Dynamic Boundary Detection for Speech Translation

Nina Zhou, Xuancong Wang and AiTi Aw
Institute for Infocomm Research (I2R), Singapore
E-mail: zhoun@i2r.a-star.edu.sg, aaiti@i2r.a-star.edu.sg

***Abstract*—Speech translation is usually the pipeline task of automatic speech recognition (ASR), translation unit segmentation and machine translation (MT). Segmenting the ASR output to translation units poses a challenge of balancing the translation quality and efficiency for real-time speech translation. In this paper, we firstly propose a parser-based semantic boundary detection method to detect all semantic boundaries based on our definition. To realize the translation of the semantic units, a word-boundary language model is secondly proposed to improve the translation quality. Experiments on English to Chinese and Chinese to English speech translation have shown that the proposed method yields improved translation quality and lower latency, when compared to the conventional punctuated methods.**

I. INTRODUCTION

Conventional speech translation consists of two basic components: automatic speech recognition (ASR), machine translation (MT). ASR usually outputs audio segmented text stream without any punctuation marks, which is largely not matched with the style of the training data used for the machine translation (MT). To address this mismatch, Paulik *et al.* (2008) suggested a number of possibilities [13]. For better readability of ASR output or for other applications like meeting notes summary, most authors put one component between ASR and MT to make ASR output suitable for MT [3]. The major role of this component is to segment ASR output. In the real-time speech translation, the performance of this middle component is measured by two key parameters, i.e., translation quality and latency. The latency is to evaluate how long the ASR transcription and the translation are displayed to the user when one word is spoken.

A considerable amount of research has focused on reducing the latency but at the same not degrading or even improving the speech translation. The earlier segmentation methods on ASR outputs include the fixed length segmentation and sentence end segmentation [14]. Rao *et al.* (2007) [14] proposed to detect the sentences end using prosody and a language model. Sridhar *et al.* (2013) proposed to use kernel-based SVM to predict the sentence boundaries [16]. For the sentence segmentation methods, the user will have to wait for the entire sentence to be finished before seeing the translation result displayed. This may leads to high latency for long sentences and will not be acceptable in the real-time speech translation system. Besides sentence segmentation, Paulik *et al.* (2008) proposed to first use the modified phrase table to recover the comma and then utilized decision tree to realize the sentence end detection [13]. Sridhar *et al.* (2013) also suggested passing the comma-separated units for speech translation [13]. Furthermore, the author also proposed a conjunction-based segmentation method [13], which was realized by a unigram POS tagger to recognize the conjunction words like “and” and ‘or’. For example, based on this method [13], the sentence “My/PRP dogs/NN like/VBZ eating/VBG sausage/NN and/CC running/VGB along/in the/DT river/NN” will be split into two units by the conjunction word “and”. But we can see that if only depending on conjunction words to determine the translation segments, it may destroy the semantic context and therefore affects the translation accordingly. On the other hand, Cho *et al.* (2015) proposed to use the monolingual translation system to insert all punctuation marks on the modified stream decoding output of the ASR system, so as the time spent for punctuation insertion can be saved by the stream decoding [2].

Different from the above mentioned sentence-separated, comma-separated, or conjunction-based methods, researchers were also focusing on a different type of information, i.e., unseen constituents, to reduce the translation delay [3, 4]. Grissom II *et al.* (2014) proposed to predict sentence-final verbs through reinforcement learning [7], so that the translation from verb-final languages to verb initial languages (such as German-English) can be started earlier. But this method only can be applied to certain particular cases as [7]. Oda *et al.* (2015) proposed to predict unseen syntactic constitutes so that the system can start translation before seeing the coming context, and thus reduce the latency [12]. This method was proposed for the benefit of using source-side parsing, e.g., tree-to-string (T2S) translation [12]. In addition, it also needs to wait for more translation units to be predicted for translation if the current predicted constituents need reordering in the translation.

