CRF-based Disfluency Detection using Semantic Features for German to English Spoken Language Translation

Eunah Cho, Thanh-Le Ha, Alex Waibel

International Center for Advanced Communication Technologies - InterACT
Institute of Anthropomatics
Karlsruhe Institute of Technology, Germany
{eunah.cho|thanh-le.ha|alex.waibel}@kit.edu

Abstract

Disfluencies in speech pose severe difficulties in machine translation of spontaneous speech. This paper presents our conditional random field (CRF)-based speech disfluency detection system developed on German to improve spoken language translation performance.

In order to detect speech disfluencies considering syntactics and semantics of speech utterances, we carried out a CRF-based approach using information learned from the word representation and the phrase table used for machine translation. The word representation is gained using recurrent neural networks and projected words are clustered using the k-means algorithm. Using the output from the model trained with the word representations and phrase table information, we achieve an improvement of 1.96 BLEU points on the lecture test set. By keeping or removing human-annotated disfluencies, we show an upper bound and lower bound of translation quality. In an oracle experiment we gain 3.16 BLEU points of improvement on the lecture test set, compared to the same set with all disfluencies.

1. Introduction

Natural language processing (NLP) tasks often suffer from disfluencies in spontaneous speech. In spontaneous speech, speakers occasionally talk with disfluencies such as repetitions, stuttering, or filler words. These speech disfluencies inhibit proper processing for other subsequent applications, for example machine translation (MT) systems.

MT systems are generally trained using well-structured, cleanly written texts. The mismatch between this training data and the actual test data, in this case spontaneous speech, causes a performance drop. A system which reconstructs the non-fluent output from an automatic speech recognition (ASR) system into the proper form for subsequent applications will increase the performance of the application.

A considerable number of works on this task such as [1] and [2] focus on English, from the point of view of the ASR systems. One of our goals is to extend this work to German, and also apply it to the MT task, in order to analyze the effect of speech disfluencies on MT.

1.1. Disfluencies in Spontaneous Speech

Filler words (e.g. “uh”, “uhm”) are a common disfluencies in spontaneous speech. Discourse markers (e.g. “you know”, “well” in English) are considered filler words as well. Another common disfluency is repetition, where speakers repeat their words. A repetition can either be an identical repetition, where speakers exactly repeat a word or phrase, or a rough repetition, where they correct themselves using similar words. Simplified examples of such repetitions from our disfluency annotated lecture data with English gloss translation are shown in Table 1, in which the identical repetition is on the upper part, and the rough repetition is on the lower part.

Table 1: *Repetitions in spontaneous speech*

Source	Das sind die Vorteile, **die Sie die Sie** haben.
En.gls	These are the advantages, **that you that you** have.
Source	**Da gibt es da gab es** nur eins.
En.gls	**There is there was** only one.

Another type of speech disfluency, where several speech fragments are dropped and new fragments are introduced, is restart fragments. As presented in Table 2, the speaker starts a new way of forming the sentence after aborting the first several utterances. Although the example shown in this table depicts a case where the context is still kept in the following new utterances, occasionally we confront other cases where the previous context is abandoned and a new topic is discussed in spontaneous speech.

Table 2: *Restart fragment in spontaneous speech*

Source	**Das ist alles, was Sie** das haben Sie alles gelernt, und jetzt können Sie...
Engl. gloss	**That is all, what you** you have learned all of this, and now can you...

1.2. Motivation

Detecting obvious filler words and simple repetitions can be more feasible than other sorts of disfluencies for automatic modeling techniques, using lexical patterns such as typical filler word tokens and repetitive part-of-speech (POS) tokens as in previous work [2, 3]. Although it is the case for obvious disfluencies (i.e. “uh”, “uhm”, same repetitive tokens, and so on), we are confronted with many other cases where it is hard to recognize or decide whether the token is a disfluency or not via automatic means. This issue can be consistent even when the disfluency is filler words or repetitive tokens. Table 3 contains a sentence from the annotated data, which depicts this issue for repetition. In the German source sentence, the word *üblicherweise*, meaning ‘customarily’ is annotated as a disfluency, as it was the speaker’s intention to change the utterance into the next word *traditionell*, which means ‘traditionally’.

