Tight Integration of Speech Disfluency Removal into SMT

Eunah Cho **Jan Niehues** **Alex Waibel**
Interactive Systems Lab
Institute of Anthropomatics
Karlsruhe Institute of Technology, Germany
{eunah.cho,jan.niehues,alex.waibel}@kit.edu

Abstract

Speech disfluencies are one of the main challenges of spoken language processing. Conventional disfluency detection systems deploy a hard decision, which can have a negative influence on subsequent applications such as machine translation. In this paper we suggest a novel approach in which disfluency detection is integrated into the translation process.

We train a CRF model to obtain a disfluency probability for each word. The SMT decoder will then skip the potentially disfluent word based on its disfluency probability. Using the suggested scheme, the translation score of both the manual transcript and ASR output is improved by around 0.35 BLEU points compared to the CRF hard decision system.

1 Introduction

Disfluencies arise due to the spontaneous nature of speech. There has been a great deal of effort to detect disfluent words, remove them (Johnson and Charniak, 2004; Fitzgerald et al., 2009) and use the cleaned text for subsequent applications such as machine translation (MT) (Wang et al., 2010; Cho et al., 2013).

One potential drawback of conventional approaches is that the decision whether a token is a disfluency or not is a hard decision. For an MT system, this can pose a severe problem if the removed token was not in fact a disfluency and should have been kept for the correct translation. Therefore, we pass the decision whether a word is part of a disfluency or not on to the translation system, so that we can use the additional knowledge available in the translation system to make a more reliable decision. In order to limit the complexity, the search space is pruned prior to decoding and represented in a word lattice.

2 Related Work

Disfluencies in spontaneous speech have been studied from various points of view. In the noisy channel model (Honal and Schultz, 2003), it is assumed that clean text without any disfluencies has passed through a noisy channel. The clean string is retrieved based on language model (LM) scores and five additional models. Another noisy channel approach involves a phrase-level statistical MT system, where noisy tokens are translated into clean tokens (Maskey et al., 2006). A tree adjoining grammar is combined with this noisy channel model in (Johnson and Charniak, 2004), using a syntactic parser to build an LM.

Fitzgerald et al. (2009) present a method to detect speech disfluencies using a conditional random field (CRF) with lexical, LM, and parser information features. While previous work has been limited to the postprocessing step of the automatic speech recogition (ASR) system, further approaches (Wang et al., 2010; Cho et al., 2013) use extended CRF features or additional models to clean manual speech transcripts and use them as input for an MT system.

While ASR systems use lattices to encode hypotheses, lattices have been used for MT systems with various purposes. Herrmann et al. (2013) use lattices to encode different reordering variants. Lattices have also been used as a segmentation tactic for compound words (Dyer, 2009), where the segmentation is encoded as input in the lattice.

One of the differences between our work and previous work is that we integrate the disfluency removal into an MT system. Our work is not limited to the preprocessing step of MT, instead we use the translation model to detect and remove disfluencies. Contrary to other systems where detection is limited on manual transcripts only, our sys-

tem shows translation performance improvements on the ASR output as well.

3 Tight Integration using Lattices

In this chapter, we explain how the disfluency removal is integrated into the MT process.

3.1 Model

The conventional translation of texts from spontaneous speech can be formulated as

$$\hat{e} = \arg\max_{e} p(e|\arg\max_{f_c} p(f_c|f)) \quad (1)$$

with

$$p(f_c|f) = \prod_{i=1}^{I} p(c_i|f_i) \quad (2)$$

where f_c denotes the clean string

$$f_c = \{f_1, \ldots, f_I \mid c_i = clean\} \quad (3)$$

for the disfluency decision class c of each token.

$$c \in \begin{cases} clean \\ disfluent \end{cases} \quad (4)$$

Thus, using the conventional models, disfluency removal is applied to the original, potentially noisy string in order to obtain the cleaned string first. This clean string is then translated.

The potential drawback of a conventional speech translation system is caused by the rough estimation in Equation 1, as disfluency removal does not depend on maximizing the translation quality itself. For example, we can consider the sentence *Use what you build, build what you use*. Due to its repetitive pattern in words and structure, the first clause is often detected as a disfluency using automatic means. To avoid this, we can change the scheme how the clean string is chosen as follows:

$$\hat{e} = \arg\max_{e,f_c}(p(e|f_c) \cdot p(f_c|f)) \quad (5)$$

This way a clean string which maximizes the translation quality is chosen. Thus, no instant decision is made whether a token is a disfluency or not. Instead, the disfluency probability of the token will be passed on to the MT process, using the log linear combination of the probabilities as shown in Equation 5.

