Building domain-independent text generation system

XinYu Deng **Sadao Kurohashi** **Jun-ichi Nakamura**
Graduate School of Informatics, Kyoto University
606-8501 Yoshida Honmachi, Sakyoku
Kyoto, Japan
{deng, kuro, nakamura}@pine.kuee.kyoto-u.ac.jp

Abstract

This paper investigates an effective method of building domain-independent text generation system. In our English text generation system, Semantic Network is used as the internal Knowledge Representation. Nodes and links of the semantic network are classified according to word class and grammatical relations respectively. Generation results prove that the system works well and can generate coherent text flexibly.

1 Introduction

Natural Language Generation (NLG) is the automatic generation of Natural Language by computer in order to meet communicative goals (Smith, 1995). Until now, almost all of the existing generation systems are domain-dependent, ie., they were built for some specific applications. For example, IDAS "produces on-line hypertext help messages for users of complex machinery" (Reiter and Dale, 2000) (p.12); STOP "is a natural language generation system which produces personalised smoking-cessation letters" (p.16). However, how to build a domain-independent generation system is still regarded to be a difficult problem.

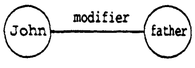
We think that the main difference between domain-dependent and domain-independent generation system is due to the problem of text structuring. Generally speaking, before building a domain-dependent text generation system, designers should make a domain model. In the model, the characteristics of the structure of text to be generated are described. For domain-dependent systems, text structure varies with application domains. In fact, no matter what domain a text belongs to, its structure reflects the common features of English text. If we make a text structure model representing the common features of English texts, the texts generated by this model are domain-independent.

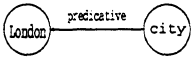
(Ozaki et al., 1997) introduced how to generate coherent Japanese text from semantic network. Based on their research, we built an English Generation System which is domain-independent. Our reserach has two stages. At the first stage, we focus on sentence generation and text generation, but we do not consider discourse structure (cue phrases, "but", "similarly", for example). At the second stage, we will explore the problem of discourse generation. In this paper, we describe the first stage of our research in detail. The rest of the paper is organized as follows: section 2 introduces semantic network of the English Generation System; section 3 introduces generating coherent text flexibly on sentence level and text level; section 4 introduces experiment and generation results; section 5 introduces future directions.

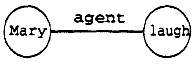
2 Semantic network

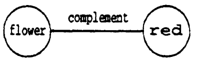
A semantic network is a graph of the structure of meaning in which nodes represent concepts and links represent relations and abstractions (Lehmann, 1992).

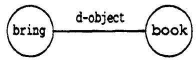
In the English Generation System, nodes are classified into two types according to the word classes: entity node (e.g. noun) and event node (e.g. verb, adjective); links are classified into three types according to the grammatical relations:

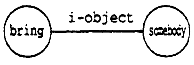
1. Type I includes modifier link and predicative link.
   (a) modifier link: the node modified is parent node, the other node is son node, eg, John's father. 

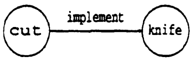
   (b) predicative link: the node representing subject is son node, the other node is parent node, eg, London is a city. 

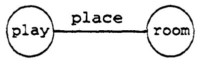
2. Type II represents the relation between entity node and event node. Event node is parent node, entity node is son node. Type II consists of 9 kinds of links.
   (a) agent link: eg, Mary laughs. 

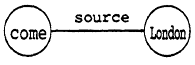
   (b) complement link: eg, The flower is red. 

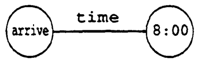
   (c) d-object (direct object) link: eg, buy a book. 

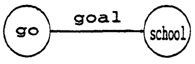
   (d) i-object(indirect object) link: eg, bring somebody a book. 

   (e) implement link: eg, cut something with a knife 

   (f) place link: eg, play in the room. 

   (g) source link: eg, come from London. 

   (h) time link: eg, arrive at 8:00. 

   (i) goal link: eg, go to school. 

3. Type III is event-event link which represents the relation between two discourse segments. We will explore this issue at the second stage of our research. Event-event link is classified into 14 kinds, elaboration, summary, contrast, time before and so on. The event node in the first sentence is parent node. Figure 1 shows how to represent "I start my meal before John arrives." by semantic network.