Table 3: *Difficulty in detecting repetitions*

Source	Die Kommunikation zwischen Mensch und Maschine, die wir so **üblicherweise** traditionell immer sehen, ist die...
Engl. gloss	The communication between man and machine, which we **customarily** traditionally always see, is the...

Discourse markers can be hard to capture, as they occasionally convey meanings in a sentence. In the same way as it is with English discourse markers such as “I mean”, “actually”, and “like”, for example, German discourse markers, as shown in Table 4, can sometimes be used as a discourse marker and sometimes as normal tokens. In this table it is shown that a German word *nun* means ‘now’ as shown in the upper part, but occasionally is used as a discourse marker like in the lower part and does not need to be translated. In the lower row, the word *nun* appears with another discourse marker *ja*, which can also mean ‘yes’ in English, depending on the context.

Table 4: *Difficulty in detecting discourse markers*

Source	Sie sehen hier unseren Simultanübersetzer, der **nun** meinen Vortrag transkribiert.
Reference	Here you see our simultaneous translator, which **now** transcribes my presentation.
Source	An einer Universität haben wir **ja nun** viele Vorlesungen.
Reference	In a university, we have many lectures.

These examples suggest that disfluency detection requires an analysis of syntactics as well as semantics. Detecting restarted fragments especially requires semantic labeling, as in some cases the restarted new fragment does not contain the same content as the aborted utterances.

In this work we aim to analyze and improve machine translation performance by detecting and removing the disfluencies in a preprocessing step before translation. For this we adopt a conditional random field (CRF)-based approach, in which the characteristics of disfluencies can be modeled using various features. In order to consider the issues discussed previously, we devised features learned from word representations and phrase tables used for the MT process in addition to lexical and language model features. The MT performance of CRF-detected output is evaluated and compared to the result of an oracle experiment, where the test data without all annotated disfluencies is translated.

This paper is organized as follows. In Section 2, a brief overview of past research on speech disfluency detection is given. The annotated data used in this work is described in Section 3, followed by Section 4 which contains the CRF modeling technique with extended features from word representation and phrase table information. Section 5 describes our experiment setups and their results along with an analysis. Finally, Section 6 concludes our discussions.

2. Related Work

In previous work, the disfluency detection problem has been addressed using a noisy channel approach [4]. In this work it is assumed that fluent text, free of any disfluencies passed a noisy channel which adds disfluencies to the clean string. The authors use language model scores and five different models to retrieve the string, where the two factors are controlled by weight. An in-depth analysis on disfluency removal using this system and its effect are provided in [5]. They find that for the given news test set, an 8% improvement in BLEU [6] is achieved when the disfluencies are removed.

In another noisy channel approach [7], the disfluency detection problem is reformulated as a phrase-level statistical machine translation problem. Trained on 142K words of data, the translation system translates noisy tokens with disfluencies into clean tokens. The clean data contains new tags of classes such as repair, repeat, and filled pauses. Using this translation model based technique, they achieve their highest F-score of 97.6 for filled pauses and lowest F-score of 40.1 for repairs.

The noisy channel approach is combined with a tree-adjoining grammar to model speech repairs in [1]. A syntactic parser is used for building a language model to improve the accuracy of repair detection. Same or similar words in roughly the same order, defined *rough copy*, are modeled using crossed word dependencies. Trained on the annotated Switchboard corpus, they achieve an F-score up to 79.7.

The automatic annotation generated in [1] is one of the features used for modeling disfluencies in [2], where they train a CRF model to detect speech disfluencies. In addition to the automatic identification by [1], they use lexical, language model, and parser information as features. The CRF model is trained, optimized and tested on around 150K words of annotated data, where disfluencies are to be classified into three different classes. Following this work, the authors offer an insightful analysis on syntactics and semantics of manually reconstructed spontaneous speech [8].

