Generating easier texts on discourse level

Xinyu Deng and **Jun-ichi Nakamura**
Kyoto University
deng@pine.kuee.kyoto-u.ac.jp

1 Introduction

At present, English has become more and more important for non-native speakers. Therefore, in Natural Language Generation, it is necessary to explore the generation mechanism for non-native users. This paper introduces how to generate texts appropriate for non-native speakers on discourse level. The domain of the texts is *natural and pure science*. Generally, non-native speakers are divided into three levels: primary (middle school level), intermediate (high school level) and advanced (university level). The users of this study are assumed to be at intermediate level.

This study focuses on building a microplanner of a text generation system (the SILK system) whose input is an RST tree (Mann and Thompson, 1988) (Figure 1). In the tree, nonternimal nodes represent discourse relations, terminal nodes represent sentences. The task of the system is to transform text representations from hierarchical tree structures into ordered individual sentences. That is, the system decides how the sentences from the tree will be ordered and punctuated, and selects one text which is appropriate for non-native users. Since the possible combinations of span order and between-text-span punctuation are very huge, the task of generation does not require a global optimum. To some degree, a combination which could be understood by users without difficulties would be enough, e.g., the text shown in Table 1. So we think a Genetic Algorithm is suitable for solving such a problem.

The rest of the paper is arranged as follows. Section 2 introduces the related work. Section 3 shows how to represent a tree by genes. Section 4 describes the features used to evaluate a tree structure. Section 5 draws a conclusion.

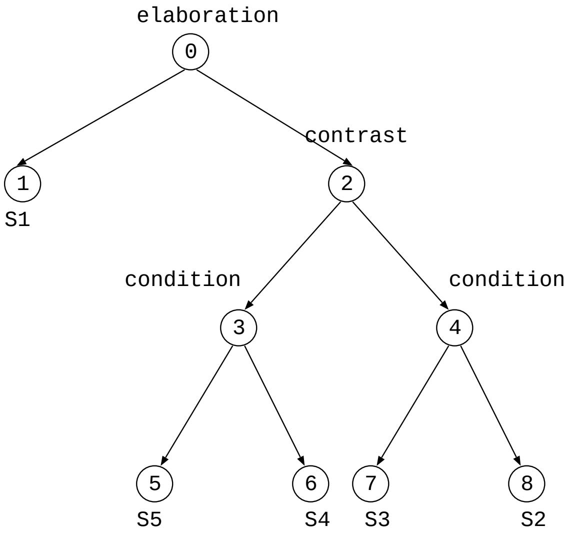

Figure 1: An example of RST tree

> Do things at the right time of year [S1]. If you grow young plants in the fields at the correct time of year [S2], the results will be better [S3]. But if you do things at the wrong time of year [S4], the results will not be so good [S5].

Table 1: A generated result of Figure 1

2 Related work

(Scott and de Souza, 1990) can be regarded as the first study for ease of comprehension in NLG output. The authors pointed out that generating discourse cue phrases whenever possible could make a text easier to comprehend. The PSET project (Devlin et al., 2000) simplified newspaper articles for aphasic readers, which focused on the simplification of syntactic structures and lexical simplification. Furthermore, the GIRL system (Williams, 2003) generates texts for poor readers and good readers of native English speakers at discourse level. To our knowledge, our study is the first one on generating texts for non-native speaker users.

3 Presenting a tree structure by genes

In an RST tree, each non-terminal node is the root of a sub-tree. We assumed that after generated, each sub-tree has two kinds of structures. One is *local structure* in which the nucleus and satellite are considered as two nodes, therefore, the generation result of a sub-tree can be represents as "child node + between-text-span punctuation + child node". Another is *linear structure* which represents ordered individual sentences. For example, in Figure 1, the sub-tree whose root is node 3 has two child nodes: node 5 and node 6. If one *local structure* is "node 6 + comma + node 5", its *linear structure* is "if you do things at the wrong time of year, the result will not be so good."

A *local structure* can be represented by three genes (Figure 2), and each gene has two scores (0 and 1). The first gene represents the position of nucleus: "0" means that nucleus is placed in the first span; "1" means that nucleus is placed in the second span. The second and the third gene represent the between-text-span punctuation: "00" means no punctuation (i.e., space); "01" means a comma; "10" means a full stop; "11" means the beginning of another paragraph. Since we do not consider the problem of generating a text with more than one paragraph, "11" is not desirable, so its score is negative in the experiments.

