A Sentence Reduction Using Syntax Control

Nguyen Minh Le
The Graduate School of
Information Science JAIST
Ishikawa, 923-1292, Japan
nguyenml@jaist.ac.jp

Susumu Horiguchi
The Graduate School of
Information Science JAIST
Ishikawa, 923-1292, Japan
hori@jaist.ac.jp

Abstract

This paper present a method based on the behavior of nonnative speaker for reduction sentence in foreign language. We demonstrate an algorithm using semantic information in order to produce two reduced sentences in two difference languages and ensure both grammatical and sentence meaning of the original sentence in reduced sentences. In addition, the orders of reduced sentences are able to be different from original sentences.

1 Introduction

Most of the researches in automatic summarization were focused on extraction or identifying the important clauses and sentences, paragraphs in texts (Inderject Mani and Mark Maybury, 1999). However, when humans produce summaries of documents, they used to create new sentences that are grammatical, that cohere with one another, and capture the most salient parts of information in the original document. Sentence reduction is the problem to remove some redundant words or some phrases from the original sentence by creating a new sentence in which the gist meaning of the original sentence was unchanged.

Methods of sentence reduction have been used in many applications. Grefenstette (G.Grefenstette, 1998) proposed removing phrases in sentences to produce a telegraphic text that can be used to provide audio scanning services for the blind. Dolan (S.H. Olivers and W.B.Dolan, 1999) proposed removing clauses in sentences before indexing document for information retrieval. Those methods remove phrases based on their syntactic categories but not rely on the context of words, phrases and sentences around. Without using that information can be reduced the accuracy of sentence reduction problem. Mani and Maybury also present a process of writing a reduced sentence by reversing the original sentence with a set of revised rules to improve the performance of summarization. (Inderject Mani and Mark Maybury, 1999).

Jing and McKeown(H. Jing, 2000) studied a new method to remove extraneous phrase from sentences by using multiple source of knowledge to decide which phrase in the sentences can be removed. The multiple sources include syntactic knowledge, context information and statistic computed from a corpus that consists of examples written by human professional. Their method prevented removing some phrases that were relative to its context around and produced a grammatical sentence.

Recently, Knight and Marcu(K.Knight and D.Marcu, 2002) demonstrated two methods for sentence compression problem, which are similar to sentence reduction one. They devised both noisy-channel and decision tree approach to the problem. The noisy-channel framework has been used in many applications, including speech recognition, machine translation, and information retrieval. The decision tree approach has been used in parsing sentence. (D. Magerman, 1995)(Ulf Hermijakob and J.Mooney, 1997) to define the rhetorical of text documents (Daniel Marcu, 1999).

Most of the previous methods only produce a short sentence whose word order is the same as that of the original sentence, and in the same language, e.g., English.

When nonnative speaker reduce a long sentence in foreign language, they usually try to link the meaning of words within the original sentence into meanings in their language. In addition, in some cases, the reduced sentence and the original sentence had their word order are difference. Therefore, two reduced sentences are performed by non-native speaker, one is the reduced sentence in foreign language and another is in their language.

Following the behavior of nonnative speaker, two new requirements have been arisen for sentence reduction problem as follows:

1) The word order of the reduced sentence may different from the original sentence.

2) Two reduced sentences in two difference languages can be generated.

With the two new perspectives above, sentence reduction task are useful for many applications such as: information retrieval, query text summarization and especially cross-language information retrieval.

To satisfy these new requirements, we proposed a new algorithm using semantic information to simulate the behavior of nonnative-speaker. The semantic information obtained from the original sentence will be integrated into the syntax tree through syntax control. The remainder of this paper will be organized as follows: Section 2 demonstrated a method using syntactic control to reduced sentences. Section 3 shows implementation and experiments. Section 4 gives some conclusions and remained problems to be solved in future.