Though most of the progress has been focused on enhancing the performance of speech recognition via disfluency detection, authors of the work [3] employ disfluency detection to achieve improved machine translation. They train three different systems. The first system combines hidden-event language models and knowledge-based rules. The second system is a CRF model, which combines lexical features and shallow syntactic features. The final system is a rule-based filler-detecting system. Five classes are used in this task. The test sets for testing MT performance are generated by manually pulling out sentences with disfluencies from all sentences available. Thus, only the sentences containing disfluencies are selected and evaluated. There are two test sets built in this way, which are 339 sentences and 242 sentences out of 1,134 sentences and 937 sentences respectively. Absolute improvements of 0.8 and 0.7 BLEU points are gained on the two selected test sets.

There are several notable differences between our disfluency detection system and previous work. Unlike [2], we deploy extended features from neural networks and a phrase table in order to capture more semantic aspects. Furthermore, in our work the CRF detection result is further processed and evaluated in an MT system. In the work in [3], three systems are combined to detect disfluencies and evaluated in an MT system. Contrary to their systems, we did not deploy any rule-based detection. Moreover, in our work the CRF-based disfluency detection is extended further using semantic features. Finally, in contrast to using only the affected 28% portion of their test data to evaluate the MT performance, we use all our available data for evaluation, including unaffected, originally clean sentences. This aims at evaluating the performance in a more fair condition.

3. Data

For training and testing our CRF model for disfluency detection, we use in-house German lecture data from different speakers, which is transcribed, annotated, and translated into English.

Disfluencies are annotated manually on a word or phrase level. There are subcategories of annotation such as filler words, repetitions, deletions, partial words, and so on. These subcategories are very fine-grained, so we later re-classify them for the CRF tagging task according to our aims. Inspired by the classes defined in previous works [1, 2], we classified these annotations into three categories; `filler`, `(rough)copy`, and `non-copy`.

The class `filler` includes simple disfluencies such as *uhm*, *uh*, *like*, *you know* in English. If source words are discourse words or do not necessarily convey meaning and are not required for correct grammar, they are also classified as filler words. Words or phrases are grouped into `(rough)copy` when the same or similar tokens reoccur, as shown in Table 1 and 3 with bold letters. Words are tagged as `non-copy` when the speaker changes their mind about how or what to say, as shown in Table 2 with bold letters. Contrary to previous work [2], extreme cases of `non-copy`, in which the restarted fragments are considered to have new contexts after aborted utterances, are not excluded from the modeling target but are also taken into account.

Compared to other works on English, we have a considerably lower amount of annotated data in German. We gathered 61K manually-annotated words from lecture data, with roughly 9% marked as disfluencies. Detailed statistics of the annotated data is given in Table 5.

Table 5: *Data statistics on classes of the annotation*

	Tokens	Percentage in the corpus
`Filler`	3,304	5.35%
`(rough)Copy`	1,518	2.46%
`Non-copy`	620	1.00%
Non-disfluency	56,264	91.18%

In order to make use of all annotated data and to enable cross validation, we divided the 61K words of annotated data as well as its translation in English into three parts, such that each part has around 20K words in the German source. For testing one corpus part out of three, the other two parts, which are around 40K words, are used as training data for the CRF model.

4. Disfluency Detection using CRF

Introduced by [9], CRF is a framework dedicated to labeling sequence data. A CRF models a hidden label sequence given the observed sequence. CRFs have been applied extensively in diverse tasks of NLP, such as sentence segmentation [10], POS tagging [9] and shallow parsing [11] due to its advantages of representing long-range dependencies in the observations.

In this work we use the linear chain CRF modeling technique to detect speech disfluencies. By using bigram features we can model first-order dependencies between words with a disfluency. We used the GRMM package [12] implementation of the CRF model. The CRF model was trained using L-BFGS, with the default parameters of the toolkit.

4.1. Features

In this work we utilize lexical, language model, word representation, and phrase table information features. Word representation and phrase table information features are devised in order to capture more syntactic and semantic characteristics of speech disfluencies. They are described in detail later on.