In this work, we use a CRF (Lafferty et al., 2001) model to obtain the disfluency probability of each token.

Since there are two possible classes for each token, the number of possible clean sentences is exponential with regard to the sentence length. Thus, we restrict the search space by representing only the most probable clean source sentences in a word lattice.

3.2 CRF Model Training

In order to build the CRF model, we used the open source toolkit CRF++ (Kudoh, 2007). As unigram features, we use lexical and LM features adopted from Fitzgerald et al. (2009), and additional semantics-based features discussed in (Cho et al., 2013). In addition to the unigram features, we also use a bigram feature to model first-order dependencies between labels.

We train the CRF with four classes; `FL` for filler words, `RC` for (rough) copy, `NC` for non-copy and `0` for clean tokens. The class `FL` includes obvious filler words (e.g. *uh*, *uhm*) as well as other discourse markers (e.g. *you know*, *well* in English). The `RC` class covers identical or roughly similar repetitions as well as lexically different words with the same meaning. The `NC` class represents the case where the speaker changes what to speak about or reformulates the sentence and restarts the speech fragments. The disfluency probability P_d of each token is calculated as the sum of probabilities of each class.

3.3 Lattice Implementation

We construct a word lattice which encodes long-range reordering variants (Rottmann and Vogel, 2007; Niehues and Kolss, 2009). For translation we extend this so that potentially disfluent words can be skipped. A reordering lattice of the example sentence *Das sind die Vorteile, die sie uh die sie haben.* (En.gls: *These are the advantages, that you uh that you have.*) is shown in Figure 1, where words representing a disfluency are marked in bold letters. In this sentence, the part *die sie uh* was manually annotated as a disfluency, due to repetition and usage of a filler word.

Table 1 shows the P_d obtained from the CRF model for each token. As expected, the words *die sie uh* obtain a high P_d from the CRF model.

In order to provide an option to avoid translating a disfluent word, a new edge which skips the word is introduced into the lattice when the word has a higher P_d than a threshold θ. During decoding the importance of this newly introduced edge is optimized by weights based on the disfluency proba-

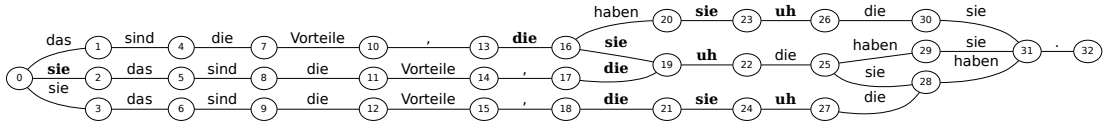

Figure 1: Reordering lattice before adding alternative clean paths for an exemplary sentence

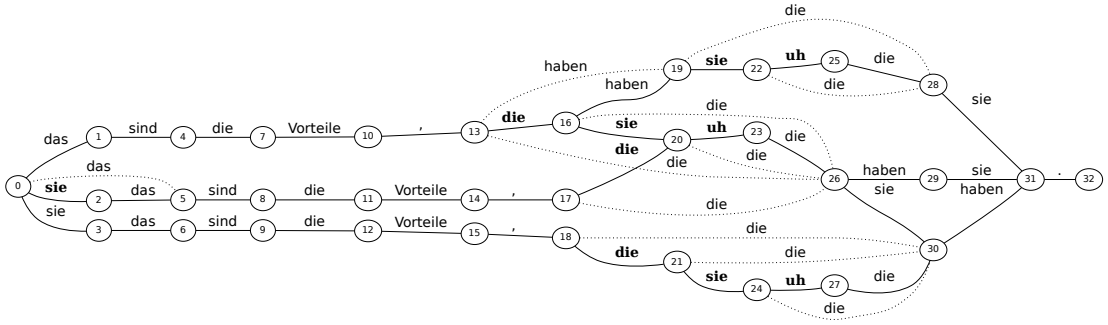

Figure 2: Extended lattice with alternative clean paths for an exemplary sentence

das	0.000732	**sie**	**0.953126**
sind	0.004445	**uh**	**0.999579**
die	0.013451	die	0.029010
Vorteile	0.008183	sie	0.001426
,	0.035408	haben	0.000108
die	**0.651642**	.	0.000033

Table 1: Disfluency probability of each word

bility and transition probability. The extended lattice for the given sentence with $\theta = 0.5$ is shown in Figure 2, with alternative paths marked by a dotted line. The optimal value of θ was manually tuned on the development set.

