Segmentation and Punctuation Prediction in Speech Language Translation Using a Monolingual Translation System

Eunah Cho, Jan Niehues and Alex Waibel

International Center for Advanced Communication Technologies - InterACT
Institute of Anthropomatics
Karlsruhe Institute of Technology, Germany
firstname.lastname@kit.edu

Abstract

In spoken language translation (SLT), finding proper segmentation and reconstructing punctuation marks are not only significant but also challenging tasks. In this paper we present our recent work on speech translation quality analysis for German-English by improving sentence segmentation and punctuation.

From oracle experiments, we show an upper bound of translation quality if we had human-generated segmentation and punctuation on the output stream of speech recognition systems. In our oracle experiments we gain 1.78 BLEU points of improvements on the lecture test set. We build a monolingual translation system from German to German implementing segmentation and punctuation prediction as a machine translation task. Using the monolingual translation system we get an improvement of 1.53 BLEU points on the lecture test set, which is a comparable performance against the upper bound drawn by the oracle experiments.

1. Introduction

With increased performance in the area of automatic speech recognition (ASR), a large number of applications arise, which use the output of ASR systems as input. It is critical for these applications to have a clean, well-constructed input.

Especially for an application such as statistical machine translation (SMT), it is expected to have sentence-like segments in the input. As a first reason, most MT systems are trained using text data with well-defined sentence boundaries. Therefore, it is necessary to have proper segmentation before the translation to match the translation models in order to achieve better translation quality. Moreover, there are algorithmic constraints as well as user preferences, such as readability. When a sentence is excessively long, it either consumes a great deal of resources and time, or readability suffers.

If the input is already augmented with punctuation in the source language, it is advantageous to the training procedure of MT. In this case, there is no need to retrain the translation system with modification on the training data, in order to match the ASR output [1]. Nevertheless, most of the current ASR systems do not provide punctuation marks.

It is one of the challenging tasks to restore segmentation and punctuation in the output of an ASR system, especially for speech translation. Sentence segmentation in the ASR system is often generated using prosodic features (pause duration, pitch, etc.) and lexical cues (e.g. language model probability). However, the performance of sentence segmentation degrades in spontaneous speech. This is because a large amount of the spontaneous utterance is less grammatical compared to written texts [2] and there are fewer sentence-like-units (SU). Moreover, the presence of disfluencies in casual and spontaneous speech increases the difficulty of this task.

In this work we aim at recovering sentence segmentation and punctuation before translation as a preprocessing step and analyze its impact on the translation quality. The first goal of this paper is to investigate the upper bound of possible improvement on the translation quality when proper sentence segmentation and punctuation are achieved. For this we implement an oracle experiment, in which the human-generated segmentation and punctuation of manual transcripts are applied to ASR output before the translation process. In the second part of the oracle experiments, we insert the segmentation according to the ASR system into manual transcripts. As a second goal of this work, we build a monolingual translation system as a method to generate segments and punctuation marks. We will evaluate the performance of our monolingual translation system against the oracle experiment.

This paper is organized as follows. In Section 2, a brief overview of past research on segmentation and punctuation prediction is given. In Section 3, we present our baseline translation system used for this work. The oracle experiments and their results are described in Section 4, followed by Section 5 which contains the strategy to recover segmentation and punctuation and its results. Section 6 concludes our discussions.

2. Related Work

In previous work, the punctuation prediction problem was addressed to improve the readability as well as subsequent natural language processing [3]. In order to annotate ASR output with punctuation marks, they developed a maximum-entropy based approach. In this approach the insertion of punctuation was considered a tagging task. A maximum entropy tagger using both lexical and prosodic features was applied and the model was used to combine the different features. Their work showed that it is hard to distinguish between commas and default tags, and periods and question marks, since there is little prosodic information (similarly short or similarly long pause durations) and the features can cover a span longer than bigrams. They achieved a good F-measure for both reference transcriptions and transcriptions produced by a speech recognition system.

In [1] the authors made an extensive analysis on how to predict punctuation using a machine translation system. In this work, it was assumed that the ASR output already has the proper segmentation, which is sentence-like units. They investigated three different approaches to restore punctuation marks; prediction in the source language, implicit prediction, and prediction in the target language. Using a translation system to translate from unpunctuated to punctuated text, they showed significant improvements in the evaluation campaign of IWSLT 2011.