Our lexical and language model features are based on the ones described in [2]. We extend the language model features on words and POS tags up to 4-grams. Parser information and JC-04 Edit results as shown in [1] are not available in

Table 6: *Sample features on the lexical level*

Source	Da	gibt	es	da	gab	es	in	uh	gab	es	nur	eins	.
Engl. gloss.		There is			there was		in	uh	there was		only	one	.
Word	Da	gibt	es	da	gab	es	in	uh	gab	es	nur	eins	.
POS	ADV	VVFIN	PPER	ADV	VVFIN	PPER	APPR	ITJ	VVFIN	PPER	ADV	PIS	$.
Word-Dist	3	365	3	47	4	4	259	9	218	821	115	933	27
POS-Dist	3	3	3	7	4	4	12	9	6	80	3	21	27
Word-Patt	0	0	0	0	1	1	0	0	0	0	0	0	0
POS-Patt	1	1	1	0	1	1	0	0	0	0	0	0	0
Annotation	–	RC	RC	RC	RC	RC	RC	FL	–	–	–	–	–

German, and therefore not used in this work. Furthermore, we add two new pattern features on the lexical level.

In Table 6, several selected features are shown for the rough repetition sentence from Table 1. The 'Word/POS-Dist' feature means the distance of a token to its next appearance. Therefore, a low 'Word/POS-Dist' number indicates that this token occurs again shortly thereafter. If two or more neighboring tokens have the same 'Word/POS-Dist', the 'Word/POS-Patt' feature of the corresponding tokens is set to 1. For example, the first three tokens have the same 'POS-Dist' number, therefore their 'POS-Patt' has a value of 1. This feature enables us to efficiently detect such blocks of repetition, where the same or roughly the same words are repeated. We use a 1 of k encoding for features. Since binary features are supported better by the toolkit, we quantize the numeric features. The POS tags are automatically generated using [13].

With the mentioned features, we can find syntactic clues for disfluency detection. For example, POS tokens and their patterns can help to figure out repetitive `(rough) copy` occurrences. However, as discussed earlier, in the annotated data we observe that in many cases it is required to include a semantic level of information as well. In addition to the mentioned features, we devised a new strategy of including word embedding features derived from a recurrent neural network (RNN) and phrase table information.

4.2. Word Representation using RNN

Word representations have gained a great deal of attention for various NLP tasks. Especially word representation using RNNs is proven to be able to capture meaningful syntactic and semantic regularities efficiently [14]. RNNs are similar to multilayer perceptrons, but an RNN has a backwards directed loop, where the output of hidden layers becomes additional input. This allows the network to effectively capture longer history compared to other feed-forward-based n-gram models.

Word embedding is a distributed word representation, where words are represented as multi-dimensional vectors. The word vectors syntactically and semantically relating to each other will be close to each other in that representation space. Thus, words within certain semantic and syntactic relations have similar vector values. Conventionally, word embeddings of a textual corpus are obtained using certain types of neural networks.

In the hope that word representation can offer insights on semantics and syntactis, in this paper we use word embedding features learned from an RNN for the CRF model. We use RNNLM [15] with 100 dimensions for word representations. In order to ensure an appropriate coverage of the representation, we use the preprocessed training data of the MT system, which contains various domains such as news and lectures. This data consists of 462 million tokens with 150K unique tokens.

4.2.1. Word Projection and Cosine Distance

Figure 1 depicts the 2-dimensional word projection from the real-valued 100-dimensional vectors representations using the RNN, we can observe word clusters being formed. This visualization is obtained using t-Distributed Stochastic Neighbor Embedding [16]. Due to memory consumption, only the most frequent 10K words are projected.

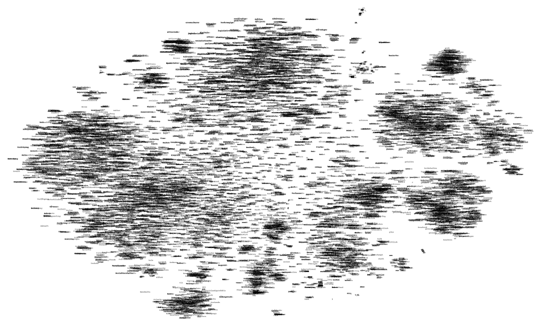

Figure 1: *Word projection of training data, with word representation obtained with an RNN*

Analyzing the details of this projection, we observe that words with the same syntactic role are projected closely to each other. For example, possessive cases corresponding to 'my', 'his', and 'our' in English are projected closely to each other. This is consistent for other grammatical components of a sentence, such as personal pronouns or relative pronouns. We observe clusters for dates, months and times.