2 Sentence reduction using syntax control

2.1 Formulation

Let E and V be two difference languages. Given a long sentence $e : e_1, e_2, ..., e_n$ in the language E. The task of sentence reduction into two languages E and V is to remove or replace some redundant words in the sentence e to generate two new sentences $e'_1, e'_2, ..., e'_m$ and $v_1, v_2, ..., v_k$ in language E and V so that their gist meanings are unchanged.

In practice, we used English language as a source language and the target language are in English and Vietnamese. However, the reader should understand that our method can apply for any pair of languages. In the following part we present an algorithm of sentence reduction using syntax control with rich semantic information.

2.2 Sentence reduction algorithm

We present an algorithm based on a semantic parsing in order to generate two short sentences into difference languages. There are three steps in a reduction algorithm using syntax control. In the first step, the input sentence e will be parsed into a syntax tree t through a syntax parser.

In the second step, the syntax tree will be added rich semantic information by using a semantic parser, in which each node of the syntax tree is associated with a specific syntax control. The final step is a process of generating two deference sentences into language E and V language from the syntax tree t that has been annotated with rich semantic information.

2.2.1 Syntax parsing

First, We parse a sentence into a syntax tree. Our syntax parser locates the subject, object, and head word within a sentence. It also recognizes phrase verbs, cue phases or expressions in English sentences. These are useful information to reduce sentence. The Figure 2 explains the equivalent of our grammar symbol with English grammar symbol.

Figure 1 shows an example of our syntax parsing for the sentence "Like FaceLift, much of ATM's screen performance depends on the underlying application".

To reduce the ambiguity, we design a syntactic parsing base on grammar symbols, which classified in detail. Part of speech of words was extended to cope with the ambiguity problem. For example, in Figure 2, "noun" was dived into "private noun" and "general noun".

The bilingual dictionary was built including about 200,000 words in English and its meaning in Vietnamese. Each English word entry includes several meanings in Vietnamese and each meaning was associated with a symbol meaning. The set of symbol meanings in each word entry is defined by using WordNet database.(C. Fellbaum, 1998) The dictionary also contained several phrases, expressions in

