoretical, or because it has many assignments... In contrast, the scale of cardinality for sets is linear.³

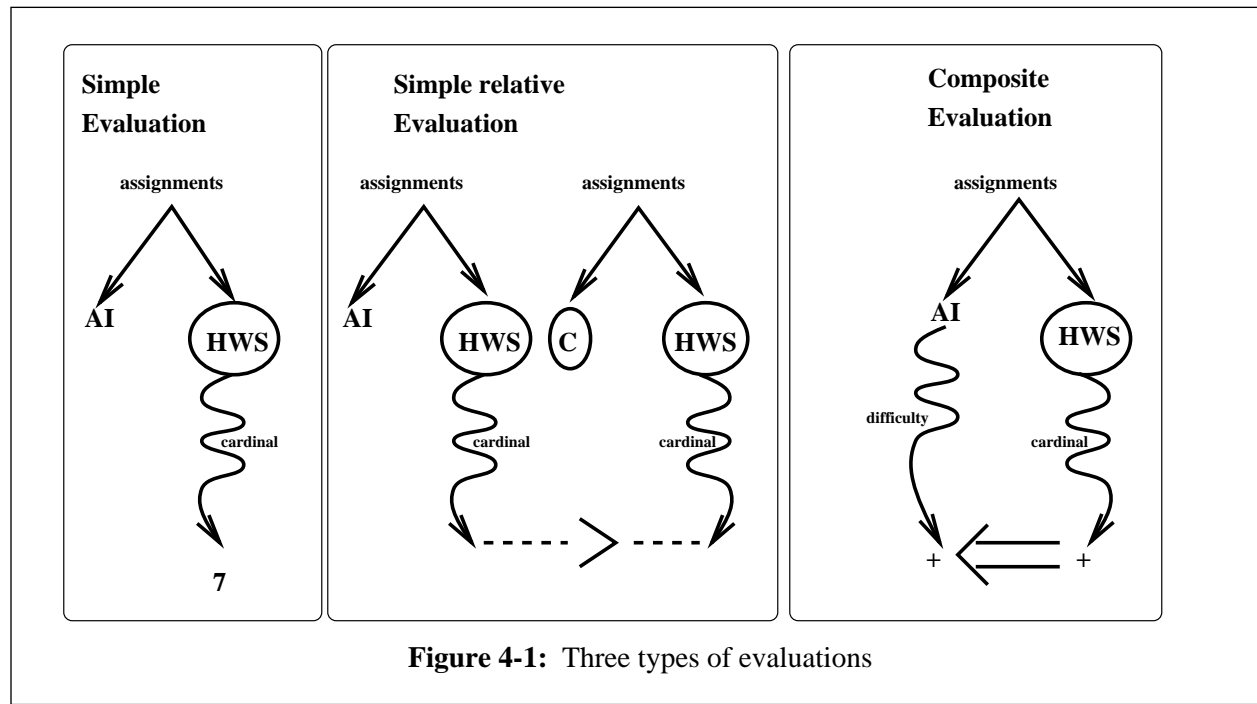


Figure 4-1: Three types of evaluations

The three different types of evaluations are depicted in Fig.4-1. In this figure, conceptual relations extracted from the knowledge base are shown with straight lines, and argumentative evaluations are

³Non-linear scales induce only a partial order on the objects that are evaluated, which relates to the notion of partially ordered set used in [20] in the context of scalar implicatures.

shown with wavy lines, labeled with the scale. Sets are represented in circles, and individuals as simple labels. The figure depicts three topical evaluations that could be phrased respectively:

- *AI has 7 assignments.*
- *AI has many assignments [more than other classes].*
- *AI requires many assignments [so it is difficult].*

A composite evaluation of the form $\langle c1, \langle c2, v \rangle \rangle$ corresponds to a canonical topos, of the form */the more c1, the more c2/*. The right-hand side of Fig.4-1 can be seen as the depiction of the topos */the more a class has assignments, the more it is difficult/*. In this figure, the four relations that make up a topos (as discussed in Sect.3) appear as four arcs: the evaluation of the entity HW on the scale of cardinal is shown with the rightmost wavy line, the evaluation of the entity AI on the scale of difficulty with the leftmost wavy line, the gradual relation between these two evaluations with the bottom implicative arrow, and the topical relation between the entities AI and HW is shown with the knowledge-base assignments-of relation at the top.

4.2. Evaluation Functions: Producing Evaluations from Observations

The role of the content determination module is to produce a set of such argumentative evaluations that relate to the ultimate argumentative intent of the answer. This intent is given as input to the content determination module. The output is a set of propositions extracted from the system's knowledge base and a set of evaluations. While evaluations can be naturally related to the desired argumentative intent of the answer (which is itself an evaluation), there is no direct way to connect the knowledge base information available to the system to this same input. There is indeed a gap between these two types of propositions: evaluations are situation dependent (they depend on the student model and eventually on the argumentative intent of the answer) while knowledge base facts are not. This gap needs to be bridged to connect the information in the knowledge base to the argumentative intent of the answer given as input to the text generation system. The task of the content determination module is therefore both to relate evaluations to facts and to select only the set of facts that are appropriate for a certain conclusion.

We represent the argumentative intent as an evaluation of the form $\langle \text{take}(\text{student}, \text{AI}), + \rangle$. It is given as input to the system, and can therefore be considered as an *a priori* bias of the answer. The output of the content determination module which eventually leads to the production of the example in Fig.2-2 is shown in Fig.4-2.

The figure shows the four argumentative derivations that link information extracted from the knowledge base and the user model to the eventual evaluation $+/-\text{take}(\text{student}, \text{AI})$. Each derivation is made up of three types of information:

1. A set of knowledge-base propositions called *observations*. An observation only contains objective facts about the classes and assumptions about the user.
2. An evaluation supported by the observations called a *judgment*. A judgment is a simple absolute evaluation on a non-linear scale, using the terminology just introduced. A judgment can be seen as the left part of a topos anchored in the knowledge base.
3. Finally, each judgment is linked to the argumentative intent of the answer by chaining through *topoi*. Topoi are used here in a way very similar to the production rules of an expert system.

Thus the content generated contains two distinct types of propositions:

1. knowledge base facts, e.g., $\text{topics-of}(\text{AI}, \text{topics1})$.
2. scalar evaluations, e.g., $\langle \text{interest}(\text{AI}), + \rangle$.

An observation in the knowledge base is linked to the argumentative intent in two steps: a first evaluation is derived from the observation, then this evaluation is linked through a *topoi-chain* to the argumentative intent. The first step is performed by *evaluation functions*, which bridge the gap between knowledge-base information and evaluations. Evaluation functions map a collection of facts stored in the knowledge base to the activation of a scale. An example of evaluation function is shown in Fig.4-3.