Inspired by the aforementioned methods, we propose a parser-based dynamic semantic boundary detection (SBD) method to segment the input stream to small semantic units for the translation challenges. Unlike the previous work mentioned, we use one model to predict all the punctuation marks and semantic boundaries for the speech translation. The translation unit obtained may be due to a punctuation mark or due to a coordinate or subordinate conjunction word and so on. The SBD is realized through conditional random fields (CRF) [9, 17, 19], which makes the SBD problem a sequential labelling task [9, 17, 19]. In this case, preparing the data with semantic boundary labels is crucial for successfully training a SBD model. In our paper, the semantic boundary labels are defined based on the data’s parsing tree [20], from which, the training data is automatically generated (Section II) and manually verified.

To evaluate the proposed SDB method, we further proposed a translation boundary language model (Section III) to take the advantage of the semantic unit input. The experiments showed that the proposed SBD method with the translation language model can reduce the latency by up to 2 to 4 times with a small improvement in translation accuracy.

Table 1. The semantic boundary categories for English

Final categories	Words in bold text with labels defined as left categories shown
Non-break	All those not defined specifically
	*I like eating ice cream **and** hotpot* *He did not go home **but** went to school*
comma_NB	*I like eating ice cream hotpot and beef*
CSC_B	coordination conjunction break (CC_B): *Will you go first **or** shall I go first*
	Subordinate conjunction break (CS_B): *We finished the **task** before boss came back*
THWH_B	th-clause. Sometimes “that” is omitted. *I told **you (that)** I like him* wh-clause: e.g., why, who, what followed by clause: *you **know** what I mean.*
comma_B	*I asked **you about him** then you did not reply*
SB_P: . SB_E: ! SB_Q: ?	Sentence end with punctuations “! . ? ”: *I went there **myself***

II. THE SBD MODULE

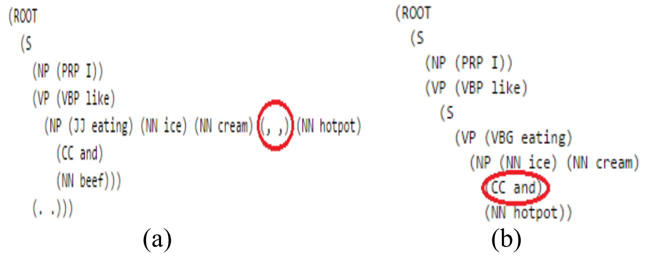
In this paper, the semantic boundary is defined as the point in which a segment delimited by two boundaries is a semantic constituent carrying a complete meaning. It includes punctuation marks as well as specific category of words.

The process of labelling the data is: 1) define all the semantic boundary labels; 2) run parser on data with punctuation marks, then get the parsing tree; 3) based on the parsing tree, the labels are determined and words are labelled; 4) remove all the punctuations from the labeled data. Then the data is ready for training.

A. Semantic boundary labels' definition and data labelling

We have mentioned that the definition of the semantic boundaries is delivered from the parsing tree. For English, we define the labels of the semantic boundaries from the open source Stanford parser [1, 20]. While for Chinese, we define the semantic boundaries from the Niuparser [18], an open source for Chinese syntactic and semantic parser.

The semantic boundary labels for English are shown in Table 1. The labels of the words in bold text are corresponding to the categories shown in first column of the table. We define the non-semantic boundaries, with two special conditions in which a comma and coordinate conjunction words are not semantic boundaries. For example, in the sentence of “*I like eating ice cream, hotpot and beef*”, the comma after ice cream will not be considered as a semantic boundary since it connects two NPs (Fig. 1 (a)). Likewise, in the sentence “*I like eating ice cream and hotpot*”, the coordinate conjunction word “*and*” is also not considered as a semantic boundary since it connects two NNs (Fig. 1 (b)). Although comma_NB is not a semantic boundary, we still treat it as an individual label because it inserts comma. Fig.1 shows the labelling of the words in the two sentences. “*cream/*comma_NB” means one comma will be inserted after “cream” in the sentence. Both of these two sentences have only one break at the position where it is labelled as SB_P.



Labelled sentence (a): “*I/none like/none eating/none ice/none cream/comma_NB hotpot/none and/none beef/SB_P*”.
Labelled sentence (b): *I/none like/none eating/none ice/none cream/none and/none hotpot/SB_P*

Figure 1. The parsing of two complete sentences for defining the semantic boundaries and the two labelled sentences.

