Practical Issues in Automatic Documentation Generation

Kathleen McKeown
450 Computer Science Building
Columbia University
New York, NY 10027
`kathy@cs.columbia.edu`

Karen Kukich
Bell Communication Research
Morristown, NJ 07960-6438
`kukich@bellcore.com`

James Shaw
450 Computer Science Building
Columbia University
New York, NY 10027
`shaw@cs.columbia.edu`

Abstract

PLANDOC, a system under joint development by Columbia and Bellcore, documents the activity of planning engineers as they study telephone routes. It takes as input a trace of the engineer's interaction with a network planning tool and produces 1-2 page summary. In this paper, we describe the user needs analysis we performed and how it influenced the development of PLANDOC. In particular, we show how it pinpointed the need for a sublanguage specification, allowing us to identify input messages and to characterize the different sentence paraphrases for realizing them. We focus on the systematic use of conjunction in combination with paraphrase that we developed for PLANDOC, which allows for the generation of summaries that are both concise–avoiding repetition of similar information, and fluent–avoiding repetition of similar phrasing.

1 Motivation

In a collaborative effort between academics and industry, we have embarked on a project that uses text generation research in service of an industrial application. Bellcore and Columbia are jointly developing a system, PLANDOC, that will document the activity of planning engineers as they study telephone routes[1]. Tele-

phone planning engineers currently use a software tool, the Bellcore LEIS[2]-PLAN system, that helps them derive 20-year capacity expansion plans based on growth forecasts and economic constraints. PLANDOC takes as input a trace of the engineer's interaction with LEIS-PLAN and produces a 1-2 page summary. The PLANDOC prototype is currently being tested by development teams and will move into use by regional planners sometime this Fall.

The role of documentation has gained increasing importance as businesses attempt to achieve higher levels of productivity, often with fewer employees. In such environments, work must be carefully documented, both to make previous business decisions readily available to current employees, and to provide management with information needed to authorize major expenditures, in the million dollar range. Network planning managers need justification for why a proposed plan is best and whether alternatives were investigated. Until recently this information was provided orally, if at all, due to time constraints. But internal auditors and public regulators have increased the demand for formal documentation. Indeed, lawsuits have made the lack of documentation extremely costly. In a recent settlement, Pacific Bell promised to provide increased documentation in lieu of an 80 million dollar rebate to rate payers. PLAN-Doc documentation also promises to be useful in training new planning engineers; it provides

---

[1] PLANDOC is being developed collaboratively by Karen Kukich and Neal Morgan of Bellcore and Kathy McKeown, James Shaw, Jacques Robin, and Jong Lim of Columbia University.

[2] LEIS is a registered trademark of Bell Communications Research, Piscataway, NJ.

a record of how experienced planning engineers arrive at their decisions, information which is not currently available.

Because telephone network planning is currently done with an automated software system that produces a trace, albeit cryptic, of the actions of both the system and the user, development of an automated documentation system is quite practical; input to a report generator is automatically produced and readily available. Our approach makes use of existing text generation tools; we adopted the FUF/SURGE package(FUF5; Elhadad 93), developed and widely used at Columbia (Robin 93; McKeown et al. 90; McKeown & Feiner 90; Elhadad 93; Paris 87; Wolz 92), which handles the generation of individual sentences. Given the PLAN trace and the FUF/SURGE sentence generation tools, development of PLANDOC requires bridging the gap between the two. The main research problems include:

- organizing the content of the report, i.e., content planning,

- mapping facts in the trace to sentence structures and choosing appropriate words, i.e., lexicalization.

To handle these appropriately, we performed a user needs analysis to gather details about the kinds of reports that users would find helpful. Our analysis revealed two overriding practical considerations for the design and implementation of the PLANDOC automatic documentation generator:

- the need for user-centered design, and

- the need for a bounded sublanguage.

The first of these was motivated by the fact that the system would eventually be used in a live production setting. The second was mandated by the need for a concise, but fluent report. The analysis showed that reports must avoid repeating similar information which occurs across input facts, while at the same time avoiding repeating exact phrasing.

