NMT-based Segmentation and Punctuation Insertion for Real-time Spoken Language Translation

Eunah Cho, Jan Niehues, Alex Waibel

Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology, Germany
{eunah.cho|jan.niehues|alex.waibel}@kit.edu

Abstract

Insertion of proper segmentation and punctuation into an ASR transcript is crucial not only for the performance of subsequent applications but also for the readability of the text. In a simultaneous spoken language translation system, the segmentation model has to fulfill real-time constraints and minimize latency as well.

In this paper, we show the successful integration of an attentional encoder-decoder-based segmentation and punctuation insertion model into a real-time spoken language translation system. The proposed technique can be easily integrated into the real-time framework and improve the punctuation performance on reference transcripts as well as on ASR outputs. Compared to the conventional language model and prosody-based model, our experiments on end-to-end spoken language translation show that translation performance is improved by 1.3 BLEU points by adopting the NMT-based punctuation model, maintaining low-latency.

Index Terms: spoken language translation, segmentation and punctuation insertion, spoken language processing

1. Introduction

Insertion of proper punctuation marks into automatically generated transcripts plays a crucial role in improving readability of the transcripts as well as the performance of subsequent applications, such as machine translation (MT). Since many of the conventional automatic speech recognition (ASR) systems do not generate reliable punctuation, there has been extensive research on segmentation and punctuation insertion models. Especially in a simultaneous spoken language translation system, the punctuation and segmentation component also has to fulfill the real-time constraints, while guaranteeing its best performance.

Language model (LM) and prosody based system has been one of the commonly used methods to insert punctuation marks into ASR output as discussed in [1, 2, 3]. While it has a strong advantage of low latency, the short context of this model often leads to underwhelming performance. Machine translation based model, which translates non-punctuated text into punctuated text [4, 5], showed its effectiveness in spoken language translation evaluation campaigns [6, 7, 8]. While phrase-based machine translation (PBMT) has been mainly used for this task, using this system in an on-line setup has an extra burden of pruning and loading of all relevant models, including phrase table and language models.

Encoder-decoder framework with attention mechanism [9, 10] is used extensively in many sequence-to-sequence mapping. Analysis on recent evaluation campaigns [11] also shows that such neural machine translation (NMT) systems achieve better performance than PBMT systems when using the same parallel data.

Inspired by this, we model segmentation and punctuation insertion system using the framework of encoder-decoder with attention. While it achieves a better performance, it also offers an advantage of compact model size.

To our knowledge, this is the first work to present segmentation and punctuation insertion scheme using encoder-decoder framework with attention mechanism, integrated into a real-time spoken language translation. In this work, we analyze the performance of NMT-based segmentation and punctuation system considering real-time constraints and adapt the model for them. In order to minimize the bottleneck at the softmax layer at the output, we deploy a compact representation of output vocabulary. The trade-off between network size and performance is also studied. The required context length without increasing latency is also investigated.

We build an NMT-based segmentation and punctuation model for English, and integrate into a real-time spoken language translation system [12]. The performance of this model is analyzed in both offline and online scenarios. Compared to the conventionally used language model based segmentation, we achieve 1.3 BLEU points of improvement in English to German translation when using the NMT-based model, maintaining the low latency.

2. Related Work

Insertion of punctuation and segmentation into ASR transcripts has been studied from various aspects. Using language model probabilities with pause duration was suggested in [1]. Another approach includes maximum entropy model [13], using lexical and prosodic features. A sequential tagging based approach was also applied in [14].

Punctuation prediction task was combined with the translation task in [15]. In this work, authors built a translation model between non-punctuated source and punctuated target languages. In [4], authors compared different mechanisms to use a machine translation framework for punctuation prediction task. It was concluded that the best performance is achieved when they insert punctuation marks within the source language, prior to the translation. In this method, however, punctuation marks are inserted within pre-defined sentences. Thus, it was assumed that proper sentence boundaries are already available.

