A MODEL OF REVISION
IN NATURAL LANGUAGE GENERATION

Marie W. Meteer (Vaughan)
David D. McDonald

Department of Computer and Information Science
University of Massachusetts
Amherst, Massachusetts 01003

COINS Technical Report 86-52

ABSTRACT

We outline a model of generation with revision, focusing on improving textual coherence. We argue that high quality text is more easily produced by iteratively revising and regenerating, as people do, rather than by using an architecturally more complex single pass generator. As a general area of study, the revision process presents interesting problems: Recognition of flaws in text requires a descriptive theory of what constitutes well written prose and a parser which can build a representation in those terms. Improving text requires associating flaws with strategies for improvement. The strategies, in turn, need to know what adjustments to the decisions made during the initial generation will produce appropriate modifications to the text. We compare our treatment of revision with those of Mann and Moore (1981), Gabriel (1984), and Mann (1983).

This paper was presented at the 24th Annual Meeting of the Association for Computational Linguistics, June 10-13, 1986, Columbia University, New York, New York, and published in the Proceedings of the Conference, p. 90-96.

1. INTRODUCTION

Revision is a large part of the writing process for people. This is one respect in which writing differs from speech. In ordinary conversation we do not rehearse what we are going to say; however, when writing a text which may be used more than once by an audience which is not present, we use a multipass system of writing and rewriting to produce optimal text. By reading what we write, we seem better able to detect flaws in the text and see new options for improvement.

Why most people are not able to produce optimal text in one pass is an open and interesting question. Flower and Hayes (1980) and Collins and Gentner (1980) suggest that writers are unable to juggle the excessive number of simultaneous demands and constraints which arise in producing well written text. Writers must concentrate not only on expressing content and purpose, but also on the discourse conventions of written prose: the constraints on sentence, paragraph, and text structure which are designed to make texts more readable. Successive iterations of writing and revising may allow the writer to reduce the number of considerations demanding attention at a given time.

The developers of natural language generation systems must also address the problem of how to produce high quality text. Most systems today concentrate on the production of dialogs or commentaries, where the texts are generally short and the coherence is strengthened by nonlinguistic context. However, in written documents coherence must be maintained by the text alone. In addition, written text must anticipate the questions of its readers. The text must be clear and well organized so that the reader may follow the points easily, and it must be concise and interesting so as to hold the reader's attention. These considerations place greater demands on a generation system.

Most natural language generation systems generate in a single pass with no revision. A drawback of this approach is that the information necessary for decision making must be structured so that at any given point the generator has enough information to make an optimal decision. While many decisions require only local information, decisions involving long range dependencies, such as maintaining coherence, may require not only a history of the decisions made so far, but also predictions of what future decisions might be made and the interactions between those decisions.

An alternative approach is a single pass system which incorporates provisions for revision of its internal representations at specific points in the generation process (Mann & Moore, 1981; Gabriel, 1984). Evaluating the result of a set of decisions after they have been made allows a more parsimonious distribution of knowledge since specific types of improvements may be evaluated at different stages. Interactions among the decisions made so far may also be evaluated rather than predicted. The problem remains, however, of not being able to take into account the interaction with future decisions.

A third approach, and the one described in this paper, is to use the writing process as a model and to improve the text in successive passes. A generation/revision system would include a generator, a parser, and an evaluation component which would assess the parse of what the generator had produced and determine strategies for improvement. Such a system would be able to tailor the degree of refinement to the particular context and audience. In an interactive situation the system may make no refinements at all, as in "off the cuff" speech; when writing a final report, where the quality of the text is more important than the speed of production, it may generate several drafts.

While single pass approaches may be engineered to give them the ability to produce high quality text, the parser-mediated revision approach has several advantages. Using revision can reduce the structural demands on the generator's representations, and thus reduce the overall complexity of the system. Since the revision component is analyzing actual text with a parser, it can assess long range dependencies naturally without needing to keep a history within the generator or having it predict what decisions it might make later.

Revision also creates an interesting research context for examining both computational and psychological issues. In a closed loop system, the generator and parser must interact closely. This provides an opportunity to examine how these processes differ and what knowledge may be shared between them. In a similar vein, we may use a computational model of the revision task to assess the computational implications of proposed psychological theories of the writing process.