In this paper, we show how PLANDOC uses a systematic combination of conjunction and paraphrasing power to achieve these goals. Further, we show how we bounded their different combinations to avoid a combinatoric explosion of possible phrasings, while still maintaining fluency and conciseness in the generated reports. The systematic use of conjunction and ellipsis to achieve conciseness, combined with paraphrasing power, is a unique feature of the PLANDOC system.

In the following sections, we first describe the user needs analysis, then turn to a description of the sublanguage and the constrained use of conjunction and paraphrasing. We close with a discussion of our current directions.

2 User-Centered Design

User-needs analysis is a common practice in the development of computer-human interface systems and other end-user software. Particularly in developing a large scale, practical system, the needs of the user must be studied if the resulting system is to be accepted and effectively used by the users. In this section, we describe the user-needs analysis and system development methodology that we are using in our ongoing development of PLANDOC.

Our analysis combined two complementary approaches. First, we interviewed a variety of different groups of people involved in the telephone network planning task. Our goal was to identify potential users of PLANDOC and to solicit their views on how such a system could be most helpful. Second, we collected a set of manually-written narratives to inform the development of the generator, providing insights on report form and content, vocabulary and sentence structure. In this section we describe how user interviews and corpus analysis shaped the design of the documentation generator. But first we provide some brief background information on the problem domain.

2.1 Problem Setting

Voice and data service is carried to telephone customers through a complex network of routes

consisting of copper or fiber cables supplemented by additional equipment such as Digital Loop Carrier (DLC) systems and fiber multiplexors. It is the telephone network planning engineer's job is to derive a capacity expansion (relief) plan specifying when, where, and how much new copper, fiber, multiplexing and other equipment to install in a route to avoid facilities exhaustion. This activity is an integral part of telephone operations. New installations are costly, but at the same time facilities exhaustion can lead to a disruption in telephone service. Currently, about 1,000 planning engineers in 8 regional and independent telephone companies produce a total of about 15,000 route studies per year.

The engineer uses PLAN to compute an optimum, cost-effective base relief plan needed to meet forecast demand over the next twenty years. The base plan, however, may not always be realizable or desirable due to political, economical, practical and other factors known to the engineer but not to the computer. The engineer uses PLAN's Interactive Refinement Module that allows 'what-if' modeling to explore the effects of various changes to the base plan. For example, an engineer might explore requesting a DLC activation for a given site, or changing a fiber activation time. After comparing the effects of different refinement scenarios, the engineer ultimately decides on a realizable relief plan to recommend to management for project authorization.

Overall interaction with PLAN thus includes an automatically generated base plan, a sequence of refinements to explore the effects of different changes to the base, and a final proposed plan which may include elements of the base plan along with a selected set of refinements.

2.2 Interviews

With the help of Bellcore Planning and Engineering staff[3] we formulated an initial proposal for PLANDOC and drafted preliminary target narratives. We then conducted a series of interviews with planning engineers, managers, auditors and PLAN support staff from several regional telephone companies in their home offices and at two PLAN training courses[4]. The work experience of the engineers we interviewed ranged from beginner to expert. Our goal was to determine how engineers actually used the PLAN system, whether they would find an automated documentation facility to be helpful, and, if so, what the form and content of the narratives should be.

We learned that novice planners often run 'bozo' refinements just to develop a feel for the process, while experienced planners sometimes run refinements they know will be suboptimal just for the record, i.e., for the benefit of managers, auditors and regulators who might ask "did you try such and such?". More critical to the need for documentation, we also learned that experienced planners keep handwritten notes on paper, listing their refinements and why they tried them; they asked for a way to enter their notes on-line to keep track of their reasoning. Inexperienced planners asked to see narratives written by experienced planners in order to learn from them; unfortunately few such narratives exist. Finally, all planners welcomed the idea of having the computer generate narratives that they could include in their documentation packages, especially if they could add to the narratives themselves.

