Splitting Long Input Sentences for Phrase-based Statistical Machine Translation

Chooi-Ling Goh and Eiichiro Sumita
Language Translation Group, MASTAR Project
National Institute of Information and Communications Technology, 619-0289 Kyoto
{chooiling.goh,eiichiro.sumita}@nict.go.jp

Abstract

Translation results suffer when a standard phrase-based statistical machine translation system is used for translating long sentences. The translation output will not have the same word order as the source. When a sentence is long, it should be partitioned into several clauses, and the word reordering during the translation done within these clauses, not between the clauses. In this paper, we propose splitting the long sentences using linguistic information, and translating the sentence piece by piece. In other words, we constrain the word reordering so that it can only be done within the pieces but not between the pieces. We then apply a language model to join the pieces back together in the original sequence in order to reduce disfluencies in the connection. By doing so, word order can be preserved and translation quality improved. Our experiments on the patent translation from Japanese to English are able to achieve better translations measured by both BLEU score and word error rate (WER).

1 Introduction

Translating long and complex sentences has been a critical problem in machine translation. A standard phrase-based statistical machine translation (SMT) system cannot solve the problem of word reordering in the target when the source sentence has a complex structure. A syntax-based machine translation system could solve the problem by running a parser on the source sentence in order to get the dependency structure, but when a sentence is long and complex, the parser may fail to give a correct parse tree. However, in this research, we found that even when a sentence is long and complex, it is possible to split a sentence into smaller units which can be translated seperately with minor consideration of the context. The main problem here is locating the best locations for the split. We use linguistic information such part-of-speech (POS) tags and commas as clues to determine the split positions. After splitting a sentence into small clauses, the clauses are translated independently. This means that word reordering can only be done within a clause, not between clauses. Finally, we apply a language model to join these translated pieces together to form a complete translation sentence.

We use the NTCIR-8 Patent Translation shared task data for Japanese to English in our experiment. The results show that splitting the long sentences into small independent clauses helps to improve the translation quality. Automatic evaluation using BLEU scores and WER shows that splitting long sentences can improve the translation, and applying a language model to join the pieces further improves fluency.

2 Previous Work

Research has been done on splitting long sentences into smaller pieces in order to improve the translation (Kim and Ehara, 1994; Furuse et al., 1998; Doi and Sumita, 2003; Sudoh et al., 2010). (Furuse et al., 1998) and (Doi and Sumita, 2003) tend to split speech output instead of text data. For speech output, the main issue is that one utterance may contain a few short sentences instead of one long sentence. Therefore, the main problem is splitting them into proper sentences for translation.

(Sudoh et al., 2010) divide the source sentence into small clauses based on a syntactic parser. Then, a non-terminal symbol serves as a place-holder for the relative clause. However, they also have to train a clause translation model which can translate the non-terminal symbols. They proposed a clause alignment method using a graph-based method.

(Kim and Ehara, 1994) proposed using a rule-based method to split the long sentences into multiple sentences. Furthermore, after splitting the sentences, they tried to identify a subject and inserted it into the subsequence sentences wherever needed. Wherever a sentence is split, the ending grammar is changed so that its conjugation (tense, aspect, modality) matches the ending of the original complete sentence.

Our research in this paper is different in the sense that we only want to split for translation long sentences where the context before and after the splitting points are independent, and we try to join these translations sequentially in a more natural manner to reduce disinfluecies. Our method is simple and does not require complicated processes like clause alignment, subject supplement or sentence ending completion.

3 Translation of Long Sentences

A standard phrase-based statistical machine translation system does not work well for translating long sentences. Fig-

ure 1 shows two examples of long sentence translation using a phrase-based SMT system. In both sentences, the word order of the translation does not follow the source sentence and the translation is bad. However, if we can split the sentences into small clauses such as those shown below the baseline translation, each clause can be translated in a better word order, and the overall translation improves. Our research here is to find out where best to split the sentences and how to join the pieces together in a more natural manner in order to keep the fluency.

4 Method

4.1 Split Conditions

Usually in Japanese written text, a comma will be used if a sentence is long and complicated in order to improve the readability. However, having a comma inserted is not compulsory and there are no strict rules on where a comma should be inserted. There is some research being done on inserting missing punctuation into the text (Murata et al., 2010; Guo et al., 2010). Punctuation could be very useful in written texts for understanding the meaning. In our case, the comma in Japanese is used as a clue for the split position. According to (Murata et al., 2010), there are more than 8 uses for commas in Japanese text and 36.32% of them are used when the content before and after the comma are independent of each other.