Later this work is extended also to predict sentence boundaries in [5]. Thereby training data is prepared differently. Sentences in the training data is cut randomly, so that sentence boundaries can be observed anywhere throughout the data. For testing, a sliding window was used. In [16], this work is revisited and studied considering the real-time constraints. It was shown that monolingual translation system can be used with a modified input mechanism in order to decrease the overall latency.

The latency issue in real-time speech translation system has been emphasized in [17]. In this work, it was shown that latency can be decreased by showing the initial hypothesis to the users and allowing updates from further contexts.

Neural networks have been used for the punctuation prediction task. In [18], the authors used a classifier based on a recurrent neural network (RNN). Authors in [19] used a bidirectional recurrent neural network with attention mechanism for the punctuation prediction task.

Our work differentiates itself from theirs in several aspects. First, our work deploys attentional encoder-decoder framework where the punctuation insertion is viewed as a translation task. Second, our model has been adapted and integrated into the real-time spoken language translation system, maintaining the low latency. For example, the system in [19] uses 200 word long slices of input sequence. Once input sequence is partitioned into 200 word long slices, each slice is then punctuated and segmented using the system. While this long context is beneficial for punctuation prediction performance, it generates long latency in a real-time application scenario. On the other hand, our work offers an analysis in adapting the model for the real-time scenario.

3. NMT-based Punctuation and Segmentation

In our work, we model the punctuation and segmentation as a translation problem. We translated from lower-cased, non-punctuated language into true-cased, punctuated language. Motivated by the success of NMT in recent evaluations, we used an attention-based encoder-decoder model as our translation system.

In this framework, the source sentence is first encoded using a bi-direction long short-term memory (LSTM) [20] network. The target sentence is then generated by a second RNN-model, the decoder. A weighted sum over the source hidden states is used as input to the decoder. Thereby, the weights are calculated by the attention layer. A detailed description can be found in [9]. In contrast to PBMT-based systems, this architecture needs a fixed vocabulary size. The most successful approach to represent an open vocabulary using a fixed number of token in NMT is the byte-pair encoding [21]. In our work, we used this technique.

Based on this baseline system, we adapted the network to the task of punctuation and segmentation. Since the model should be used in a real-time speech translation system, our first focus was to allow fast decoding. We addressed this problem by using a compact representation of the output space. The details will be discussed in Section 3.1. Different from machine translation for text input, the input in this task is not properly segmented. Therefore, we needed to adapt the NMT system to this condition. Detailed description on how our input streams are constructed is given in Section 3.2.

3.1. Compact Representation

When analyzing the complexity of the different layers of the neural network, the softmax layer at the output is known to be the most complex one. For example, we used a vocabulary size of 40K tokens in our conventional NMT setup. If we use this setup, the calculation at the softmax layer at the output will be very expensive. It is worth noting that in our application scenario GPUs are only available during training, not in the testing scenario. Therefore, this issue is especially problematic.

In standard NMT between two different languages, the target words are completely different from the input words. However, in our scenario, the only difference between the input and the output sequence is the casing and the punctuation marks. Therefore, we introduced a compact representation of the target words using tags.

Each word in a target sentence is represented in either *U*, for a uppercased word, or *L*, for a lowercased word, concatenated with the punctuation mark following the word. In order to keep the number of tokens the same for source and target side, we concatenated all trailing punctuation marks with the tags. The target sentence, thus, is a sequence of *U*, *L*, and one of these concatenated with a sequence of trailing punctuation marks, i.e. *L?* or *L,”*. In our case, this was altogether 60 tokens. Compared to the 40K tokens in the original output vocabulary, this is around 1.5% of the original size. For example, for an excerpt *stuff. and we said, “Well what about play and recess?” and* we will have the target sequence of *L. L L L,“ U L L L L L?” L*.

The output sequence is then replaced into a sequence of uppercased/lowercased words and punctuation marks. If a word exists in a pre-defined list for special casing, its uppercasing map is applied. Otherwise, only the first character is uppercased.