We defined the second type of semantic boundary: CSC_B, which can be described in the two sub-categories of coordinate conjunction break (CC_B) and subordinate conjunction break (SC_B) (Table 1). CC_B indicates the break caused by CC words, e.g., “and”,’or’,’but’, when they connect two independently subordinate clauses (i.e., SBAR). For the example of the sentence in Table 1, “*Will you go first **or** shall I go first*”, the CC ‘***or***’ will be labelled as CC_B according to the Stanford parser. SC_B indicates the break caused by subordinate conjunction (SC) words, e.g., “before”, “after”, “as soon as” and “when”, when they are followed by a SBAR. During the labelling, we will label the word before the SC words as SC_B instead of the SC word itself. For example in Table 1, the “***task***” before the SC ‘***before***’ is labelled as SC_B. The similar function of CC_B and SC_B in the sentence and their distribution ratio over other labels make us decide to merge CC_B and CS_B as one category.

THWH_B is the third type of semantic boundary. It refers to the break caused by ‘that’ and “wh-” words, e.g., ‘why’, ‘what’ and ‘which’. As to that- clause, the word “that” can be omitted. No matter what, we can determine the labelling of THWH_B by checking the parsing tree of SBAR-> S or SBAR-> {IN S} and so on.

When a comma connects two SBARs/Ss, it is considered as a break, i.e., comma_B. For example, in the context of “*I asked you about **him**, then you did not reply*”, the comma after “***him***” is connecting two independent “S” in the parsing tree, so this “**him**” is labelled as comma_B. All sentence end punctuation marks are easy to label. “SB_P” indicates the punctuation ‘.’; “SB_Q” indicates the punctuation “?”, and “SB_E” indicates the punctuation “!”.

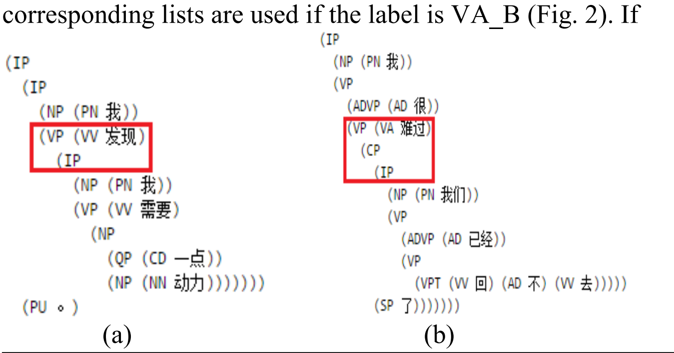
Due to the characteristics of the Chinese language, the definition of Chinese semantic boundary labels will be different from the English semantic boundary labels. With reference to Table 2, we did not set any conjunction word breaks. This is because the CC word like “和/and” is always omitted in Chinese context. For example, in the sentence of “*小/little 宝宝/baby 爱/love 喝/drink 牛奶/milk 吃/eat 香蕉/banana 吃/eat 肉/meat*”, the word ‘和/and’ is omitted. We count up the number of other Chinese CC words used in Chinese contexts and decided not to consider CC words in Chinese SBD. We define one semantic boundary for Chinese i.e., VA_B in Table 2. The total number of labels defined in the Chinese SBD model is 6.

Table 2. Semantic boundary definition for Chinese

Labels	Example of sentences (red highlighted word tokens are labelled as the one in the left column.
None	All other not defined specifically
Comma_B	*你/you 不/not* ***回家/go home*** *那/then 我/I 也/also 晚 点 /later 回/go back 啦*
comma_NB	*我/i 先/first* ***出门/go out*** *先/first 走一步/go 啦 。 小/little 宝宝/baby 爱/love 喝/drink* ***牛奶/milk*** *吃/eat* ***香蕉/banana*** *吃/eat 肉/meat*
VA_B	*我/i* ***发现/found*** *我/i 需要/need 一点/a little bit 动力 /motivation 。 我/i 很/very* ***难过/sad*** *我们/we 已经/already 回 不 去 了/(been not like before).*
SB_P	*你/ 不/no 回 家/go home 那/then 我/i 也/also 晚 点 /later* ***回/back***
SB_Q	*为啥/why 你/you 不/not 一起/together* ***来/come***
SB_E	*你/you 是 are 多/how 爱/love 看/watch* ***电视/TV***