2. DEFINING THE PROBLEM

In order to make research into the problem of revision tractable, we need to first delimit the criteria by which to evaluate the text. They need to be broad enough to make a significant improvement in the readability of the text, narrow enough to be defined in terms of a representation a parser could build today, and have associated strategies for improvement that are definable in terms understood by the text planner and generator. In addition, we would like to delegate to the revision component those decisions which would be difficult for a generator to make when initially producing the text. As textual coherence often requires awareness of long range dependencies, we will begin by considering it an appropriate category of evaluation for a revision component.

Coherence in text comes from a number of different sources. One is simply the reference made to earlier words and phrases in the text through anaphoric and cataphoric pronominal references; nominal, verbal and clausal substitution of phrases with elements such as 'one', 'do', and 'so'; ellipsis; and the selection of the same item twice or two items that are closely related. Coreferences create textual cohesion since the interpretation of one element in the text is dependent on another (Halliday and Hansan, 1976).

Scinto (1983) describes a narrower type of cohesion which operates between successive predicational units of meaning (roughly clauses). These units can be described in terms of their "theme" (what is being talked about) and "rheme" (what is being said about it). Thematic progression is the organization of given and new information into theme-rheme patterns in successive sentences. Preliminary studies have shown (Glatt, 1982) that thematic progressions in which the theme of a sentence is coreferential with the theme or the rheme of the immediately preceding sentence are easier to comprehend than those with other thematic progressions. This ease of comprehension can be attributed to the fact that the connection of the sentence with previous text comes early in the sentence. It would appear that the longer the reader must wait for the connection, the more difficult the integration with previous information will be.

Another source of coherence is lexical connectives, such as sentential adjuncts ('first', 'for example', 'however'), adverbials ('subsequently', 'accordingly', 'actually'), and subordinate and coordinate conjunctions ('while', 'because', 'but'). These connectives are used to express the abstract relation between two propositions explicitly, rather than leaving it to the reader to infer. Other ways of combining sentences can function to increase coherence as well. Chafe (1984) enumerates the devices used to combine "idea units" in written text, including turning predications into modifications with attributive adjectives, preposed and postposed participles, and combining sentences using complement and relative clauses, appositives, and participle clauses. These structures function to increase connectivity by making the text more concise.

Paragraph structure also contributes to the coherence of a text. "Paragraph" in this sense (Longacre, 1979) refers to a structural unit which does not necessarily correspond to the orthographic unit indicated by an indentation of the text. Paragraphs are characterized by closure (a beginning and end) and internal unity. They may be marked prosodically by intonation in speech or orthographically by indentation in writing, and structurally, such as by initial sentence adjuncts. Paragraphs are recursive structures, and thus may be composed of embedded paragraphs. In this respect they are similar to Mann's rhetorical discourse structures (Mann, 1984).

3. A MODEL OF GENERATION AND REVISION

In this section we will outline a model of generation with revision, focusing on improving textual coherence. First we establish a division of labor within the generation/revision process. Then we look at the phases of revision and consider the capabilities necessary for recognizing deficiencies in cohesion and how they may be repaired. In the fourth section, we apply this model to the revision of an example summary paragraph.

The initial generation of a text involves making decisions of various kinds. Some are conceptually based, such as what information to include and what perspectives to take. Others are grammatically based, such as what grammatical form a concept may take in the particular syntactic context in which it is being realized, or how structures may be combined. Still others are essentially stylistic and have many degrees of freedom, such as choosing a variant of a clause or whether to pied pipe in a relative clause.

The decisions that revision affects are at the stylistic level; only stylistic decisions are free of fixed constraints and may therefore be changed. Changes to conceptually dictated decisions would shift the meaning of the text. During initial generation, heuristics for maintaining local cohesion are used, drawing on the representations of simple local dependencies. By "local", we mean specifically that we restrict the scope of information available to the generator to the sentence before, so that it can use thematic progression heuristics, letting revision take care of longer range coherence considerations.

The revision process can be modeled in terms of three phases:

1) recognition, which determines where there are potential problems in the text;

2) editing, which determines what strategies for revision are appropriate and chooses which, if any, to employ;

3) re-generation, which employs the chosen strategy by directing the decision making in the generation of the text at appropriate moments.

This division reflects an essential difference in the types of decisions being made and the character of representations being used in each phase.

