A NEW APPROACH TO QUALITY TEXT GENERATION

Jyun-Sheng Chang
Hwei-Ming Kou

Institute of Computer Science
National Tsing Hua University
Hinchu, Taiwan, Republic of China

1. INTRODUCTION

The need to study text generation is obvious. It is one of the two ways in which people communicate with one another. Thus, if we can simulate this capability computationally, we would be able to use the techniques in many applications such as (1) automatic generation of reports, manuals, and letters, (2) providing a natural language interface from a system to its users, (3) an nature language interpreter for reading and debugging information encoded in some formal notation such as a knowledge base and a software specification.

Text generation is already established as a research area within computational linguistics [Mann 1982]. It has a rather late start compared to other areas in computational lingustics. During 1970's, out of disatification for pre-prepared text (canned text), researchers began to study ways of generating sentences automatically [Goldman 1975, Grishman 1979 and Spapiro 1979]. In the 1980's, the focus has shifted toward the generation of discourse (cohesive text with many sentences, either in producing a monologue or engaging in a dialogue) [Derr-McKeown 1984, Man

1984, McDonald-Pustejovesky 1985 and McKeown 1985] and the methodology used in text generation [Danlos 1984 and Vaughan-McDonald 1986]. Thus far, there have been only a few experimental systems that generate text in a technically interesting way or are based on sound linguistic theories. However the progress in bettering the understanding of human text production and techniques for simulating this capabiliy, is considerable.

2. PREVIOUS WORK

The generally accepted model of text generation consists of mainly two phases: deep generation and surface generation. The deep generation phase takes the representation of meaning or knowledge and produces a sequence of ordered messages. The surface generation phase then convert each of these messages into a sentence.

The deep generation phase determines what to say (content determination) and when to say what (discourse structure). There are two major tasks in surface generation: syntactical choice and lexical choice. A syntactical pattern must be chosen to realize the sentence. And each entity in the message must have a proper wording for it.

Systems requiring text generation include dialogue systems, question-answering systems, systems that validate natural language input by paraphrasing, expert systems with explanation capability, story and document generation systems.

2.1 CONTENT DETERMINATION

The first task in text generation is to determine what to say which generally depends on the context and purpose of the system. A randon sentence generator obviously could not care less about the content that it produces [Fredman 1969 and Parisi-Giorgi 1985]. The content of a paraphrase is whatever the user has just input [Goldman 1975 and McKeown 1979]. In a question-answering system, the question is parsed, tranformed into some kind of query for the underlaying database. And the result of the query is the content of the answer [Grishman 1979]. McDonal and Conklin pointed out that in describing a picture, a good stategy is to say whatever are most salient [Conklin-McDonald 1982]. Sometimes, the content is diminishable when the purpose of text generation is merely to say something and passes as a person [Boden 1976].

The information resulting from this phase may be represented in various forms: predicate calculus [Grishman 1979], conceptual dependency structures (CD) [Goldman 1975], semantic nets [Minsky 1981, Simmons-Slocum 1972 and McKeown 1985], or frames [Woods 1975 and Mauldin 1984].

2.2 DISCOURSE STRUCTURE

There are many systems in the literature capable of generating multi-sentence text: Simmons and Slocum built a system to generate English sentences from semantic networks [Simmons-Slocum 1972]. Davey's PROTEUS takes the sequence of moves played in a tic-tactoe game and produces a paragraph of commentary of the game [Davey 1975, Mann 1982, and Richie 1984]. The BLAH system generates multi-sentence explanation of the reasoning process taken by the system [Weiner 1980]. Meehan's TALESPIN produces multi-paragraph stories [Meehan 1977]. However these systems focus on knowledge needed for generation and its representation. The organization of text is either given (as in PROTEUS) or fixed (as in TALESPIN).

The systems that have a module determining discourse structure and represent the knowledge about discourse structure explicitly, include the Knowledge Delivery System (KDS) [Mann-Moore 1979 and 1981], BLAH, the explanation module for an expert system [Weiner 1980], and TEXT, a system that answers questions about the structure of a database [McKeown 1985].

BLAH mimics the simple way that people use to explain something or justify a statement to organize the text for explanation. TEXT goes considerably behind BLAH's simple formulation of discourse structure in the three ways: (1) TEXT includes discourse strategies in the form of ATN graphs, for many more discourse goals in addition to explanation. (2) These discourse strategies are based on naturally occurring text. Thus they reflect the discourse patterns which are effective and most often used. (3) Due to the nondeterminism of ATN's, these discourse strategies capture a notion of variability which is resolved by focus of attention and semantic relations in the text. The text generated in a top-down, goal-directed fashion, looks well-structured, cohesive, and with a purpose.