3 Generating coherent text

This section includes two parts. In the first part, we discuss how to generate single sentence; in the second part, we describe domain-independent text structure model and discuss how to realize this model. Actually, many text generation systems were built from different perspectives. Some approaches are strongly driven by engineering point of view, others are more concerned with human language. In our research, we integrate the human language factor into the engineering

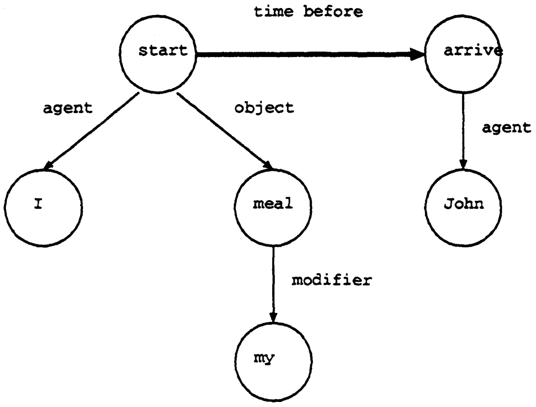


Figure 1: Example of discourse generation.

approach. We hope that it can enhance the overall quality (coherence, flexibility, and readablity) of the generated texts.

3.1 Generation on sentence-level

"English is commonly described as a 'fixed word-order language'" (Quirk et al., 1985)(p.51). "A comprehensive grammar of the English language"(p.720) classifies clause constitutes into five functional categories: subject(S), verb(V), object(O) (direct object–O_d, indirect object–O_i), complement(C) (subject complement–C_s, object complement–C_o), and adverbial(A) (subject-related–A_s, object-related–A_o). Based on the permissible combinations of these functional categories, seven basic structures of clause are established (Quirk et al., 1985) (p.721): SV, SVO, SVC, SVA, SVO_iO_d, SVOC, SVOA. According to these structures, we decided generation order of the links between the root and its sons. The order constraint is: (agent link) > (i-object link) > (d-object link) > (complement link) > (source link) > (goal link) > (time link) > (implement link) > (place link). We created a compare fuction to decide generation order of links.

We regard one single sentence as a tree whose leaf nodes are words and whose root is an event node. Figure 2 shows how to describe a sentence ("John's sister goes to school by bike.") by semantic network. What we want to explain here is that one clause constitute does not mean one node. For example, "S" refers to "subject", it can be represented by one node such as "school", it can be represented by two nodes such as "John's sister" as well. On the other hand, one son node can also be viewed as the root of a sub-tree (or sub-sub-tree, ...), e,g, node "sister" can be regared as the root of a sub-tree, node "John" can be regarded as the root of a sub-sub-tree. We use Figure 2 to explain the generation algorithm on sentence level.

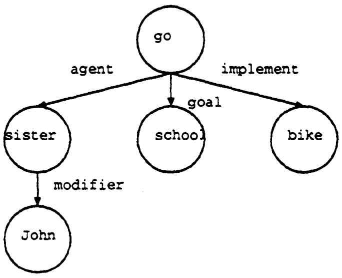


Figure 2: Example of sentence generation.

1. In order to decide generation order of links, the system compares and orders the links between root "go" and its son nodes by compare function. Generation order of the links is: agent link – goal link – implement. So, agent link is first generated.

2. In agent link, node "sister" is son node. On the other hand, node "sister" can be regarded as the root of a sub-tree whose son node is "John" as well. Generation system compares the links between root "sister" and its sons by compare function. Root "sister" has only one son node "John", so the system goes to sub-sub-tree whose root is node "John".

3. Since node "John" has no son node, word "John" is generated. Then the system signal the modifier link between "sister" and "John" by "'s" (because "John" is a proper noun).

4. The system generates the sub-tree whose root is node "sister". Generation result is: "John's sister". That is to say, for any tree (or sub-tree, sub-sub-tree, ...), root is generated at last. Here, generation of agent link is over.

5. Goal link and implement link are generated. The results are "to school" and "by bike" respectively.

6. The tree whose root is node "go" is generated. The result is "John's sister to school by bike goes."

7. The system inserts verb "goes" between "John's siter" (subject) and "to school". Generation result is "John's sister goes to school by bike."

3.2 Generation on text-level

In this part, we introduce the method of text structuring and the algorithm of how to realize this model. Generation results prove that the generated texts are coherent and the method we use is effective.

3.2.1 Principle of END-FOCUS

In this section, we describe the general characteristics of English text structure. Firstly, we introduce principle of END-FUCUS. Please look at the following sentence:

Example(3.2.1-1):
Mary invited me to her birthday party. It was held in a big hotel where I met her brother. He was a dentist.

Each sentence of Example (3.2.1-1) ends with a new information (e,g, "her birthday party") and begins with an old information (e,g, "It"). The information value of "new information" is the highest, the information value of "old information" is the lowest. In English, it is common to process information in a sentence "so as to achieve a liner presentation from low to high information value". This is called the principle of END-FOCUS (Quirk et al., 1985)(p.1357).