The recognition phase is responsible for parsing the text and building a representation rich enough to be evaluated in terms of how well the text coheres. Since in this model the system is evaluating its own output, it need not rely only on the output text in making its judgements; the original message input to the generator is available as a basis for comparing what was intended with what was actually said. The goal is to notice the relationships among the things mentioned in the text and the degree to which the relationships appear explicitly. For example, the representation must capture whether a noun phrase is the first reference to an object or a subsequent reference, and if it is a subsequent reference, where and how it was previously mentioned. The recognition phase analyzes the text as it proceeds using a set of evaluation criteria. Some of these criteria look through the representation for specific flaws, such as ambiguous referents, while others simply flag places where optimizations may be possible, such as predicate nominal or other simple sentence structures which might be combined with other sentences. Other criteria compare the representation with the original plan in order to flag potential places for revision such as parallel sub-plans not realized in parallel text structure, or relations included in the plan which are expressed implicitly, rather than explicitly, in the text.

Once a potential problem has been noted, the editing phase takes over. For each problem there is a set of one or more strategies for correcting it. For example, if there is no previous referent for the subject of a sentence, but there is a previous reference to the object, the sentence might be changed from active to passive; or if the subject has a relation to previous referent which is not explicitly mentioned in the text, more information may be added through modification to make that implicit connection explicit. The task of the editing phase is to determine which, if any, of these strategies to employ. (It may, for example decide not to take any action until further text has been analyzed.) However, what constitutes an improvement is not always clear. While using the passive may strengthen the coherency, active sentences are generally preferred over passives. And while adding more information may strengthen a referent, it may also make the noun phrase too heavy if there are already modifications. The criteria that choose between strategies must take into account the fact that the various dimensions along which the text may be evaluated are often in conflict. Simple evaluation functions will not suffice.

The final step is actually making the change once the strategy has been chosen. This essentially involves "marking" the input to the generator, so that it will query the revision component at appropriate decision points. For example, if the goal is to put two sentences into parallel structure, the input plan which produces the structure to be changed would be marked. Then, when the generator reached that unit, it would query the revision component as to where the unit should be put in the text (e.g. a main clause or a subordinate one) and how it should be realized (e.g. active or passive).

Note that as the revision process proceeds, it is continually dealing with a new text and plan, and must update its representations accordingly. New opportunities for changes will be created and previous ones blocked. We have left open the question of how the system decides when it is done. With a limited set of evaluation criteria, the system may simply run out of strategies for improvement. The question will be more easily answered empirically when the system is implemented.

An important architectural point of the design is that the system is not able to look ahead to consider later repercussions of a change; it is constrained to decide upon a course of action considering only the current state of the textual analysis and the original plan. While this constraint obviates the problems of the combinatorial explosion of potential versions and indefinite lookahead, we must guard against the possibility of a choice causing unforeseen problems in later steps of the revision process. One way to avoid this problem is to keep a version of the text for each change made and allow the system to return to a previous draft if none of the strategies available could sufficiently improve the text.

4. PARAGRAPH ANALYSIS

In this section we use the model outlined above to describe how the revision component could improve a generated text. What follows is an example of the incremental revision of a summary paragraph. The discussion at each step gives an indication of the character of information needed and the types of decisions made in the recognition, editing, and regeneration phases.

The example is from the UMass COUNSELOR Project, which is developing a natural language discourse system based on the HYPO legal reasoning system (Rissland, Valcarce, & Ashley, 1984). The immediate context is a dialog between a lawyer and the COUNSELOR system. Based on information from the lawyer, the system has determined that the lawyer's case might be argued along the dimension "common employee transferred products or tools". The system summarizes a similar case that has been argued along the same dimension as an example.

The information to be included in the summary is chosen from the set of factual predicates that must be satisfied in order for the particular dimension to apply.

In the initial generation of the summary, the overall organization is guided by a default paragraph organization for a case summary. The first sentence functions to introduce the case and place it as an example of the dimension in question. The body presents the facts of the case organized according to a partial ordering based on the chronology of the events. The final sentence summarizes the case by giving the action and decision. The choice of text structure is guided by simple heuristics which combine sentences when possible and choose a structure for a new sentence based on thematic progression, so that the subject of the new sentence is related to the theme or rheme of the previous sentence.

> (1) The case Telex vs. IBM was argued along the dimension "common employee transferred products or tools". IBM developed the product Merlin, which is a disk storage system. Merlin competes with the T-6830, which was developed by Telex. The manager on the Merlin development project was Clemens. He left IBM in 1972 to work for Telex and took with him a copy of the Merlin code. IBM sued Telex for misappropriation of trade secret information and won the case.

The recognition phase analyzes the text, looking for both flaws in the text and missed opportunities. The repetition of the word "develop" in the second and third sentences alerts the editing phase to consider whether a different word should be chosen to avoid repetition, or the repetition should be capitalized on to create parallel structure. By examining the input message, it determines that these clauses were realized from parallel plans, so it chooses to realize them in parallel structure.