Evaluation functions have the form of a query/conclusion pattern. The rule in Fig.4-3 can be read as

QUESTION: Should I take AI?

```
(1) KB Observations:  assignments-of(ai, assignments1)
                      member-of(assignments1, a1),
                      activity-of(a1, paper-writing),
                      user-profile(experience, paper-writing, -)
Judgment:             <workload(AI), +>
Argumentative Linking: /+ workload(AI), + difficult(AI)/
                      /+ difficult(AI), - take(student, AI)/

(2) KB Observations:  topics-of(ai, topics1),
                      member-of(topics1, nlp),
                      user-profile(interest, nlp, +)
Judgment:             <interest(AI), +>
Argumentative Linking: /+ interest(AI), + take(student, AI)/

(3) KB Observations:  topics-of(ai, topics1),
                      member-of(topics1, logic),
                      area(logic, theory),
                      user-profile(experience, theory, -)
Judgment:             <theory(AI), +>
Argumentative Linking: /+ theory(AI), + difficult(AI)/
                      /+ difficult(AI), - take(student, AI)/

(4) KB Observations:  assignments-of(ai, assignments1),
                      subset-of(assignments1, prog-assignments,
                                activity(X, programming)),
                      cardinal(prog-assignments) > 2,
                      user-profile(interest, programming, +) ]
Judgment:             <programming(AI), +>
Argumentative Linking: /+ programming(AI), + interest(AI)/
                      /+ interest(AI), + take(student, AI)/
```

Figure 4-2: Output of the content determination module

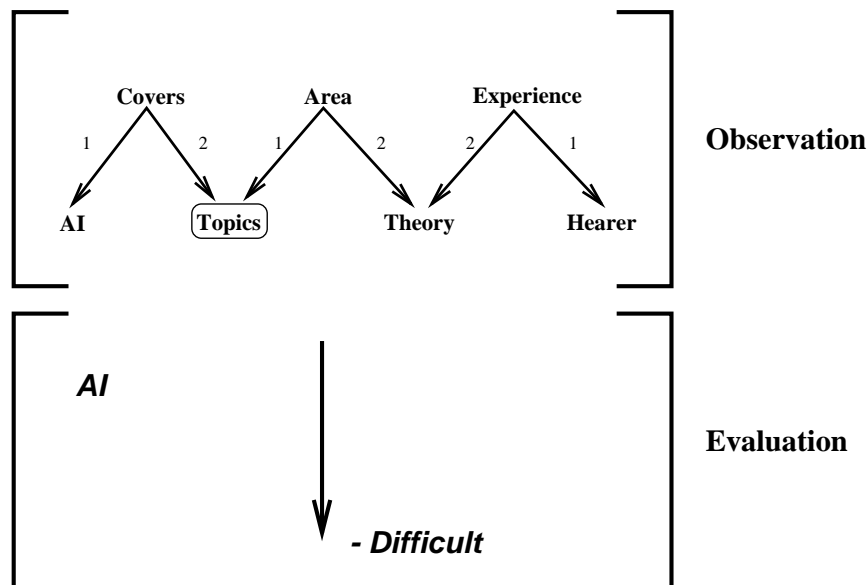
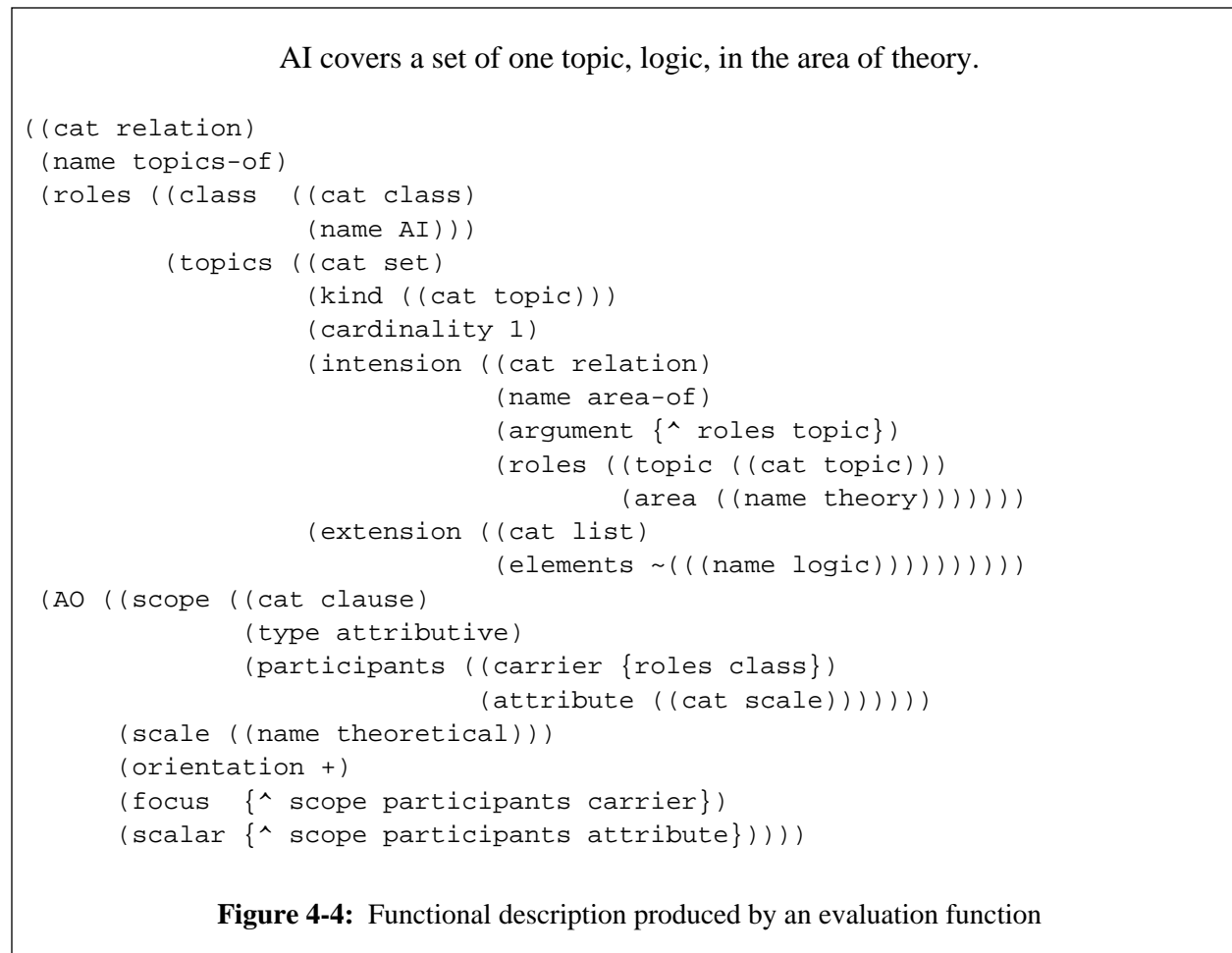


Figure 4-3: An evaluation function

follows: if there is in the knowledge base a set of topics ts such that (1) AI covers all ts , (2) all ts are in the area of theory, and (3) the user model indicates that the user has no experience in the field of theory, then AI can be evaluated as high on the scale of being theoretical. In generation, we can use this rule in

the other direction: in order to support the conclusion that AI is theoretical, it is sufficient to find three propositions in the knowledge base or user model matching the pattern. When three such propositions can be found, they are added to the pool of propositions that the explanation must convey, as shown in the third chunk of Fig.4-2.

In addition, all propositions are annotated with an argumentative feature indicating that they all serve as arguments for the given conclusion. A proposition extracted from the knowledge base is called an *observation*, while we call an annotated proposition a *judgment* (this terminology is derived from [7]). The formal representation of an annotated proposition is shown in Fig.4-4.



The figure uses the FUF syntax to encode functional descriptions, which are lists of attribute-value pairs. The syntax is fully explained in [13, Chap.3]. To understand the figure note that the { } notation indicates pointers to other features within the structure. The figure shows the FUF functional description representation of the following complex knowledge base observation:

```

topics-of(AI, topics1) where
  for all t ## topics1, area-of(t, theory) and
  topics1 = {logic}

```

Literally, this expression means that there is a set of one topic, logic, such that all topics in the set are in the area of theory and all topics in the set are covered by AI. The most important part of the figure is that this knowledge base description, of category *relation* is also annotated with a feature called *AO* for argumentative orientation. In this case, the *AO* specifies that the role *class* of the relation is being evaluated on the scale of being theoretical; that is, the absolute simple evaluation $\langle \text{theoretical}(\text{AI}), + \rangle$ must be expressed when conveying the content encoded in this description. In other words, the linguistic realization component must present this fact as an argument supporting the stated conclusion.

Thus a fact is never passed “alone” to the linguistic generation component: it is always annotated with an *AO* feature, and this annotated structure is called a *judgment*. Evaluation functions are the tools that map from objective observations to judgments that are context-dependent, gradual and goal-oriented.