Among different motivations for the sentence segmentation, [4] split long sentence pairs in the bilingual training corpora to make full use of training data and improved model estimation for statistical machine translation (SMT). For the splitting they used the lexicon information to find splitting points. They showed that splitting sentences improved the performance for Chinese-English translation task. Similarly, to improve the performance of Example-based machine translation (EMBT) systems, [5] suggested a method to split sentences using sentence similarity based on edit-distance.

Combining prosodic and lexical information to detect sentence boundaries and disfluencies was demonstrated in the work of [6], where decision trees are used to model prosodic cues and N-grams for the language model. The authors suggested that having large amounts of recognizer output as training data for the models can improve the prediction task as it lowers the mismatch between training data and test set. The necessity of resegmentation for the ASR output was investigated in [2]. They trained a sentence segmenter based on pause duration and language model probabilities. It was emphasized that it is important to have commas in addition to periods within a sentence boundary, as it defines independently translatable regions and eventually improves translation performance.

Segmentation and punctuation issues are addressed together in [7]. The authors modified phrase tables so that the target side contains commas, but the source side does not contain any. Thus, when this modified phrase table was applied during translation, it recovered commas on the target side. For the segmentation and periods after each new line, they used a sentence segmenter based on a decision tree on the source side. They applied this method to three language pairs and achieved a significantly improved translation performance.

Table 1: *Information on the preprocessed source side of the test set*

ASR output	Sentences	2393
	Words without punctuation marks	27173
	WER	20.79%
Manual Transcript	Sentences	1241
	Words	29795
	Words without punctuation marks	26718
	Periods	1186
	Commas	1834
	Question marks	55

3. System Description

In this section we briefly introduce the statistical MT system that we use in this experiment.

As we work on translating speech in this experiment, we use the parallel TED[1] data and manual transcripts of lecture data containing 63k sentences as indomain data and adapt our models at the domain. The lecture data is collected internally at our university, and the domain of each lecture differs from the others. To better cope with domain-specific terminologies in university lectures, Wikipedia[2] title information is used as presented in [8].

For development and testing, we use the lecture data from different speakers. These are also collected internally from university classes and events. They consist of talks of 30 to 45 minutes and the topic varies from one speech to the other. For the development set we use manual transcripts of lectures, while for testing we use the transcripts generated by an ASR system. The development set consists of 14K parallel sentences, with 30K words on the source side and 33K words on the target side including punctuation marks. Detailed information on the source side of the test set, including the word error rate (WER) of the recognition output, can be found in Table 1.

The translation system is trained on 1.8 million sentences of German-English parallel data including the European Parliament data and News Commentary corpus. Before the training, the data is preprocessed and compound splitting for the German side is applied. Preprocessing consists of text normalization, tokenization, smartcasing, conversion of German words written according to the old spelling conventions into the new form of spelling.

[1] http://www.ted.com
[2] http://www.wikipedia.org

The Moses package [9] is used to build the phrase table. The 4-gram language model is trained on the English side of the above data with nearly 425 million words using the SRILM toolkit [10]. To extend source word context, a bilingual language model [11] is used. The POS-based reordering model as described in [12] is used for word reordering in order to account for the different word orders in source and target language. To cover long-range reorderings, we apply the modified reordering model as described in [13]. The translation hypotheses are generated using an in-house phrase-based decoder [14] and the optimization is performed using minimum error rate training (MERT) [15].

Translation models are built using the punctuated source side. Also for the other experiments, where there are no punctuation marks on the source side available, phrase tables are prepared in the same way.

4. Oracle Experiments

To investigate the impact of segmentation and punctuation marks on the translation quality, we conduct two experiments.

In the first experiment, we apply human-transcribed segments and punctuation marks to the output of the speech recognition system. Thus, words are still from an ASR system, but the segments and punctuation marks are reused from a human-generated transcript. In the second experiment, the segments in the output of the speech recognition system are applied to the human-generated transcripts. In this case, words are transcribed by human transcribers, but segmentation and punctuation are from an ASR system.

From these experiments we can observe how much impact the better segmentation and punctuation have for the performance of ASR output translation. We can also find how the segmentation according to an ASR system affects manual transcripts.

