Segmentation and Disfluency Removal for Conversational Speech Translation

Hany Hassan, Lee Schwartz, Dilek Hakkani-Tür, Gokhan Tur

Microsoft Research

{hanyh,leesc}@microsoft.com, {dilek,gokhan.tur}@ieee.org

Abstract

In this paper we focus on the effect of on-line speech segmentation and disfluency removal methods on conversational speech translation. In a real-time conversational speech to speech translation system, on-line segmentation of speech is required to avoid latency beyond few seconds. While sentential unit segmentation and disfluency removal have been heavily studied mainly for off-line speech processing, to the best of our knowledge, the combined effect of these tasks on conversational speech translation has not been investigated. Furthermore, optimization of performance given maximum allowable system latency to enable a conversation is a newer problem for these tasks. We show that the conventional assumption of doing segmentation followed by disfluency removal is not the best practice. We propose a new approach to do simple-disfluency removal followed by segmentation and then by complex-disfluency removal. The proposed approach shows a significant gain on translation performance of up to 3 Bleu points with only 6 second latency to look ahead, using state-of-the art machine translation and speech recognition systems.

Index Terms: speech translation, disfluency removal, segmentation, sentence units, speech processing

1. Introduction

Conversational speech translation (S2S) systems should provide real-time translations with acceptable latency. This is a challenging task due to the interaction of the three components that compose these systems, namely automatic speech recognition (ASR), machine translation (MT), and text-to-speech (TTS).

ASR systems usually segment the stream of recognized words based on pause duration, which might not be adequate for translation systems. Machine translation systems would provide more accurate translation when provided with full sentences[1]. Similarly, TTS requires full sentences that are short enough to provide acceptable latency between turn-to-turn times. Breaking the word stream into sentence units that can be consumed by MT and TTS systems is crucial to the overall system performance.

Spontaneous conversational speech often has numerous disfluencies, such as filler words, repetitions, revisions and stuttering. These disfluencies usually affect the performance of any MT system[2] since they are generally trained on well-formed text. For example, a phrase-based MT system would suffer from disfluencies that break the phrases and prevent the system from matching longer phrases from the phrase table. Similarly, a syntactic MT system would not be able to get a good parse for the utterances, due to the disfluencies between words.

By way of example, Figure 1 presents an English transcript that is hard to interpret due to its disfluencies. It is obvious that the translation accuracy is significantly affected. When segmentation and disfluency removal are performed on the transcripts, the translation accuracy is much better. The third example in the figure shows how imperfect segmentation leads to less accurate disfluency removal which, in turn, leads to poorer translation quality.

Transcripts:
um no i mean yes but i am i've never done it myself have you done that uh yes
Spanish MT:
Um no i decir si pero estoy nunca lo hice me has hecho eso si

Segmented and Disfluency Removed:
Yes, but i've never done it myself.
Have you done that? Yes.
Spanish MT:
Si, pero nunca lo hice yo mismo.
Has hecho eso? Si.

Imperfect Segmentation and Disfluency Removal:
No. i mean yes. But i am.
I've never done it myself.
Have you done that yes.
Spanish MT:
NO. Quiero decir que si.
Pero yo soy. Nunca lo he hecho yo mismo. Has hecho que si.

Figure 1: Disfluency, Segmentations and Translation

In this paper, we investigate the interaction between segmentation and disfluency removal and their impact on each other and on translation accuracy as well. Previous work on S2S has considered these tasks in isolation: [1] studied the impact of sentence segmentation on machine translation for broadcast news speech. [2] investigated impact of disfluency removal for offline translation of broadcast conversations. Two distinct characteristics of an S2S system make these tasks more challenging: i) spontaneous conversational speech is full of disfluencies, resulting in a genre mismatch with the corresponding MT system ii) for more natural interactions, real-time translation of conversations requires low system latency, maybe few to several seconds, so that the other party does not need to wait for the whole turn to end.

We found that the best approach to achieve better MT quality is to have two stages for disfluency handling; simple-disfluency removal done before segmentation and complex-disfluency removal done after segmentation. To the best of our knowledge, this is the first study that examines the effect of both segmentation and disfluency handling on online MT quality for conversational speech. Our results show an improvement of up to 3 Bleu points on a Switchboard test set translated from English into Spanish.