The list is learned from the parallel training data (TED). Throughout the corpus, we examine the most frequent uppercasing form for each word (e.g. i→I, obama→Obama). If the most frequent uppercasing form has only the first character uppercased, we consider this our default uppercasing format and therefore do not keep it in our list. Only if the most frequent form includes special casing (e.g. youtube→YouTube, ted→TED), the word is added to the list. By containing only special casings, we obtain a compact list. Since most of words have its uppercasing format of only first letter uppercased, the list contains around 0.1% of all vocabularies in the training data.

This tag-based representation assumes that the number of input tokens is the same as the number of output tokens. The problem is that this cannot be assured by the standard NMT model. Furthermore, we are using BPE-encoding for the input. Therefore, without any specific modifications, there is a possibility that a word is split into multiple tokens and then internal parts of the tokens are uppercased or a punctuation mark is inserted between them. We address this problem by using the tags for the original, complete words, not for the split subword tokens. Therefore, punctuation marks can only inserted between words, not between sub-words. Secondly, during decoding we only consider hypothesis which have the same length as the original source sentence. Therefore, there can be no mismatch between the number of tags and words in a sentence.

3.2. Input Data Stream

In machine translation scenarios, the input has a format of properly segmented sentences. However, this is not available in the test case of our online scenario. We addressed this challenge by modifying input data stream differently for training and testing (offline/online) of the model.

In training, we used an approach suggested for phrase-based monolingual translation systems in [5]. The training data is randomly split into segments between 20 and 30 words. Thereby the system is able to learn to put punctuation marks at any position within the segment.

This approach, however, cannot be applied during decoding as the context at the beginning of each segment is very limited. With the limited context, the system would not be able to make a well-grounded decision for the initial parts of each segment. For this reason, it was suggested in [5] to use a sliding window for testing condition. In this approach, every word is processed several times, observed in various contexts. Although this often generates reliable and well-performant punctuation for the subsequent applications, we can not apply this approach in the real-time use case due to latency. For offline tests, instead, we apply the same procedure as the training data, where the test data is randomly cut. This also simulates our online setup more closely, where we can no longer have the benefit of overlapping windows.

In our application, we used a framework which allows for dynamic updates of all models as described in [17]. In this scenario, authors in [16] presented an approach to efficiently use a punctuation system. The main idea is to always keep a context of the previous l_w words. For words in this context, we will reuse the previously generated punctuation marks. Therefore, the initial words of each segment would have enough context for the punctuation prediction in the previous step.

4. System Description

In this section, we briefly describe the punctuation prediction systems used and compared throughout this work.

4.1. Neural Machine Monolingual Translation Setup

All punctuation and segmentation insertion models are built using the NMT framework `lamtram` [22], with an attention-based encoder-decoder model. The system is trained on English TED data, around ∼197k sentences. We generated the sub-word units using byte-pair encoding (BPE), as described in [21], with the BPE merging operations at 40k.

Same as in [5], the training data is randomly cut so that sentence boundaries as well as punctuation marks can be observed in any location throughout the segment. For the source side, all punctuation marks are removed and all letters are lowercased. The target side is cleaned up so that we have only considering punctuation marks (sentence boundary marks e.g. .?!, commas, and double quotation marks) left. We also adopted a compact representation on the target side, which decreases output vocabulary size dramatically as discussed in Section 3.1. Test data is prepared in the same way, abandoning the sliding window approach due to the latency issue, as discussed in [16]. The details on test data preparation for offline and online scenario can be found in Section 3.2.

The models were all trained with Adam, where we restarted the algorithm twice and early stopping is applied. Details of system architecture and our preliminary results on comparing different network sizes are described in Section 5.1.