4.1. Oracle 1: Insertion of manual segments and punctuation marks into ASR output

Applying manual segments to the output of an ASR system requires the time stamp information for each utterance. We use this information from manual transcripts and segment the output stream generated by the ASR system according to it. The alignment information between ASR test sets and their manual transcripts is learned in order to insert punctuation marks. As punctuation marks, we consider period, comma, question mark, and exclamation mark. Punctuation marks such as period, question mark, and exclamation mark are usually followed by a new segment in manual transcripts, and commas are useful to define independently translatable regions [2].

Depending on which punctuation marks are inserted, three hypotheses are considered in this experiment.

- MTSegment: correct segments from a manual transcript are applied to the ASR test set.
- MTSegmentFullStop: correct segments and ".?!" from a manual transcript are applied to the ASR test set.
- MTSegmentAllPunct: correct segments and ".,?!" from a manual transcript, including commas, are applied to the ASR test set.

Therefore, the results in the hypothesis MTSegment show the boundary of performance improvement when the proper segmentation is given, while the hypothesis MTSegmentAllPunct shows the scenario when we also have good punctuation marks additionally. With the hypothesis MTSegmentFullStop, we intend to investigate how helpful it is for the translation quality to have commas or not.

To show the impact of the difference of the segmentation according to the ASR system and according to the hypothesis MTSegmentAllPunct, several consecutive segments are extracted from our test set. The translation of these two texts with different segmentation is presented in Table 2. The two source texts contain the same recognized words from an ASR system, but different segmentation and punctuation are applied. We can observe that when the text is with manual transcripts' segmentation, the translated text conveys the meaning of the sentence substantially better, as well as it provides improved readability. For example, the German participle *gesprochen*, which was translated into *spoken* using MTSegmenatAllPunct, is lost in the first segment in the ASR system and segmented into the next line. This leads to the loss of the

Table 2: *Translation using different segmentation according to ASR output and MTSegmentAllPunct hypothesis*

Segmentation	Translation
ASR	> We see here is an example from the European Parliament, the European Parliament 20 languages > And you try simultaneously by help human translator translators the > Talk to each of the speaker in other languages to translate it is possible to build computers > The similar to provide translation services
MTSegment-AllPunct	> We see here is an example from the European Parliament. > The European Parliament 20 languages are spoken, and you try by help human translator to translate simultaneously translators the speeches of the speaker in each case in other languages. > It is possible to build computers that are similar to provide translation services?

Table 3: *Disfluency and its affect on the automatic segmentation*
(Reference translation: Thus we consequently also have a third foot hold in Asia, in the Chinese region, in Hong Kong.)

System	
ASR output	> wir haben somit also auch ein drittes Standbein in Asien in > in chinesischen Raum in Hongkong
reference	> wir haben somit also auch ein drittes Standbein in Asien, im chinesischen Raum, in Hongkong.

information about this participle during the translation. An article and its following noun, *die Reden*, are also split using the original segmentation of the ASR system. It becomes the reason why the more suitable word *(the) speeches* in this context is not chosen, but *Talk*.

4.2. Oracle 2: Insertion of ASR output segments into manual transcripts

In addition to the insertion of proper segmentation and punctuation into the output of the ASR system, we perform another experiment where the segmentation in the output of the ASR system is applied to manual transcripts.

Although the segmentation from ASR output is obtained by incorporating language model probability and prosodic information such as pause duration, it is often not the best segmentation especially for spontaneous speech. This is caused by its nature of having less organized sentences and more disfluencies.

Table 3 depicts an example of incorrect automatic segmentation caused by disfluencies. As the speaker stutters, the automatic segmenter of the ASR system based on pause duration and a language model trained on clean texts inserts a new line.

In this experiment, we analyze the following three scenarios.

- ASRSegment: a manual transcript was segmented according to the segmentation of the ASR output.
- ASRSegmentComma: a manual transcript was segmented according to the segmentation of the ASR output, and commas are removed.
- ASRSegmentAllPunct: a manual transcript was segmented according to the segmentation of the ASR output, and all four punctuation marks are removed.

The four punctuation marks correspond to ".,?!" as in the first oracle experiment. To segment a manual transcript as in the ASR output, we use an algorithm which is commonly used for evaluating machine translation output with automatic sentence segmentation [16]. This method is based on the Levenshtein edit distance algorithm [17]. By backtracing the decisions of the Levenshtein edit distance algorithm, we can find the Levenshtein alignment between the reference words and the words in the ASR output.