We have empirically identified the following 8 scales in our domain:⁴

- Goodness
- Importance
- Level
- Difficulty
- Workload
- Programming
- Interest
- Theoretical

The 15 evaluation functions implemented in the current ADVISOR II prototype, provide a semantics for these 8 scales by defining which objective facts, contextual factors and argumentative intentions can activate an evaluation on each scale.

Practically, evaluation functions also serve as the single interface between the knowledge base and the generation system. They therefore implement the translation between the different formalisms used in each system (CLASSIC and FUF-based functional descriptions).

In summary, evaluation functions are important in bridging the gap between argumentative intent and propositional content. The use of evaluation functions also distinguishes our method from previous generation systems in an important aspect: the content derived from the knowledge-base is annotated with argumentative orientation features that indicate its function in the interaction. These annotations are used when phrasing the explanation to select words and constructions that reflect this argumentative function, and allow the lexical chooser to make decisions that would not be possible without this extra information.

4.3. Knowledge Sources and their Acquisition

Evaluation functions require the specification of two knowledge sources: a knowledge base and a set of scales on which objects of the knowledge base can be evaluated. We now describe how this knowledge is represented in the system, and how it has been modeled.

Information about the classes and the student is represented in a knowledge base implemented in the CLASSIC formalism [40, 2], a KL-ONE type of language. The knowledge base contains a collection of objects of different types: class, course, student, teacher, topic and assignment are the most important in our domain. Each object is identified by a set of attributes. For example, the class object has for attributes `area`, `prerequisites`, `topics-of`, `assignments-of`, `followup-class`, `required`. Each attribute can be seen as a binary relation between an object and a set of other objects. For example, a `class` object is related to a set of `tasks` through the `assignments-of` relation.

The second knowledge source needed is a set of evaluation scales along which classes can be evaluated. Since the use of such scalar information is quite specific to our approach, we now explain how it can be acquired and modeled. As explained above, the semantics of the scales is defined by the evaluation functions which determine under which conditions a scale can be activated, and by the topoi which determine how the scales are related to each other. Our implementation contains 14 primitive topoi linking the 8 scales of the domain. More topoi can be produced by applying transformations to the primitive topoi (e.g., `/+ , - /` to `/- , + /`) and by linking topoi.

Scales and topoi have been identified by analyzing a corpus of real advising sessions that we have recorded by looking for linguistic clues of scalar reasoning. We focused our analysis on uses of scalar adjectives, judgment determiners such as “*many*” or “*a lot*” and connectives such as “*but*” or “*therefore*”.

In particular, we performed an exhaustive analysis of the 40,000 words corpus for the analysis of adjectives. In this corpus, we identified approximately 1400 occurrences of 240 distinct adjectives. We

⁴The way the scales have been identified is discussed below. We want to thank Chaya Ochs for her assistance in the identification of evaluation functions and in their implementation.

focused on all occurrences of adjectives modifying a course, in both predicative and attributive positions. We found 150 such occurrences, of 26 distinct adjectives. The next step was to cluster these adjectives in semantic classes, each class providing a first approximation of scale. The results are shown in Fig.4-5.

Figure 4-5 highlights an important property of the domain: a lot of what is said about a class in real advising sessions is scalar evaluations. It also shows that an easy to apply, quite reliable procedure can be designed to identify salient scales in a specific domain. Focusing on adjectives to identify the semantic scales underlying our domain is motivated partly by the technique of the *semantic differential* of Osgood (cf. [35, p.20] for a discussion of the use of scalar adjectives for "meaning measurement", cf. also Sect.3.2 for a discussion of Osgood's theory).

Note that a similar procedure of exhaustive corpus analysis and clustering was performed on nouns and verbs in order to identify the classes and relations needed in the knowledge representation of the domain.

5. The Role of Topoi in Content Organization

Once the content of the answer has been determined and a set of semantic propositions with their argumentative annotations has been produced, the generator must organize and structure the answer. This is the task of the content organization module. We first present in this section what issues are involved in content organization, by identifying a set of dimensions along which a paragraph structure can vary. We then briefly survey existing work in content organization and finally describe how topoi can serve as the basis of a content organization procedure operating on argumentatively annotated semantic propositions.

5.1. The Task of Content Organization

The input to the content organization module is a pool of propositions, annotated with argumentative orientation features, as shown in Fig.4-2. The output is a network where each node is a proposition and each link a rhetorical relation. Figure 5-1 illustrates different answers generated from the same pool of propositions (shown in Fig.4-2), but using different rhetorical structures.

Figure 4-2 shows that four evaluation functions matched the query. For each match, the evaluation function identifies a set of knowledge-base observations, a judgment (*i.e.*, an evaluation supported by the observations) and a chain of argumentative inferences linking from the judgement to the eventual conclusion. The topoi used in this chain are assumed to be shared by the speaker and the hearer.

Semantic class	Adjective	Occurrences	Semantic class	Adjective	Occurrences
Difficulty [24]	advanced	1	Importance [24]	important	10
	basic	1		needed	1
	challenging	1		recommended	5
	difficult	4		required	5
	easy	5		suggested	1
	hard	11		useful	1
	high-level	1		valuable	1
Domain [8]	mathematical	2		Evaluative [10]	interesting
	programming	4	perfect		1
	theory	1	good		5
	computing	1	Misc [3]	traditional	1
		new		1	
		interdisciplinary		1	

Figure 4-5: Adjectives modifying courses in corpus

AI covers NLP, a topic that interests you.
 SO it should be interesting.
 I would recommend it.

AI should interest you,
 BECAUSE it offers lots of programming assignments.

AI can be difficult,
 BECAUSE it requires a lot of work
 AND it can be pretty mathematical,
 BUT it is a very interesting course,
 BECAUSE it covers many exciting topics,
 AND it includes a good number of programming assignments.

AI covers nlp, a topic that interests you,
 AND it has plenty of programming assignments.
 SO, it is a very interesting course,
 EVEN THOUGH it can be rather difficult,
 BECAUSE you would have to write some papers,
 AND it covers some pretty theoretical stuff.

Figure 5-1: Different rhetorical structures

Figure 5-1 shows possible rhetorical structures corresponding to this input, assuming that the argumentative intent of the answer is to recommend taking the AI course. In this figure, indentation is used to indicate the subordination relations between main points and elaborations, and connectives expressing the articulation of the answer are printed in all caps. The list shown only hints at the wide variety of options available to the content organization module. To better define the task of content organization, we have identified dimensions along which the structure can vary. We have characterized two main dimensions: mode of selection of propositions and mode of aggregation. The different options which determine how to map from the content shown in Fig.4-2 to one of the structures shown in Fig.5-1 are summarized in Fig.5-2.

The first dimension concerns selection of content out of the input pool. There are several options: one might always convey all the propositions of the input in the answer, or select only a subset of those. When selecting a subset, different criteria can be used: select a subset of a fixed size (*e.g.*, only one proposition, probably the “best” according to some metric), select only the arguments that support the argumentative intent of the answer (as in the first two paragraphs in Fig.5-1) or include also contradictory arguments (as in the last two paragraphs).

