A Transformation-based Sentence Splitting Method for Statistical Machine Translation

Jonghoon Lee, Donghyeon Lee and Gary Geunbae Lee
Department of Computer Science and Engineering
Pohang University of Science & Technology (POSTECH)
{jh21983, semko, gblee}@postech.ac.kr

Abstract

We propose a transformation based sentence splitting method for statistical machine translation. Transformations are expanded to improve machine translation quality after automatically obtained from manually split corpus. Through a series of experiments we show that the transformation based sentence splitting is effective pre-processing to long sentence translation.

1 Introduction

Statistical approaches to machine translation have been studied actively, after the formalism of statistical machine translation (SMT) is proposed by Brown et al. (1993). Although many approaches of them were effective, there are still lots of problems to solve. Among others, we have an interest in the problems occurring with long sentence decoding. Various problems occur when we try to translate long input sentences because a longer sentence contains more possibilities of selecting translation options and reordering phrases. However, reordering models in traditional phrase-based systems are not sufficient to treat such complex cases when we translate long sentences (Koehn et al, 2003).

Some methods which can offer powerful reordering policies have been proposed like syntax based machine translation (Yamada and Knight, 2001) and Inversion Transduction Grammar (Wu, 1997). Although these approaches are effective, decoding long sentences is still difficult due to their computational complexity. As the length of an input sentence becomes longer, the analysis and decoding become more complex. The complexity causes approximations and errors inevitable during the decoding search.

In order to reduce this kind of difficulty caused by the complexity, a long sentence can be paraphrased by several shorter sentences with the same meaning. Generally, however, decomposing a complex sentence into sub-sentences requires information of the sentence structures which can be obtained by syntactic or semantic analysis. Unfortunately, the high level syntactic and semantic analysis can be erroneous and costs as expensive as SMT itself. So, we don’t want to fully analyze the sentences to get a series of sub-sentences, and our approach to this problem considers splitting only compound sentences.

In the past years, many research works were concerned with sentence splitting methods to improve machine translation quality. This idea had been used in speech translation (Furuse et al, 1998) and example based machine translation (Doi and Sumita, 2004). These research works achieved meaningful results in terms of machine translation quality. Unfortunately, however, the method of Doi and Sumita using n-gram is not available if the source language is Korean. In Korean language, most of sentences have special form of ending morphemes at the end. For that reason, we should determine not only the splitting position but also the ending morphemes that we should replace instead of connecting morphemes. And the Furuse et al’s method involves parsing which requires heavy cost.

In this paper we propose a transformation based splitting method to improve machine translation quality which can be applied to the translation tasks with Korean as a source language.

2 Methods

Our task is splitting a long compound sentence into short sub-sentences to improve the performance of phrase-based statistical machine translation system. We use a transformation based approach to accomplish our goal.

2.1 A Concept of Transformation

The transformation based learning (TBL) is a kind of rule learning methods. The formalism of TBL is introduced by Brill (1995). In past years, the TBL approach was used to solve various problems in natural language processing such as part of speech (POS) tagging and parsing (Brill, 1993).

A transformation consists of two parts: a triggering environment and a rewriting rule. And the rewriting rule consists of a source pattern and a target pattern. Our consideration is how to get the right transformations and apply them to split the long sentences.

A transformation works in the following manner; some portion of the input is changed by the rewriting rule if the input meets a condition specified in the triggering environment. The rewriting rule finds the source pattern in the input and replaces it with the target pattern. For example, suppose that a transformation which have a triggering environment A, source pattern B and target pattern C. We can describe this transformation as a sentence: if a condition A is satisfied by an input sentence, then replace pattern B in the input sentence with pattern C.

2.2 A Transformation Based Sentence Splitting Method

Normally, we have two choices when there are two or more transformations available for an input pattern at the same time. The first choice is applying the transformation one by one, and the second choice is applying them simultaneously. The choice is up to the characteristics of the problem that we want to solve. In our problem, we choose the former strategy which is applying the transformations one by one, because it gives direct intuition about the process of splitting sentences. By choosing this strategy, we can design splitting process as a recursive algorithm.

At first, we try to split an input sentence into two sub-sentences. If the sentence has been split by some transformation, the result involves exactly two sub-sentences. And then we try to split each sub-sentence again. We repeat this process in recursive manner until no sub-sentences are split.