In this work, the ASR output plays the role of a reference and using this algorithm we are able to find a resegmentation of the human reference transcript based on the original segmentation of the ASR output.

4.3. Results

Table 4 depicts the results of the two experiments in numbers. The scores are reported as case-insensitive BLEU [18] scores, without considering punctuation marks. This aims at analyzing the impact of the segmentation and punctuation solely on the translation quality.

Table 4: *Influence of oracle segmentation and punctuation on the speech translation quality*

System		BLEU
ASR		**20.70**
Oracle 1	MTSegment	21.42
	MTSegmentFullStop	22.18
	MTSegmentAllPunct	22.48
Transcripts		**27.99**
Oracle 2	ASRSegment	26.38
	ASRSegmentComma	26.36
	ASRSegmentAllPunct	25.54

For the hypotheses MTSegment, ASRSegmentAllPunct and tests on the ASR output, we create phrase tables removing punctuation marks on the source side in order to make a better match between the test set and the phrase table. To evaluate the translation hypotheses of ASR output and the ASRSegmentation experiments, we resegmented our translation hypotheses to have the same number of segments as the reference as shown in [16].

From this table we observe that having the correct segmentation and punctuation improves the translation quality significantly. When the human-transcribed segmentation and punctuation are available, an improvement of 1.78 BLEU is observable on the test set.

Another interesting point is when we compare MTSegmentAllPunct to MTSegmentFullStop, we see the steady improvement of 0.3 BLEU in translation from having commas on the source side. This is congruent with the findings in [2], that inserting commas in addition to periods improves translation quality. In our case, the scores are evaluated ignoring punctuation marks. Thus, the improvement on BLEU means

that by having proper punctuation marks the translation quality itself can be improved.

On the other hand, we can observe from Table 4 that by simply changing the segmentation of the transcripts we lose 1.6 BLEU scores in translation performance. As shown in Table 1, there are almost twice as many segments in the ASR output compared to the manual transcript. This can be one reason of the drastic drop of the translation quality. We also observed from this translation that incorrect reordering of words occasionally happens within a segment, when the segment is not a sentence-like unit but a part of a sentence.

Removing commas from ASRSegment does not result in a big performance drop in ASRSegmentComma. Often, the segments from the ASR system do not match with the phrase boundaries learned in the text translation system, which results in having fewer independently translatable regions separated by commas. In addition to this, losing all punctuation information leads to a further performance drop of 0.84 BLEU scores.

5. Monolingual Translation System

In this section we introduce our monolingual translation system that we used to predict the segmentation and punctuation.

Inspired by [1], we build a monolingual translation system to predict segmentation and punctuation marks in the translation process. This monolingual translation system translates non-punctuated German into punctuated German. Using this system we predict punctuation marks as well as segmentation before the actual translation of the test sets. The output of this system becomes the input to our regular text translation system which is trained using training data with punctuation marks.

When translating the output of the monolingual translation system, no preprocessing is applied as the test set is already preprocessed before going through the monolingual translation system. The monolingual translation system does neither alter any words nor reorder words, but it is used solely for changing segments and inserting punctuation marks.

In order to build this system, we first process the training data to make the source side not contain any punctuation marks, but the target side contain all punctuation marks. The training data statistics on the target side is shown in Table 5.

Table 5: *Information on the preprocessed punctuated German side of the training data*

Words	46.32M
Periods	1.76M
Commas	2.88M
Question marks	0.10M
Exclamation marks	0.07M

For a language model, we use 4-grams and it is trained on the punctuated German data. Also, no reordering model is used as we use the monotone alignment.

The difference of our monolingual translation system to the work in [1] is that in our work the monolingual translation system is used to predict sentence segmentation additionally. In their work, it was assumed that the segmentation of the speech recognition output was given and corresponded to at least sentence-like units. Therefore, their monolingual translation system was used to reconstruct punctuation marks only with using three different strategies.

It was shown in the previous section that the segmentation generated from an ASR system is not necessarily the best segmentation, especially when the recognized text is spontaneous speech with less grammatical sentences and more disfluencies. In this work, we aim at improving segmentation in addition to inserting punctuation marks using this monolingual translation system. To perform this it is required to modify the training data as well as development and test sets.