In the regeneration phase, the message is marked so that the revision component can be queried at the appropriate moments to control when and how the information unit for "Telex developed the T-6830" will be realized. After generation of the second sentence, the generator has the choice of attaching either <develop Telex T-6830> or <compete Merlin T-6830> as the next sentence. As one of these has been marked, the revision component is queried. Its goal is to make this sentence parallel to the previous one, so it indicates that the marked unit, <develop ...>, should be the next main clause and should be realized in the active voice. Once that has been accomplished, the default generation heuristics take over to attach <competes with...> as a relative clause:

(2) The case Telex vs. IBM was argued along the dimension "common employee transferred products or tools". IBM developed the product Merlin, which is a disk storage system. **Telex developed the T-6830, which competes with Merlin.** The manager on the Merlin development project was Clemens. He left IBM in 1972 to work for Telex and took with him a copy of the Merlin code. IBM sued Telex for misappropriation of trade secret information and won the case.

Once the change is completed, the recognition phase takes over once again. It notices that sentence four no longer follows a preferred thematic progression as "Merlin" is no longer a theme or rheme of the previous sentence. It considers the following possibilities:

-- Create a theme-theme progression by moving sentence five before sentence four and beginning it with "Telex", as in: "Telex was who Clemens worked for after he left IBM in 1972." (Note there are no other possibilities for preferred thematic progressions without changing previous sentences.)

-- Reject the previous change which created the parallel structure and go back to the original draft.

-- Leave the sentence as it is. Although there is no preferred thematic progression, cohesion is created by the repetition of "Merlin" in the two sentences.

-- Create an internal paragraph break by using "in 1972" as an initial adjunct. This signals to the reader that there is a change of focus and reduces the expectation of a strong connection with the previous sentences.

The editor chooses the fourth strategy, since not only does it allow the previous change to be retained, but it imposes additional structure on the paragraph. Again during the regeneration phase the editor marks the information unit in the message which is to be realized differently in the new draft. Default generation heuristics choose to realize "Clemens" as a name, rather than a pronoun as it had been, and to attach "the manager..." as an appositive.

(3) The case Telex vs. IBM was argued along the dimension "common employee transferred products or tools". IBM developed the product Merlin, which is a disk storage system. Telex developed the T-6830, which competes with Merlin. **In 1972, Clemens, the manager on the Merlin development project, left IBM to work for Telex and took with him a copy of the Merlin code.** IBM sued Telex for misappropriation of trade secret information and won the case.

5. OTHER REVISION SYSTEMS

Few generation systems address the question of using successive refinement to improve their output. Some notable exceptions are KDS (Mann & Moore, 1981), Yh (Gabriel, 1982), and Penman (Mann, 1983). KDS and Yh use a top down approach where intermediate representations are evaluated and improved before any text is actually generated; Penman uses a cyclic approach similar to that described here.

KDS uses a hill climbing module to improve text. Once a set of protosentences has been produced and grossly organized, the hill climber attempts to compose complex protosentences from simple ones by applying a set of aggregation rules, which correspond roughly to English clause combining rules. Next, the hill climber uses a set of preference rules to judge the relative quality of the resulting units and repeatedly improves the set of protosentences on the basis of those judgements. Finally, a simple linguistic component realizes the units as sentences.

There are two main differences between this system and the one described in this paper. First, KDS uses a quantitative measure of evaluation in the form of preference rules which are stated independently of any linguistic context. The score assigned to a particular construction or combination of units does not consider which rules have been applied in nearby sentences. Consequently, intersentential relations cannot be used to evaluate the text for more global considerations. Secondly, KDS evaluates an intermediate structure, rather than the final text. Therefore, realization decisions, such as those made by KDS's Referring Phrase Generator, have not yet been made. This makes evaluating the strength of coherence difficult, since it is not possible to determine whether a connection will be made through modification.

Yh also uses a top down improvement algorithm, however rather than having a single improvement module which applies one time, it evaluates and improves throughout the generation process. The program consists of a set of experts which do such things as construct phrases, construct sentences, and supply words and idioms. The "planner" tries to find a sequence of experts that will transform the initial situation (initially a specification to be generated) to a goal situation (ultimately text). First, experts which group the information into paragraph size sets are applied; then other experts divide those sets into sentence size chunks; next, sentence schemata experts determine sentence structure; and finally experts which choose lexical items and generate text apply. After each expert applies, critics evaluate the result and may call an expert to improve it. Like KDS, this type of approach makes editing of global coherence considerations difficult since structural decisions are made before lexical choices.