In the above process, a sentence is split into at most two sub-sentences through a single trial. In a single trial, a transformation works in the following manner: If an input sentence satisfies the environment, we substitute the source pattern into the target pattern. That is, replace the connecting morphemes with the proper ending morphemes. And then we split the sentence with pre-defined position in the transformation. And finally, we insert the junction word that is also pre-defined in the transformation between the split sentences after the sub sentences are translated independently.

From the above process, we can notice easily that a transformation for sentence splitting consists of the four components: a triggering environment, a rewriting rule, a splitting position and a junction type. The contents of each component are as follows. (1) A triggering environment contains a sequence of morphemes with their POS tags. (2) A rewriting consists of a pair of sequences of POS tagged morphemes. (3) A junction type can have one of four types: ‘and’, ‘or’, ‘but’ and ‘NULL’. (4) A splitting position is a non-negative integer that means the position of starting word of second sub-sentence.

2.3 Learning the Transformation for Sentence Splitting

At the training phase, TBL process determines the order of application (or rank) of the transformations to minimize the error-rate defined by a specific measure. The order is determined by choosing the best rule for a given situation and applying the best rule for each situation iteratively. In the sentence splitting task, we maximize the machine translation quality with BLEU score (Papineni et al., 2001) instead of minimizing the error of sentence splitting.

During the training phase, we determine the order of applying transformation after we build a set of transformations. To build the set of transformations, we need manually split examples to learn the transformations.

Building a transformation starts from extracting a rewriting rule by calculating edit-distance matrix between an original sentence and its split form from the corpus. We can easily extract the different parts from the matrix.

```
BaseBLEU := BLEU score of the baseline system
S := Split example sentence
T := Extracted initial transformation
for each t∈ T
    for each s∈S
        while true
            try to split s with t
            if mis-splitting is occurred
                Expand environment
            else exit while loop
            if environment cannot be expanded
                exit while loop
    S' := apply t to S
    Decode S'
    BLEU := measure BLEU
    Discard t if BLEU < BaseBLEU
sort T w.r.t. BLEU
```

Figure 1. Modified TBL for sentence splitting

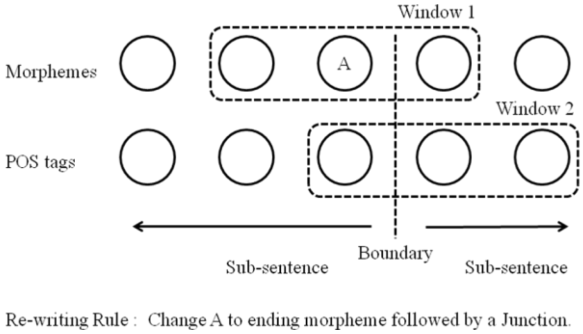
Window 1
Morphemes
A
Window 2
POS tags
Boundary
Sub-sentence
Sub-sentence
Re-writing Rule : Change A to ending morpheme followed by a Junction.

Figure 2. Window-based heuristics for triggering environments

From the difference pattern, we can make the source pattern of a rewriting rule by taking the different parts of the original sentence side. Similarly, the target pattern can be obtained from the different parts of split form. And the junction type and splitting position are directly obtained from the difference pattern. Finally, the transformation is completed by setting the triggering environment as same to the source pattern. The set of initial transformations is obtained by repeating this process on all the examples.

The Transformations for sentence splitting are built from the initial transformations through expanding process. In the expanding process, each rule is applied to the split examples. We expand the triggering environment with some heuristics (in section 2.4), if a sentence is a mis-split.

And finally, in order to determine the rank of each transformation, we sorted the extracted transformations by decreasing order of resulted BLEU scores after applying the transformation to each training sentence. And some transformations are discarded if they decrease the BLEU score. This process is different from original TBL. The modified TBL learning process is described in figure 1.

2.4 Expanding Triggering Environments

Expanding environment should be treated very carefully. If the environment is too specific, the transformation cannot be used in real situation. On the other hand, if it is too general, then the transformation becomes erroneous.

Our main strategy for expanding the environment is to increase context window size of the triggering environment one by one until it causes no error on the training sentences. In this manner, we can get minimal error-free transformations on the sentence splitting corpus.