4.2. PBMT-based Monolingual Segmentation and Punctuation

Same as the NMT system, the PBMT-based segmentation system is trained on the TED data. Training data is prepared in the same way as described in Section 4.1. However, since PBMT system is compared only in offline mode, we used sliding window for test data as described in [5]. Using the sliding window, each word can be observed in various contexts. It yielded better performant punctuation prediction as shown in previous spoken language translation tasks [8].

4.3. LM-based Segmentation and Punctuation

In this system, a 4-gram language model is used to measure the probability of a punctuation given the previous two and following two words. If the probability exceeds an empirically chosen threshold, the punctuation mark is inserted. For prosody, pause information is used. The details can be found in [23].

Due to its low latency, this segmentation method has been used extensively in real-time applications such as Lecture Translator [3, 12]. In this work, we compare the online module of our NMT-based segmentation and punctuation against the LM-based system.

5. Experiments and Results

In order to analyze the performance of NMT-based segmentation and punctuation model, we compared offline and online scenarios.

For offline scenario, we used a phrase-based monolingual translation system and compared the performance with the NMT-based system. In this scenario, we used two input conditions. First, we used manual transcript of the test data in order to see the punctuation prediction performance without any ASR errors and other online application constraints. The ASR transcript of the test data is also fed into the offline systems to show the impact of ASR errors in the offline setup. The ASR system description can be found in [17].

In case of the test on the manual transcripts, the punctuation prediction performance is measured in F-scores for prediction accuracy compared to the human-generated reference, and BLEU [24] for its impact on machine translation performance. In case of the test on the ASR transcript, we translated the automatically punctuated test data into German and compared the translation performance in BLEU. For machine translation, we used a system shown in [25].

For online scenario, we used a language model and prosody based segmentation method. In order to consider the potential latency, we limit the decoding time to the length of audio files. Therefore, if there were any delayed translation due to latency, which was not fully decoded during the audio time, it would have not been included in our final hypothesis. However, we have not encountered with any of undecoded sentences during the test.

As test data, we used test2013 from IWSLT evaluation campaign. The manual transcript of this test data is 993 sentences in English and the audio reaches around 2 hours and 16 minutes.

5.1. Tradeoff: Network Size vs. Performance

In a first series of preliminary experiments we analyze the tradeoff between network size and performance.

In the Baseline system, the encoder uses word embeddings of size 256 and a bidirectional LSTM [20] with 256 hidden layers for each direction. For the attention, we used a multi-layer perceptron with 512 hidden units. The decoder uses conditional GRU units with 512 hidden units in order to benefit from context information. We then halved the overall network size and compared the performance, except for the word embeddings dimension.

Table 1 shows the results and word embedding dimension size for each setup. For clarity, we halved the word embeddings dimension only for the *Quarter* system.

Results show that while the overall performance drops slightly as we decrease the network size, the performance of the *Quarter* system is still comparable to the baseline. In our

System	F-score
Baseline, wd256	61.13
Half, wd256	59.85
Quarter, wd128	59.04

Table 1: *Comparison of performance when adopting different size of network.*

preliminary experiments on another language, decreasing the network size one step further down costed us around 10 in F-score. For our online system, we chose *Quarter* system.

5.2. Context Length

In the second preliminary experiment, we wanted to analyze the impact of the context length used during decoding in the neural network. As described in Section 3.2, one parameter of the system is the context length during decoding. In the baseline configuration, we used a context length of four. That means, that the last four words are always kept, as a context, and punctuation is reapplied for these words in the next step. Therefore, a new word has a context of four previous words.

Context length	En→De BLEU
4	13.18
16	14.20

Table 2: *Comparison of online segmentation and punctuation performance in on-line setup, with different context length permitted.*

The results can be found in Table 2. When we use this method, we achieve a BLEU score of 13.18 in the subsequent translation. In a second experiment, we increases this context up to 16 words. In this case, the BLEU score is increased to 14.20. Therefore, we can conclude that the neural network based model is able to facilitate long dependencies and only works well, if this context is also available during decoding.