Dimensions of variability in content organization

1. Selection:

- 1.1 all
- 1.2 n only
- 1.3 + only
- 1.4 + and -

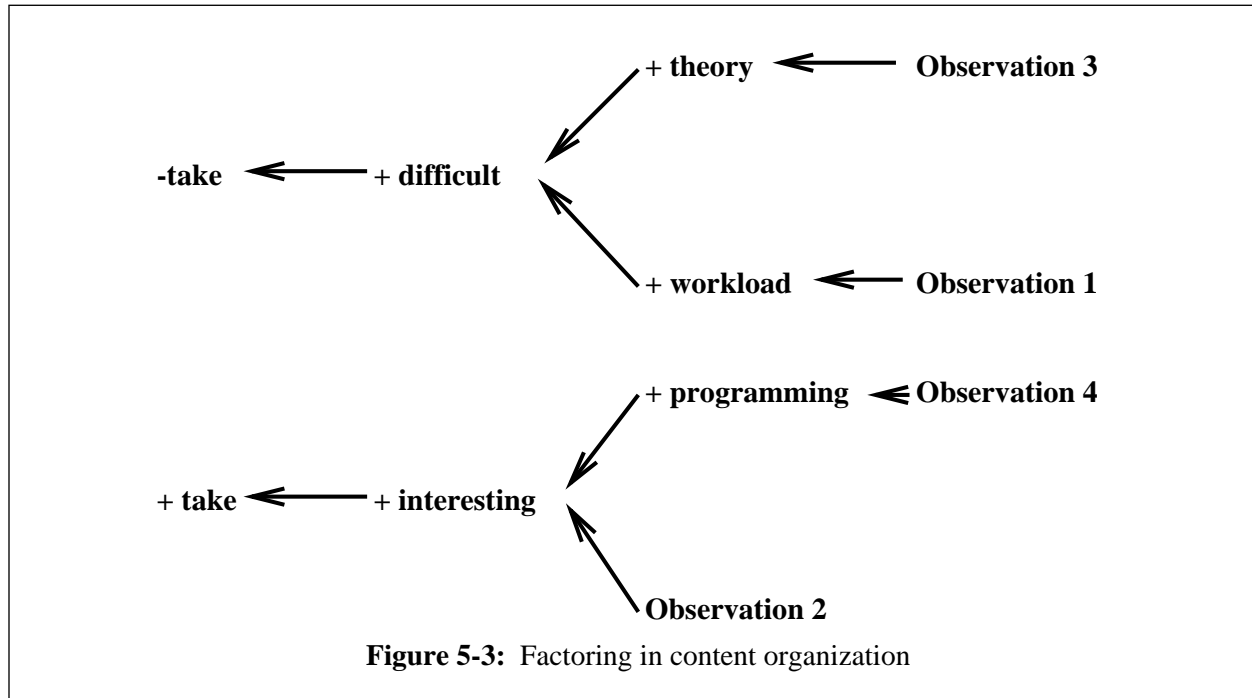
2. Aggregation:

- 2.1 factoring (factor out redundant information)
- 2.2 filtering (redundant and recoverable information)
- 2.3 connective
- 2.4 embedding

Figure 5-2: Issues in content organization

The second dimension concerns aggregation and structure. The input in Fig.4-2 is made up of chains of propositions, connected by argumentative relations. The content organization module can factor out

common parts of several chains. For example, chains 1 and 3 both end with the same links `/+difficult,-take/`. It is therefore possible to build a tree-like structure, as shown in Fig.5-3. In this tree, the leaves correspond to the observations found by the evaluation functions; that is, to propositions extracted from the knowledge base. Each link corresponds to a topos relation between two evaluations. Given the tree structure, one can avoid to repeat redundant information and use instead conjunction to group arguments supporting the same conclusion (as in the last two paragraphs of Fig.5-1). Another way to deal with redundant or recoverable information is to just filter it out and leave it implicit. For example, in the second paragraph, the conclusion that the advisor recommends taking AI is not stated explicitly.



When several arcs reach the same node in the tree shown in Fig.5-3, the linguistic realizations corresponding to the several conceptual predicates can be combined into a single linguistic constituent. For example, in the top part of the figure, two arcs reach the node `+ difficult` and one arc leaves it. Two methods of combination can be considered: by using connectives, several propositions can be combined (either in a conjunction as in *AI is difficult: it is quite theoretical and requires a lot of work* - or in a subordination as in *AI is difficult because it requires a lot of work*). Often, the connective can be left implicit and the clauses realizing the propositions are just put in sequence. Another method for combining propositions is to embed one proposition as a modifier of one of the noun phrases of a clause. Embedding can be used when two propositions share a conceptual entity that is realized by a noun phrase as in *AI which covers some theoretical topics is quite difficult*.

The task of the content organization module can therefore be described as follows: traverse the tree shown in Fig.5-3 and at each node choose a selection strategy (include the arguments or not) and an aggregation strategy (how to add new propositions to the structure being built). In our system, we have not investigated how this choice is made and what factors influence it.⁵ We have instead studied the range of decisions that must be made to structure a paragraph, and most importantly, how these decisions affect the linguistic realization of each clause within the paragraph.

5.2. Previous Work on Content Organization

In previous work in text generation, two techniques have emerged to handle the task of content organization: rhetorical schemas and RST (for Rhetorical Structure Theory).

Schemas, introduced in [31], encode conventional patterns of text structure. A schema is associated

⁵The factors determining the choice of a content organization strategy are mainly conceptual: what argument will have the most effect on the hearer, do I need to counter some preconceptions of the hearer, which argument should be given more salience in the structure. We focus in our work on the more linguistic decisions.

with a communicative goal and describes how this goal is conventionally satisfied. For example, the *constituency schema* is used to describe the parts of an object, and the *process schema* [36] is used to describe a complex process. Such schemas are devised by empirical observation. A schema describes a sequence of *rhetorical predicates* where each predicate is either a primitive communicative function, which can be fulfilled by a single proposition, or recursively another schema. Example of primitive predicates include *attributive* to attribute a property to an object and *illustration* to elaborate a claim by an example.

When activated, a schema is applied to an input concept from an underlying knowledge-base. Each predicate in the schema is mapped to a query in the knowledge base which is evaluated on the input concept, thus determining the content of each proposition in the paragraph. The output of a schema application is therefore a sequence of propositions labeled by the name of the rhetorical predicates they instantiate. In most implementations, transition words and connectives are also added by the schema traverser and selected before the generation of each clause starts. Schemas therefore address at the same time the tasks of content determination and content organization.

When the schema leaves a choice under-determined, other pragmatic factors are taken into account to commit to a choice: for example, in [31] focus transition rules derived from [42] constrain the order in which concepts can be referred to, and in [36] a user-model is consulted.

While schemas label each proposition as the instantiation of a predicate, RST attempts to label the *relation* between propositions. RST, as introduced in [30], was first a descriptive theory aiming at enumerating possible rhetorical relations between discourse segments. RST relations include *elaboration*, *anti-thesis* and *solutionhood*. A relation connects two text spans, which can be either single propositions or recursively embedded rhetorical relations. One argument of the relation is marked as its “nucleus” - and conveys the main point of the text span - while the others are the “satellites” (subordinate).

RST was made operational as a technique for planning the structure of paragraphs in [22] and [33], elaborating an idea first introduced in [30]. The idea is to attach a communicative intent with each RST relation and to view the combining of relations into paragraphs as a planning process, decomposing a high-level intention into lower-level goals that eventually can be mapped to the utterance of single propositions. The communicative goals associated with the leaves of the structure are then used to determine the content of each propositions. By making the intentional structure of the paragraph explicit, this work combines easily with the planning-based content determination techniques reviewed in the previous section. Note also that, since in RST with planning, the structure of paragraphs is dynamically derived, it is possible to view schemas as the compilation of RST configurations with some information abstracted out, as pointed out in [34] or [24, p.31]. In addition, [25] introduces the notion of *growth point* in RST relations: a growth point is an opportunity for the text planner to attach more information to one of the nodes of the structure being constructed. An example of annotated RST relation is shown in Fig.5-4. This example shows how the RST relation of sequence is formalized as a plan which can be applied to communicate the information that a *sequence-of* relation holds between two input propositions (the variables ?PART and ?NEXT). When the relation is selected, a choice of connectives is also determined (no connective, *then* or *next*). Note that this definition crucially depends on the domain-specific relation called *NEXT-ACTION* which determines when it is possible to view two events as members of the same sequence.⁶

5.3. Argumentation and Content Organization

Topoi can be used for the task of content organization when one notices that topoi can be directly interpreted as rhetorical relations. While topoi are different from RST-like relations, they account for many of the phenomena for which RST relations were invented: they explain the coherence between two text spans in terms of a conceptual relation; they can be related to communicative goals such as argumentative intent and a form of planning can compose several topoi into extended chains. For example, in order to answer whether the student should take AI, our system uses backward chaining through the topoi base to link a possible conclusion like *+take(ai)* to an evaluation function, and eventually to some observations in the knowledge base, to produce the chains shown in Fig.4-2. This backward chaining is similar to the process of top-down planning discussed in [25]. In addition, when merging several topoi chains together, to build a tree as shown in Fig.5-3, each merging point plays a role similar to the growth points of [25]. So topoi account for at least all the phenomena that RST relations are supposed to explain.