5.1. Data preparation

Usually training data for conventional text translation systems is segmented by human transcribers so that it has punctuation such as a full stop, a question mark, or an exclamation mark at the end of each line. Therefore, if we use this training data to translate the ASR test sets, translation models would more likely insert a punctuation mark at the end of every line of the ASR test set during translation. From this observation, we resegment training corpora randomly so that every segment is not necessarily one proper sentence-like unit. The development set is modified in the same way.

The test sets for this monolingual translation system are also prepared differently, using the idea of a sliding window. Examplary sentences from our test set are shown in Table 6. In this table, each line contains 8 words and the first line starts with a word *der*. In the second line, we have the next starting word *bildet*, which was the second word in the first line. At the same time, we have a new encountering word *gesehen* at the end of the line. When the length of a sliding window is *l*, each line consists of *l-1* words from the previous line and 1 new word. Thus, the *n*th line contains the *n*th to *n+l-1*th word of a test set. The test set prepared in this way has the same length as the number of words in the original test set. In this way we can have up to *l* spaces between words. For those spaces we want to investigate how probable it is to have a punctuation mark in that word space. In this experiment, we constrain the length of sliding window *l* to 10.

This differently formatted test set enters the monolingual translation process in a normal way, line by line. The translation of the test set shown in Table 6 using our monolingual translation system is illustrated in Table 7. We see that words such as *Normalform* and *gesehen* are followed by certain punctuation marks.

Table 6: *Test set preparation for the monolingual translation system*

der	bildet	die	sogenannte	konjunktive	Normalform	wir	haben
bildet	die	sogenannte	konjunktive	Normalform	wir	haben	gesehen
die	sogenannte	konjunktive	Normalform	wir	haben	gesehen	dass
sogenannte	konjunktive	Normalform	wir	haben	gesehen	dass	wir
konjunktive	Normalform	wir	haben	gesehen	dass	wir	diese
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 7: *Translation using the monolingual translation system*

der	bildet	die	sogenannte	konjunktive	**Normalform.**	Wir	haben
bildet	die	sogenannte	konjunktive	**Normalform.**	Wir	haben	**gesehen,**
die	sogenannte	konjunktive	**Normalform.**	Wir	haben	**gesehen,**	dass
sogenannte	konjunktive	**Normalform.**	Wir	haben	**gesehen,**	dass	wir
konjunktive	**Normalform.**	Wir	haben	**gesehen,**	dass	wir	diese
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

5.2. Punctuation prediction criteria

A punctuation mark is chosen if the same punctuation mark is found same or more often than a given threshold. If more than one punctuation mark appears more than the threshold in the same word space, the most frequent one is chosen. There are some cases where we have the same frequency for multiple punctuation marks; in this case we put a different priority on punctuation marks. For example, in this experiment we put higher priority for a period over a comma.

In this experiment, we evaluate the translation quality over a varying threshold, from 1 to 9. We exempt the case when the threshold is 10, the length of the sliding window. In this case, one punctuation mark has to appear all the 10 word spaces after a word in order to be inserted. This condition is so restrictive that only few full stops are generated, which causes unaffordable computational time consumption for the translation procedure.

In the same way as in the oracle experiment, we consider four punctuation marks here: period, comma, question mark, and exclamation mark. A new segment is introduced when either a period, question mark, or exclamation mark is predicted, in order to have congruence with the manual transcripts.

To make the hypotheses comparable with the oracle experiments, we considered three different hypotheses of reconstructing segmentation and punctuation.

- MonoTrans-Segment: monolingual translation system is used for segmentation prediction only.
- MonoTrans-FullStop: monolingual translation system is used for segmentation and full stop prediction.
- MonoTrans-AllPunct: monolingual translation system is used for segmentation and all punctuation marks prediction.