VA_B indicates those verbs and adjectives which are followed by a clause. A clause is usually represented by IP or CP in the parsing tree. For example, in the sentence of "我/i ***发现/found*** 我/i 需要/need 一点/(a little bit) 动力/motivation", the verb "***发现/found***" is followed by a statement clause IP (see Fig. 2). Therefore, the word "***发现/found***" will be labelled as "VA_B". In the sentence of "我/i 很/very ***难过/sad*** 我们/we 已经/already 回 不 去 了/(been not like before)", the adjective "***难过/sad***" works in the same way (Fig. 2).

Prior to labelling, we first check the parsing tree and find out those sub-trees of VP-> {VV, IP} or VP->{VA,CP} and so on. If those sub-trees are found, we further detect the leaves of VV or VA, i.e., "发现" or "难过". The leaves of VV or VA can constitute two different word lists in an accumulative way. When the two lists are formed, we manually check those lists to remove the wrong words caused by parsing errors. During the word labelling, the two corresponding lists are used if the label is VA_B (Fig. 2). If



Labelled sentence a): *我/none 发现/VA_B 我/none 需要/none 一点/none 动力/SB_P*
Labelled sentence b): 我/none 很/none 难过/VA_B 我们/none 已经/none 回/none 不/none 去/none 了/SB_P

Figure 2. The parsing of two complete sentences for defining the semantic boundaries and the corresponding labelled sentences.

the detected word is not in the corresponding list, we will not label them as "VA_B".

B. Data setup for the SBD experiment

For real-time speech translation, we focus on social media data domain. We use an in-house social media data for the SBD model training and testing. The size of the original data is 100K (sentences) for Chinese and 30K for English (sentences). We select sentences that have a minimum length of 10, provided there is no comma in the sentence, or at least there is one comma inside the sentence. After data selection, we obtained 80K Chinese sentences and 19K English sentences. For both data, we split 80% for training and 20% for testing. There is one sentence in each line. In the experiments, to balance the number of sentence-end punctuation marks at the end of a sentence and in the middle of a sentence, we concatenated N=2,3,4,5… utterances for the training and testing.

C. The SBD model training and testing

For the SBD model training, we extracted the features of word n-gram and POS (part-of-speech) n-gram features up to tri-grams, in a context of 5 words from the current word [10]. Since the labelling of the current word is based on the last few words and also the next few words, the setting of the n-gram is very important. With larger n, the model can store more information for better prediction, while larger n also means longer waiting time for the labelling of the current word. In our work, we will allow the model to wait for 3 future words. Feature pruning is done at min occur=3. Gaussian regularization is tuned on a tuning set selected from the training set.

Table 3 shows a comparison about the punctuation marks between the SBD model and a punctuation insertion (PI) model. Both models are trained on the data with one sentence per line (N=1), and tested on the data with N=1 sentences per line. From the results, we can see that the performance of both comma_B and comma_NB for the SBD are worse than that of comma for the PI model. To group comma_B and comma_NB, we recalculate the number of correct prediction of comma_B and comma_NB. If comma_B is predicted as itself or comma_NB, we consider it is a correct prediction. For comma_NB, it is calculated in the same way. After grouping, the prediction of comma for SBD is better than the individual comma_B and comma_NB, but it is still worse

Table 3. Result comparison between SBD and PI model, train model (N=1) and test (N=2)

Categories	Test (N=1)	
	SBD	PI
comma_B	67.17	71.66
comma_NB	40.67	
Group (comma_B, comma_NB)	68.26	
SB_P	93.47	86.84
SB_Q	81.93	82.87
SB_E	49.79	56.55
SB (all "." + "!"+ "?")	95.24	89.29
SB_EOL (all SB at EOL)	95.24	89.29
SB_notEOL	null	null

than PI. For sentence end punctuation marks, the SBD performs better on SB_P than PI, but for SB_Q and SB_E, the SBD performs a little bit worse than the PI. However, if we group all the sentence end punctuations marks (i.e., SB), it is better for the SBD than for the PI (Table 3). SB_notEOL indicates those sentence boundaries not at the end of line (EOL) and SB_EOL indicates the sentence boundaries at the end of line (EOL). The result is reasonable because the PI model only have 5 punctuation marks to distinguish, while the SBD has a few more categories to distinguish besides the punctuation marks.