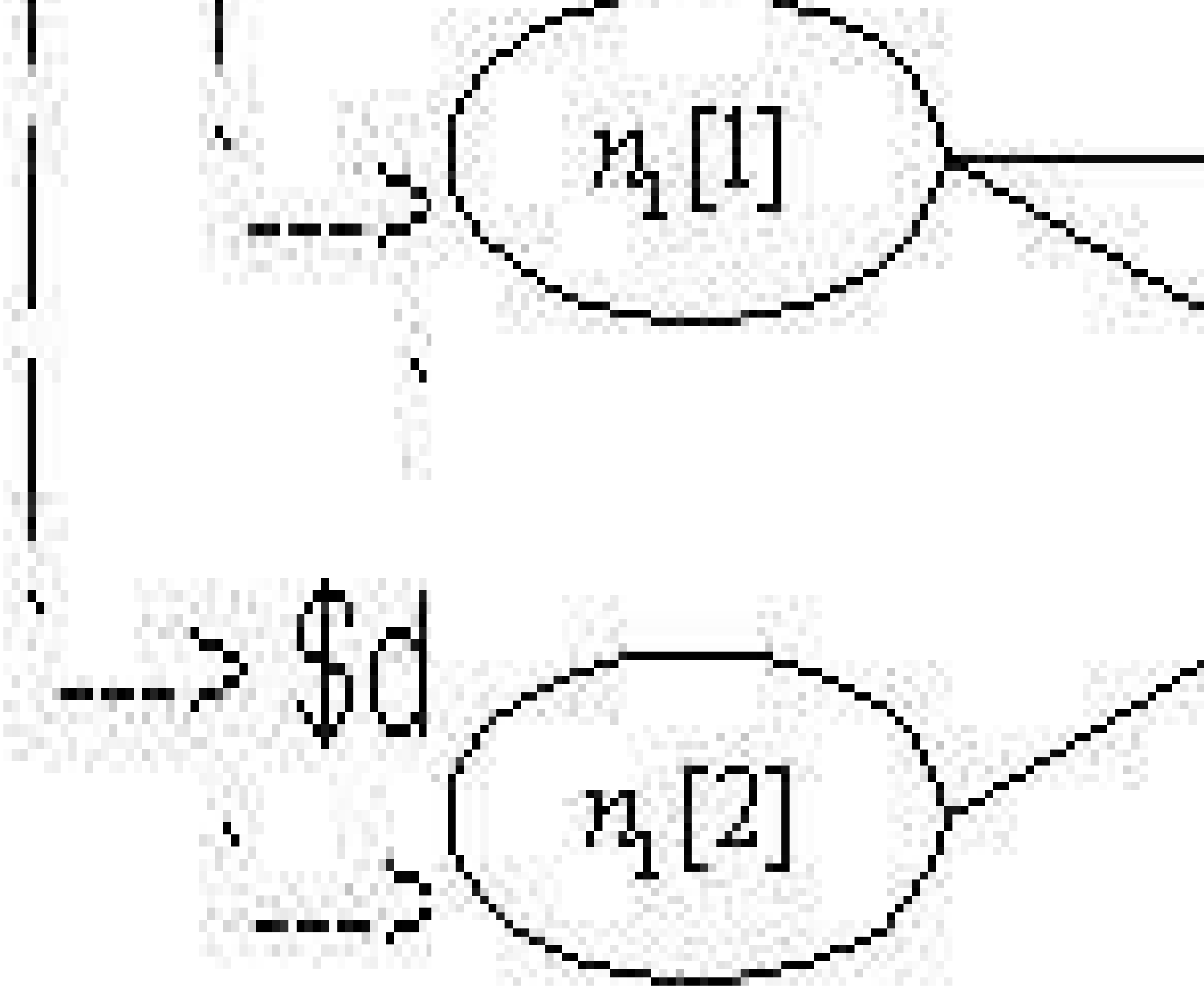

Figure 1: An example of syntax tree of "Like FaceLift, much of ATM's screen performance depends on the underlying application"

English and its equivalent to Vietnamese.

2.2.2 Semantic parsing using syntax control

After producing a syntax tree with rich information, we continue to apply a semantic parsing for that syntax tree.
Let N be an internal node of the syntax tree t and N has k children nodes: $n_1, n_2, ...n_k$.

The node N based on semantic information from its n children nodes to consider what the remained part in the reducing sentence should be.

When parsing semantic for the syntax tree t, each N must be used the information of children nodes to define its information. We call that information is semantic-information of the node N and define it as $N.sem$. In addition, each semantic-information of a given node N was mapped with a meaning in the

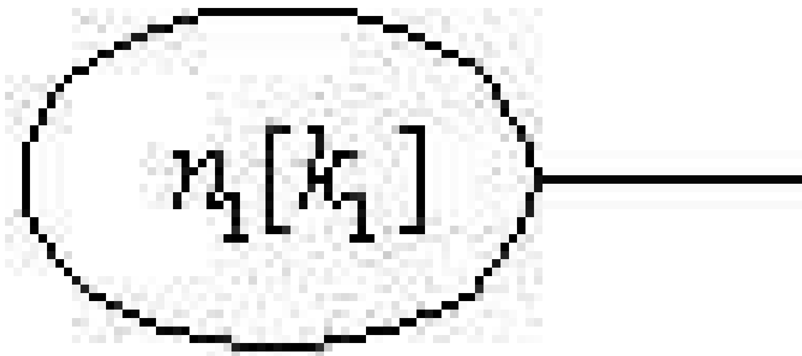

Figure 3: Syntax control

Figure 3 shows two choices for sequence meanings of the node N in a reduction process .
It is easy for human to understand exactly which

meaning of n_i should be and then decoding them as objects to memorize. With this basic idea, we design a control language to do this task.