Similar to (Kim and Ehara, 1994), a rule-based approach is proposed to split a sentence into multiple clauses. First, the sentence is POS tagged by ChaSen[1] using the IPAdic dictionary. In many cases, if there is a comma, the context before and after the comma are possibly independent and can be translated separately, making a comma a very important clue for locating splitting position candidates. We therefore combine the POS tags and commas as clues to determine the split position for long sentences. Table 1 shows some of the POS tags that have been used for splitting. These POS tags were analyzed and found to be good markers for splitting position candidates, as the clauses before and after they occur may be independent of each other and thus able to be translated seperately. Two rules are used:

> If a POS tag in the head position is found after a comma, then the head will be a split position.
>
> If a POS tag in the tail position is found before a comma, then the word after the comma will be a split position.

If a comma is found after a dependency particle (助詞-係助詞), it is hard to say that the context before the particle is independent of the context after the comma. However, by observing the corpus data, we found that when the sentence is long, it is better if the text before the dependency particle to be translated seperately if it happens to be a very long noun phrase. Therefore, we leave it here as one of the POS tags for splitting the sentences. The other POS tags indicate places that are very likely to be good for splitting.

[1]http://chasen-legacy.sourceforge.jp/

POS tag	Description
Head Position	
副詞-助詞類接続	adverb-particle_conjunction
接続詞	conjunction
Tail Position	
名詞-副詞可能	noun-adverbial
名詞-非自立-副詞可能	noun-affix-adverbial
動詞-自立	verb-main
動詞-非自立	verb-auxiliary
動詞-接尾	verb-suffix
助動詞	auxiliary
助詞-格助詞-連語	particle-case-compound
助詞-接続助詞	particle-conjunctive
助詞-係助詞*	particle-dependency
助詞-副詞化	particle-adverbializer

Table 1: POS tags used for split

4.2 Joining Conditions

After a long sentence has been split into multiple clauses, the clauses are independently translated as usual. However, when these translated pieces are joined back together sequentially, we face the problem of unsatisfactory fluency. In order to make the translation more fluent, we apply a language model when joining the pieces back together.

$$\begin{aligned} e_{best} &= \underset{e}{\operatorname{argmax}}\, p(e|f) \\ &= \underset{e}{\operatorname{argmax}}\, p(f|e)p(e) \end{aligned}$$

$$\begin{aligned} p(e|f) = p_\phi(f|e)^{\lambda_\phi} \times p_{LM}(e)^{\lambda_{LM}} \times p_D(e,f)^{\lambda_D} \\ \times \omega^{length(e)\lambda_{\omega(e)}} \end{aligned} \quad (1)$$

Equation 1 shows the standard statistical translation equation from source language f to target language e using translation model, language model, distortion model and word penalty. If we split the sentence into multiple clauses $f_1, ..., f_n$ and translate these separately, we can use Equation 2 to calculate the translation probability, but this results in the the translation output suffering from the connectivity between clauses. Therefore, a language model is applied to the whole translated sentence instead of to individual clauses[2] as shown in Equation 3.

$$p(e|f) = \prod_{i=1}^{n} p(e_i|f_i) \quad (2)$$

$$\begin{aligned} p(e|f) = \prod_{i=1}^{n} p_\phi(f_i|e_i)^{\lambda_\phi} \times \prod_{i=1}^{n} p_D(e_i,f_i)^{\lambda_D} \\ \times \prod_{i=1}^{n} \omega^{length(e_i)\lambda_{\omega(e_i)}} \times p_{LM}(e)^{\lambda_{LM}} \end{aligned} \quad (3)$$

[2]This should be similar to the *wall* constraint in Moses (Koehn and Haddow, 2009).