In the following sections, we describe the sentence boundary detection in Section 2. The disfluency removal systems are described in Section 3. Then in Section 4, we present the gen-

eral architecture of the S2S system employed. In Section 5, we present experimental results with various configurations of these two components in an S2S framework.

2. Sentence Unit Boundary Detection

Following the speech processing literature, we treat sentence unit boundary detection as a sequence classification problem [3, 4, 5, 6, 7, 8]. After each word we identify if there should be a sentence boundary or not. We restrict the classification problem to binary classification.

For training the model, we use human transcriptions of conversational speech data with manually annotated sentence boundaries. We do not distinguish between a period and a question mark at this stage. All sentence boundaries are considered the same in the training data.

Our Sentence Unit Boundary Detector (SUD) is very similar to the one proposed in [5] with an extended set of features to fit the translation task at hand.

For this sequence classification problem, Conditional Random Field (CRF) classifier [9] that utilizes lexical and speech pause features is employed. CRF is an undirected graphical model with log linear distribution of the label sequence y given the input words and features F with associated learned weights, λ_k, using the L-BFGS algorithm:

$$\mathbb{P}(y|F) = \frac{1}{Z(W,F)} \sum_k \lambda_k G(y,F) \quad (1)$$

where Z is the normalization factor.

2.1. Data

We used Switchboard [11] and Fisher [12] data sets to train the sentence boundary detector. We remove all non-boundary punctuation and keep only the sentence boundaries. As an example, consider the transcripts shown below, in which the end of sentence punctuation is used to indicate a sentence boundary while we ignore any other punctuation in the transcripts:

Punctuated Transcripts: *no, i've never done it myself. have you done that?*
SU Training Data: *no i've never done it myself [SB] have you done that [SB]*

2.2. Sentence Boundary Detection Features

The extracted features for SUD, in a sliding window of two to the left and two to the right of the current word, are as follows:

- *Lexical features*: the actual identity of the word.
- *Word clusters*: Brown word clusters [13] trained on $500M$ words with 1000 classes.
- Part-of-speech (POS) tags: POS tags as described below.
- *Speech based pause duration features*: binning the pause gap between any two utterances with 10 bins form 0 to 9 corresponds to 0 second to 1 or more seconds.
- *Phrase translation table feature*: whether or not the sequence of words exists as a phrase in the translation phrase table. This feature should discourage inserting sentence boundaries in the middle of phrases for which we have good translation.

2.3. POS model

In all the models, we deploy a POS feature from a POS tagger trained on conversational data. We use Switchboard data (LDC99T42) which is annotated with POS tags for conversational speech. We opted to train a POS tagger specifically for conversational data style since conventional POS taggers would not perform well with text characterized by disfluencies and spontaneous speech artifacts. Our POS tagger is another CRF classifier with the following features:

- *Lexical features*: the actual identity of the word.
- *Word clusters*: Brown word clusters as described above.
- *Word suffixes*: up to last three characters of the word.

All features are used in a sliding window of two to the left and two to the right of the current word. The classifier has 40 POS tags. Its accuracy is 95.96 F-Score on Switchboard development data according to the split defined in [5].

3. Disfluency Removal

Conversational speech has many types of disfluencies, as detailed in [10]. In this work, we focus on two categories of disfluencies as follows:

- *Simple-Disfluencies*: Filler Pauses (FP), i.e. "uh" , "um", "oh". Discourse Markers (DM), i.e. : "i mean", "you know", "anyway".
- *Complex-Disfluencies*: complex edits which represent revisions, correction, or repetition of syntactically similar units, as in the string *"yes i'm i've done this before sorry after him"*

Disfluency removal for speech translation has been addressed previously in the literature. For example, [14] employed a noisy channel approach to map from disfluent broadcast news transcripts into fluent ones. More recently, [2] used three systems in cascade to handle disfluency removal, the first being based on a hidden event language model and rules that detect interruption words, the second being a CRF classifier that detects edit terms, and the third being a rule-based system that detects filler words.