We use two different windows to define a triggering environment: one for morpheme and another for its part of speech (POS) tag. Figure 2 shows this concept of two windows. The circles correspond to sequences of morphemes and POS tags in a splitting example. Window 1 represents a morpheme context and window 2 represents a POS tag context. The windows are independently expanded from the initial environment which consists of a morpheme 'A' and its POS tag. In the figure, window 1 is expanded to one forward morpheme and one backward morpheme while window 2 is expanded to two backward POS tags.

In order to control these windows, we defined some heuristics by specifying the following three policies of expanding windows: no expansion, forward only and forward and backward. From those three polices, we have 9 combinations of heuristics because we have two windows. By observing the behavior of these heuristics, we can estimate what kind of information is most important to determine the triggering environment.

		SMT		Splitting	
		Korean	English	Before Split	After Split
Train	# of Sentences	123,425		1,577	1,906
	# of Words	1,083,912	916,950	19,918	20,243
	Vocabulary	15,002	14,242	1,956	1,952
Test	#of Sentences	1,577		-	-

Table 1. Corpus statistics

Test No.	Window1 policy	Window2 policy
Test 1	No expansion	No expansion
Test 2		Forward only
Test 3		Free expansion
Test 4	Forward only	No expansion
Test 5		Forward only
Test 6		Free expansion
Test 7	Free expansion	No expansion
Test 8		Forward only
Test 9		Free expansion

Table 2.Experimental setup

Test No.	# of affected sentences	BLEU score	
		Before splitting	After splitting
Test 1	209	0.1778	0.1838
Test 2	142	0.1564	0.1846
Test 3	110	0.1634	0.1863
Test 4	9	0.1871	0.2150
Test 5	96	0.1398	0.1682
Test 6	100	0.1452	0.1699
Test 7	8	0.2122	0.2433
Test 8	157	0.1515	0.1727
Test 9	98	0.1409	0.1664

Table 3. BLEU scores of affected sentences

We have at most 4 choices for a single step of the expanding procedure: forward morpheme, backward morpheme, forward POS tag, and backward POS tag. We choose one of them in a fixed order: forward POS tag, forward morpheme, backward POS tag and backward morpheme. These choices can be limited by 9 heuristics. For example, suppose that we use a heuristic with forward policy on morpheme context window and no expansion policy for POS tag context window. In this case we have only one choice: forward morpheme.

3 Experiments

We performed a series of experiments on Korean to English translation task to see how the sentence splitting affects machine translation quality and which heuristics are the best. Our baseline system built with Pharaoh (Koehn, 2004) which is most popular phrase-based decoder. And trigram language model with KN-discounting (Kneser and Ney, 1995) built by SRILM toolkit (Stolcke, 2002) is used.

Table 1 shows the corpus statistics used in the experiments. The training corpus for MT system has been built by manually translating Korean sentences which are collected from various sources. We built 123,425 sentence pairs for training SMT, 1,577 pairs for splitting and another 1,577 pairs for testing. The domain of the text is daily conversations and travel expressions. The sentence splitting corpus has been built by extracting long sentences from the source-side mono-lingual corpus. The sentences in the splitting corpus have been manually split.

The experimental settings for comparing 9 heuristics described in the section 2.4 are listed in table 2. Each experiment corresponds to a heuristic.

To see the effect of sentence splitting on translation quality, we evaluated BLEU score for affected sentenced by the splitting. The results are shown in table 3. Each test number shows the effect of transformation-based sentence splitting with different window selection heuristics listed in table 2. The scores are consistently increased with significant differences. After analyzing the results of table 3, we notice that we can expect some perfor-

Test No.	# of transformations (rules)	# of changes (sentences)	# of superior changes	# of inferior changes	# of insignificant changes	Ratio Sup/Inf	Ratio trans/change
1	34	209	60	30	119	2.00	6.15
2	177	142	43	9	90	4.78	0.802
3	213	110	29	9	72	3.22	0.516
4	287	9	4	1	4	4.00	0.031
5	206	96	25	4	67	6.25	0.466
6	209	100	23	8	69	2.88	0.478
7	256	8	3	1	4	3.00	0.031
8	177	157	42	10	102	4.20	0.887
9	210	98	21	4	73	5.25	0.467

Table 4. Human evaluation results

Superior change	Reference	I saw that some items are on sale on window . what are they ?
	Baseline	What kind of items do you have this item in OOV some discount, I get a discount ?
	Split	You have this item in OOV some discount . what kind of items do I get a discount ?
Insignificant change	Reference	What is necessary to be issued a new credit card?
	Baseline	I ‘d like to make a credit card . What do I need?
	Split	I ‘d like to make a credit card . What is necessary?
Inferior change	Reference	I ‘d like to make a reservation by phone and tell me the phone number please .
	Baseline	I ‘d like to make a reservation but can you tell me the phone number , please .
	Split	I ‘d like to make a reservation . can you tell me the , please .