4 System Description

The training data for our MT system consists of 1.76 million sentences of German-English parallel data. Parallel TED talks[1] are used as in-domain data and our translation models are adapted to the domain. Before training, we apply preprocessing such as text normalization, tokenization, and smartcasing. Additionally, German compound words are split.

To build the phrase table we use the Moses package (Koehn et al., 2007). An LM is trained on 462 million words in English using the SRILM Toolkit (Stolcke, 2002). In order to extend source word context, we use a bilingual LM (Niehues et al., 2011). We use an in-house decoder (Vogel, 2003) with minimum error rate training (Venugopal et al., 2005) for optimization.

For training and testing the CRF model, we use 61k annotated words of manual transcripts of university lectures in German. For tuning and testing the MT system, the same data is used along with its English reference translation. In order to make the best use of the data, we split it into three parts and perform three-fold cross validation. Therefore, the train/development data consists of around 40k words, or 2k sentences, while the test data consists of around 20k words, or 1k sentences.

[1] http://www.ted.com

5 Experiments

In order to compare the effect of the tight integration with other disfluency removal strategies, we conduct different experiments on manual transcripts as well as on the ASR output.

5.1 Manual Transcripts

As a baseline for manual transcripts, we use the whole uncleaned data for development and test. For "No *uh*", we remove the obvious filler words *uh* and *uhm* manually. In the CRF-hard experiment, the token is removed if the label output of the CRF model is a disfluency class. The fourth experiment uses the tight integration scheme, where new source paths which jump over the potentially noisy words are inserted based on the disfluency probabilities assigned by the CRF model. In the next experiments, this method is combined with other aforementioned approaches. First, we apply the tight integration scheme after we remove all obvious filler words. In the next experiment, we first remove all words whose P_d is higher than 0.9 as early pruning and then apply the tight integration scheme. In a final experiment, we conduct an oracle experiment, where all words annotated as a disfluency are removed.

5.2 ASR Output

The same experiments are applied to the ASR output. Since the ASR output does not contain reliable punctuation marks, there is a mismatch between the training data of the CRF model, which is manual transcripts with all punctuation marks, and the test data. Thus, we insert punctuation marks and augment sentence boundaries in the ASR output using the monolingual translation system (Cho et al., 2012). As the sentence boundaries differ from the reference translation, we use the Levenshtein minimum edit distance algorithm (Matusov et al., 2005) to align hypothesis for evaluation. No optimization is conducted, but the scaling factors obtained when using the correponding setup of manual transcripts are used for testing.

5.3 Results

Table 2 shows the results of our experiments. The scores are reported in case-sensitive BLEU (Papineni et al., 2002).

System	Dev	Text	ASR
Baseline	23.45	22.70	14.50
No *uh*	25.09	24.04	15.10
CRF-hard	25.32	24.50	15.15
Tight int.	25.30	24.59	15.19
No *uh* + Tight int.	25.41	24.68	15.33
Pruning + Tight int.	25.38	**24.84**	**15.51**
Oracle	25.57	24.87	-

Table 2: Translation results for the investigated disfluency removal strategies

Compared to the baseline where all disfluencies are kept, the translation quality is improved by 1.34 BLEU points for manual transcripts by simply removing all obvious filler words. When we take the output of the CRF as a hard decision, the performance is further improved by 0.46 BLEU points. When using the tight integration scheme, we improve the translation quality around 0.1 BLEU points compared to the CRF-hard decision. The performance is further improved by removing *uh* and *uhm* before applying the tight integration scheme. Finally the best score is achieved by using the early pruning coupled with the tight integration scheme. The translation score is 0.34 BLEU points higher than the CRF-hard decision. This score is only 0.03 BLEU points less than the oracle case, without all disfluencies.

Experiments on the ASR output also showed a considerable improvement despite word errors and consequently decreased accuracy of the CRF detection. Compared to using only the CRF-hard decision, using the coupled approach improved the performance by 0.36 BLEU points, which is 1.0 BLEU point higher than the baseline.

System	Precision	Recall
CRF-hard	0.898	0.544
Pruning + Tight int.	0.937	0.521

Table 3: Detection performance comparison

Table 3 shows a comparison of the disfluency detection performance on word tokens. While recall is slightly worse for the coupled approach, precision is improved by 4% over the hard decision, indicating that the tight integration scheme decides more accurately. Since deletions made by a hard decision can not be recovered and losing a meaningful word on the source side can be very critical, we believe that precision is more important for this task. Consequently we retain more words on the source side with the tight integration scheme, but the numbers of word tokens on the translated target side are similar. The translation model is able to leave out unnecessary words during translation.