The previous work on disfluency removal mainly focused on offline speech processing, mostly in broadcast news [15, 16, 17, 18]. They assumed perfect, or human-provided, segmentations. They did not address the effect of imperfect segmentation on disfluency removal and translation, and some assumed that segmentation should be done before disfluency removal.

In this paper, we investigate the combined effects of segmentation and disfluency removal, and their effect on translation. To the best of our knowledge, this is the first work to study this problem on conversational speech translation.

We propose two independent systems to handle disfluency removal. The first system handles simple-disfluencies and needs local contextual information, but not sentence boundary units. Moreover, simple-disfluency detection before sentence boundary detection can actually help in improving the overall system performance, as we will show in the experiments. The second system is responsible for handling the more sophisticated edits, which need non-local context information, syntactic information and sentence boundary information.

3.1. Simple-Disfluency Removal

The simple-disfluency removal system is a CRF classifier with the following disfluency classes, as defined in [5]

- FP: filler pause such as "uh"
- DM: discourse marker such as "you know"
- CC: such as "and" when used as a starter
- EE: edits such as "i mean"

We acquire the training data for this classifier from Switchboard LDC99T42 data, which is annotated with disfluencies. We restrict the classifier types to those mentioned above, avoiding the more complex-disfluencies and edits, which are much harder to detect.

The classifier uses the following features in a sliding window of two to the left and two to the right of the current word:

- *Lexical features*: the actual identity of the word.
- *Word clusters*: Brown word clusters as described above.
- *POS tags*: POS tags as described above.

3.2. Complex-Disfluency Removal

Our complex-disfluency removal system is composed of two systems; the first is a CRF classifier that inserts punctuation and the second is based on a knowledge-based parser.

The purpose of the punctuation annotation system is twofold; first, to provide initial punctuation of sentences, which can help during translation; second, to provide markers for the parser to highlight possible disfluencies in the sentence. The punctuation annotation can be handled as a tagging problem as proposed in [19], where we annotate each word with the possible punctuation to insert after it.

We restrict the annotation to three classes only, period, comma or nothing. The CRF classifier used for punctuation annotation is very similar to the sentence unit detection classifier described above. The main difference between the classifier outcomes is whether to insert a comma or a period after each word. The punctuation classifier uses the same lexical features as the sentence boundary detector without any speech features.

The punctuated text becomes the input to the parser. The main objective of the parser for disfluency removal is to convert input strings into "reasonably" grammatical strings. We used a broad-coverage rule-based parser, NLPWin[20], to help identify disfluencies. The parser is forgiving; it does not require grammatically correct input or input that is correctly punctuated to produce a parse. However, the better the punctuation, the better the parse. Therefore, the disfluency detection task is easier on automatically punctuated ASR than on straight ASR input.

The procedure for removing disfluencies is to parse the punctuated input string, identify disfluencies, remove them, and create a new, modified string. That new string is then parsed, disfluencies are identified and removed, and the same procedure applies iteratively until a preset limit of parses is reached or no more disfluencies can be identified for removal. This is in part, similar to the main idea in the Johns Hopkins University Summer Workshop study on reranking sentence segmentation using parsing [8].

To exemplify the process, we trace the main steps taken in removing disfluencies from the following sentence and its parse: *well, of course, it's, you know, it's the last thing in the world, you want to do you know.*
FITTED[VP_1[VP_2[well, of course its]$_2$, VP_3[you know, VP_4[it is NP_6[the last thing PP_5[in the world]$_5$]$_6$ VP_7[you want to do you]$_7$]$_4$]$_3$]$_1$ VP_8[know .]$_8$]

The sentence is parsed, producing a tree with the top-level node, FITTED, which means that a coherent parse of the string could not be attained. The disfluency removal component looks for its first disfluency candidates. These are the ones that are easiest to identify, i.e., fillers and discourse markers (i.e.: um, uh, you know, well). The strings "you know" and "well" are difficult to identify as disfluencies. In the example, the commas following "well" and the commas surrounding the first "you know" are helpful to the parser in producing an analysis in which the strings can be easily identified as disfluencies. However, even without a preceding comma, the final "you know" is detected as a filler. The removal component uses the following information to determine that the strings are disfluencies:

- "well" is an interjection, not a modifier of a verb.
- The first instance of "you know" is not only surrounded by commas, but it also has been parsed as part of a VP_3 construction, that is characterized by a comma splice (rather than a full conjunction)
- The parse of the final instance of "you know" has "know" in a disconnected VP_8 with no internal arguments

With the above evidence, the identified fillers and discourse markers are removed to produce the modified string: *of course, it's,,, it's the last thing in the world, you want to do.*

The parse of the modified input string is no longer a FITTED parse, though it is still not optimal. The grammaticality of the input is improving. No fillers or discourse markers are found in the new parse, but a repetition is found. The repeated *"it's"* is deleted to produce the string:

of course, it's the last thing in the world, you want to do.

This new string is parsed and no disfluencies are identified, so it is the final output of the disfluency removal component. It is worth noting the disfluency removal does not delete all repetitions it finds. So, for example, the input string *"it is a big, big fish"* is not modified by the disfluency component because the parser produces a legitimate structure for *"big, big"* as a premodifying adjective.

4. Systems and Data

Microsoft Research S2S Translation System is a large vocabulary real-time robust speech translation system, covering multiple languages, consisting of the ASR and MT components as described below.

4.1. Speech Recognition System

The ASR system is an HMM-based triphone/trigram large-vocabulary continuous speech recognition system that is standard, except that it uses a deep neural network for acoustic modeling; specifically a context-dependent deep-neural network hidden Markov model [21], [22] . The system is speaker-independent and trained on 2000h of data (SWBD and Fisher corpora), as described in [23].

4.2. Machine Translation System

The MT system is a typical phrase-based system similar to [24]. The details of the decoder can be found here [25]. The system is a large scale English to Spanish system trained on 29M sentence pairs from a variety of sources, including UN data, WMT, Europarl and web crawled data. The language model is a 5-gram model trained on 600M sentences.

4.3. Data

The Sentence Boundary Detector was trained on both Switchboard and Fisher Data. The same data was used to train the punctuation annotator for the complex-disfluency removal system. The simple-disfluency removal system has been trained on the rich annotated SwitchBoard data (LDC99T42).

The test set is the SwitchBoard test set according to the split in [5]. A bilingual annotator translated the English transcripts into fluent Spanish. The test set has 67 conversations with total of 4522 sentences; each one represents a turn-taking in the conversation. When the sentence is empty on the English source side, i.e., composed of non-audible segments or just "uh", we remove it from the set. We report case-insensitive Bleu score[26] on all systems ignoring any punctuation.

5. Experiments

We present experimental results in two sets: First we analyze the effect of sentence boundary detection alone on MT performance *given latency* of few seconds. Then we present the results with various combinations of disfluency removal and sentence segmentation.

5.1. Sentence Boundary Detection

In this set of experiments, we evaluate the effect of sentence boundary detection on translation performance. Table 1 summarizes the results using various segmentation methods.

Segmentation	Transcripts	ASR	F-score	Latency
OffLine	22.11	19.39	NA	300 sec
Turn Taking	22.13	19.13	NA	6 sec
Chunk	19.75	17.16	10.81	5 sec
Pause	20.32	18.78	36.02	4 sec
SU1	22.60	19.46	80.53	6 sec
SU2	22.67	19.48	80.91	6 sec
SU3	22.54	19.28	78.36	8 sec

Table 1: Sentence Boundary Detection effect on translation. SU1:CRF lexical feature, SU2:CRF lexical + pause features and SU3:CRF lexical features on SWBD+Fisher. BLEU score is reported on ENU-ESN SWBD testset with one reference translation for both Human Transcripts and ASR output. F-score and latency for SU detection are reported on SWBD dev.