The projection seems to convey semantic relations to some extent. When it comes to adjectives, they are projected according to their stem and occasionally meanings. Verbs are clustered with other verbs with the same tense or stem.

In order to compare the closeness of words numerically, we calculate their cosine similarity.

Table 7: *Cosine similarity of words in word representations*

Word in German	Meaning in English	Cosine Distance
schnell	fast, quick	1
rasch	quick, rapid	0.8394
bald	soon, shortly	0.6245
effektiv	effective	0.6092
zügig	efficient, speedy	0.6088
wahrscheinlich	probable	1
vermutlich	probably	0.9066
möglicherweise	maybe, possibly	0.8938
sicherlich	certainly	0.8937
vielleicht	maybe, possibly	0.8827

Table 7 depicts a couple of examples. For each bold-lettered word, the four words with the highest cosine similarity are presented. Evidently, these four words are sharing a high semantic closeness with each given word, which will provide a quality feature for the task of disfluency detection. From this analysis, we conclude that RNNs can offer syntactic and semantic clues for disfluency detection.

4.2.2. Word Clustering

In order to use the word representation vectors as features in the CRF model more efficiently, we cluster the word representations with the k-means algorithm. From preliminary experiments, the number of clusters k is chosen to be 100.

Therefore every word of the RNN training data falls into the 100 clusters. For every word in the test data, it is checked whether this word has been observed in the word representations. If it has been observed, the word is assigned with the corresponding cluster code as a binary feature. If it has not been observed, the cluster code 0 is assigned. Also, the distance to the next identical cluster code and the repetitive pattern of it are also used as CRF model features, as shown in Table 6 for word and POS tokens.

4.3. Phrase Table Information

One of the common effects of disfluencies on the MT process is that often the translation contains repetitive words or phrases. When identical tokens in the source sentence are the reason for this, the original source sentence can be corrected using lexical features. However, often we observe other cases where two words, which are different on the lexical level, generate two identical translated words. Table 8 depicts one example for this from our data.

In this example, the German word *jetzt* (Engl. gloss. ‘now’) is annotated as a disfluency, followed by a word *inzwischen* (Engl. gloss. ‘meantime’, ‘now’). Translating this source sentence as it is generates the translation containing two identical tokens in a row in English. We expect to solve this problem by examining the meaning of the source words in a phrase table. Thus, the target words for given source words in a phrase table are examined.

An advantage from using phrase table information is that we can detect semantic closeness of words or phrases in a source sentence independent from their syntactic roles. As shown in Table 7, word representation tends to group those words together which are syntactically and semantically closely related. However, using the phrase table information, words which are only semantically related, but not necessarily syntactically related, can also be grouped together. Considering that many of the repetitions also have different POS tags in a sentence, this phrase table feature is expected to capture such disfluencies.

In order to derive this feature, we examine the bilingual language model [17] tokens in the phrase table. The bilingual language model tokens consist of target words and their aligned source words. Using this information, we count how often a given source word is aligned to a certain target word and list the three most frequently used target words. We compare the aligned target words of the current and the following word. If the same target word(s) appears in both lists, the current word is given a phrase table feature.

An equivalent feature is introduced for the phrase level. As an example, we can consider consecutive source words f_1, f_2, and f_3 in one phrase. This phrase is aligned to a target token e_1. If the next source token f_4 is also aligned to the target token e_1, the first three tokens, namely f_1, f_2, and f_3, are given the phrase level phrase table feature. The coverage of the phrase level feature can be expanded upto three consecutive words as one phrase on the source side. Thus, the source tokens f_1, f_2, and f_3 are examined as one phrase, and this can be also narrowed down to f_1 and f_2 only. The target token(s) aligned to the source phrase, consists of upto f_1, f_2, and f_3, is compared to the target token(s) aligned to the potential repetitive phrase, which can consist of also upto next three tokens f_4, f_5, and f_6. The German source words with split compounds are also considered in this way.