Source	また、図１中の第１のスイッチ素子１３は放電用のＮＭＯＳトランジスタ１７からなり、このＮＭＯＳトランジスタ１７のゲートは制御回路２３により制御される。
Reference	In addition , the first switch element 13 in FIG . 1 comprises an NMOS transistor 17 , and a gate electrode of the NMOS transistor 17 is controlled by a control circuit 23 .
Baseline	Further , the first NMOS transistor 17 , and the gate of NMOS transistor 17 is controlled by the control circuit 23 in FIG . 1 a switch element 13 for discharge from .
Split into multiple clauses	また、図１中の第１のスイッチ素子１３は放電用のＮＭＯＳトランジスタ１７からなり、 Further , the first switch element 13 is for discharging NMOS transistor 17 in FIG . 1 , このＮＭＯＳトランジスタ１７のゲートは制御回路２３により制御される。 The gate of NMOS transistor 17 is controlled by the control circuit 23 .
Source	以上説明した第一実施形態によると、許容範囲Ｗθにおいては、第一及び第二回転カム５２，５３がそれぞれローラ４３及び接触部材４４と常に接触する。
Reference	In this embodiment , the first cam 52 is constantly in contact with the roller 43 and second cam 53 is constantly in contact with the contact member 44 in the allowable range W .theta. .
Baseline	According to the above described first embodiment , the first and second rotating cam 52 and 53 are brought into contact with the contact member 44 and each roller 43 and allowable range W .theta. always .
Split into multiple clauses	以上説明した第一実施形態によると、 As described above , according to the first embodiment , 許容範囲Ｗθにおいては、 In the allowable range W .theta. 第一及び第二回転カム５２，５３がそれぞれローラ４３及び接触部材４４と常に接触する。 The first and second rotary cams 52 and 53 are always in contact with the roller 43 and contact portion material 44 .

Figure 1: Long sentence translation examples

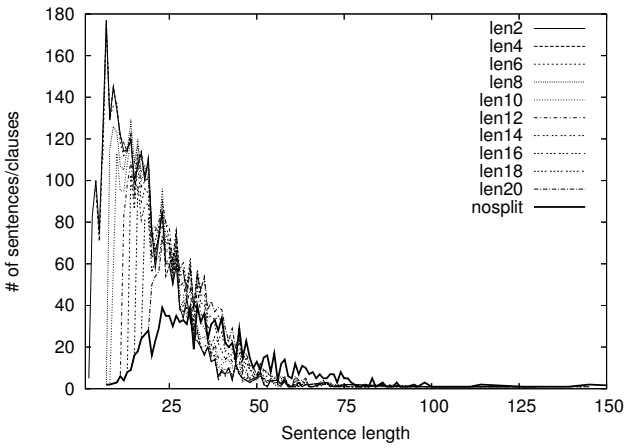

Figure 2: Distribution of sentence length before and after split

Split length	# of sentences /clauses	Average length
len2	2,892	17.83
len4	2,804	18.36
len6	2,643	19.42
len8	2,385	21.41
len10	2,173	23.40
len12	1,999	25.35
len14	1,857	27.22
len16	1,720	29.31
len18	1,605	31.33
len20	1,520	33.03
Nosplit	1,251	38.92

Table 2: Number of sentences/clauses after split

5 Experiment Results

We used the Patent corpus provided by the NTCIR-8 Translation Campaign for Japanese to English translation. The training corpus contains about 3 million sentence pairs, the development set has 1,000 sentence pairs and the test set has 1,251 sentence pairs. The MT system used is an in-house phrase-based statistical machine translation system, CleopATRa. The development set is used to tune the parameter weights using MERT.

Figure 2 shows the distribution of sentence lengths before and after splitting for the test set. The length threshold is the minimum length of a clause after a split. As you can see, if we fix the split length to be shorter, more clauses are generated and the number of long sentences decreases. Table 2 shows the number of sentences/clauses for each length category after splitting. Before splitting, the average length of the sentences is about 39 words but after splitting it ranges from 18 to 33 words. We will examine which split length is best for translation in the following paragraph.

Figure 3 and Figure 4 show the translation results evaluated in BLEU scores and word error rate (WER) by the