We provided a number of baseline performance figures. First, we do not employ any sentence segmentation, discarding the latency requirement, i.e., performing offline translation of the whole conversation. This setup resulted in a Bleu score of 22.11. A variation of this is using turn boundaries as translation units, which resulted in similar result. However, the turn taking may not be a feasible assumption in another setting with lighter interaction, such as lecture translation. Another baseline is using chunks of 10 words, which resulted in a loss of 2 Bleu points on ASR output, as expected. Pause-motivated segmentation recovered most of this loss, segmenting whenever there is a pause duration of more than 0.5 seconds.

Comparing the segmentation models with these baselines, we see that the CRF classifiers outperform these simpler segmentation methods. This is true for both human transcriptions and ASR output. When pause duration is added as a feature, SU2 does slightly better than SU1 in terms of F-score and BLEU, and performed the best amongst these CRF models.

The additional improvement using pause duration however is not consistent with previous research on combining pause duration and lexical information [27]; though our findings are similar to results in [19]. This may be due to the fact that we use very strong lexical features, which outperform the pause duration feature. With Fisher data added to the classifier data (SU3), scores did not improve on SWBD data, possibly due to overfitting the model to the switchboard data style.

5.2. Disfluency Effect

The simple disfluencies represent 21% of the SwitchBoard data, while the complex disfluencies represent 16% of the data. The F-score of the simple disfluency removal classifier is 98.27% which is very accurate.

We tried different scenarios with disfluency removal. Table 2 shows various experiments. Using SU2 followed by disfluency removal does help the translation. When we have two systems for disfluency removal, i.e., simple-disfluency removal before segmentation and complex-disfluency removal after segmentation, the system improves significantly, with +2.6 Bleu points over the turn-taking case and almost +3 Bleu points gain compared to the simple pause-based segmentation system.

Segmentation	Disfluency	Transcripts	ASR
Turn Taking	None	22.13	19.13
Turn Taking	SA+CA	23.46	20.49
Pause	None	20.32	18.78
Pause	SA+CA	22.53	19.32
SU2	None	22.67	19.48
SU2	SA+CA	25.11	21.24
SU2	SB	24.79	20.95
SU2	SB+CA	25.65	21.76

Table 2: Disfluency Removal effect, reported BLEU score on ENU-ESN SWBD testset. SA: Simple Disfluency Removal After Segmentation, SB: Simple before Segmentation, CA: Complex after Segmentation.

6. Conclusions

In this work, we have investigated the interaction between sentence boundary detection and disfluency removal and its effect on conversational speech translation. We show that the conventional practice of doing segmentation followed by disfluency removal is not optimal. Instead, we have showed that translation quality improves with simple-disfluency removal followed by segmentation and then complex-disfluency removal relying on sentence unit determination. The proposed approach achieves a gain of almost 3 Bleu points over pause-based segmentation and 2.6 Bleu points over human-segmented data.

As future work, we will investigate the possibility of having real time translation with variable latency depending on disfluency, and explore the possibility of refining the ASR output adjacent to the disfluent parts.

7. Acknowledgements

We would like to thank Frank Seide for providing the ASR system and for helpful discussions, and the anonymous reviewers for helpful comments.

8. References

[1] Matusov, E. Hillard, D., Magimai-Doss, M., Hakkani-Tür, D. Ostendorf, M. , Ney, H., "Improving Speech Translation with Automatic Boundary Prediction", In Proceedings of the Interspeech, 2007.

[2] Wang, W., Tur, G., Zheng, J. , Ayan, N. Automatic Disfluency Removal for Improving Spoken Language Translation, in ICASSP, 2010.

[3] Shriberg, E., Stolcke A., Hakkani-Tür, D, Tur, G., "Prosody-Based Automatic Segmentation of Speech into Sentences and Topics", in Speech Communication, 127-154, 2000.

[4] Ang, J., Liu, Y., Shriberg, E., "Automatic Dialog Act Segmentation and Classification in Multiparty Meetings", in Proceedings of the ICASSP, 2005.

[5] Liu, Y. and Shriberg, E. and Stolcke, A. and Hillard, D. and Ostendorf, M. and Harper, M. " Enriching speech recognition with automatic detection of sentence boundaries and disfluencies", IEEE Transcriptions on Audio, Speech & Language Processing, 14(5):1526–1540, 2006.