These findings shaped the content of PLAN-DOC narratives and the design of the system. Specifically, they indicated that planners may not want all refinements that they tried to appear in the narrative. For example, novice planners do not want to include their 'bozo' refinements, while experienced planners do want to include the suboptimal refinements they ran to show that their final refinements were superior. Thus, PLANDOC includes a facility that lets the planner select a subset of refinements to be

---

PART 1 Route Input Data Summary (Tabular)

PART 2 Narrative (Textual)

- Base Plan Summary
- Refinements Summary with Engineering Notes
- Proposed Plan Summary

Figure 1: PLANDoc Report Format

---

included in the final narrative. Planners made it clear that they use knowledge not included in PLAN to make their decisions (e.g., corporate strategies) and they wanted a way to record that knowledge on-line, while they were working. This gave rise to PLANDOC's facility to prompt for manually-written engineer's notes at crucial points. We instituted only two user-visible changes to PLAN's original, successful interface, one to prompt for engineering notes and another to allow the engineer to request a narrative and select a subset of refinements to be included. Both options are presented using familiar PLAN interface commands and screen formats. Reports are generated off-line.

2.3 Corpus Analysis

We also arranged for an experienced retired planning engineer, Jim Phillips, who is also a PLAN expert, to write a corpus of target narratives based on PLAN runs of actual routes. Based on the findings from our interviews and on the target narratives, we arrived at the report format shown in Figure 1. It consists of two parts, a tabular summary of route input data and a narrative that integrates machine-generated text with the engineer's manually-entered notes.

Our corpus of target narratives provided information on what should be included in the report and its overall structure. Thus, it directly influenced development of both the Lexicalizer and Content Planner modules of PLANDOC. An analysis of PLAN's menu of refinement actions and the sentences in the target narratives

allowed us to specify a set of 31 different possible message types for refinement sentences including, for example, fiber extensions to CSAs (Carrier Serving Areas), or DLC (Digital Loop Carrier) equipment activations or denials for CSAs.

We then systematically categorized the sentences in our corpus to reveal all the different phrasings for each message type. This categorization showed that there was tremendous variety in the possible sentences for each message type with respect to sentence structure and lexical choice. Indeed, our first implementation of PLANDOC's sentence generator[5], resulted in more than 150 paraphrases for some message classes.

The target narratives also informed the design of PLANDOC's Content Planner. Our analysis revealed that choosing a specific paraphrase for use in a summary depends on what has already been mentioned (i.e., the choice is based on previous discourse). Furthermore, the narratives provided examples of how multiple messages were frequently combined to form complex compound sentences. In order to avoid a combinatorial explosion from combining many different sentences forms, we needed to specify a bounded sublanguage for PLAN's domain that ensured the sentence variety needed to maintain discourse coherence and fluency while enabling the construction of complex sentences. Before discussing this problem, we provide an actual sample of some PLANDOC output in Figure 2.

2.4 Sample PLANDoc Output

At present, the tabular Input Summary generator[6] and the textual Refinements Summary generator of the PLANDOC system are fully implemented. Figure 2 is an abbreviated sample of a Refinements Summary generated by PLANDOC. The incorporated Engineering Notes were entered manually by the Planning Engineer at run time and automatically integrated into the narrative by PLANDOC.

3 Sublanguage Specification

In this section we first provide a brief overview of PLANDoc's architecture and functioning. We then illustrate the large number of possible sentence combinations, describe the sub-

---

[5]written in FUF by J. Lim
[6]written in C by N. Morgan

```
RUNID reg1 DLC 5/7/93 act yes
CSU 3122 idlc272 idlc272
SAT 3122 3 1994 3 1994
SAT 3130 3 1994 3 1994
SAT 3134 3 1994 3 1994
SAT 3208 3 1994 3 1994
SAT 3420 3 1994 3 1994
END. 2110.1 1064.0
```

Figure 4: A portion of tracking file

```
((cat message)
 (admin ((PLANDoc-message-name RDA)
         (track-tag SAT)
         (seq-num 3)
         (runid r-reg1)
         (prev-runid BASE)
         (status act)
         (saved yes)))
 (class refinement)
 (ref-type DLC)
 (action activation)
 (equipment-type  all-dlc)
 (csa-site 3122)
 (date ((year 1994) (quarter 3))))
```

Figure 5: Output of the Message Generator

RUNID: REG1
Run-ID REG1 started at the BASE plan. This saved refinement activated DLC for CSAs 3122, 3130, 3134, 3208 and 3420 in the third quarter of 1994. It demanded that PLAN use DLC system IDLC272 for all placements in CSA 3122. The 20 year PWE was $2110.1K, a $198.6K savings over the BASE plan and the 5 year IFC was $1064.0K, a $64.5K penalty over the BASE plan.