Unlike BLAH and TEXT, KDS lacks an explicit representation of knowledge about discourse structure. It employs a rule-based planning strategy to organize information into discourse. The rules used have a strong bottom-up, data-driven flavor.

Finally there is the Rhetoric Structure Theory (RST) [Mann 1983], a descriptive theory on discourse structure. The author claimed that it can be turned into a constructive process through the use of a planning strategy with various rhetoric structures reguarded as means of realizing the goal of text generation. But that remains to be seen.

2.3 SEMI-SURFACE GENERATION

Going from deep generation to surface generation, there is an issue that needs to be resolved. That is the problem of how much information to put in a sentence: Whether to put a lot of information in one complex sentence or put them in 2 or 3 simple sentences. We call this consideration the *semi-surface generation*.

Davey used a fixed strategy in this regard [Davey 1975]. Derr and McKeown recognized the interplay of this decision and shifting in focus of attension in the text [Derr-McKeown 1984]. McDonald studied the encyclopedia articals on African tribes and found that this decision has a lot to do with the prose style of the text generated [McDonald 1985].

2.4 SURFACE GENERATION

There are essential three methods for surface generation: canned text, template, and direct translation. Error message generated by compilers is typical example of canned text. Early expert systems use templates to generate explanations; a template is associated with each rule and the explanation consists of the templates associated with the rules fired. Canned text and templates are fast, easy to construct but must be anticipated in advanced. Consistency and closure are difficult to achieve.

In order to generate sentences of higher quality consistently and to ensure closure, one needs to translate directly a message into a sentence using some form of grammar. Three components are needed for this process: (1) a formal representation of the sentence structure in the language, (2) a dictionary containing various information such that proper words may be chosen to represent concepts and entity conveyed in the message, (3) a way of doing syntactical and lexical choice.

Several grammar formalisms have been used for surface generation: (1) Systemic grammar [Halliday 1973, 1976, and 1985], (2) Transformational grammar founded by Noam Chomsky, (3) ATN grammar [Woods 1970], (4) The Linear String Parser (LSP) [Sager 1981].

The BABEL paraphrasing system uses discriminate nets to help select lexical items and rely on an ATN grammar to generate sentence structure [Goldman 1975]. The The question-answering system by Grishman uses LSP [Grishman 1979] as the grammartical formalism for generation as well as parsing. The sentence generator Kafka [Mauldin 1984] in the XCON/XSEL system uses transformational grammar [Mauldin 1984]. So do the CO-OP paraphraser [McKeown 1979] and a system that generates English sentences for instructional purpose [Bates-Ingria 1981]. An explanation module for a student advisor expert system uses functional grammar [Winograd 1983], which a grammatical formalism based on many ideas from systemic grammar.

System grammar is used by the sentence generators in PROTEUS, the PENMAN/Nigel system, and Patten's system [Davey 1975, Mann 1983, Matthiessen 1983, and Patten 1985]. There is a growing consensus among researchers that systemic grammar is the grammar of choice for text generation.

2.5 METHODOLOGY FOR TEXT GENERATION

Most text generation systems are a one-pass process through the content determination, deep generation, semi-surface generation, and surface generation phases. Vaughan and McDonald observed that people write and then rewrite again and again; revision seem to be a large part of writing process for people. Thus they propose a *revisional model* of text generation to simulate this human strategy of writing [Vaughan-McDonald 1986].

Within this model, text is first generated in a straightforward fashion, without attempting much global arrangement. Then the system iterates through a process of recognition, editing, and regeneration. The recognition phase essentially finds out the places where changes can be made to enhance cohesion of the text. The editing and regeneration phases implement these changes. The authors argued that using this kind of model will reduce the complexity of the text generator. The KDS system follows the revisional model. A revisional module is also planned for the Penman system. However, the feasibility of this model is difficult to assess at this point of time.

Danlos observed that the syntactical choice is sometimes influenced by the choice of lexical items and suggested against strict separation of lexical and syntactical choices [Danlos 1984].

3. A MODEL FOR TEXT GENERATION

This section presents the new approach taken in our text generation system. The system is intended as a test bed for experimenting with new ideas and for understanding the text generation needs in different environments and for different languages.

Our objectives in implementing the system include (1) to base the system in solid linguistic theories [Halliday 1973, Halliday-Hasan 1976, and Hudson 1971], (2) the ability to handle situations where the information needed for text generation is not pre-stored within but rather needs to be acquired from the user, (3) the ability to generate text whose pattern may be determined more or less beforehand (goal-driven) [McKeown 1985] or may be dictated by the information available (data-driven), (4) the adaptability to different styles intended for different kinds of users or purposes [McDonald-Pustejovesky 1985].

Currently, we are considering the following as our domains for text generation: (1) English business letters [keshi 1987], (2) User's manuals for computer systems in English and Chinese, and (3) A paragraph-level target language (Chinese or English) generator for a machine translation system.