Table 4. The F-measure results (%) for the English SBD model (N=2) on N=1,2,3,4,5 test sentences merged

Categories	N=1	N=2	N=3	N=4	N=5
comma_NB	38.51	39.14	38.39	38.16	38.23
CSC_B	76.39	76.13	76.07	75.89	76.19
THWH_B	71.65	71.29	71.23	71.16	71.06
comma_B	65.34	65.15	64.28	64.04	63.69
SB_P (.)	91.64	90.79	89.09	88.22	87.74
SB_Q (?)	79.21	75.05	71.86	70.36	68.92
SB_E (!)	46.40	28.30	27.88	16.75	15.96
SB (all "." + "!"+ "?")	93.81	93.02	91.38	90.74	90.19
SB_EOL (all SB at EOL)	93.81	98.64	98.56	98.56	98.75
SB_notEOL	null	88.88	88.49	88.57	88.37

In the real applications, there are usually multiple sentences in one line. Therefore, in the following experiments we will report the results for the SBD model trained on N=2 sentence merged in this paper. In Table 4, the SBD model is tested on N=1,2,3,4,5 sentences merged. With increasing N, the result has a slight degradation and the change is consistent for all the categories. Specifically, when N is increased from 1 to 2, the number of the sentence boundary not at EOL (i.e., SB_notEOL), is increased from 0 to nearly half of the previous number of SB. We can see the result for SB_notEOL is very stable when N is increased from 2 to 3, 4 and 5.

For Chinese social media data, as many sentences have subject omission, after merging two sentences, it will make it difficult to distinguish the comma and period. Prior to Chinese SBD, an in-house Chinese word segmentation tool is applied on the Chinese ASR output. From the results in Table 5, we can see that there is a big drop in the performance of the two labels "comma_B" and "SB_P", when N is increased from N=1 to 2. In the real-time situation, this problem can be weakened by the pause between different speakers or between different content of talks.

Table 5. The F-measure (%) results for the Chinese SBD model (N=2) on N=1,2,3,4,5 test sentences merged.

Breaks	N=1	N=2	N=3	N=4	N=5
Comma_B	72.16	65.62	63.25	62.04	61.58
Comma_NB	57.90	53.92	52.26	51.56	51.03
VV_B	66.15	64.60	63.97	63.82	63.75
SB_P	87.23	76.30	72.18	70.10	68.56
SB_Q	76.26	72.85	71.38	70.69	69.42
SB_E	18.74	14.94	14.09	13.53	13.29
S_B	99.98	84.99	81.05	79.11	78.36
SB_EOL	99.98	99.99	99.98	99.99	99.99
SB_notEOL	null	70.37	71.69	71.20	71.05

III. TRANSLATION: EXPERIMENTS AND RESULTS

A. Translation boundary language model (TBLM)

We often encounter the scenario when combining translation spans, a worse translation is selected instead although decoding each individual span in isolation gives decent results. For example:

I basically wrote ➔ *我 基本上 写 了* √
that I am available ➔ 我 有 空 √
I basically wrote that I am available ➔ **有 空 我 基本上 写 的** ×