Engineer's note:
These CSA's are beyond 28 kf and need range extenders to provide service on copper. Moving them to 1994 will negate a job adding a reg bay to the office.
...

RUNID: 3234-2
This saved refinement included all DLC changes in Run-ID REG1E. It requested the activation of DLC for CSA 3234 in the second quarter of 1994 and for CSA 3233 in the fourth quarter of 1994. DLC systems DLC96SS and DLC96M2 were used for all placements in CSAs 3233 and 3234. For this refinement, the 20 year route PWE was $1925.3K, a $383.4K savings over the BASE plan and the 5 year IFC was $833.9K, a $165.6K savings over the BASE plan.

Engineer's note:
I didn't need to demand the activation of these systems in the refinement as they were activated at this time in the BASE plan. The 'idlc272' was demanded because of the high demand. The non-integrated systems in CSA 3234 because it is a business area.
...

Figure 2: PLANDoc Refinements Summary

language specification solution and PLANDoc's paraphrasing and conjunction capabilities.

3.1 PLANDoc System Overview

PLANDoc's architecture, which is shown in Figure 3, draws on our previous text generation and report generation work (McKeown 85; Kukich 83).

The PLANDoc system consists of five sequential modules: a Message Generator, an Ontologizer, a Content Planner, a Lexicalizer, and a Surface Generator. Since PLAN itself is implemented in C and PLANDoc's text generation modules are implemented in Lisp, a Message Generator module[7] serves as an interface between PLAN and PLANDoc. Input to the Message Generator comes from PLAN tracking files which record the user's actions during a planning session. Figure 4 is a portion of a
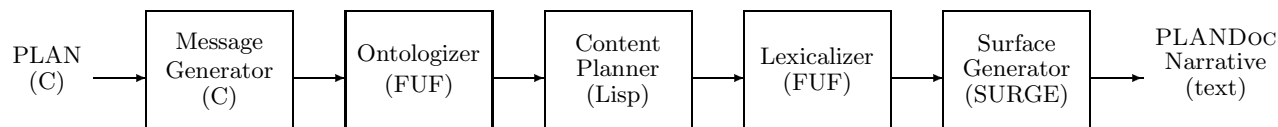
---
[7]written in C by N. Morgan

Figure 3: PLANDOC **System Architecture**

tracking file; it corresponds to the paragraph labeled RUNID REG1 in the sample PLANDOC narrative above. Shown below it (Figure 5) is a Lisp representation, ie., a message in attribute-value format, for one refinement action in the tracking file produced by the Message Generator. Output messages are first passed to an Ontologizer[8]. The complete set of enriched messages is then passed to a Content Planner[9] whose job is to determine which information in the messages should appear in the various paragraphs and to organize the overall narrative. This involves combining individual messages to produce the input for complex sentences, choosing cue words and determining paraphrasing forms that maintain focus and ensure coherence. The output of the Content Planner is a 'condensed' set of complex messages, each still in hierarchical attribute-value format.

We are using the FUF/SURGE package (FUF5; Elhadad 93; Kay 79; Halliday 85) for the Lexicalizer and Surface Generator modules of PLANDOC. We used FUF to write a lexicalization grammar for PLAN's sublanguage. The task of the Lexicalizer module[10] is twofold: 1) to map the attributes of the messages into systemic/semantic case roles, such as agent, beneficiary, process, circumstance, etc., and 2) to select content words to express the values of the attributes, all the while maintaining constraints imposed by the Content Planner. Finally, the FUF/SURGE Surface Generator takes the lexicalized messages, maps case roles into syntactic roles, builds the constituent structure of the sentence, fills in function words such as pronouns,

prepositions, conjunctions, etc., ensures agreement, and ultimately realizes the structure as a linear surface sentence.