In order to achieve the above goals, we take the following approach to implement our system: (1) The system uses a representation which can reflect existence as well as the lack of information. (2) In stead of producing an ordered sequence of messages, the deep generator produces a partilly ordered sequence of propositions with functional marking representing the rhoritical or cohesive relation among the elements in the propositions. (3) The system uses a hybrid strategy which covers the whole spectrum of goal-driven and data-driven strategies. (4) An intermediate phase, called *semi-surface generation*, between deep generation and surface generation, is included, to serve the purpose of reflecting different prose styles. (5) The surface generator produces sentences using a systemic grammar [Hudson 1971].

3.1 CONTENT DETERMINATION

We propose using a frame-based knowledge base to represent knowledge known a priori and as a information acquisition scheme.

If all there is to say is known before text generation, then the representation will be all filled up. On the other hand, if there is information yet unknown at generation time, there will be unfilled slots in the frame system. Then this knowledge base can be used to drive an input module to acquire needed information from the user. Thus we can use a uniform representation for situations where the information needed to generate the text is either pre-stored in the system or needed to be acquired from the user.

3.2 DEEP GENERATION

We propose a deep generation method which is inspired by Discourse Strategies (DS) in McKeown's TEXT system and Mann's Rhetoric Structure Theory (RST). There are two processes in our deep generation phase: a top-down, goal-directed process and bottom-up, data-driven process.

The top-down process uses an ATN-like representation of overall strategy of determine content and organization of the text and go through the recursive representation like travelling a tree from the root (the goal or purpose of the text) down to its leaves. A leaf of the tree is either a single proposition (message) to be converted to a sentence or a demo to start an instance of the bottom-up process.

The bottom-up process uses rules based on (1) focus of attention, (2) identities among the elements of propositions, and (3) semantic relavance between propositions, to pick out propositions and give them a partial order. This is not only a process of finding what propositions should be included in a sentenec but also a process finding and marking the cohesive links in the content. These markings are subsequently used in surface generation for such activitie as pronomial, demonstractive, verbal, and clausal substitutions, ellipsis, selection of conjunction, and lexical choice [Halliday-Hasan 1976]. The result of this phase is an totally ordered sequence of packages where each package contains a set of marked propositions in partial order.

This 2-phase , mixed-strategy approach covers the wole spectrum from the strictly goal-directed strategies to the strictly data-driven strategies. Thus the system can be tuned to adapt to diiferent text generation situations.

3.3 SEMI-SURFACE GENERATION

The partially ordered propositions produced in the deep generation phase subsequently go through a filtering phase. We propose a rule-based approach to determine whether to pack propositions in a sentence or to leave them along so one proposition will produce one sentence in the final surface generation phase.

There should be two kinds of rules: (1) rules that pack propositions based on shifting of focus of attention and degree of identity among propositions (2) meta rules that assign priorities to the first kind of rules.

The first kind of rules will enhance local cohesion in the text, while the second kind of rules can be used to reflect prose style.

3.4 SURFACE GENERATION

Systemic grammar is used in the surface generation for the following reasons: (1) It is based on function of language and emphasizes the mechanism of choice according to function. That corresponds closely to the nature of the generation process. (2) The phases before surface generation produce a lot of functional feactures on which the system grammar is mainly structured. (3) Systemic grammar encode lexical choice and syntactial choice in one notation. Thus the problem pointed out by Danlos can be handle more easily using systemic grammar.

4. CONCLUSION

We have proposed in this paper a framework for text generation which is based the system in solid linguistic theories. The system can handle different situations and generate text either by following a pre-determined pattern or by adjusting to what is present to be conveyed. And it is possible to tune the system so that text of different styles can be generated to serve different kinds of users or purposes.

References

Appelt 1983 *Telegram: A Grammar Formalism for Language Planning*, Proceedings of the 21st Annual Meeting of the ACL pp. 74-78.

Bates 1981 *Controlled Transformational Sentence Generation*, Proceedings of the 19th Annual Meeting of the ACL, pp. 153 158.

Boden 1976 *Artificial Intelligence and Natural Man*, Basic Books, New York, pp. 95 111.

Davey 1975 *Discourse Production*, Edinburgh University Press, Edinburgh.

Danlos 1984 *Conceptual and Linquistic Decisions in Generation*, Proceedings of the 22nd Annual Meeting of the ACL, (COLING 84), pp. 501-504.

Derr-McKeown 1984 *Using Focus to Generate Complex and Simple Sentences*, Proceedings of the 22nd Annual Meeting of the ACL, (COLING 84), pp. 319-326.

Goldman 1975 *Sentence Paraphrasing from a Conceptual Database*, Comm. ACM 18:2, pp. 96 106.