6 Conclusion

We presented a novel scheme to integrate disfluency removal into the MT process. Using this scheme, it is possible to consider disfluency probabilities during decoding and therefore to choose words which can lead to better translation performance. The disfluency probability of each token is obtained from a CRF model, and is encoded in the word lattice. Additional edges are added in the word lattice, to bypass the words potentially representing speech disfluencies.

We achieve the best performance using the tight integration method coupled with early pruning. This method yields an improvement of 2.1 BLEU points for manual transcripts and 1.0 BLEU point improvement over the baseline for ASR output.

Although the translation of ASR output is improved using the suggested scheme, there is still room to improve. In future work, we would like to improve performance of disfluency detection for ASR output by including acoustic features in the model.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

References

Eunah Cho, Jan Niehues, and Alex Waibel. 2012. Segmentation and Punctuation Prediction in Speech Language Translation using a Monolingual Translation System. In *Proceedings of the International Workshop for Spoken Language Translation (IWSLT)*, Hong Kong, China.

Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2013. CRF-based Disfluency Detection using Semantic Features for German to English Spoken Language Translation. In *Proceedings of the International Workshop for Spoken Language Translation (IWSLT)*, Heidelberg, Germany.

Chris Dyer. 2009. Using a Maximum Entropy Model to Build Segmentation Lattices for MT. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado, USA, June. Association for Computational Linguistics.

Erin Fitzgerald, Kieth Hall, and Frederick Jelinek. 2009. Reconstructing False Start Errors in Spontaneous Speech Text. In *Proceedings of the European Association for Computational Linguistics (EACL)*, Athens, Greece.

Teresa Herrmann, Jan Niehues, and Alex Waibel. 2013. Combining Word Reordering Methods on different Linguistic Abstraction Levels for Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation*, Altanta, Georgia, USA, June. Association for Computational Linguistics.

Matthias Honal and Tanja Schultz. 2003. Correction of Disfluencies in Spontaneous Speech using a Noisy-Channel Approach. In *Eurospeech*, Geneva.

Mark Johnson and Eugene Charniak. 2004. A TAG-based Noisy Channel Model of Speech Repairs. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the Association for Computational Linguistics (ACL), Demonstration Session*, Prague, Czech Republic, June.

Taku Kudoh. 2007. CRF++: Yet Another CRF Toolkit.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilitic Models for Segmenting and Labeling Sequence Data. In *ICML*, Massachusetts, USA.

Sameer Maskey, Bowen Zhou, and Yuqing Gao. 2006. A Phrase-Level Machine Translation Approach for Disfluency Detection using Weighted Finite State Tranducers. In *Interspeech*, Pittsburgh, PA.

Evgeny Matusov, Gregor Leusch, Oliver Bender, and Herrmann Ney. 2005. Evaluating Machine Translation Output with Automatic Sentence Segmentation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Boulder, Colorado, USA, October.

Jan Niehues and Muntsin Kolss. 2009. A POS-Based Model for Long-Range Reorderings in SMT. In *Proceedings of the 4th Workshop on Statistical Machine Translation*, Athens, Greece.

Jan Niehues, Teresa Herrmann, Stephan Vogel, and Alex Waibel. 2011. Wider Context by Using Bilingual Language Models in Machine Translation. In *Proceedings of the 6th Workshop on Statistical Machine Translation*, Edinburgh, UK.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. Technical Report RC22176 (W0109-022), IBM Research Division, T. J. Watson Research Center.

Kay Rottmann and Stephan Vogel. 2007. Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model. In *TMI*, Skövde, Sweden.

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. Denver, Colorado, USA.

Ashish Venugopal, Andreas Zollman, and Alex Waibel. 2005. Training and Evaluation Error Minimization Rules for Statistical Machine Translation. In *WPT-05*, Ann Arbor, MI.

Stephan Vogel. 2003. SMT Decoder Dissected: Word Reordering. In *Int. Conf. on Natural Language Processing and Knowledge Engineering*, Beijing, China.

Wen Wang, Gokhan Tur, Jing Zheng, and Necip Fazil Ayan. 2010. Automatic Disfluency Removal for Improving Spoken Language Translation. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.