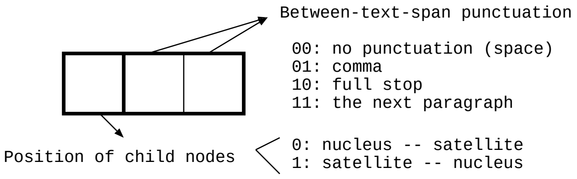

Figure 2: Represent a *local structure* by genes

4 Features used to evaluate a tree structure

A key requirement of the GA approach is the ability to evaluate the quality of a possible solution. In this research, one sub-tree is assigned a score which is the sum of the scores for four particular features the sub-tree may have. The four features are: 1) the position of nucleus in a *local structure*, 2) the between-text-span punctuation in a *local structure*, 3) the complex multiple cue phrases in a *linear structure*, 4) and the punctuation pattern in a *linear structure*. The fitness of a whole candidate tree is the sum of the scores of each sub-tree it has. We think that it is the relative preferences among features rather than the features themselves that determine the ease of a text. In this section, we discuss the preferences among the features mentioned above.

4.1 Feature 1: Position of the nucleus in the *local structure*

In order to investigate the cue usage for non-native speakers, we created corpus CNNSE (Corpus for Non-Native Speakers of English), which contains 200,000 words. The domain of CNNSE is *natural and pure science.* The texts of CNNSE were extracted from the books published in China and in Japan, and the target audience of these books was *high school student.*

At present, the SILK system can generate six discourse relations: namely, "explanation", "contrast", "example", "condition", "elaboration" and "list" relation. We use the cues *because*, *but*, *for example* and *if* to signal the first four relations, and no cue phrase is used to signal the later two. By machine learning program C4.5 and corpus analysis, we obtained the information on nucleus position of each relation. We divided the nucleus position into three possible states.

- State 1: "Good position". It refers to the position which is the result of machine learning. For example, for "condition" relation signaled by cue phrase *if*, the results of machine learning showed that the nucleus is always placed in the second span (i.e., "If you study hard, you can master English"). So the second span is a "good position".

- State 2: "Normal position". It refers to a position which is not a good one but can be found in corpus CNNSE. For example, for "condition" relation signaled by cue phrase *if*, we found that the nucleus sometimes occured in the first span (i.e., "You can master English, if you study hard."). So, the first span is a "normal position".

- State 3: "Bad position". It refers to the position which is neither the results of machine learning nor of corpus analysis. For example, for the "concession" relation signaled by the cue *but*, the nucleus never occurs in the first span (i.e., “But her sister didn’t, Mary went to the Party.”).

Heuristic 1 Preferences among the possible states of nucleus position: good position > normal position > bad position

4.2 Feature 2: Between-text-span punctuation in the *local structure*

We used the machine learning program C4.5 to induce the classification of between-text-span punctuation as well. In fact, between-text-span punctuation is determined by the nucleus position. For example, when the cue phrase *if* occurs in the first span, the between-text-span punctuation should be a comma, otherwise, a comma or a space (no punctuation) is used. Based on the results of machine learning and corpus analysis, we divided the between-text-span punctuation into three possible states.

- State 1: “Good punctuation”. It refers to the punctuation which is the result of machine learning. For example, in the sentence “If you work hard, you can master English”, the comma is a “good punctuation”.
- State 2: “Normal punctuation”. It refers to a punctuation which is not a good one. However, it exists in CNNSE. For example, in the sentence “Mary went to the party, but her sister didn’t.”, the comma is a “normal punctuation”.
- State 3: “Bad punctuation”. It refers to the punctuation which is neither a result of machine learning nor exists in CNNSE. For example, in the sentence “If you work hard. You can master English.” The “full stop” is a “bad punctuation”.

Heuristic 2 Preferences among the possible states of between-text-span punctuation: good punctuation > normal punctuation > bad punctuation

4.3 Feature 3: Multiple complex cue phrases in the *linear structure*

Until now, most of the studies on cue phrases assume that in one sentence there is only one cue phrase which is used to signal a discourse relation. Actually, in an embedded structure, two phrases are sometimes used to signal two discourse relations. This kind of cue phrases are called *complex multiple cue phrases* (CMCPs) (Oates, 2001). In this study, an embedded structure in which CMCPs occur is defined to have two cue phrases and three propositions. CMCPs are divided into two classes (Table 2). *Class 1* represents the first cue phrase immediately precedes the second one and both cue phrases are attached to the second proposition. *Class2* represents the cue phrases precede the second and the third proposition.

– *Class 1* of CMCPs:
You failed the exam. ***But if*** you study hard, you can master English.
– *Class 2* of CMCPs:
You failed the exam. ***But*** you can master English ***if*** you study hard.