The Penman System is the most similar to the one described in this paper. The principle data flow and division of labor into modules are the same: planning, sentence generation, improvement. However, an important difference is that Penman does not parse the text in order to revise it. Rather it uses quantitative measures, such as sentence length and level of clause embeddings to flag potential trouble spots. While this approach may improve text along some dimensions, it will not be capable of improving relations such as coherence, which depend on understanding the text. A similarity between Penman's revision module and the model described in this paper is that neither has been implemented. As the two systems mature, a more complete comparison may be made.

6. CONCLUSION

Using the writing process as a model for generation is effective as a means of improving the quality of the text generated, especially when considering intersentential relations such as coherence. Decisions which increase coherence are difficult for a generator to make on a first pass without keeping an elaborate history of its previous decisions and being able to predict future decisions. Once the text has been generated however, revision can take advantage of the global information available to evaluate and improve coherence.

The next steps in the development of the system proposed in this paper are clear: For the recognition phase, a more comprehensive set of evaluation criteria need to be enumerated and the requirements they place on a parser specified. For the editing phase, the relationships between strategies for improving text, and changes in generation decisions and variation in output text need to be explored. Finally, a prototypical model of the system needs to be implemented so that the actual behavior of the system may be studied.

7. ACKNOWLEDGEMENTS

We would like to thank John Brolio and Philip Werner for their helpful commentary in the preparation of this paper.

8. REFERENCES

Chafe, Wallace L. (1985) "Linguistic Differences Produced by Differences Between Speaking and Writing", in Olson, David K., Nancy Torrance, & Angela Hildyard, eds. *Literacy, Language and Learning: The nature and consequences of reading and writing,* Cambridge University Press, pp. 105-123.

Clippinger, John, & David D. McDonald (1983) "What makes Good Writing Easier to Understand", *IJCAI Proceedings,* pp.730-732.

Collins, Allan & Dedre Gentner (1980) "A Framework for a Cognitive Theory of Writing", in Gregg & Steinburg, eds, pp. 51-72.

Flower, Linda & John Hayes (1980) "The Dynamics of Composing: Making Plans and Juggling Constraints", in Gregg & Steinberg, eds, pp. 31-50.

Gabriel, Richard (1984) "Deliberate Writing", to appear in McDonald & Bolc, eds. *Papers on Natural Language Generation,* Springer-Verlag, 1987.

Glatt, Barabara S. (1982) "Defining Thematic Progressions and Their Relationships to Reader Comprehension", in Nystrand, Martin, ed. *What Writers Know: the language, process, and structure of written discourse,* New York, NY: Academic Press, pp. 87-104.

Gregg, L. & E.R. Steinberg, eds. (1980) *Cognitive Processes in Writing* , Hilldale, NJ: Lawrence Erlbaum Associates.

Halliday, M.A.K., & Ruqaiya Hasan (1976) *Cohesion in English,* London: Longman Group Ltd.

Hayes, John, & Linda Fower (1980) "Identifying the Organization of Writing Processes", in Gregg & Steinberg (Eds), pp. 3-30.

Longacre, R.E. (1979) "The Paragraph as a Grammatical Unit", in *Syntax and Semantics, Vol. 12: Discourse and Syntax,* Academic Press, pp. 115-134.

Mann, William C. & James Moore (1981) "Computer Generation of Multiparagraph English Text", *American Journal of Computational Linguistics*, Vol.7, No.1, Jan-Mar, pp.17-29.

Mann, William C. (1983) *An Overview of the Penman Text Generation System*, USC/ISI Technical Report RR-83-114.

Mann, William C. (1984) *Discourse Structures for Text Generation*, ISI Technical Report ISI/RR-84-127.

McDonald, David D. (1985) "Recovering the Speaker's Decisions during Mechanical Translation", *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Colgate University, pp.183-199.

McDonald, David D. & James Pustejovsky (1985) "Description-directed Natural Language Generation". *IJCAI Proceedings*, pp.799-805.

Rissland E., E. Valcarce, & K. Ashley (1984) "Explaining and Arguing with Examples", *Proceedings of AAAI-84*.

Scinto, Leonard, F.M. (1983) "Functional Connectivity and the Communicative Structure of Text", in Petofi, Janos S. & Emel Sozer, eds. (1983) *Micro and Macro Connexity of Texts*, Hamburg: Buske, pp.73-115.