⁶The notation BMB is the “mutual belief” operator: (BMB S H P) means that P is mutually believed by S and H

Name: SEQUENCE
Results: ((BMB SPEAKER HEARER (SEQUENCE-OF ?PART ?NEXT)))
Nucleus requirements/subgoals: ((BMB SPEAKER HEARER (TOPIC ?PART)))
Satellite requirements/subgoals: ((BMB SPEAKER HEARER (TOPIC ?NEXT)))
Nucleus+Satellite requirements/subgoals: ((NEXT-ACTION ?PART ?NEXT))
Nucleus growth points:
 ((BMB SPEAKER HEARER (CIRCUMSTANCE-OF ?PART ?CIR))
 (BMB SPEAKER HEARER (ATTRIBUTE-OF ?PART ?VAL))
 (BMB SPEAKER HEARER (PURPOSE-OF ?PART ?PURP)))
Satellite growth points:
 ((BMB SPEAKER HEARER (ATTRIBUTE-OF ?NEXT ?VAL))
 (BMB SPEAKER HEARER (DETAILS-OF ?NEXT ?DETS))
 (BMB SPEAKER HEARER (SEQUENCE-OF ?NEXT ?FOLL)))
Order: (NUCLEUS SATELLITE)
Relation-phrases: (" "then" "next")
Activation-question:
 "Could ~A be presented as start-point, mid-point, or end-point of some
 succession of items along some dimension? -- that is, should the hearer
 know that ~A is part of a sequence?"

Figure 5-4: The RST relation/plan SEQUENCE (from [24, p.22])

But in addition, topoi also provide further benefits to the problem of content organization, that derive from the two key aspects in which they differ from RST-like relations: topoi are both domain and situation specific and topoi are linked to the lexicon.

Because topoi are domain dependent and user dependent, it is possible to build an operational system that determines whether a relation is applicable in a certain situation of enunciation. In fact, when using RST or a related theory, system implementors must also attach domain-dependent predicates to each generic relation to determine when it can be used (for example, the `NEXT-ACTION` relation in Fig.5-4). So while the theory of RST claims to have uncovered generic rhetorical relations, each implementation must redefine what it means to be an example, what it means to be a solution etc. in domain specific terms. In [30], this task is described in terms of “oracles” - black boxes that are able to answer questions like “*if I say X, will the reader believe it?*”. But RST provides little guidelines to build such oracles in practical systems. And indeed, implementors are often left struggling, trying to find operational nuances between RST relations like “*cause*”, “*reason*”, “*justification*” and “*evidence*” (all four relations are used for example in [21]). In contrast, when using topoi as a primary form of rhetorical relation, one does not need to distinguish between the generic level and the domain dependent. There is no need to develop a vocabulary like “*evidence*”, “*justification*” etc. One can implement a text planner where the domain relations themselves can be used to account for the rhetorical structure.

The second aspect in which topoi differ from RST-like relations is that topoi are related to the lexicon and to the rules of syntax, as is detailed in Sect.6 (cf. also [14] for a development of this argument). Therefore, when selecting a topos, the system not only creates a complex text span (a text structuring operation), but also constrains the linguistic realization of this text span. Because topoi annotations are kept with each proposition sent to the lexical choice module, the rhetorical function of a text segment can have an effect on all aspects of its realization. In contrast, in an RST-based text structurer, the position of a text span within the rhetorical tree only affects the choice of connectives, as indicated by the `relation-phrases` feature of Fig.5-4 (this limitation is partially acknowledged in [25, p.96]). But the rhetorical function of a proposition affects its linguistic realization at many levels. Consider the following utterances, extracted from a corpus of real advising sessions:

- *It requires quite a lot of programming*
- *It does involve some programming, but nothing outrageous.*

Both sentences can be generated from the *same* observations in the knowledge base, but the difference between the two forms is determined by the argumentative function of the clause. This argumentative difference explains the selection of “*a lot*” vs “*some*”, “*require*” vs. “*involve*”, of the concessive construction “*does include ... but ...*”. In that sense again, topoi can serve as a bridge between early decisions in the generation process (content organization) and later decisions (lexical choice), as we now

show.

6. The Role of Topoi in Lexical Choice

The task of the lexical choice module is to select which words will eventually appear in the answer. We first present a brief overview of the issues involved in lexical choice. We then focus on how argumentation can be used to help perform lexical choice, and show how a verb conveying an appropriate connotation is represented in the lexicon and how it gets selected by the generator.

6.1. The Task of Lexical Choice

Input to the lexical choice module is the rhetorical structure built by the content planner; that is, a network of conceptual propositions that are annotated by their argumentative function. A comprehensive survey of the research in lexical choice is provided in [41]. We enumerate here some constraints that have been identified on lexical choice in previous work.

There are two aspects to the lexical choice task:

- Syntagmatic: The lexical chooser selects lexical heads out of the conceptual network, and for each head, determines its syntactic category and which syntactic constituents depend on the head.
- Paradigmatic: The lexical chooser selects an entry from the lexicon to map each head to an actual word.

The first aspect interacts closely with the syntactic decisions to be made when generating a sentence. In previous work, this was done by associating predicate-argument relations of the conceptual network with phrasal templates. For example, [6] relied on a lexicon-grammar and [26] on pattern-concept pairs to perform this task.

The second task of a lexical chooser is, given a conceptual head and its dependents, to choose an appropriate lexicalization. There is indeed a many-to-many mapping between conceptual entities and relations and the lexical entries. There are several ways to choose between different lexicalizations. In certain cases, the choice has no impact on the appropriateness of the generated text. For example, the node `AI` can be realized by “*AI*”, “*the AI course*”, “*Artificial Intelligence*”. In most situations, the choice between these alternatives has little impact.

In other cases, the difference between alternative lexicalizations can be related to stylistic factors: length of the expression, formality etc. Such factors have been used in [23] for example.

In yet other cases, alternative lexicalizations can be determined by the semantics of the arguments of the lexical head. For example, if the conceptual entity is `produce-noise`, then depending on the semantic type of the agent, the words “*bark*”, “*resound*”, “*meow*” etc... would be selected. Such selections have been handled in previous work by building discrimination trees, which describe how to progressively refine a generic concept into a specific lexicalization (cf. [19] for the original implementation of this idea and [6, 5] for recent uses of the technique).

Finally, and this the motivation of this work, we have also found that one can choose between alternative lexicalizations by looking at the argumentative intent of the utterance. We have presented some of the issues involved for connectives in [32, 16], for adjectives in [12], for the selection of verbs and adverbials in [17], and for the selection of determiners in [15].

6.2. Using Argumentation for Lexical Choice

There are two demands a generator must meet when mapping a conceptual predicate-argument structure to a linguistic structure: completeness - each conceptual predicate-argument must be “covered” by a linguistic predicative relation - and coherence - no unwanted semantic information should be conveyed. For example, the relation `assignments-of` between the entity `AI` and the set `HW` in Fig.4-1 is “covered” in the sentence “*AI has many assignments*”, but not in the sentence “*John takes AI and does the homework*” (it is therefore not a complete rendering of the input) and the proposition `DO(John, HW)` is expressed by the second sentence while it is not part of the input (it is therefore not a coherent rendering of the input).