That is, the change of information value in English text has a regular pattern, which is shown by Figure 3. On the basis of this feature, we decided generation strategies:

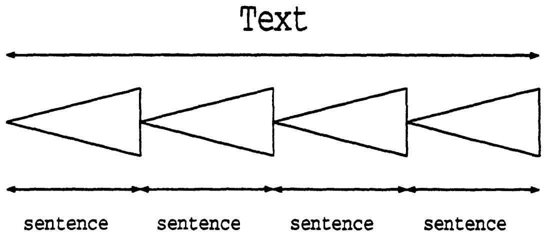


Figure 3: Regular change of Information value in English text.

Strategy 1

The system selects an entity node having high information value from the first generated sentence, and generates the second sentence using the selected node as its subject. For example, if the first sentence is "Tom lives in Canada", the entity nodes which are potential for use as the subject of the second sentence are "Canada" and Tom", we call them "subject candidate". The subject candidates are arranged in order, that is, the entity node who has higher information value is selected first. So, "Canada" is selected by the system first. If no sentence can be generated by using "Canada" as the subject, the system will select the next candidate till the second sentence is generated, as Figure 4 shows. At last, if no sentence can be generated using the first strategy, the system will go to the second strategy.

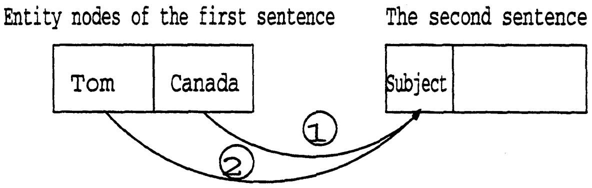


Figure 4: Generation strategy 1.

Strategy 2

The system selects an entity node from the subject candidates, and uses this node as input to generate the second sentence. At this time, the selected entity node is not the subject of the second sentence, as Figure 5 shows. If no sentence can be generated, the system will go to strategy 3.

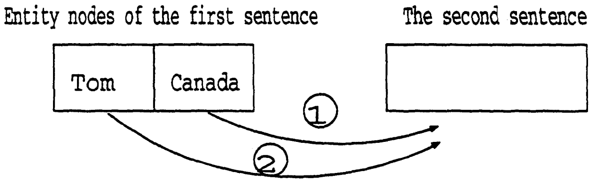


Figure 5: Generation strategy 2.

Strategy 3

The system selects an entity node from the unused entity nodes randomly, and uses this node as input to generate the second sentence. At this time, there is no connection between the generated sentence and its preceding one, as Figure 6 shows:

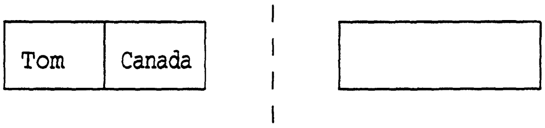


Figure 6: Generation strategy 3.

3.2.2 Attentional state of the readers

Text structure influences on attentional state of the readers. First, let us analyze the following two discourses (Grosz et al., 1995):

(1) a. John went to his favorite music store to buy a piano.
b. He had frequented the store for many years.
c. He was excited that he could finally buy a piano.
d. He arrived just as the store was closing for the day.

(2) a. John went to his favorite music store to buy a piano.
b. It was a store John had frequented for many years.
c. He was excited that he could finally buy a piano.
d. It was closing just as John arrived.

Discourse(1) is more coherent than Discourse(2), though they convey the same information. The main reason that contributes to the difference between these two discourses is: from the perspective of attentional state of the readers, "Discourse(1) centers around a single individual, describing various actions he took and his reactions to them"; "Discourse(2) seems to flip back and forth among several different entities.".and has no single clear center of attention." (Grosz et al., 1995). Concretely speaking, in Discourse(2), the subject (the center of attention) of sentence(2a) is about "John", while the store becomes center in sentence(2b); and then, the center shift to "John" from sentence(2b) to sentence(2c); at last, "the store" becomes central again in sentence(2d).

Figure 7 represents the issue in attentional state by semantic network. Generation algorithm is:

1. Generating all the sentences.
2. Calculating the length of each sentence (node quantity).
3. Ordering the sentences by length, the shorter one is prior. If two sentences have the same length, they are ordered randomly.
4. Enclosing the 3 sentences in brackets. In fact, the brackets will be used while generating pronouns.