5.3. Results

First experiment is devoted to compare phrase-based monolingual translation system and neural machine translation system in an off-line mode. All punctuation marks of manual transcript of test data are removed prior to the experiment. Once the test data is punctuated using either one of the two systems, the accuracy of inserted punctuation is measured in F-score. The punctuated transcript is then translated into German, in order to see the impact of inserted punctuation marks on a subsequent application.

System	F-score	En→De BLEU
PBMT-seg	59.59	18.84
NMT-seg	61.30	19.21

Table 3: *Comparison of offline segmentation and punctuation methods on manual transcript.*

Table 3 shows the result. We can see that the NMT-based system not only excels in the intrinsic evaluation, but also in the extrinsic one, improving BLEU by 0.4 points.

In order to show the impact of ASR errors, we used the ASR transcript of the test data as an input to PBMT-seg and NMT-seg. The results can be found in Table 4. Due to ASR errors, the BLEU scores drop around 4-5 points.

System	En→De BLEU
PBMT-seg	13.88
NMT-seg	13.96

Table 4: *Comparison of offline segmentation and punctuation methods on ASR transcript.*

Finally, Table 5 shows the results of end-to-end speech translation performance using two different segmentation methods. As mentioned in Section 5, the decoding time is constrained to the audio length.

System	En→De BLEU
LM-seg	13.37
NMT-seg	14.67

Table 5: *Comparison of end-to-end real-time speech translation performance on different segmentation methods.*

The results show that while LM-based segmentation achieved 13.37 BLEU for the given test data, the NMT-based segmentation achieved 14.67 BLEU points. We can observe that by replacing the segmentation and punctuation module from language model based one to NMT-based one improved the translation quality by 1.3 BLEU points. Since the decoding time was restricted for both conditions, no latency was added.

6. Conclusions

In this paper, we presented our recent work in NMT-based segmentation and punctuation model. To our knowledge, this is the first work to use the encoder-decoder framework with attention mechanism for a punctuation prediction task and integrate it for a real-time spoken language translation system. In order to ensure low-latency, we minimized the bottleneck at the softmax layer by decreasing output vocabulary. We also offer an in-depth analysis on design choices, including network size and context length, for an improved performance in real-time application with low latency.

Experiments show that NMT-based segmentation and punctuation model outperforms the conventional language model and prosody based model by 1.3 BLEU points of an end to end spoken language translation, maintaining low latency.

Future work includes expansion of this model into different source languages. Also, this model can be combined with other preprocessing of machine translation, i.e. disfluency removal.

7. Acknowledgements

This work was supported by the Carl-Zeiss-Stiftung.

8. References

[1] S. Rao, I. Lane, and T. Schultz, “Optimizing Sentence Segmentation for Spoken Language Translation,” in *Proceedings of the eighth Annual Conference of the International Speech Communication Association (INTERSPEECH 2007)*, Antwerp, Belgium, 2007.

[2] M. Ostendorf, B. Favre, R. Grishman, D. Hakkani-Tur, M. Harper, D. Hillard, J. Hirschberg, H. Ji, J. G. Kahn, Y. Liu *et al.*, “Speech segmentation and spoken document processing,” *Signal Processing Magazine, IEEE*, vol. 25, no. 3, pp. 59–69, 2008.

[3] E. Cho, C. Fügen, T. Herrmann, K. Kilgour, M. Mediani, C. Mohr, J. Niehues, K. Rottmann, C. Saam, S. Stüker, and A. Waibel, "A real-world system for simultaneous translation of german lectures," in *Proceedings of the 14th Annual Conference of the International Speech Communication Association (INTERSPEECH 2013)*, Lyon, France, 2013.

[4] 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 2011)*, San Francisco, California, USA, 2011.

[5] E. Cho, J. Niehues, and A. Waibel, "Segmentation and punctuation prediction in speech language translation using a monolingual translation system," in *Proceedings of the International Workshop for Spoken Language Translation (IWSLT 2012)*, Hong Kong, China, 2012, pp. 252–259.