3.2 Combinatorial Explosion

Two of the most salient characteristics of the text in our corpus are the great degree of paraphrasing found and the frequent use of conjunction and ellipsis. Both characteristics arise from the fact that the domain of discourse is limited to 31 message types, but user interactions include many variations and combinations of those message types. Paraphrasing is used to avoid repetition and to maintain focus; conjunction and ellipsis are used to combine messages with similar attributes to form concise summary sentences. While the number of paraphrase combinations actually occurring in the target narratives was small, the different combinations the user might invoke was beyond our control and potentially quite large.

The scope of naturally occurring paraphrasing is illustrated by the sentences derived for one message class in terms of their mapping of semantic attributes to lexical roles[11] (such as *agent, beneficiary, location, etc.*) and syntactic roles (such as *subject, direct object, object of preposition, etc.*) It is the job of the PLAN-Doc lexicalizer to chose lexical roles for semantic attributes; the SURGE surface generator then maps lexical roles into syntactic roles.

The main semantic attributes of the fiber-service-extension message are:

```
(class refinement)
(ref-type fiber)
(action service-extension)
```

---

[8]The Ontologizer simply enriches each message with semantic knowledge from PLAN's domain of discourse.
[9]written in Lisp by J. Robin and J. Shaw
[10]written in FUF by J. Shaw with input from J. Lim. J. Robin, M. Elhadad, D. Radev and D. Horowitz

[11]Lexical roles are often referred to as semantic roles of a sentence, where sentence semantic roles are distinct from domain semantic attributes. We use "lexical roles" to avoid confusion.

```
(extension-type T-1)
(from-fiber-hub 2113)
(to-csa 2115)
```

Some of the paraphrases derived from our corpus for this message are:

1. "This refinement extended T-1 service from fiber hub 2113 to CSA 2115."

2. "This refinement demanded that PLAN extend T-1 service from fiber hub 2113 to CSA 2115."

3. "This refinement called for PLAN to extend T-1 service from fiber hub 2113 to CSA 2115."

4. "This refinement requested a T-1 service extension from fiber hub 2113 to CSA 2115."

5. "This refinement called for a T-1 service extension from fiber hub 2113 to CSA 2115."

6. "This refinement served CSA 2115 by T-1 extension from fiber hub 2113."

7. "This refinement demanded that PLAN serve CSA 2115 by T-1 extension from fiber hub 2113."

8. "This refinement called for PLAN to serve CSA 2115 by T-1 extension from fiber hub 2113."

9. "This refinement demanded service to CSA 2115 by T-1 extension from fiber hub 2113."

10. "This refinement called for service to CSA 2115 by T-1 extension from fiber hub 2113."

Note that the lexical and syntactic roles filled by the semantic attributes in the message vary across paraphrases. For example, although the semantic attribute *to-csa* is most often realized in the lexical role *location* which gets mapped to the syntactic role *object of preposition* (e.g., 1, 2, 3), in some paraphrases (e.g., 6, 7, 8) it appears in the lexical role *beneficiary* which gets mapped to the syntactic role *direct object*. More dramatically, two main lexical variants occur for the semantic attribute *action*, namely the head verbs 'extend' and 'serve'. These in turn give rise to a variety of syntactic constructions, e.g., simple sentences, nominalizations of the head verbs in participial clauses, infinitive clauses, etc. Since passive is sometimes needed to maintain focus or coherence within a paragraph(McKeown 85), the number of possible paraphrases doubles.

When paraphrasing is combined with conjunction, the problems compound. Complex

1) "This refinement activated DLC for CSAs 2111, 2112, 2113, 2114, 2115 and 2116 in 1996 Q1."

2) "This refinement activated DLC for CSA 2111 in 1995 Q3, for CSAs 2112 and 2113 in 1995 Q4, and for CSAs 2114, 2115 and 2116 in 1996 Q1."

3) "It requested the placement of a 48-fiber cable from the CO to section 1103 and the placement of 24-fiber cables from section 1201 to section 1301 and from section 2201 to section 2301 in the second quarter of 1995."