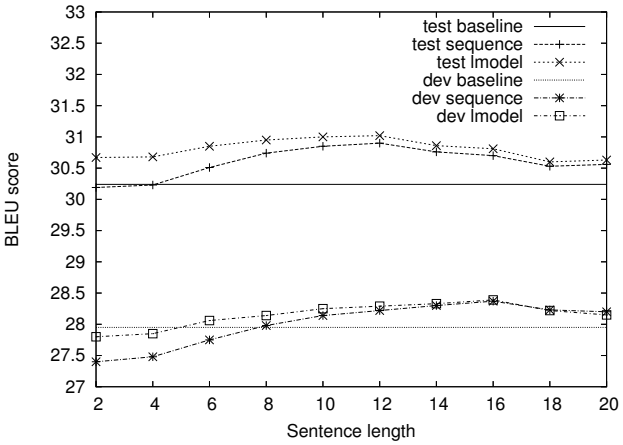

Figure 3: Translation evaluation results by BLEU scores

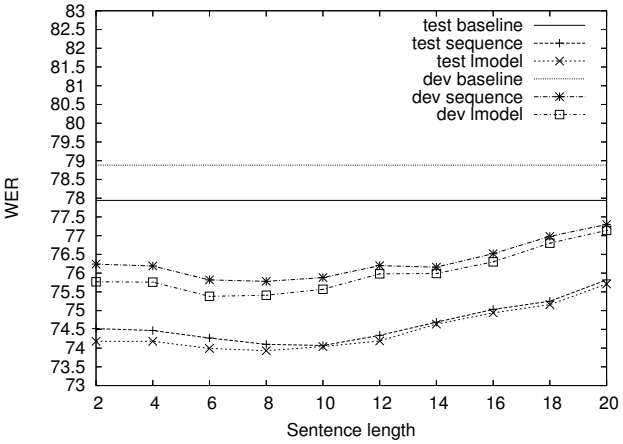

Figure 4: Translation evaluation results by WER

length of clauses. There is only one reference translation and the evaluation is case-sensitive. "Baseline" is the standard translation using Equation 1, "sequence" is using Equation 2 and "lmodel" is using Equation 3. In general, our proposed method shows improvements in both evaluation metrics. Applying a language model when joining the translated pieces together could further improve the results. From our experimental experience, we found that although splitting long sentences into smaller pieces could improve the translation results, it is better not to split them if the clauses are too short as the translations may no longer be independent of each other. Based on the BLEU scores, the development set did best with clauses of a minimum of 16 words and the test set did best with a minimum of 12. WER scores, however, were best at 6 and 8 words, respectively. (Gerber and Hovy, 1998) has fixed the minimum sentence length as 7 words for splitting, but they could not prove that this is the best length. In our case, we still cannot conclude which case is the best and we will leave the problem for future study.

6 Conclusion and Future Work

It is difficult to translate long sentences using a phrase-based statistical machine translation system due to context word order being badly preserved. We proposed splitting the long sentence into multiple short clauses that could be translated independently. POS tags and commas are used as clues to determine the splitting positions. In order to reduce the disinfluencies when rejoining the translated pieces, a language model is applied to the whole translated sentence instead of to the individual pieces. Our experiment results for the patent translation from Japanese to English showed some improvements on the translation quality measured by BLEU score and WER. In the future, we would like to discover optimum clause length for individual sentences instead of an overall blanket length. Furthermore, we will work on inserting commas into the sentences whenever they are needed for sentences that are missing them.

References

Takao Doi and Eiichiro Sumita. 2003. Input Sentence Splitting and Translating. In *Proceedings of the HLT/NAACL: Workshop on Building and Using Parallel Texts*.

Osamu Furuse, Setsuo Yamada, and Kazuhide Yamamoto. 1998. Splitting Long and Ill-formed input for Robust Spoken-language Translation. In *Proceedings of COLING-ACL*, pages 421–427.

Laurie Gerber and Eduard Hovy. 1998. Improving Translation Quality by Manipulating Sentence Length. In *Proceedings of AMTA*, pages 448–460.

Yuqing Guo, Haifeng Wang, and Josef van Genabith. 2010. A Linguistically Inspired Statistical Model for Chinese Punctuation Generation. *ACL Transactions on Asian Language Information Processing*, 9(2):6:1–6:27.

Yeun-Bae Kim and Terumasa Ehara. 1994. A Method for Partitioning of Long Japanese Sentences with Subject Resolution in J/E Machine Translation. In *Proceedings of International Conference on Computer Processing of Oriental Languages*, pages 467–473.

Philipp Koehn and Barry Haddow. 2009. Edinburgh 's Submission to all Tracks of the WMT2009 Shared Task with Reordering and Speed Improvements to Moses. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 160–164.

Masaki Murata, Tomohiro Ohno, and Shigeki Matsubara. 2010. Automatic Comma Insertion for Japanese Text Generation. In *Proceedings of the EMNLP*, pages 892–901.

Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, Tsutomu Hirao, and Masaaki Nagata. 2010. Divide and Translate: Improving Long Distance Reordering in Statistical Machine Translation. In *Proceedings of the Joint 5th Workshop on SMT and MetricsMATR*, pages 418–427.