This is mostly due to the language model (LM) being unaware of the translation boundaries. For the above example, in the case of "*wrote a letter*", or "*wrote the book*", where the object constituent "*a letter*"/"*the book*" does not form a complete semantic unit by itself, n-gram conditioning applies well because not everything can follow "*wrote*". However, if the object forms a complete semantic unit such as a clause, n-gram conditioning should be much more relaxed across the boundary because any complicated clause can follow wrote. To solve this problem, we propose translation boundary LM (TBLM) in addition to the standard n-gram LM. The n-gram log-likelihood of TBLM is defined as:

$$\log P(w_i|h_i) = \log P(w_i|h_i\text{--}) - \log P(w_i|h_i\text{-})$$

where h_i-- is the i^{th}word's full history up to n-gram order and h_i- is the i^{th} word's partial history constrained by the translation boundary. For example, for a 5-gram LM, at the word "*空*" of the hypothesis "*我基本上写了<b> 我有空*",

$$logP(\text{空}|\text{h}) = logP(\text{空}|\text{写了我有}) - logP(\text{空}|\text{我有})$$

The rationale is to provide some sort of n-gram isolation at the translation boundary, by relaxing the n-gram conditioning across translation boundaries. In the above example, the clause "我 /I 有 /have 空 /availability" (meaning "I am available") forms a complete semantic unit and it can be well predicted by the n-gram language model. However, across the semantic boundary, it could be "I think I am available", or "He says I am available", or "I do n't know whether I am available", or etc. There are numerous possibilities. So intuitively, the strength of n-gram conditioning within a semantic unit and across a semantic unit, should be weighted differently.

B. Experiments and Discussion

For the translation experiment of English to Chinese, we select data from corpora provided by NIST OpenMT15 to train, tune and test based on SMT. For training, we select domain 280K utterances and out-domain 2.4M utterances. For tuning and testing, 6.6K and 774 utterances are used.

We use perplexity minimization method [15] to combine 4 models: alignments from GIZA++ [8] and fast align models [5] from in-domain and out-domain training data. We used two 5-gram language models: one on the parallel in-domain text, the other one on Chinese Gigaword Corpus. The Chinese sentences are automatically segmented into words. However, MT scores are computed at the character level for tuning and evaluation.

To simulate the real-time translation, we concatenate every N utterance in the test set, remove all punctuations, predict

Table 6. Accuracy and speed comparison for flat-boundary constrained mode by concatenating every N utterances, and re-predicting punctuation and translation boundaries. Tested using multi-thread on CTS.

Model	N	BLEU	TER	METEOR	Time/s
PI+ Baseline Model	1	27.42	53.74	30.50	202
	2	26.19	54.09	30.40	284
	3	25.55	54.11	30.63	347
	4	25.48	54.27	30.59	391
	5	25.34	54.47	30.65	432
SBD+TBLM	1	27.45	53.69	30.48	65
	2	26.46	53.83	30.40	76
	3	25.59	53.97	30.47	78
	4	25.44	53.84	30.53	86
	5	25.39	53.51	30.74	90

semantic boundaries including punctuation marks, split every utterance into sentences according to full-stop punctuation, and then pass each sentence to the decoder. We did not re-tune the translation model. From the results in Table 6, the proposed method has substantially increased the decoding speed with no degradation in accuracy. The degradation from N=1 to N=2 is due to the increase of the proportion of SB in the middle of an utterance by concatenating utterances. We call this as flat-boundary constrained mode.

As Cho. et.al. (2014) has mentioned that long sentences translation is a challenge for neural machine translation (NMT)[4]. For Chinese to English translation, we decide to use an in-house NMT system to test the proposed SBD on in-house Chinese broadcast news data. Table 7 shows the translation experiment results. True punc indicates the translation is tested on the test data with gold standard punctuations. PI indicates the punctuation marks in the test data is inserted by the PI model. SBD indicates the proposed method is used to segment the data for translation. Considering the SBD model used here is trained on social media data, the translation result can be better if the SBD was trained on broadcast news domain. This will be shown in our future work.

Table 7. Translation accuracy results on Chinese data with average length >20 for NMT system

System	BLEU	NIST	TER	METEOR
True punc	0.1961	5.7416	0.6952	0.2634
PI	0.1570	4.9902	0.7525	0.2233
Proposed SBD	0.1644	5.1458	0.7749	0.2375

IV. Conclusions and Future Work

In this paper, we propose a parser-based dynamic semantic boundary detection method to improve the speech translation in terms of the translation quality and efficiency. Based on parsing tree, we defined three additional semantic boundary labels for English context and one additional semantic boundary label for Chinese context, besides the basic punctuation marks. With the two SBD models trained from CRF, the ASR output stream can be labelled very quickly. From the viewpoint of being semantic complete, we do not always treat comma as a translation boundary.