The k children nodes $n_1, n_2, ...n_k$ are associated with a set of a syntax control to conduct the reducing sentence process. The node N and its children are associated with a set of rules. To present the set of rules we used a simple syntax of a control language as follows:
1) Syntax to present the order of children nodes and nodes to be removed.
2) Syntax to constraint each meaning of a children node with meanings of other children nodes.
3) Syntax to combine sequence meanings into one symbol meaning (this process called a inherit process from the node N to its children).
A syntax rule control will be encoded as one-generation rules and a set of condition rules so that the generation rule has to satisfy. With a specification condition rule, we can define its generation rule directly.

Condition rule

A condition rule is formulated as follows: if $n_{j_1}.sem = v_1 \wedge n_{j_2}.sem = v_2 ... \wedge n_{j_m}.sem = v_m$ then $N.sem = v$ with v and $v_j \in SI$

Generation rule

A generation rule is a sequence of symbols in order to transfer the internal node N into the internal node of a reduced sentence. We used two generation rules, one for E and other one for V . Given a sequence symbols $g : g_1 g_2 ... g_m$, in which g_i is an integer or a string. The equation $g_i = j$ means the children node be remained at position j in the target node. If $g_i = "v_1 v_2 ... v_l"$, we have that string will in the children node n_i of the target node.
Figure 1 shows a syntax tree of the input sentence: "Much of ATM's performance depends on the underlying application.". In this syntax tree, the syntax rule:*"S1=Bng-daucau Subj cdgt Bng-cuoicau"* will be used the syntax control bellow to reduce
$< Con > default < /Con >$
$< Gen > 1\ 2 < /Gen >$
The condition rule is "default" mean the generation rule is applied to any condition rule. The generation rule be "1 2" mean only the node (Subj) in the index 1 and the node (cdgt) in the index 2 of the rule *"S1=Bng-daucau Subj cdgt Bng-cuoicau"* are remained in the reduced sentence.

If the syntax control is changed to
$< Con > Subj = HUMAN < /Con >$
$< Gen > 1\ 2 < /Gen >$
This condition rule means that only the case the semantic information in the children node "$Subj$" is "$HUMAN$" the generation rule "1 2" is applied for reduction process. Using the default condition rule the reduced sentences to be generated as follows.
Original sentence: *Like FaceLift, much of ATM's screen performance depends on the underlying application.*
Reduced sentence in English: *Much of ATM's performance depends on the underlying application.*
Reduced sentence in Vietnamese: *Nhieu hieu suat cua ATM phu thuoc vao nhung ung dung tiem an.*

In order to generating reduced sentence in Vietnamese language, the condition rule and generation is also designed. This process is used the same way as transfer translation method.
Because the gist meaning of a short sentence is unchanged in comparing with the original sentence, the gist meaning of a node after applying the syntax control will be unchanged. With this assumption, we can reuse the syntax control for translating the original sentence into other languages (English into Vietnamese) for translating the reduced sentence. Therefore, our sentence reduction program can produce two reduced sentences in two difference languages. Our semantic parsing used that set of rules to select suitable rules for the current context. The problem of selecting a set of suitable rules for the current context of the current node N is to find the most likely condition rule among the set of syntax control rules that associated with it. Thus, semantic parsing using syntax control problem can be described mathematically as follows:

Given a sequence of children nodes $n_1, n_2, ..., n_k$ of a node N, each node n_i consist of a list of meaning, in which each meaning was associated with a symbol meaning. The syntax rule for the node N was associated with a set of condition rules. In addition, one condition rule is mapped with a specification generation rule.
Find the most condition rules for that node sequences.
This problem can be solved by using a variant of the

Viterbi algorithm (A.J. Viterbi, 1967).
Firstly, we define each semantic-information of a children node with all index condition rules. Secondly, we try to find all sequences that come from the same condition rules.