Similarly, argumentative evaluations present in the input must be “covered” by some linguistic marker to be satisfied. In our system, the use of topoi in content determination and organization has produced argumentative information and attached it to the input to the lexical chooser. The task we now consider is how to satisfy these new demands on the lexicalization module: how to choose words that both convey a given semantic information and express an argumentative intent.

In our work on lexicalization and syntactic generation, we have looked at how the following decisions can be made non-arbitrarily by the generator, and motivated by argumentative factors:

- Choice of scalar adjectives in the domain (words like “*interesting*” or “*difficult*”).
- Choice of judgment determiners (words like “*many*” or “*few*”).
- Choice of argumentative connectives (words like “*but*” or “*so*”) between clauses.
- Selection between quasi-synonyms that differ only by their connotations

We have developed and implemented ADVISOR II as a complete generation system that produces sentences all the way from a formal query and that includes a comprehensive grammar of English. A complete account of the method proposed for lexicalization is given in [13]. We focus here on the last one of these choices: the selection of a lexical entry differing from other close synonyms by its connotations.

As an example, consider the contrast between the following sentences:

- *AI requires/necessitates/demands many assignments.*
- *AI has/includes/involves many assignments.*

All 6 verbs shown express (more or less successfully) the conceptual relation between a class and its assignments, roughly translated in language as a possessive relation. As discussed in Sect.3.2, there is however a difference between the first group and the second: *require*, *necessitate* and *demand* all three carry an argumentative connotation. In addition to conveying the conceptual relation of assignments-of between AI and a set of assignments, these verbs also express an evaluation of AI as a difficult class. Thus, choosing one of these verbs is sufficient to realize this argumentative evaluation when it is present in the input to the lexical chooser.

Similarly, one can contrast between the following sentences:

- *You enjoyed AI.*
- *You took AI.*
- *You struggled with AI.*

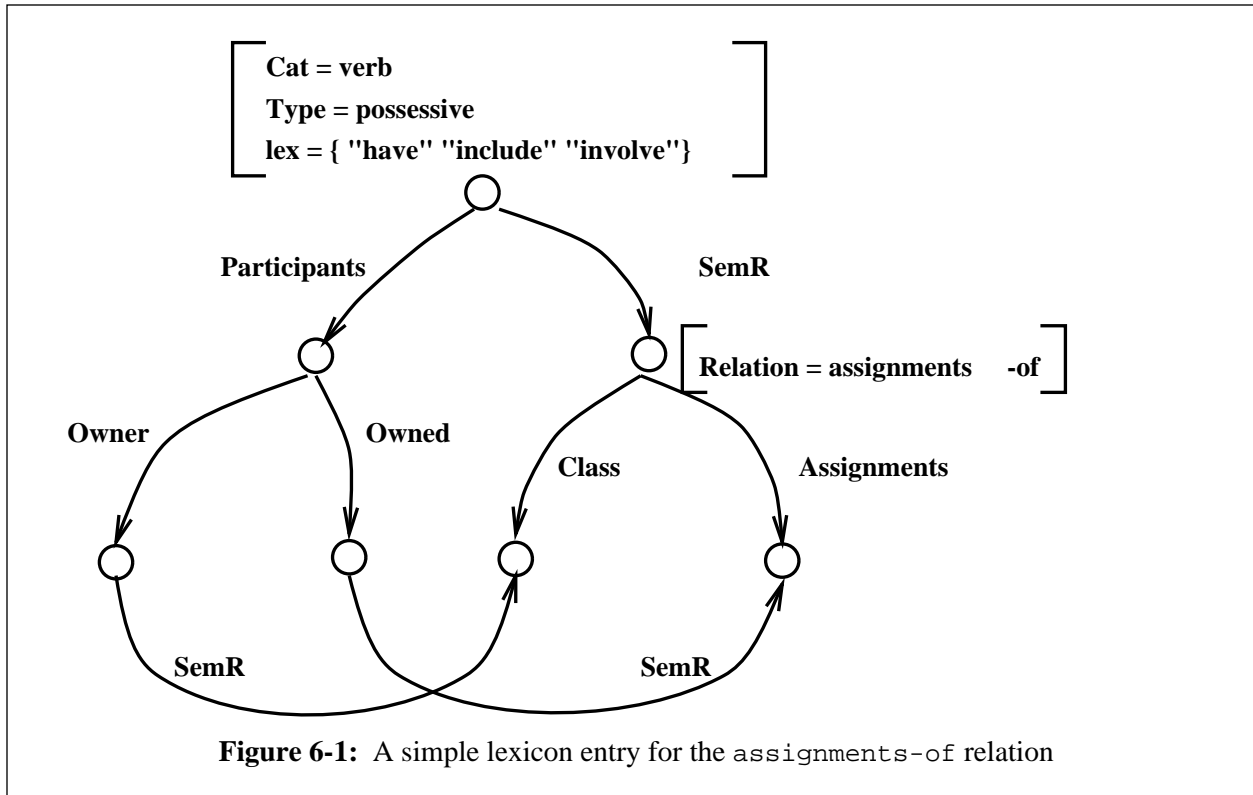
The verbs *enjoy*, *take* and *struggle* all cover the relation that the student has taken the AI class. But *enjoy* and *struggle* also express an argumentative evaluation of the class (as an interesting or difficult class).

Information on such argumentative connotations is stored in the lexicon so that the lexical chooser can consider these connotations as an additional resource to cover argumentative evaluations in the input. We now look at this information.

We use a lexical formalism inspired by HPSG [37] in our implementation, using FUF, our version of functional unification formalism [10, 13, Chap.4]. The information in this formalism is uniformly represented by sets of attribute-value pairs called *functional descriptions* or *FDs*. Curly braces indicate a disjunctive value. For example, in Fig.6-1, the value of the `lex` feature of the top node can be either “*have*”, “*include*” or “*involve*”.

Figure 6-1 shows how in the lexicon the semantic relation assignments-of is mapped to the choice of a verb, when argumentation plays no role. Each lexical head in the formalism has a feature `SemR` (for semantic representation) which contains a partial description of the semantic content covered by the word. For generation, lexical entries are indexed by conceptual objects - classes or relations. Thus, in Fig.6-1, the entry is indexed by the `[Relation = assignments-of]` feature. This relation is mapped to a lexical head of syntactic category `verb` and of type `possessive`. In addition, the entry specifies how the two entities of the semantic representation are mapped to the lexical arguments of the verb, the constituents labeled `owner` and `owned`. This mapping is indicated by the `SemR` connections between the linguistic subconstituents and the semantic arguments.

The mapping shown in Fig.6-1 therefore expresses a phrasal pattern of the form: Concept:



assignments-of(Class, Assignments) Linguistic Structure:

possessive(Owner, Owned) Phrase:

Owner {have, include, involve} Owned Mapping:

Class->Owner, Assignments->Owned

Figure 6-2 indicates how argumentation is represented in the lexicon. It shows the more complex example of the lexicon entry of the verb “to require” with its argumentative connotation. The figure actually shows how the lexical entry is instantiated when applied to the semantic input corresponding to the composite evaluation in the right part of Fig.4-1. The top part of the figure is very similar to Fig.6-1. It describes how the verb “require” covers the semantic relation of assignments-of. The only difference is that the verb “require” is not a member of the class of “possessive” verbs, which share many properties and are known to the grammar. Instead, its syntactic properties are all represented in the lexicon (but not shown in the figure). This is indicated by the feature [type lexical].