Figure 6: Conjunction Examples

messages arise because it is often necessary to combine multiple messages with some common and other distinct attributes into a single message in order to avoid repeating similar information. For example, if a user activates six CSA sites for DLC in one refinement scenario, those six messages, with four common attributes and one distinct attribute, *csa-site*, can be expressed succinctly using conjunction and ellipsis (example 1 Figure 6). Messages with two distinct attributes can also be easily conjoined depending on where in the sentence they occur (example 2). A group of messages with more than two distinct attributes results in a complex compound sentence (example 3).

Each of PLANDoc's 31 message types has five or more semantic features; six of those 31 message types are stand-alone messages; all of the remaining 25 messages can be combined to form compound messages with at least one distinct feature, half of those with at least two distinct features, and a few with three or four distinct features. Recall that there were at least ten active and ten passive sentence forms for the fiber-service-extension message, which is typical of most of the 31 message types. Given the number of possible message combinations multiplied by the number of possible paraphrases for each message, the need to limit the paraphrasing power of the PLANDoc generator should be clear.

3.3 Sublanguage Solution
Since many of the naturally occurring paraphrases involved minor variations in syntax or

substitution of synonyms that formed valid collocations in some contexts but awkward phrases in others, we chose to constrain PLANDoc's paraphrasing power to the following four active sentence forms for most of the 31 message types and their four corresponding passive forms:

1. **simple sentence**: "This refinement <verb-ed> <object-np>."

2. **nominalization**: "This refinement requested the <action-nominalization> of <object-np>."

3. **participial**: "This refinement demanded that PLAN <verb> <object-np>."

4. **infinitive**: "This refinement called for PLAN to <verb> <object-np>."

So, for example, the active and passive nominalization forms of the fiber-activation message are:

- "This refinement requested the activation of fiber for CSA 2115 in 1996 Q1."

- "The activation of fiber for CSA 2115 in 1996 Q1 was requested."

Recall that the job of the PLANDoc Lexicalizer is to manage the mapping of semantic attributes to lexical roles for all possible combinations of common and distinct attributes in compound and complex messages. Constraining the sublanguage to at most eight paraphrases greatly reduces the complexity of that mapping. It also eliminates the need to specify a complex set of collocation constraints for synonym substitutions. At the same time, eight potential paraphrases provide enough flexibility for the Lexicalizer to make choices that maintain focus and coherence and that avoid repetition. Similar sublanguage specifications related to the use of names, pronouns and deictic expressions for subsequent references, modifier constructions for noun phrases (e.g., "This saved DLC refinement ..."), and discourse cue words (e.g., "also, finally", etc.), provide the same manageability and flexibility benefits.

3.4 Conjunction and Paraphrasing

Determining when conjunction is to be used and what type of paraphrasing is required are both handled by the Content Planner. The Content Planner is given as input a list of messages which form the full content of the report. Its task is to use knowledge of the overall semantic content to determine how to order messages and where to form sentence boundaries. While it could generate a separate sentence for each input message, a common solution in many language generators, this would result in a verbose and repetitive report. In order to avoid repeating similar information, PLANDoc uses conjunction, grouping together semantically related attributes, to control how messages are ordered in the report and to form sentence boundaries. Note that this approach to content planning, relying on opportunistic grouping of information based on how it can be realized in concise linguistic form, is quite different from other systems which tend to use either rhetorical (McKeown 85; Moore & Paris; Hovy 91; Wahlster et al. 89) or domain dependent(Paris 87; Rambow & Korelsky 92) strategies to order information.

To do this, the Content Planner first groups together related messages and tries to find those with the maximum number of common attributes. It groups these by common action and within this, by common date. When all but one or two attributes are common, ellipsis can be used for every common attribute, resulting in a concise form that uses a list-like structure for one or two roles of the sentence. To generate this form, the Content Planner builds a message where one semantic role has as its value a list and the Lexicalizer selects conjunction for the lexical role. Examples 1 and 2 in Figure 6 illustrate these cases.