[6] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Frederico, "Report on the 10th IWSLT evaluation campaign," in *Proceedings of the Tenth International Workshop on Spoken Language Translation*, ser. IWSLT 2013, Heidelberg, Germany, 2013.

[7] ——, "Report on the 11th IWSLT evaluation campaign, iwslt 2014," in *Proceedings of the Eleventh International Workshop on Spoken Language Translation*, ser. IWSLT 2014, Lake Tahoe, CA, USA, 2014.

[8] ——, "Report on the 12th IWSLT evaluation campaign," in *Proceedings of the Eleventh International Workshop on Spoken Language Translation (IWSLT 2015)*, Da Nang, Vietnam, 2015.

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2015.

[10] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, Doha, Qatar, 2014.

[11] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, V. Logacheva, C. Monz *et al.*, "Findings of the 2016 conference on machine translation (wmt16)," in *Proceedings of the First Conference on Machine Translation (WMT 2016)*, Berlin, Germany, 2016.

[12] M. Müller, T. S. Nguyen, J. Niehues, E. Cho, B. Krüger, T.-L. Ha, K. Kilgour, M. Sperber, M. Mediani, S. Stüker *et al.*, "Lecture translator speech translation framework for simultaneous lecture translation," *NAACL HLT 2016*, p. 82, 2016.

[13] J. Huang and G. Zweig, "Maximum Entropy Model for Punctuation Annotation from Speech," in *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, Denver, Colorado, USA, 2002.

[14] W. Lu and H. T. Ng, "Better punctuation prediction with dynamic conditional random fields," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*. Cambridge, Massachusetts, USA: Association for Computational Linguistics, 2010, pp. 177–186.

[15] M. Paulik, S. Rao, I. Lane, S. Vogel, and T. Schultz, "Sentence Segmentation and Punctuation Recovery for Spoken Language Translation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008)*, Las Vegas, Nevada, USA, April 2008.

[16] E. Cho, J. Niehues, K. Kilgour, and A. Waibel, "Punctuation insertion for real-time spoken language translation," in *Proceedings of the Eleventh International Workshop on Spoken Language Translation (IWSLT 2015)*, Da Nang, Vietnam, 2015.

[17] J. Niehues, T. S. Nguyen, E. Cho, T.-L. Ha, K. Kilgour, M. Müller, M. Sperber, S. Stüker, and A. Waibel, "Dynamic transcription for low-latency speech translation," *Interspeech 2016*, pp. 2513–2517, 2016.

[18] M. Kazi, B. Thompson, E. Salesky, T. Anderson, G. Erdmann, E. Hansen, B. Ore, K. Young, J. Gwinnup, M. Hutt, and C. May, "The MITLL-AFRL IWSLT 2015 systems," in *Proceedings of the Eleventh International Workshop on Spoken Language Translation (IWSLT 2015)*, Da Nang, Vietnam, 2015.

[19] O. Tilk and T. Alumäe, "Bidirectional recurrent neural network with attention mechanism for punctuation restoration," *Interspeech 2016*, pp. 3047–3051, 2016.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2015.

[22] G. Neubig, "lamtram: A toolkit for language and translation modeling using neural networks," http://www.github.com/neubig/lamtram, 2015.

[23] E. Cho, J. Niehues, and A. Waibel, "Machine Translation of Multi-party Meetings: Segmentation and Disfluency Removal Strategies," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2014)*, Lake Tahoe, California, USA, 2014.

[24] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2002)*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, 2002, pp. 311–318.

[25] I. Slawik, M. Mediani, J. Niehues, Y. Zhang, E. Cho, T. Herrmann, T.-L. Ha, and A. Waibel, "The KIT Translation Systems for IWSLT 2014," in *Proceedings of the eleventh International Workshop for Spoken Language Translation (IWSLT 2014)*, Lake Tahoe, California, USA, 2014.