Table 2: Two classes of CMCPs

We used a questionnaire to explore if there is significant difference in *comprehensibility* between texts containing *Class 1* and texts containing *Class 2* for non-native speakers. The results indicated that the texts containing *Class 1* are easier to understand.

We divided CMCPs into three states:

- States 1: “Good CMCPs”. They refer to *Class 1* of CMCPs, such as “*but if*”.
- States 2: “Normal CMCPs”. They refer to *Class 2* of CMCPs, for example, “*for example,...if*”.
- States 3: “Bad CMCPs”. They refer to CMCPs which could not be found in CNNSE, for example, “*for example, but*”.

Heuristic 3 Preferences among the three states of CMCPS: good CMCPs > normal CMCPs > bad CMCPs

4.4 Feature 4: Punctuation patterns in the *linear structure*

We focused on comma and the full stop, because they are more often used than any other punctuation marks. Generally, the comma is a useful and valuable punctuation device because it is used to delimit a sentence into more than one parts, which may be words, phrases, clauses, or sentences. We only investigated the comma which is used to delimit sentence. We analysed the first 900 sentences within CNNSE to find the patterns of punctuation used to delimit sentences.

No.	Pattern	Frequency
1	< S >.	653
2	< S >, < S >.	143
3	< S >< S >.	73
4	< S >, < S >, < S >.	18
5	< S >, < S >< S >.	10
6	< S >< S >, < S >.	3
Total		900

Table 3: Patterns of punctuation

The results (Table 3) indicated that pattern 1, 2 and 3 often occur and Pattern 4, 5, and 6 seldom occur. It can infer that Pattern 4, 5, and 6 can not improve ease of texts. We therefore divided the patterns of punctuation into three states:

- State 1: “Good pattern”. It refers to a *linear structure* which contains Pattern 1, 2, or 3. For example, “If your answer is right, you may enter.”.
- State 2: “Normal pattern”. It refers to a *linear structure* which contains Pattern 4, 5, or 6, besides (or without) pattern 1, 2 or 3. For example, “The shape of metals can be changed because the layers of atoms can slide past or over each other. When they do this, some bonds are broken, but an equal number are made.”.
- State 3: “Bad pattern”. It refers to a *linear structure* which contains none of the patterns mentioned in Table 3.

Heuristic 4 Preferences among patterns of punctuation in the *linear structure*: good pattern > normal pattern > bad pattern

4.5 Implementation

We ran the GA algorithm for 5000 iterations on the input like Figure 1 for 10 times, then we chose the text with the highest score, which could be regarded as the best text among the ten generated ones. Figure 3 shows the scores of the best text (2000 iterations). The scores keeps on improving and gets stable at around 1500 iterations. At this moment, the best text (Table 1) was obtained.

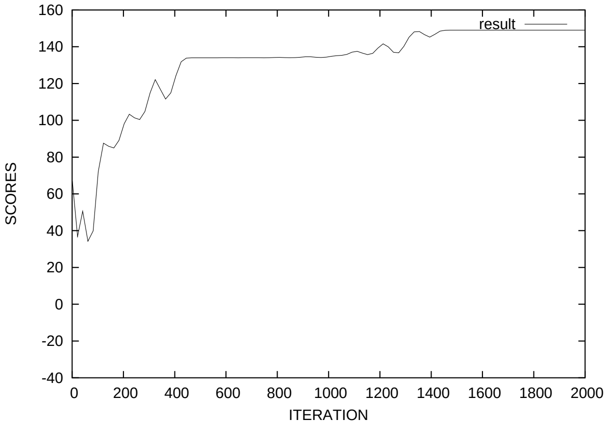

Figure 3: Scores of the best text

5 Conclusion

This paper focuses on generating texts on discourse level for intermediate level non-native speakers. The architecture based on the Genetic Algorithm also can be applied to generating texts satisfying users with different levels, for example, children, middle school student. To realize these aims, it is necessary to do more research on exploring the influence of the features mentioned above on the reading ability of those users.

Reference

William Mann and Sandra Thompson (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3).

Donia Scott and Clarisse de Souza (1990). Getting the message across in RST-based text generation. in *Current Research in Natural Language Generation*, Academic Press.

Siobhan Devlin, Yvonne Canning, John Tait, John Carrol, Guido Minnen, and Darren Pearce (2000). Making Accessible international communication for people with language comprehension difficulties. In *the 7th ICCHP*.

Sandra Williams (2003). Language choice models for microplanning and readability. in *Proceedings of HLT-NAACL Student Workshop*.

Sarah Oates (2001). Generating Multiple Discourse Markers in Text. PhD thesis, University of Brighton.