5.3. Results

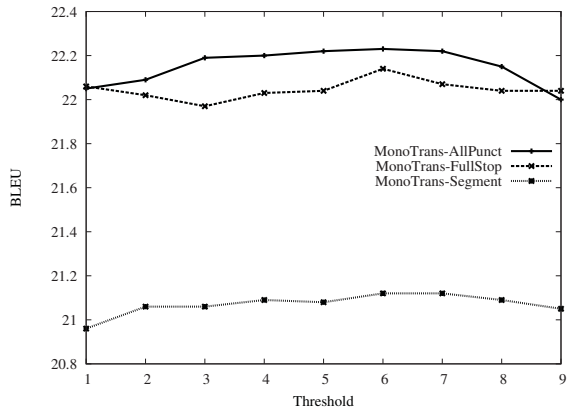
In order to analyze the effect of the varying threshold for the monolingual translation system, first we use the same threshold value for all punctuation marks. The number of punctuation marks predicted using the same threshold are shown in Table 8. As shown in the table we could predict periods and commas, but we could not generate question marks and exclamation marks. A reason might be that question mark and exclamation mark are already rare in the manual transcript. In addition, we do not have many of them appearing in the training corpora, compared to the frequency of the other punctuation marks. The number of periods in Table 8, therefore, is the same as the number of segments predicted.

Figure 1 presents the translation performance of the three hypotheses in BLEU over different threshold values. In this experiment as well, the same threshold value is used for all the different punctuation marks. Even though we ob-

Table 8: *Punctuation marks predicted using the monolingual translation system, with a different threshold. The number of punctuation marks in the manual transcript is also given as a comparison.*

Threshold	1	2	3	4	5	6	7	8	9	Manual Transcript
Periods	1,273	970	881	861	851	841	817	736	464	1,186
Commas	2,741	2,190	1,973	1,915	1,904	1,889	1,857	1,773	1,486	1,834

Figure 1: *Translation performance with varying threshold values*



tain more segments the lower we set the threshold value, each hypothesis still outperforms the translation of ASR output (20.70 in BLEU). The threshold value can go down to 1 without any significant loss in BLEU. As shown by the curve of MonoTrans-FullStop, the performance is already good when having segments from periods only. When we compare MonoTrans-AllPunct and MonoTrans-FullStop, the performance of MonoTrans-AllPunct fluctuates relatively more while that of MonoTrans-FullStop stays more stagnant. From this observation we notice the necessity of another experiment where different threshold values for period and commas are used, as the performance can be improved with fewer commas when there are more segments.

Table 9 presents how close we can get toward the oracle experiments when using the segmentation and punctuation predicted output from the monolingual translation system. The numbers from an oracle experiment and ASR output are also shown for comparison. The condition Test1 represents the results where the threshold 6 was used for both period and comma.

As depicted in this table, all three hypotheses of our monolingual translation system beat the translation quality using the ASR output with a significant difference. When both segmentation and punctuation are predicted using our monolingual translation system, we gain 1.53 BLEU points on our test set, which is only 0.25 BLEU points less than a result from the oracle experiment.

In order to maintain a similar number of segments to the manual transcript, but still have the "helpful" number of commas for translation, we separate the threshold value for period and comma. Test2 in Table 9 depicts the translation performance when we use the threshold value 1 for period and 6 for comma. Thus, a comma is chosen when it is found more than 5 times at the space between words. Compared to the case where the same threshold value of 6 for both punctuation marks is used, we obtain more than 150% of the original number of segments. However, we can still maintain a similar translation performance, showing only a drop of 0.06 BLEU points in the hypothesis MonoTrans-AllPunct.

Table 9: *Results of using monolingual translation system to reconstruct segmentation and punctuation, compared to the oracle experiment*

System	BLEU	
	Test1	Test2
ASR	20.70	
MonoTrans-Segment	21.12	20.97
Oracle 1: MTSegment	21.42	
MonoTrans-FullStop	22.14	22.06
Oracle 1: MTSegmentFullStop	22.18	
MonoTrans-AllPunct	22.23	22.17
Oracle 1: MTSegmentAllPunct	22.48	
Number of segments	851	1,292

Predicting a new line only after a period performs well for the translation. However, the numbers shown in Table 1 indicate that inserting a new line only after a period provides half of the number of segments that our ASR system produced for the test set. Therefore, to compare the performance of the ASR segmenter in a fair condition, we conduct another experiment where a new line is inserted whenever a punctuation mark, including comma, is predicted. For this experiment we use the same threshold 8 for all punctuation marks, so that we can have similar number of segments as in the ASR output. By doing so we could obtain 2,509 segments, which is nearly 200 segments more than the ASR output. From this we gained 21.67 BLEU points for the MonoTrans-AllPunct hypothesis. Although the score of the hypothesis MonoTrans-AllPunct is 0.5 BLEU points lower than previous two tests, the score is still around 1 BLEU point higher than the translation quality of raw ASR output.