Algorithm 1 A definition of condition rules algorithm. FindRule(N)

Require: Input: N is a node
Ensure: A syntax control for a rule
{Initialization step:}
for $i = 1$ to k **do**
  for $j = 1$ to K_i **do**
    Set stack $s[i]$=all index rules in the set of condition rules satisfy $n_i.sem = n_i[j]$
  end for
  for $i = 1$ to K_1 **do**
    $Cost[0][i] = 1$;
    $Back[0][i] = 0$;
  end for
end for
{Interaction step:}
for $i = 1$ to k **do**
  for j=1 to K_i **do**
    $Cost[i][j] = \max Cost[i-1][l] \times Value(s[i][j], s[i-1][l])$ with $l = 1, K_i$
    $Back[i][j]$= all the index gave max
  end for
end for
{Identification step:}
Set a list LS= all index rules gave max values $Cost[k][j]$ with $j = 1, K_k$.
Update all semantic-information of each condition rule in the list LS to node N.
Function Value (i, j)
Begin
If i==j return 2;
Else return 1;
End

After defining a set of semantic-information for each internal node, we have a frame of semantic parsing algorithm as shown in Algorithm 2. Our semantic parsing using syntax control is fast because of finding syntax control rule for each node tree is applied dynamic programming.

Algorithm 2 Semantic parsing algorithm

Require: Given a syntax tree , a set of syntax control for each node of the syntax tree.
Ensure: a syntax tree with rich semantic information
{SemanticParsingTree}
if N is leaf **then**
  Update all symbol-meaning in word entry
else
  FindRules(N);
end if{main procedure}
SemanticParsingNode(root);

2.2.3 Generation reduced sentences

The input of this process is a syntax tree which associated with rich information after applying the semantic parsing process. Browsing the syntax tree following bottom-up process, in which, a node tree can be generated a short sub-sentence by using the corresponding generation rule. Because we have two generation rules for each node tree, so we have two reduced sentences in two difference languages.

3 Experiments and Discussion

3.1 Experiment Data

We used the same corpus(K.Knight and D.Marcu, 2002) with 1067 pair of sentence and its reduction. We manually changed the order of some reduced sentences in that corpus while keep their meaning. We manually build a set of syntax control for that corpus for our reduction algorithm using syntax control. The set of semantic symbols was described such as, HUMAN, ANIMAL, THINGS, etc. We make 100 pair of sentences with the order of a reduction sentence is different from its original sentence. Afterward, those sentences are to be combined with the corpus above in order to confirm that our method can deal with the changeable word order problem.

3.2 Experiment Method

To evaluate our reduction algorithms, we randomly selected 32 pair of sentences from our parallel corpus, which will refer to as the Test corpus. We used 1035 sentence pairs for training with the reduction based decision tree algorithm. We used test corpus to confirm that our methods using semantic-information will outperform than the decision tree method without semantic-information (K.Knight and D.Marcu, 2002). We presented each original sentence in the test corpus to three judges who are Vietnamese and specialize in English, together with three sentence reductions of it: The human generated reduction sentence, the outputs of the sentence reduction based syntax control and the output of the baseline algorithm. The judges were told that all outputs were generated automatically. The order of the outputs was scrambled randomly across test cases. The judges participated in two experiments. In the first experiment, they were asked to determine on a scale from 1 to 10 how well the systems did with respect to selecting the most important words in the original sentence. In the second experiment, they were asked to determine on a scale from 1 to 10 how grammatical the outputs were. The outputs of our methods include both reduced sentences in English and Vietnamese. In the third experiment, we tested on the randomly of 32 sentences from 100 sentences whose had word order between input and output are different.

3.3 Experiment Results

Using the first and the second experiment method, we had two table results as follows.