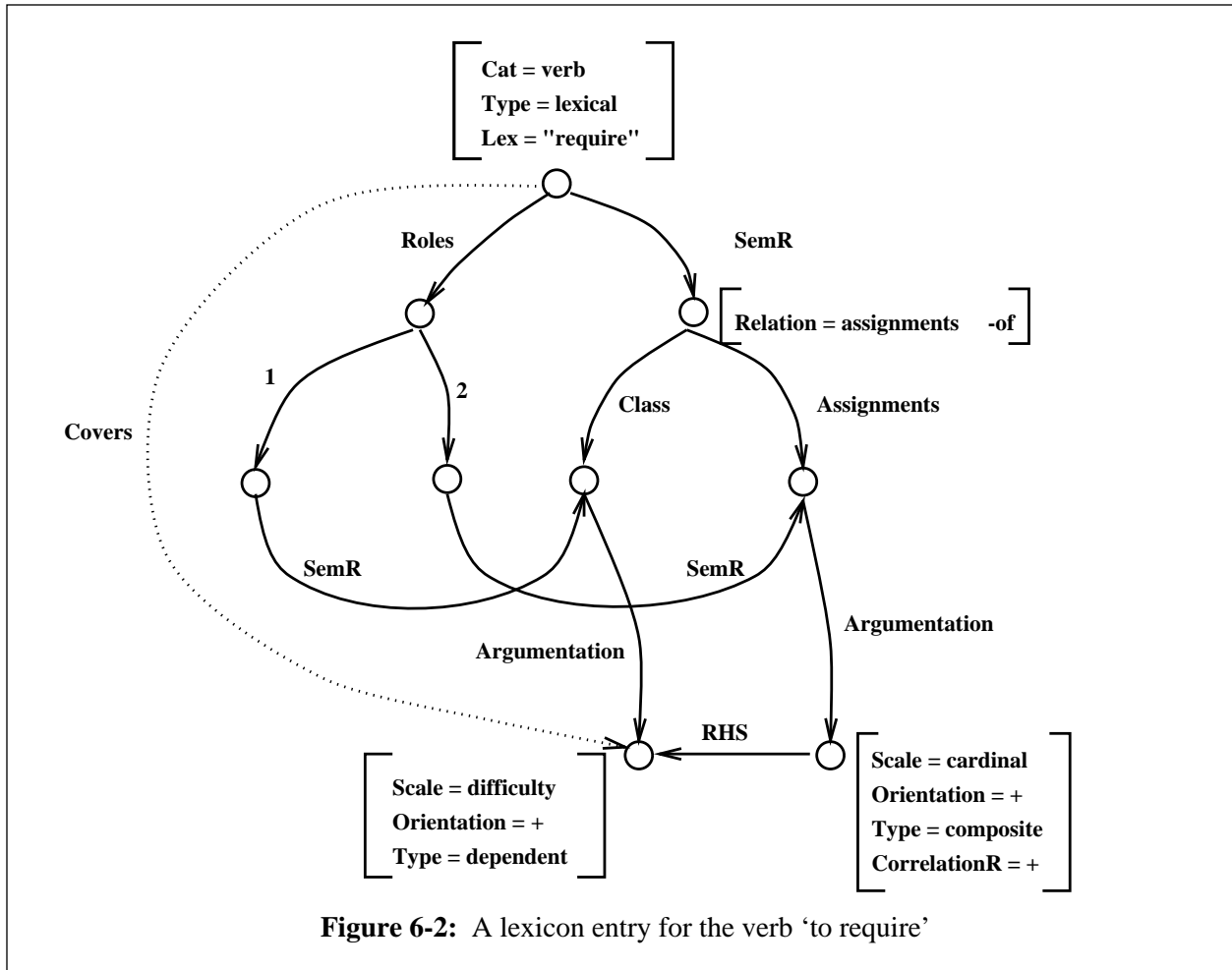
The other addition is that both of the semantic arguments of the relation assignments-of are now the object of a composite evaluation, of the form /the more assignments, the more difficult/. This evaluation is a composite evaluation, which we write down as:

```
<cardinal(assignments), + <difficult(AI), +>>
```

The argumentation feature of the assignments argument specifies that it is evaluated on the cardinality scale with orientation +. In addition, the feature also points to the RHS (right-hand side) of the composite evaluation, the embedded evaluation of AI which determines the value of this composite evaluation. The (CorrelationR +) feature indicates that the composite evaluation is of the form /+, +/ or /-, -/.

The lexical entry for “require” indicates that the choice of the verb covers the argumentative evaluation of the semR of the complement 1. This relation is indicated by the dashed line labeled covers in the figure. Because this evaluation is covered by the verb, no other linguistic device is needed to convey it. In contrast, the evaluation of the assignments argument still needs to be covered after the verb has been selected. This can happen when the complement 2 is lexicalized, by choosing for example a determiner like in “many assignments”. The covers links constitute the principal technique by which linguistic decisions are related to semantic constraints. The same links are used throughout the grammar and lexicon and account for many decisions made by the generator.

In summary, this section has shown how the use of argumentative features can help in the task of lexical choice by accounting for the difference between lexical entries that differ by argumentative con-



notation while covering the same semantic content (choice between ‘include’ and ‘require’).

7. Conclusion

We have shown in this paper how the theory of argumentation in language helps in the task of text generation. Our approach relies on a view of language where all decisions, from the high-level choice of speech-acts to the most detailed lexical distinctions can be related to the intentions of the speaker, a view very close to the model advocated in [18, p.534] and underlying the theory of argumentation in language. We have proposed an architecture where the same abstract formalism can be used throughout the generation process. As a consequence, the same intention can be satisfied by devices at different levels. For example, to convince the hearer that AI is difficult, all the following decisions can be made:

- Include the information that AI has many assignments. (*a content determination decision*).
- Express the relation between number of assignments and difficulty through a rhetorical relation such as “AI has many assignments, so it is quite difficult”. (*a content organization decision*)
- Choose a verb conveying an evaluation of the AI class like “require” in *AI requires many assignments*. (*a lexicalization decisions*)

All these decisions are motivated by the same original speaker intention and can interact with one another.

We have shown how the theory of argumentation in language and the descriptive tool of topoi can provide the uniform representation device needed to implement such an architecture. Topoi effectively serve as a bridge between the conceptual decisions and the linguistic decisions made by the generator. Topoi have been found to be abstract and expressive enough to be used for content determination, content organization and lexicalization.

In our work, we have ignored some of the more complex issues related to the expression of argumentation. In particular:

- We do not deal with the notion of “argumentative force” or “degree”. As a consequence, we do not distinguish between expressions varying only in degree, such as “*difficult*” vs. “*very/extremely/immensely difficult*” or “*some/many/most*”.
- We have ignored the effects of negation on argumentation, negative contexts and effects like hedges and understatements.
- We do not deal with the complex relation between different perspectives and argumentation, aspects related to the theory of *polyphony* [8]. Such effects appear in sentences like “*of course, it would be easy for you*” with the complex planning involved in presenting an image of the hearer.

Acknowledgments

The research reported in this article was conducted while the author was at Columbia University, NY. The work was partially supported by grants from the National Science Foundation under Grant IRT-84-51438, the New York State Center for Advanced Technology under Contract NYSSTF-CAT(88)-5, the Office of Naval Research under Contracts N00014-89-J-1782, DARPA under contract N00039-84-C-0165, GE and Bellcore. I want to thank Kathleen R. McKeown, Jacques Robin and Frank Smadja for their help in this work.