6. Conclusion

In this paper, we first presented the impact of segmentation and punctuation on the output of speech recognition systems by implementing oracle experiments. Experiments have shown that we can gain up to 1.78 BLEU points of improvement on the translation quality if we apply the manual segmentation and punctuation to the ASR output. On the other hand, when we apply the segmentation and punctuation of speech recognition output to the manual transcripts, we have an overall loss of 2.45 BLEU points on the translation quality. Therefore we show that the segmentation produced by ASR systems may not assure the best translation performance, but a separate process to segment the ASR stream before the translation can help the translation performance.

In the second part of the paper, the monolingual translation system is used to predict segmentation and punctuation

in ASR output. In order to implement this system, we change the format of the training corpora as well as the development and test set. By using the monolingual translation system, we gain more than 1.5 BLEU points on the ASR test set.

In future work, we would like to pursue on developing the monolingual translation system with different ways to extract relevant phrases for the task. Furthermore, the analysis on disfluencies in speech is necessary to improve the segmentation and punctuation prediction.

7. Acknowledgements

This work was achieved as part of the Quaero Programme, funded by OSEO, French State agency for innovation. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

8. References

[1] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling Punctuation Prediction as Machine Translation," in *Proceedings of the eight International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, 2011.

[2] S. Rao, I. Lane, and T. Schultz, "Optimizing Sentence Segmentation for Spoken Language Translation," in *Proc. of Interspeech*, Antwerp, Belgium, 2007.

[3] J. Huang and G. Zweig, "Maximum Entropy Model for Punctuation Annotation from Speech." in *Proc. of ICSLP*, Denver, CO, USA, 2002.

[4] J. Xu, R. Zens, and H. Ney, "Sentence Segmentation using IBM Word Alignment Model," in *EAMT 2005*, Budapest, Hungary, 2005.

[5] T. Doi and E. Sumita, "Splitting Input Sentence for Machine Translation Using Language Model with Sentence Similarity," in *Coling 2004*.

[6] A. Stolcke, E. Shriberg, R. Bates, M. Ostendorf, D. Hakkani, M. Plauche, G. Tür, and Y. Lu, "Automatic Detection of Sentence Boundaries and Disfluencies Based on Recognized Words," in *Proc. of ICSLP*, Sidney, Australia, 1998.

[7] M. Paulik, S. Rao, I. Lane, S. Vogel, and T. Schultz, "Sentence Segmentation and Punctuation Recovery for Spoken Language Translation," in *ICASSP*, Las Vegas, Nevada, USA, April 2008.

[8] J. Niehues and A. Waibel, "Using Wikipedia to Translate Domain-specific Terms in SMT," in *Proceedings of the eight International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, 2011.

[9] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *ACL 2007, Demonstration Session*, Prague, Czech Republic, June 23 2007.

[10] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit." in *Proc. of ICSLP*, Denver, Colorado, USA, 2002.

[11] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *Sixth Workshop on Statistical Machine Translation (WMT 2011)*, Edinburgh, UK, 2011.

[12] K. Rottmann and S. Vogel, "Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model," in *TMI*, Skövde, Sweden, 2007.

[13] J. Niehues and M. Kolss, "A POS-Based Model for Long-Range Reorderings in SMT," in *Fourth Workshop on Statistical Machine Translation (WMT 2009)*, Athens, Greece, 2009.

[14] S. Vogel, "SMT Decoder Dissected: Word Reordering." in *Int. Conf. on Natural Language Processing and Knowledge Engineering*, Beijing, China, 2003.

[15] A. Venugopal, A. Zollman, and A. Waibel, "Training and Evaluation Error Minimization Rules for Statistical Machine Translation," in *Workshop on Data-drive Machine Translation and Beyond (WPT-05)*, Ann Arbor, MI, 2005.

[16] E. Matusov, G. Leusch, O. Bender, and H. Ney, "Evaluating Machine Translation Output with Automatic Sentence Segmentation." in *Internat. Workshop on Spoken Language Translation*, Pittsburgh, USA, 2005.

[17] V. I. Levenshtein, *Binary Codes Capable of Correcting Deletions, Insertions and Reversals.*, ser. 10(8). Soviet Physics Doklady, 1966, vol. pp. 707-710.

[18] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation." IBM Research Division, T. J. Watson Research Center, Tech. Rep. RC22176 (W0109-022), 2002.