CUCS-39-83

Recursion in Text and its Use in Language Generation

Kathleen R. McKeown

Department of Computer Science
Columbia University
New York, NY 10027

This paper also appears in *Proceedings of the National Conference on Artificial Intelligence*, Washington, D.C., 1983.

RECURSION IN TEXT AND ITS USE IN LANGUAGE GENERATION[1]

Kathleen R. McKeown
Department of Computer Science
Columbia University
New York, NY 10027

ABSTRACT

In this paper, I show how *textual structure* is recursive in nature; that is, the same rhetorical strategies that are available for constructing the text's macro-structure are available for constructing its sub-sequences as well, resulting in a hierarchically structured text. The recursive formalism presented can be used by a generation system to vary the amount of detail it presents for the same discourse goal in different situations.

1 Introduction

Texts and dialogues often contain embedded units which serve a sub-function of the text or dialogue as a whole. This has been noted both by Grosz [GROSZ 77] in her observations on task dialogues and by Reichman [REICHMAN 81] in analyses of informal conversations. In this paper, I show how *textual structure* is recursive in nature; that is, the same rhetorical strategies that are available for constructing the text's macro-structure are available for constructing its sub-sequences as well, resulting in a hierarchically structured text. This complements Grosz's view of hierarchical text structure as a mirror of hierarchical task structure. A generation system can use recursion to generate a variety of different length texts from a limited number of discourse plans which specify appropriate textual structures. In the following sections, I present a formulation of recursive text structure, an example of its use in the fully implemented TEXT generation system [MCKEOWN 82A], and finally, a description of some recent work on the application of this mechanism to automatically generating the appropriate level of detail for a user.

2 What is Textual Recursion?

Rhetorical predicates (also termed *coherence relations*) have been discussed (e.g., [GRIMES 75], [HIRST 81], [HOBBS 78]) as a means for describing the predicating acts available to a speaker. They delineate the structural relations between propositions in a text. Some examples are "identification" (identify an item as member of a generic class), "analogy" (compare with a familiar object), and "particular-illustration" (exemplify point). In earlier papers [MCKEOWN 80], [MCKEOWN 82B], I showed how such predicates could be combined to form a longer textual sequence serving a single discourse purpose (for example, definition). These combinations were formalized as *schemata* which embody text structures commonly used in naturally occurring texts, as determined by empirical analysis.

This analysis also indicated that the predicates may be applied recursively to describe the structure of a text at many levels. A predicate may characterize the structural relation of a single sentence or of a longer sequence of text, such as a paragraph, to preceding text. Schemata merely indicate how predicates may be combined to form longer sequences of texts having specific functions. Thus, they describe combinations of predicates which serve the function of a *single* predicate. Textual recursion is achieved by allowing each predicate in a schema to expand to either a single proposition (e.g., a clause or a sentence) or to its associated schema (e.g. a text sequence).

As an example, consider the sequences of text shown in Examples 1 and 2 below. The structure of both of these texts is captured by the *identification schema*, a schema which describes the combination of predicates that are commonly used to provide definitions.[2] In the first text, sentence 1 identifies the hobie cat, 2 describes characteristic attributes, and 3 provides an example. The second text contains the same basic structure, except that the identification of the hobie cat is achieved by a textual sequence instead of a single sentence. This textual sequence (sentences 1-4) is also described by an instantiation of the identification schema. Note that any of the other predicates of either the higher level identification schema or the embedded definition could have been expanded by their associated schemata if the

[1]This work was partially supported by NSF grant #MCS81-07290, awarded to the Dept. of Computer and Information Science of the University of Pennsylvania.

[2]The schema itself is not shown here. That schemata allow for optional predicates accounts for the variations in the *instantiations* of the identification schema shown here. See [MCKEOWN 82] for a full description of the schemata themselves.

author/speaker preferred to provide more detail.[3]

--------- Example 1 ---------

Identification Schema

1. Identification
2. Attributive
3. Particular illustration

1. A hobie cat is a brand of catamaran, manufactured by the Hobie Company. 2. Its main attraction is that it's cheap. 3. A new one goes for about $5000.

--------- Example 2 ---------

Identification Schema **Identification Schema**

1. Identification
    2. Identification
    3. Analogy
    4. Particular illustration
5. Attributive
6. Particular illustration

1. A hobie cat is a brand of catamaran, manufactured by the Hobie Company. 2. Catamarans are sailboats with two hulls instead of the usual one. 3. A catamaran is typically much faster than a sailboat. 4. Hobie cats, tiger cats, and pacific cats are all catamarans. 5. As for the hobie cat, its main attraction is that it's cheap. 6. A new one goes for about $5000.