However, the more messages that are grouped together, the greater the number of potentially distinct attributes. PLANDoc groups such long compound messages into several separate sentences, where each sentence has a different common partition. It then combines these compound sentences together into a single conjunction. Example 3, Figure 6, illustrates this case. To generate these complex forms, the Content Planner indicates for each message which attributes are common and which are distinct. It then indicates which common attributes should be gapped; depending on the attribute and its position, sometimes only the first reference is

1. "This refinement used a cutover strategy of ALL for CSAs 1111, 1112 and 1113, of MIN for CSAs 2221 and 2222 and of GROWTH for CSA 3331."

2. *"A cutover strategy of ALL was used for CSAs 1111, 1112 and 1113, of MIN for CSAs 2221 and 2222 and of GROWTH for CSA 3331."

Figure 7: Paraphrasing and Conjunction

ungapped, while in other cases all but the last is gapped. SURGE generates the full sentence for each message, but suppresses the gapped constituents when linearizing the syntactic tree representing the sentence. While this approach is less efficient, it is highly general since it can handle any combination of attributes without specifically anticipating it.

Conjunction and ellipsis cannot be generated blindly, however. When conjunction is used for certain paraphrases, ambiguity and/or invalid sentences can result. The examples in Figure 7 show how conjunction using one paraphrase form (active with verb "use") is appropriate for conjunction with two distinct attributes (cutover strategy and CSA site), but a passive paraphrase for the same input produces an infelicitous result. This is because one distinct attribute occurs to the left of the verb ("ALL" in the first clause) and the other (CSA site) to the right of the verb. Unless no ellipsis at all is used (in which case there is no point in using conjunction), it is impossible to generate a reasonable sentence. Thus, while we have implemented a general algorithm, there are still cases that are exceptions to our approach. By limiting paraphrasing we have also limited the number of these cases to a manageable amount.

4   Related Work

Other natural language text generation systems designed to summarize quantitative data include: ANA (Kukich 83), SEMTEX (Roesner 87), LFS (Iordanskaja et al. 92), GOSSIP (Iordanskaja et al. 91), STREAK (Robin 93), and FoG (Bourbeau et al. 90). All were influenced by early work on sublanguage definition (Kittredge et al. 83). ANA, a stock market report generator, achieves a high degree of fluency for complex sentences by relying on a phrasal lexicon; SEMTEX and LFS each generate bilingual summaries of labor force statistics, French/English by the former, German/English by the latter; GOSSIP generates paragraph-length reports describing operating system usage using a semantic net formalism; STREAK generates basketball summaries, packing as much information into a single sentence as possible, using complex sentence structures such as multiple modifiers of a noun or verb, conjunction and ellipsis; FoG generates marine weather forecasts from meteorological data and remains to date the only generator in everyday industrial use. However, none of these systems make extensive use of conjunction and paraphrasing in a systematic way.

5   Future work

PLANDoc will move into actual use in Fall 1994. At this point, we will be able to fully evaluate how well its output meets user needs. Furthermore, we plan to augment the system so that it can produce summaries of both the base plan and the proposed plan. Of these, the proposed plan summary presents somewhat more of a challenge. It should be about a paragraph in length but succinctly summarize the recommendations made by the planning engineer. Thus, the system must work within tighter space constraints to include all information. A second problem for this summary is that it must include information from multiple sources. The proposed plan will include elements of the base plan as well as a subset of the refinements the engineer carried out. PLANDoc must determine how to integrate these different pieces of information, with emphasis on the resulting plan and less information on how it was derived. While we can use some of the same techniques currently used to make the refinements summary both more concise and more fluent (i.e., the combined use of conjunction and paraphrase), more research will be required in discourse plan-

ning and selection of textual focus.

6   Conclusion

PLANDoc demonstrates how text generation tools developed in a research environment are ready for commercial use. A fully implemented system, PLANDoc generates 1-2 page summaries of interactions between planning engineers and a developed software tool. In this paper, we have shown how PLANDoc uses a systematic combination of conjunction and paraphrase to avoid repetition both of information and of phrasing. The ability to systematically combine and group together related sentences in a wide variety of ways is a unique feature of our automated documentation system. Finally, through a user needs analysis we identified and implemented features to improve usability of the resulting system. In particular, by allowing engineers to add their own refinements notes and to modify system generated text, PLANDoc can also be viewed as an aid to documentation that will help engineers more quickly create needed justification of why increased expenditures are necessary.