To realize the translation, we also propose a translation boundary language model to take the advantage of the boundary information. Our experiments show that the proposed SBD with the TBLM can improve the translation quality and reduce the latency. According to the translation boundary, some modifications on the decoder can allow translation of incomplete sentences. More details for this part will also be in our future work.

In summary, our experiments show that the idea of SBD helps to determine more complete semantic units including punctuated units for better and quicker translation. This SDB method can be applied to phrase-based machine translation as well as neural machine translation (NMT)[4] to efficiently fix the translation challenge, especially for long sentences.

References

[1] Danqi Chen and Christopher D Manning. “A Fast and Accurate Dependency Parser using Neural Networks”. *Proceedings of EMNLP 2014*

[2] Eunah Cho, Jan Niehues, Kevin Kilgour, Alex Waibel, “Punctuation Insertion for Real-time Spoken Language Translation” in IWSLT, Da Nang, Vietnam, 2015

[3] E. Cho, J. Niehues, and A. Waibel, “Segmentation and Punctuation Prediction in Speech Language Translation using a Monolingual Translation System,” in IWSLT, Hong Kong, China, 2012.

[4] Kyunghyun Cho, Bart van Merri¨enboer, Dzmitry Bahdanau, and Yoshua Bengio. “ On the properties of neural machine translation: Encoder-decoderapproaches.” arXiv preprint arXiv:1409.1259, 2014

[5] Chris Dyer, Victor Chahuneau, and Noah A Smith. “A simple, fast, and effective reparameterization of IBM model 2”, HLT-NAACL, pp. 644-648, 2013

[6] M. Federico, L. Bentivogli, M. Paul, and S. St¨uker. “Overview of the IWSLT 2011 evaluation campaign”. In Proceedings of IWSLT, 2011.

[7] Alvin Grissom II, He He, Jordan Boyd-Graber, John Morgan, and Hal Daume III. “Dont until the final verb wait: Reinforcement learning for simultaneous machine translation”. In Proc. EMNLP, 2014.

[8] Franz Josef Och and Hermann Ney. “Giza++: Training of statistical translation models.” 2000

[9] John Lafferty, Andrew McCallum, and Fernando CN Pereira. “Conditional random fields: Probabilistic models for segmenting and labelling sequence data”. In Proceedings of the international conference on machine learning, 2001.

[10] Yang Liu, et al., “Enriching speech recognition with automatic detection of sentence boundaries and disfluencies”. IEEE Transactions on Audio, Speech, and Language Processing,14(5):1526–1540, 2006

[11] Evgeny Matusov, Arne Mauser, and Hermann Ney. “Automatic sentence segmentation and punctuation prediction for spoken language translation.” In Proc. IWSLT,2006

[12] Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, Satoshi Nakamura, “Syntax-based simultaneous

translation through prediction of unseen syntactic constituents", ACL, 2015

[13] 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.

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

[15] Rico Sennrich. "Perplexity minimization for translation model domain adaptation in statistical machine translation". In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 539–549. ACL 2012

[16] V. K. R. Sridhar, J. Chen, S. Bangalore, A. Ljolje, and R. Chengalvarayan, "Segmentation strategies for streaming speech translation," in HLT-NAACL, 2013, pp. 230–238.

[17] Nina Zhou, Aw Aiti, Nattadaporn Lertcheva, Xuancong Wang, "A Word Labeling Approach to Thai Sentence Boundary Detection and POS Tagging", Coling 2016.

[18] Jingbo Zhu , Muhua Zhu , Qiang Wang , Tong Xiao. "NiuParser: A Chinese Syntactic and Semantic Parsing Toolkit", In *Proc. of ACL*, demonstration session, 2015

[19] CRF++ Yet Another toolkit: https://taku910.github.io/crfpp/

[20] Stanford Parser – Stanford Natural Language Processing group: https://nlp.stanford.edu:8080/parser/