[6] Zimmerman, M., Hakkani-Tür, D., Fung, J., Mirghafori, N., Gottlieb, L., Shriberg, E., Liu, Y., "The ICSI+ Multilingual Sentence Segmentation System", in Proceedings of the ICSLP, 2006.

[7] Favre, B., Hakkani-Tür, D., Petrov, S., Klein, D., "Efficient Sentence Segmentation Using Syntactic Features." in Proceedings of the IEEE SLT Workshop, 2008.

[8] Roark, B., Liu, Y., Harper, M., Stewart, R., Lease, M., Snover, M., Shafran, I., Dorr, B., Hale, J., Krasnyanskaya, A., Yung, L., "Reranking for Sentence Boundary Detection in Conversational Speech", in Proceedings of the ICASSP, 2006.

[9] Lafferty, J., McCallum, A. and Pereira, F.,"Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", in Proceedings of ICML, 282-289, 2001.

[10] Shriberg, E., " Preliminaries to a theory of Speech Disfluencies". PhD thesis, University of California at Berkeley, 1994.

[11] Godfrey, J., Holliman, E., McDaniel, J., "Switchboard: Telephone Speech Corpus for Research and Development", in Proceedings of the ICASSP, 517-520, 1992.

[12] Cieri, C. , Miller, D., Walker, K., "The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text.", In Proceedings of LREC, 69–71, 2004.

[13] Brown, P. , de Souza, P., Mercer, R. Della Pietra, V., Lai ,J., "Class-based n-gram models of natural language.", Computational Linguistics, 467479, 1992.

[14] Rao, S., Lane, I., Schultz, T., "Improving Spoken Language Translation by Automatic Disfluency Removal: Evidence From conversational Speech Transcripts", In proceedings of Machine Translation Summit, 2007.

[15] Stolcke, A., Shriberg, E., "Statistical language modeling for speech disfluencies", in Proceedings of the ICASSP, 2006.

[16] Shriberg, E, Bates, R., Stolcke, A. "A prosody only decision-tree model for disfluency detection.", In Proceedings of Eurospeech, 1997.

[17] A. Stolcke, A., Shriberg, E., Bates, R., Ostendorf, M. Hakkani-Tur, D., Plauche, M., Tur, G. , Lu, Y., "Automatic detection of sentence boundaries and disfluencies based on recognized words.", In proceedings of ICSLP, 1998.

[18] Snover, M., Dorr, B., Schwartz,R., "A lexically-driven algorithm for disfluency detection", In Proceedings of HLT-NAACL, 2004.

[19] Huang,J., Zweig, G., "Maximum Entropy Model For Punctuation Annotation From Speech", In Proceedings of ICSLP, 2002.

[20] Heidorn, G., "Intelligent Writing Assistance", In Dale et al. Handbook of Natural language processing, Marcel Dekker, 2000.

[21] Yu, D., Deng, L. , Dahl, G. , "Roles of Pretraining and Fine-Tuning in Context-Dependent DNN-HMMs for Real-World Speech Recognition", in NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2010.

[22] Seide, F., Li, G., Yu, D., "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks", in Interspeech, 2011.

[23] Seide, F., Li, G. Chen, X., Yu, D. "Feature Engineering in Context-Dependent Deep Neural Networks for Conversational Speech Transcription", in ASRU, 2429, 2011

[24] R. C. Moore and C. Quirk, "Faster Beam-search Decoding for Phrasal Statistical Machine Translation", in In Proceedings of MT Summit XI. Citeseer, 2007.

[25] 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, J. A. Carroll, A. van den Bosch, and A. Zaenen, Eds. The Association for Computational Linguistics, 2007.

[26] Papineni, K., Roukos, S. , Ward, T., Zhu, W., "Bleu: a Method for Automatic Evaluation of Machine Translation", in Proceedings of ACL, 311–318, 2002.

[27] Guz, U., Hakkani-Tür, D., Cuendet, S., Tur, G., "Co-training Using Prosodic and Lexical Information for Sentence Segmentation" in Proceedings of the INTERSPEECH, 2007.