References

Bourbeau, L. and Carcagno, D. and Goldberg, E. and Kittredge, R. and A. Polguere. 1990. Bilingual generation of weather forecastes in an operations environment. In *Proceedings of the 13th International Conference on Computational Linguistics*, COLING.

Dale, R. 1992. *Generating Referring Expressions.* ACL-MIT Press Series in Natural Language Processing, Cambridge, Ma.

Elhadad, Michael. 1991. FUF: The universal unifier - user manual, version 5.0. *Tech Report CUCS-038-91*, Columbia University.

Elhadad, Michael. 1993. Using argumentation to control lexical choice: a unification-based implementation. Ph.D. thesis, Computer Science Department, Columbia University.

Halliday, M.A.K. 1985. *An introduction to functional grammar.* Edward Arnold, London.

Hovy, Edward. 1991. Approaches to the planning of coherent text. In Paris, C. and Swartout, W. and Mann. W.C. (editors), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, Kluwer Academic Publishers.

Iordanskaja, L., Kittredge, R., and Polguere, A. 1991. Lexical Selection and Paraphrase in a Meaning-Text Generation Model. In Paris, C. and Swartout, W. and Mann. W.C. (editors), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, Kluwer Academic Publishers, pp. 293-312.

Iordanskaja, L., Kim, M., Kittredge, R., Lavoie, B. and Polguere, A. 1992. Generation of Extended Bilingual Statistical Reports. In *Proceedings of COLING-94*, COLING, pp. 1019-1023.

Kay, Martin. 1979. Functional Grammar In *Proceedings of the 5th Annual Meeting of the Berkeley Linguistic Society.*

Kittredge, Richard, and Lehrberger, John. 1983. *Sublanguages: Studies of Language in Restricted Semantic Domains*, Walter DeGruyter, New York.

Kukich, Karen. 1983. The design of a knowledge-based text generator. In *Proceedings of the 21st Conference of the Association for Computational Linguistics*, Massachusetts Institute of Technology, Cambridge, Mass., pp 145-150.

McKeown, K. R. 1985. *Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*, newblock Studies in Natural Language Processing series, Cambridge University Press.

McKeown, K., Elhadad, M., Fukumoto, Y., Lim, J., Lombardi, C., Robin, J., and Smadja, F.. 1990. Text generation in COMET In Dale, R. and Mellish, C.S. and Zock, M. (editors), *Current Research in Natural Language Generation.* Academic Press.

McKeown, K.R., and Feiner, S.. 1990. Interactive Multimedia Explanation for Equipment Maintenance and Repair. In *Proceedings of the DARPA Speech and Natural Language Workshop*, DARPA, Hidden Valley, Pa.

Moore, J.D. and Paris, C.L. 1989. Planning Text for Advisory Dialogues. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Vancouver, B.C., pp. 203-11.

Paris, C.L. 1987. The Use of Explicit User Models in Text Generation: Tailoring to a User's Level of Expertise. Columbia University.

Rambow, O. and Korelsky, T. 1992. Applied Text Generation. In *Proceedings of the 3rd Conference on Applied Natural Language Processing.* Association for Computational Linguistics, Trento, Italy, pp. 40-47.

Robin, Jacques. 1993. A Revision-Based Generation Architecture for Reporting Facts in their Historical Context. In Horacek, H. and Zock, M. (editors), *New Concepts in Natural Language Generation: Planning, Realization and Systems.* Frances Pinter, London and New York.

Roesner, D. 1987. SEMTEX: A Text Generator for German. In Gerard Kempen (editor), *Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics.* Martinus Ninjhoff Publishers, pp. 133-148.

Wahlster, W., Andre, E., Hecking, M., and T. Rist. 1989. *WIP: Knowledge-based Presentation of Information.* German Research Center for Artificial Intelligence, Saarbruecken, FRG.

Wolz, Ursula. 1992. Extending User Expertise in Interactive Environments Department of Computer Science, Columbia University.