Granville 1984 *Controlling Lexical Substitution in Computer Text Generation*, Proceedings of the 22nd Annual Meeting of the ACL, (COLING 84), pp. 381-384.

Grishman 1979 *Response Generation in Question-Answering Systems*, Proceedings of the 17th Annual Meeting of the ACL, pp. 99-101.

Halliday 1973 *Explorations in the Functions of Language*, Edward Arnold, London.

Halliday 1975 *System and Function in Language*, Oxford University Press, London.

Halliday 1985 *An Introduction to Functional Grammar*, Edward Arnold, London.

Halliday-Hasan 1976 *Cohesion in English*, Longman, London.

Hudson 1971 *English Complex Sentences: an introduction to systemic grammar*, North-Holland, Amsterdam.

Keshi 1987 *A Knowledge-based Framework in an Intelligent Assistant for Making Document*, Abstracts of the International Conference on AI, (AI 87 JAPAN), Osaka, Japan, pp. 286-294.

Mann-Moore 1979 *A Snapshot of KDS: A knownledge Delivery System*, Proceedings of the 17th Annual Meeting of the ACL, pp. 51-52.

Mann-Moore 1981 *Computer Generation of Multiparagraph English Text*, AJCL 7:1, pp. 17-29.

Mann 1982 *Applied Computational Linguistics in Perspective: Proceedings of the Workshop - Text Generation*, AJCL 8, pp. 62-69.

Mann 1983 *An Overview of the Nigel Text Generation Grammar*, Proceedings of the 21st Annual Meeting of the ACL, pp. 79-84.

Mann 1984 *Discourse Structures for Text Generation*, Proceedings of the 22nd Annual Meeting of the ACL, (COLING 84), pp. 367-375.

Matthiessen 1983 *Systemic Grammar in Computation: The Nigel Case*, Proceedings of the 22nd Annual Meeting of the ACL (COLING 84), pp. 155-164.

Mauldin 1984 *Semantic Rule Based Text Generation*, Proceedings of the 21st Annual Meeting of the ACL (COLING 84), pp. 376-380.

McCoy 1982 *Augmenting a Database Knowledge Representation for Natural Language Generation*, Proceedings of the 20th Annual Meeting of the ACL, pp. 121-128.

McDonald 1985 *A Computation Theory of Prose Style for Natural Generation*, Proceedings of the 2nd Conference of the European Chapter of the ACL, pp.187-193.

McDonald-Conklin 1982 *Salience: The Key to the Selection Problem in Natural Language Generation*, Proceeding of the 20th Annual Meeting of the ACL, pp. 129-135,

McDonald-Pustejovesky 1985 *A Computational Theory of Prose Style for Natural Language Generation*, Proceedings of the 2nd Conference of the European Chapter of the ACL, pp.

187-193.

McKeown 1985 *Discourse Strategies for Generating English Text*, Artificial Intelligence 27, pp. 1-41.

Minsky 1981 *A Framework for Representing Knowledge*, in *Mind Design*, edited by J. Haugeland, MIT Press, pp. 95-128.

Parisi-Giorgi 1985 *GEMS: A Model of Sentence Production*, Proceedings of the 2nd Conference of the European Chapter of the ACL, pp. 258-262.

Patten 1985 *A Problem Solving Approach to Generatin Text from Systemic Grammar*, Proceedings of the 2nd Conference of the European Chapter of the ACL, pp. 187-193.

Ritchie 1984 *A Rational Reconstruction of the Proteus Sentence Planner*, Proceedings of the 21st Annual Meeting of the ACL, (COLING 84), pp. 327-329.

Sager 1981 *Natural Language Information Processing*, Addison-Wesley, Reading.

Shapiro 1979 *Generalized Augmented Transition Network Grammars for Generation from Semantic Networks*, Proceedings of the 17th Annual Meeting of the ACL pp. 25-30.

Simmons-Slocum 1972 *Generating English Discourse from Semantic Networmks*, Comm. ACM 13:1, pp. 15-30.

Vaughan-McDonald 1986 *A Model of Revision in Natural Language Generation* Proceedings of the 24th Annual Meeting of the ACL, pp. 90-96.

Weiner 1980 *BLAH, A System Which Explains its Reasoning*, Artificial Intelligence 15, pp. 19-48.

Winograd 1983 *Language as a Cognitive Process, Volume 1: Syntax*, Addison-Wesley, Reading.

Woods 1970 *Transition Network Grammars for Natural Language Analysis*, Comm. ACM 13:10, pp. 591-606.

Woods 1975 *What is in a Link: Foundation for Semantic Network*, in *Studies in Cognitive Science*, D.G. Bobrow and A.M. Collins, eds., Acdemic Press, New York, pp. 35-82.