Table 5. Example translations (The sentences are manually re-cased for readability)

mance gain when the average sentence length is long.

The human evaluation shows more promising results in table 4. In the table, the superior change means that the splitting results in better translation and inferior means the opposite case. Two ratios are calculated to see the effects of sentence splitting. The ratio ‘sup/inf’ shows the ratio of superior over inferior splitting. And ratio trans/change shows how many sentences are affected by a transformation in an average. In most of the experiments, the number of superior splitting is over three times larger than that of inferior ones. This result means that the sentence splitting is a helpful pre-processing for machine translation.

We listed some example translations affected by sentence splitting in the table 5. In the three cases, junction words don’t appear in the results of translation after split because their junction types are NULL that involves no junction word. Although several kinds of improvements are observed in superior cases, the most interesting case occurs in out-of-vocabulary (OOV) cases. A translation result has a tendency to be a word salad when OOV’s are included in the input sentence. In this case, the whole sentence may lose its original meaning in the result of translation. But after splitting the input sentence, the OOV’s have a high chance to be located in one of the split sub-sentences. Then the translation result can save at least a part of its original meaning. This case occurs easily if an input sentence includes only one OOV. The Superior change of table 5 is the case. Although both baseline and split are far from the reference, split catches some portion of the meaning.

Most of the Inferior cases are caused by mis-splitting. Mis-splitting includes a case of splitting a sentence that should not be split or splitting a sentence on the wrong position. This case can be reduced by controlling the heuristics described in section 2.4. But the problem is that the effort to reducing inferior cases also reduces the superior cases. To compare the heuristics each other in this condition, we calculated the ratio of superior and inferior cases. The best heuristic is test no. 5 in terms of the ratio of sup/inf.

The test no. 4 and 7 show that a trans-formation becomes very specific when lexical information is used alone. Hence the ratio trans/change becomes below 0.01 in this case. And test no. 1 shows that the transformations with no environment expansion are erroneous since it has the lowest ratio of sup/inf.

4 Conclusion

We introduced a transformation based sentence splitting method for machine translation as a effective and efficient pre-processing. A transformation consists of a triggering environment and a rewriting rule with position and junction type information. The triggering environment of a transformation is extended to be error-free with respect to training corpus after a rewriting rule is extracted from manually split examples. The expanding process for the transformation can be generalized by adding POS tag information into the triggering environment.

The experimental results show that the effect of splitting is clear in terms of both automatic evaluation metric and human evaluation. The results consistently state that the statistical machine translation quality can be improved by transformation based sentence splitting method.

Acknowledgments

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment) (IITA-2006-C1090-0603-0045). The parallel corpus was courteously provided by Infinity Telecom, Inc.

References

Eric Brill. 1993. Transformation-based error-driven parsing. *In Proc. of third International Workshop on Parsing.*

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics 21(4):543-565.*

Peter F. Brown, Stephen A. Della Pietra, Vincent J.Della Pietra and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter estimation. *Computational Linguistics, 19(2):263-312.*

Takao Doi and Eiichiro Sumita. 2004. Splitting input sentence for machine translation using language model with sentence similarity. *In Proc. of the 20th international conference on Computational Linguistics.*

Osamu Furuse, Setsuo Yamada and Kazuhide Yamamoto. 1998. Splitting Long or Ill-formed Input for Robust Spoken-language Translation. *In Proc of the 36th annual meeting on Association for Computational Linguistics.*

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram lnguage modeling. *In Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP).*

Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. *In Proc. of the 6th Conference of the Association for Machine translation in the Americas*.

Philipp Koehn, Franz Josef Och and Kevin Knight. 2003. Statistical Phrase-Based Translation. *In Proc of the of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: A method for automatic evaluation of Machine Translation. *Technical Report RC22176, IBM.*

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. *In Proc. of the 7th International Conference on Spoken Language Processing (ICSLP).*

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics 23(3):377-404.*

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation Model. *In Proc. of the conference of the Association for Computational Linguistics (ACL).*