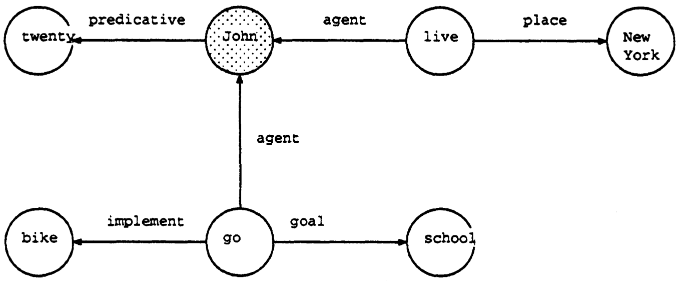


Figure 7: Representing issues in attentional state by semantic network.

Generation result of Figure 7 is:
1. John is twenty.
2. John lives in New York.
3. John goes to school by bike.

3.2.3 Generation strategies about pronouns

Generating pronouns is an important issue in text generation. We introduce two generation strategies about pronouns:

Strategy 1

We use the discourse history to keep the entity nodes that appear in the previous sentence. The discourse history is empty when the system begins, and its content changes during the text being generated. The generation strategy is: if an entity node is last mentioned in the previous sentence, then we substitute a corresponding pronoun for this entity node.

Example(3.2.5-1):
(a). Iceland is very rich in natural heat. Earthquakes are frequent in Iceland.
(b). Iceland is very rich in natural heat. Earthquakes are frequent in it.

Strategy 2

If sentences are encloused in brackets, it means that all of the sentences have the same subject. At this time, we substitute a corresponding pronoun for the subject from the second sentence.

Example(3.2.5-2):
(a). John is twenty. John lives in New York. John goes to school by bike.
(b). John is twenty. He lives in New york. He goes to school by bike.

4 Experiment and generation results

We input the information which is stored in the semantic network (shown by Figure 8) and the information about nouns(e.g. singular, plural, proper noun, etc.). Then the system can generate coherent text according to the input of the user.

Input: Tom
Output:
Tom lives in Canada. He manages three hotels. He goes to church on Sunday. Canada possesses numerous lakes. Many tourists enter it from USA.

Input: Canada
Output:
Canada possesses numerous lakes. Many tourists enter it from USA. Tom lives in Canada. He manages three hotels. He goes to church on Sunday.

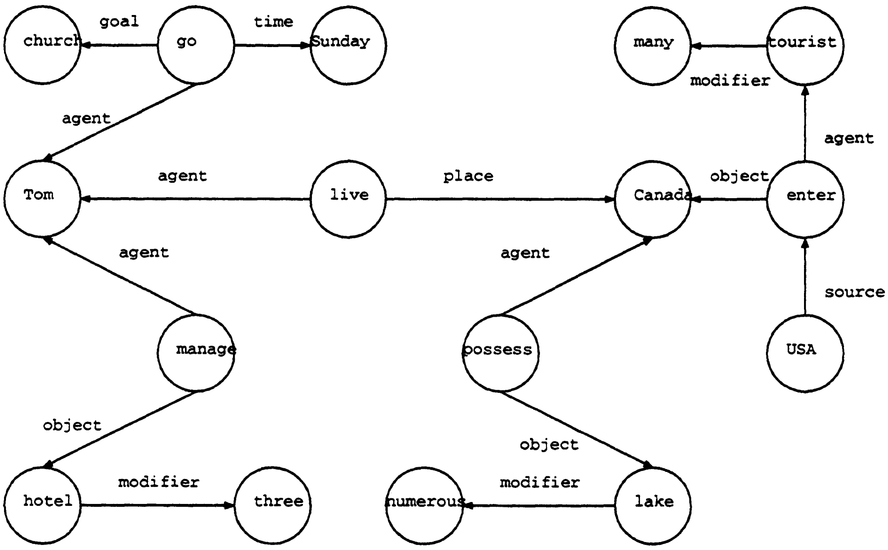


Figure 8: Example of Semantic Network.

5 Future directions

Future work will concentrate on discourse generation, i.e., how to generate sentences containing cue phrases, "because", "after", "before", and so on. On the other hand, we will go on to explore some issues in sentence-level generation, passive structure sentence, paraphrase, aggregation etc. We hope that the generation results of future work will further prove that the method put forward in this paper is effective.

References

Grosz, B., A. Joshi, and S. Weinstein. 1995. Centering: A Framework for Modelling the Local Coherence of Discourse. *Computational Linguistics*, 21(2):203–225.

Ozaki, S., S. Kurohashi, and M. Nagao. 1997. Text generation using semantic network. Technical report, The Institute of Electronics, Information and Communication Engineers, Japan, 7.

Quirk, R., S. Greenbaum, G. Leech, and J. Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman Group Limited, England.

Reiter, E. and R. Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge, UK.

Smith, M. H. 1995. *Natural Language Generation in the LOLITA System: An Engineering Approach*. Ph.D. thesis, Durham University, UK.