References

1. Anscombe, J.C. and O., Ducrot. *Philosophie et langage*. Volume :*L'argumentation dans la langue*. Pierre Mardaga, Bruxelles, 1983.
2. Brachman, R.J. and Resnick, L.A. and Borgida, A. and McGuiness, D.L. and Patel-Schneider, R.F. Living with CLASSIC: When and how to use a KL-ONE-like language. In *Principles of semantic networks*, Sowa, J., Ed., Morgan Kaufman Publishers, 1990.
3. Bruxelles, S. and Raccach P.Y. Argumentation et Semantique: le parti-pris du lexique. In *Actes du Colloque 'Enonciation et parti-pris'*, Forthcoming, 1991.
4. Bruxelles, S., Carcagno, D. and Fournier, C. "Vers une construction automatique des topoi a partir du lexique". *CC AI - Journal for the integrated study of Artificial Intelligence cognitive science and applied epistemology* 6, 4 (1989), 309-328.
5. Buchberger, E. and H. Horacek. VIE-GEN: a generator for German texts. In *Natural Language Generation Systems*, McDonald, D. and L. Bloc, Ed., Springer-Verlag, 1988, pp. 166-204.
6. Danlos, L. *The Linguistic Basis of Text Generation*. Studies in Natural Language Processing. Cambridge University Press, Cambridge, England, 1987.
7. Dispaux, G. *La Logique et le Quotidien: Une Analyse Dialogique des Mechanismes d'Argumentation*. Les Editions de Minuit, Paris, 1984.
8. Ducrot, O. *Le sens commun*. Volume :*Le dire et le dit*. Les editions de Minuit, Paris, 1983.
9. Ducrot, O. *et al. Le sens commun*. Volume :*Les mots du discours*. Les editions de Minuit, Paris, 1980.
10. Elhadad, M. Types in Functional Unification Grammars. Proceedings of 28th Meeting of the ACL (ACL'90), Pittsburgh, 1990, pp. 157-164.
11. Elhadad, M. FUF User Manual - Version 5.0. (Tech. Rep. CUCS-038-91) CUCS-038-91, Columbia University - Computer Science, 1991.
12. Elhadad, M. Generating Adjectives to Express the Speaker's Argumentative Intent. Proceedings of 9th National Conference on Artificial Intelligence (AAAI 91), Anaheim, 1991, pp. 98-104.
13. Elhadad, Michael. *Using Argumentation to Control Lexical Choice: a Functional Unification-based approach*. Ph.D. Th., Columbia University, Department of Computer Science, New York, NY 10027, 1992.
14. Elhadad, M. Generating Argumentative Paragraphs. Proceedings of COLING-92, Nantes, France, July, 1992.
15. Elhadad, M. Generating Argumentative Judgment Determiners. Proceedings of 11th National Conference on Artificial Intelligence (AAAI 93), Washington DC, 1993.
16. Elhadad, M. and K.R. McKeown. Generating Connectives. Proceedings of COLING'90 (Volume 3), Helsinki, Finland, 1990, pp. 97-101.
17. Elhadad, M. & Robin, J. Controlling Content Realization with Functional Unification Grammars. In *Aspects of Automated Natural Language Generation*, R. Dale, E. Hovy, D. Roesner and O. Stock, Ed., Springer Verlag, 1992, pp. 89-104.
18. Fillmore, C.J, Kay, P. and O'Connor C. "Regularity and idiomaticity in grammatical constructions: the case of *let alone*". *Language* 64, 3 (1988).
19. Goldman, N.M. Conceptual generation. In *Conceptual Information Processing*, Schank, R.C., Ed., North Holland, Amsterdam, 1975.
20. Hirschberg, J.B. *A Theory of Scalar Implicature*. Ph.D. Th., University of Pennsylvania, 1985.

21. Horacek, Helmut. An Integrated View of Text Planning. In *Proceedings of the Sixth International Workshop on Natural Language Generation, Trento, Italy*, R. Dale, E. Hovy, D. Roesner and O. Stock, Ed., Springer Verlag, 1992. To appear.
22. Hovy, E.H. Planning Coherent Multisentential Text. Proceedings of the 26th ACL Conference, ACL, Buffalo, June, 1988, pp. P.163-169.
23. Hovy, E.H. *Generating natural language under pragmatic constraints*. L. Erlbaum Associates, Hillsdale, N.J., 1988. Based on the author's thesis (doctoral--Yale University, 1987).
24. Hovy, E. Unresolved issues in paragraph planning. In *Current Research in Natural Language Generation*, Dale, R., Mellish, C.S. and Zock, M., Ed., Academic Press, 1990, pp. 17-45.
25. Hovy, E. Approaches to the planning of coherent text. In *Natural Language Generation in Artificial Intelligence and Computational Linguistic*, Paris, C.L., Swartout, W.R. and Mann, W.C., Ed., Kluwer Academic Publishers, 1991, pp. 83-102.
26. Jacobs, P.S. "Knowledge-intensive natural language generation". *Artificial Intelligence* 33 (1987), 325-378.
27. Kerbrat-orecchioni, C. *La connotation*. Presses Universitaires de Lyon, Lyon, 1977.
28. Klein, E. "A semantics for positive and comparative adjectives". *Linguistics and Philosophy* 4, 1 (1980), 1-45.
29. Ludlow, P. "Implicit comparison classes". *Linguistics and Philosophy* 12 (1989), 519-533.
30. Mann, W.C. and S.A. Thompson. Rhetorical Structure Theory: Description and Construction of Text Structures. In *Natural Language Generation*, Gerard Kempen, Ed., Martinus Nijhoff, 1987, pp. 85-96.
31. McKeown, K.R. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Studies in Natural Language Processing. Cambridge University Press, Cambridge, England, 1985.
32. McKeown, K. and M. Elhadad. A Contrastive Evaluation of Functional Unification Grammar for Surface Language Generators: A Case Study in Choice of Connectives. In *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, Cecile L. Paris, William R. Swartout and William C. Mann, Eds., Kluwer Academic Publishers, 1991, pp. 351-396.
33. Moore, J.D. and C.L. Paris. Planning Text for Advisory Dialogues. Proceeding 27th ACL, ACL, Vancouver, BC, June, 1989, pp. 203-211.
34. Moore, J.D. & C.L. Paris. Discourse Structure for Explanatory Dialogues. AAAI Spring Symposium on Discourse Structure, AAAI, 1991.
35. Osgood, C.E., Suci, G.S. and Tannenbaum, P.H. *The measurement of meaning*. University of Illinois Press, Urbana, 1957.
36. Paris, C.L. *The Use of Explicit User models in Text Generation: Tailoring to a User's level of expertise*. Ph.D. Th., Columbia University, 1987.
37. Pollard, C. and I.A. Sag. *CSLI Lecture Notes*. Volume 13: *Information-based Syntax and Semantics - Volume 1*. University of Chicago Press, Chicago, IL, 1987.
38. Raccah, P.Y. "Modelling argumentation and modelling with argumentation". *Argumentation* (1987).
39. Rayner, M. and A. Banks. "An implementable semantics for comparative constructions". *Computational Linguistics* 16, 2 (June 1990), 86-112.
40. Resnick, L.A. and Borgida, A. and Brachman, R.J. and McGuiness, D.L. and Patel-Schneider R.F. *CLASSIC: description and reference manual for the COMMON LISP implementation version 1.02*. 1990.

41. Robin, J. Lexical Choice in Natural Language Generation. Tech. Rept. CUCS-040-90, Columbia University, 1990.
42. Sidner, C.L. *Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse*. Ph.D. Th., MIT, 1979.

Table of Contents

1. Introduction	1
2. The Domain Problem: Generating Advice-Giving Paragraphs	2
3. Previous Work: A Brief Review of the Theory of Argumentation	4
3.1. Argumentation in Language	4
3.2. Scalar Lexical Semantics and Scalar Connotations	5
4. The Role of Topoi in Content Determination	7
4.1. Three Types of Argumentative Evaluation	7
4.2. Evaluation Functions: Producing Evaluations from Observations	9
4.3. Knowledge Sources and their Acquisition	12
5. The Role of Topoi in Content Organization	13
5.1. The Task of Content Organization	13
5.2. Previous Work on Content Organization	15
5.3. Argumentation and Content Organization	16
6. The Role of Topoi in Lexical Choice	18
6.1. The Task of Lexical Choice	18
6.2. Using Argumentation for Lexical Choice	18
7. Conclusion	21
Acknowledgments	22