A question raised by the above two examples is that of when recursion is necessary. Clearly, there are situations where a simple sentence is sufficient for fulfilling a communicative goal, while in other cases, it may be necessary to provide a more detailed explanation. One test for recursion hinges on an assessment of a user's knowledge. In the above example, a more detailed identification of the hobie cat might be provided if the speaker assumed the listener knew very little about sailing. An investigation of the possible tests for recursion is currently being undertaken.

3 Use of Recursion for Generation

Recursion is a mechanism that can be used to allow a generation system to uniformly provide varying amounts of detail. In the TEXT system, which generates paragraph length responses to questions about database structure, some limited use has been made of recursion. In certain cases, the user's question alone indicates that the user has a lack of knowledge and requires more detail. For example, when asking a question about the difference between two very different objects, the user indicates a total lack of knowledge about the items in question. In this case, lack of knowledge triggers the need to expand the identification of each item, using the identification schema to provide more detail.

The system's response to the question "What is the difference between a destroyer and a bomb?"[4] illustrates this feature. In this example, sequence 1-2 results from application of the identification schema for destroyer, sequence 3-4 from the identification schema for bomb, and the entire sequence (1-5) from application of a different schema (compare and contrast) which accesses the identification schema (see [MCKEOWN 82A] for more details). The destroyer and the bomb are each defined by providing two identifications (the second a result of recursion). No additional predicates (such as attributive or particular-illustration) from the identification schema are included for this response because the system has determined by other mechanisms that only generic class information is relevant [MCKEOWN 80].

--------- Example 3 ---------

(difference DESTROYER BOMB)[5]

;
; What is the difference between a destroyer and a bomb?
;
; 1. Identification destroyer
; 2. Identification ship
; 3. Identification bomb
; 4. Identification free-falling projectile
; 5. Inference

1. A destroyer is a surface ship with a draft between 15 and 222. 2. A ship is a vehicle 3. A bomb is a free-falling projectile that has a surface target location. 4. A free-falling projectile is a lethal destructive device. 5. The bomb and the destroyer, therefore, are very different kinds of entities.

----

[3]As another example, note that the structure of the last three paragraphs is also captured by the identification schema. Here, schemata are identified in the first paragraph on p 2, their recursive attribute specified (second paragraph, p 2), and an example given (third paragraph, p 2).

[4]The TEXT system was implemented on an ONR database containing information about military vehicles and weapons. The example is taken from this domain.

[5]TEXT generates the paragraph as shown (but without sentential numbers) in response to the functional question notation (TEXT has no facility for parsing English questions). Comments show the English version of the question and the predicates used in the response

4 Limits on Recursion

It has been suggested (e.g., [CONKLIN 83]) that the phenomenon described here may not actually be recursion per se since 1) there may be bounds on how many recursive pushes can be taken and 2) a speaker may not return in reverse order to every higher level dialogue from which a push was taken.

The existence of bounds on the depth of recursion is not motivated by the occurrence of recursion in naturally occurring texts. My analysis suggests instead that many levels of nesting are possible, but that when such nesting occurs the text grows in length and may cover several pages [MCKEOWN 82A]. Grosz's analysis of hierarchically nested dialogues also indicates that nesting can occur to many levels. Placing arbitrary bounds on the depth of recursion could conceivably *limit* a generation system in its ability to provide the kind of detail needed by a user in some given situation. The absence of limits on recursive depth, on the other hand, does not have detrimental side-effects as long as the system is capable of determining in what situations recursion is not necessary.

Bounds on recursion are even more severely limiting on the generality of a generation system than this suggests. Note that if no recursion is allowed, the system will only be capable of producing texts of a uniform length unless further changes are made in the system. A single schema will consistently produce paragraph length text if its predicates are always expanded as single propositions. To generate longer texts, the system must either be capable of combining schemata appropriately (requiring further theoretical work on legal combinations of schemata) or new schemata must be developed which will generate longer sequences of text.

If, on the other hand, recursion is allowed, then a limited number of schemata can be used to generate an infinite number of different length texts. A *single* schema produces infinitely many texts if its different predicates are expanded to their associated schemata instead of single propositions and this expansion occurs at all levels of the text. The use of unlimited recursion, therefore, allows for less work to be done in determining possible text orderings and, in theory, for the generation of arbitrarily long texts from a small number of schemata. Currently, schemata for 4 predicates have been developed for the TEXT system which uses a total of 10 predicates.

In the written texts that were analyzed, writers did return in reverse order to higher level texts from which a push was taken, with the exception of cases where a push was taken on the last predicate in a schema. I would speculate that whether a speaker does return to every dialogue from which a push was taken may be affected by his/her memory for the past discourse. That memory is not perfect may cause higher level unfinished discourses to be skipped when finishing a sub-dialogue. If memory is the cause, then well-planned *writing* should exhibit the phenomenon of imperfect recursion less often since planning, re-reading, and re-writing is possible. This hypothesis could be empirically tested.