Table 1: Experiment results with outputs in English

Method	comp	Grammatically	Importance
Baseline	57.19	8.6 ± 2.8	7.18 ± 1.92
Syn.con	6.5	8.7 ± 1.2	7.3 ± 1.6
Human	53.33	9.05 ± 0.3	8.5 ± 0.8

Table 2: Experiment results with outputs in Vietnamese

Method	comp	Grammatically	Importance
Baseline	x	x	x
Syn.con	67	6.5 ± 1.7	6 ± 1.3
Human	63	8.5 ± 0.3	8.7 ± 0.7

Using the third experiments method we achieved a result to be shown in Table5.

Table 3: Experiment results with the changeable order

Method	comp	Grammatically	Importance
Baseline	56.2	7.4 ± 3.1	6.5 ± 1.3
Syn.con	66	8.4 ± 2.1	7.2 ± 1.7
Human	53.33	9.2 ± 0.3	8.5 ± 0.8

3.4 Discussion

Table 1 shows the compression of three reduction methods in comparing with human for English language. The grammatically of semantic control achieved a high results because we used the syntax control from human expert. The sentence reduction decision based is yielded a smallest result. We suspect that the requirement of word order may affect the grammatically. Table 1 and Table 3 also indicates that our new method achieved the importance of words are outperform than the baseline algorithm due to semantic information. This was because our method using semantic information to avoid deleting important words. Following our point, the base line method should integrate with semantic information within the original sentence to enhance the accuracy.

Table 2 shows the outputs of our method into Vietnamese language, the baseline method cannot generate the output into Vietnamese language. The syntax control method achieved a good enough results in both grammatically and importance aspects.

The comparison row in the Table 1 and the Table 2 also reported that the baseline yields a shorter output than syntax control method.

Table 3 shows that when we selected randomly 32 sentence pairs from 100 pairs of sentences those had words order between input and output are different, we have the syntax method change a bit while the baseline method achieved a low result. This is due to the syntax control method using rule knowledge based while the baseline was not able to learn with that corpus that.

4 Conclusions

We have presented an algorithm that allows rewriting a long sentence into two reduced sentences in two difference languages. We compared our methods with the other methods to show advantages as well as limits of the method. We claim that the semantic information of the original sentence through using syntax control is very useful for sentence reduction problem.

We proposed a method for sentence reduction using semantic information and syntactic parsing so-called syntax control approach. Our method achieved a higher accuracy and the outputted reduction sentences in two different languages e.g. English and Vietnamese. Thus, it is closed to the outputs of non-native speaker in reduction manner.

Investigate machine learning to generate syntax control rules automatically from corpus available are promising to enhance the accuracy of sentence reduction using syntax control .

Acknowledgements

We would like to thank to Dr. Daniel Marcus about the data corpus for sentence reduction task. This research was supported in part by the international research project grant, JAIST.

References

Indeject Mani and Mark Maybury. 1999. Advances in Automatic Text Summarization. *The MIT press*.

G.Grefenstette. 1998. Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind. *In Working notes of the AAAI Spring Symposium on Intelligent Text summarization*, pp.111-118.

S.H.Olivers and W.B.Dolan. 1999. Less is more; eliminating index terms from subordinate clauses. *In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistic*, pp.349-356.

H.Jing. 2000. Sentence reduction for automatic text summarization. *In Proceeding of the First Annual Meeting of the North American Chapter of the Association for Computational Linguistics NAACL-2000*.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: A Probabilistic approach to sentence compression. *Artificial Intelligent*, 139: 91–107.

D. Magerman. 1995. Statistical decision tree models for parsing. *In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistic*, pp.276-283.

Ulf Hermijakob and Raymond J. Mooney. 1997. Learning parse and translation decision from examples with rich context. *In Proceeding of ACL/EACL'97*, pp 482-489.

Daniel Marcu. 1999. A decision- based approach to Rhetorical parsing. *In Proc. Of ACL'99*, pp.365-372.

C.Fellbaum. 1998. WORDNET: An Electronic Lexical Database. *The Mit Press*.

A.J.Viterbi. 1967. Error bounds for convolution codes and an asymptotically optimal decoding algorithm. *IEEE Trans on Information Theory*,13: 260–269.