5 Current Directions

The recursive mechanism can be used to allow a generation system to provide either a detailed or succinct response to the same question under different circumstances. Clearly, an analysis of the factors that trigger or inhibit recursion is critical for use of this capability and this is an endeavor that is currently underway. A preliminary analysis indicates that these factors would at least include the following:

1. The user's level of expertise: A user comes to a system with apriori knowledge on the subject in question. The system's knowledge of that level (whether deduced from interaction or explicitly stated) will influence how much it should say. Note that this is not a simple influence. An expert may in certain situations be able to handle more detail than a novice.

2. The past discourse: What the user has learned through the past discourse influences level of detail since previous discussion of a subject may mean that less can be said about it in a current response. What the system has learned through the past discourse affects level of detail as well: the user's acceptance of detail or request for detail may indicate to a system that it can provide a particular type of detail without being asked.

3. The user's overall goal in interacting with the system: Whether the user is using the system, for instance, to quickly retrieve a specific fact or to learn about or from the system will require different levels of detail.

4. The user's specific goal in asking a particluar question: If the user's question is only one step towards acquiring the information necessary for a higher level goal, that goal may dicate how much information is required.

5. Feedback from the user: While the goal of this research is to anticipate the user's needs for detail before s/he states them explicitly, in actual conversation, people often do explictly state that they have absorbed information and are ready for more (e.g., backchannel noises such as "um-hum") or that they have not understood. Such feedback can also be used in a system.

While some of these factors are very difficult to implement (e.g., determining the user's goal), others are, in fact, tractable. Tracking of past discourse, for example, has been used previously to avoid repetition [MCDONALD 80; DAVEY 79]. The recursive mechanism is also viewed as an important element in providing re-explanations. That is, a user's dissatisfaction with a given response may provide the trigger to recurse on a predicate that was previously unexpanded.

This effort is being conducted with the goal of implementing an information/expert system that can provide explanations in the domain of advising students about course schedules. This domain requires the capacity for communicating at different levels of detail and for providing re-explanations since students as users may frequently be dissatisfied with an explanation (for example, why they cannot take a course), may simply want to talk at length about a course of action, and may want to explore alternate solutions to a problem.

6 Conclusions

In this paper, a formalism which represents the hierarchical nature of texts in terms of recursive textual structure has been presented. This augments previous work on the structure of sub-dialogues by capturing another dimension along which sub-sequences of text are related to the text as a whole. Furthermore, this formulation of text structure allows a generation system to use the same schema to generate both short and more detailed descriptions. While this has already been used in a limited way in the TEXT generation system, the eventual goal is to develop a full analysis of decision mechanisms for recursion and embody this in a generation system which can provide explanations at varying levels of detail as well as re-explanations in response to a user's dissatisfaction.

ACKNOWLEDGEMENTS

I would like to thank Michael Lebowitz, Bonnie Webber, and Kathy McCoy for their comments on drafts of this paper.

7 References

[CONKLIN 83] Conklin, J., forthcoming dissertation. University of Massachusetts, Amherst, Mass., 1983.

[DAVEY 79] Davey, A., *Discourse Production*, Edinburgh University Press, Edinburgh, 1979.

[GRIMES 75] Grimes, J. E. *The Thread of Discourse*. Mouton, The Hague, Paris. (1975).

[GROSZ 77] Grosz, B. J., The representation and use of focus in dialogue understanding. Technical note 151, SRI International, Menlo Park, Ca. (1977).

[HIRST 81] Hirst, G., Discourse-oriented anaphora resolution: a review. *American Journal of Computational Linguistics*, Vol. 7, No. 2, (1981), pp. 85-98.

[HOBBS 79] Hobbs, J. R., Coherence and coreference. *Cognitive Science*. 3(1), January-March (1979), pp. 67-90.

[MCDONALD 80] McDonald, D.D., Natural language production as a process of decisions making under constraint. PhD. Dissertation, MIT, Cambridge, Mass. 1980.

[MCKEOWN 80] McKeown, K.R., Generating relevant explanations: natural language responses to questions about database structure. *Proceedings of the First Annual National Conference on Artificial Intelligence*. Stanford, Ca., 1980, pp. 306-9.

[MCKEOWN 82A] Mckeown, K.R., Generating natural language text in response to questions about database structure. Ph.D. Thesis, MS-CIS-82-05, University of Pennsylvania, Philadelphia, 1982.

[MCKEOWN 82B] McKeown, K.R., The TEXT system for natural language generation: an overview. *Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics*, Toronto, Ontario, Canada, 1982, pp. 113-20.

[REICHMAN 81] Reichman, R., Plain speaking: a theory and grammar of spontaneous discourse. Ph.D. Thesis, Harvard University, Cambridge, Ma., 1981.