In our phrase table the word *inzwischen* in Table 8 is aligned to ‘now’ most frequently, followed by ‘meantime’ and ‘meanwhile’. The most frequently appeared translation for the next appearing word *jetzt* is ‘now’, followed by ‘currently’, and ‘just’. Thus, by using the phrase table features, it will be indicated that the first word *jetzt* is aligned to a same target word with its next appearing word.

5. Experiments

5.1. System Description

In this section we introduce the SMT system used in our experiments. The translation system is trained on 1.76 million sentences of German-English parallel data including the European Parliament data and the News Commentary corpus.

Table 8: *Necessity of using phrase table information for disfluency detection*

Source Engl. gloss	Diese Vorlesungen sind natürlich **jetzt inzwischen** alle abgespeichert, die liegen auf unserem Server. These lectures are of course **now meantime** all stored, they lie on our server.
MT output	This lecture series are, of course, **now now** all stored, which lie on our server.
Reference	These lectures have of course all been saved in the meantime, they are on our server.

We also use the parallel TED data[1] as in-domain data to adapt our models to the lecture domain. Preprocessing which consists of text normalization, tokenization, and smartcasing is applied before the training. For the German side, compound splitting and conversion of words written according to the old spelling conventions into the new form of spelling are applied additionally.

As development data, manual transcripts of lecture data collected internally at our university are used. The talks are 14K parallel sentences from university classes and events.

In order to build the phrase table, we use the Moses package [18]. Using the SRILM Toolkit [19], a 4-gram language model is trained on 462 million words from the English side of the data. A bilingual language model [17] is used to extend source word context. In order to address the different word orders between German and English, the POS-based reordering model as described in [20] is applied. This is further extended as described in [21] to cover long-range reorderings. We use Minimum Error Rate Training (MERT) [22] for the optimization in the in-house phrase-based decoder [23].

5.2. Results

To investigate the impact of disfluencies in speech translation quality, we conduct four experiments.

In the first experiment, the whole data, including annotated disfluencies, is passed through our statistical machine translation (SMT) system.

For the second experiment, we remove the obvious filler words *uh* and *uhm* manually in order to study the impact of the filler words which can be captured systematically. Although there are a great number of other filler words, many of these filler words are not removed in this experiment, since they are not always disfluencies.

In the third experiment, we use the output from the CRF model trained with features from word representations and phrase table information, which will be noted as CRF-Extended. We also translate the output from the CRF model trained without any word representation and phrase table features. This will be denoted as CRF-Baseline. If the CRF models detect a token as either of the three classes, `filler`, `(rough)copy`, or `non-copy`, the word token is assumed to be a disfluency and is removed. The three classes are trained in the same model together. As mentioned previously, training and testing the CRF model is done with three-fold cross-validation. Thus, both of the CRF models are trained on around 40K annotated words, and tested on around 20K annotated words. The performance is evaluated on the joined three sub-test sets.

In the last experiment, all disfluency-annotated words are removed manually. As all annotation marks are generated manually, this experiment shows as an orcale experiment the maximum possible improvement we could achieve.

All four experiments are conducted on manually transcribed texts, in order to disambiguate the effects from errors of an ASR system. The experiments considers all available data, which is 61K words, or 3K sentences.

Table 9 depicts the results of our experiments. The scores are reported as case-sensitive BLEU scores, including punctuation marks.

Table 9: *Influence of disfluency in speech translation*

System	BLEU
Baseline	19.98
+ no *uh*	21.28
CRF-Extended	21.94
Oracle	23.14

The result of the first experiment is presented as the Baseline system, where all disfluencies are kept in the source text. When we remove all *uh*s and *uhm*s in the source text manually, we gain 1.3 BLEU points.

Apart from this, we use the output of the CRF-Extended as an input to our machine translation system. Words tagged as disfluencies are all removed. The translation score using the CRF-Extended is almost 2 BLEU points better than translating the text with all disfluencies. Compared to the second experiment where we remove *uh* and *uhm*, the performance is improved by around 0.7 BLEU points. As the BLEU score does not show a significant difference between the CRF-Extended and CRF-Baseline, here only the CRF-Extended score is shown. An in-depth analysis of the impact of the two systems will be given in the following chapter.

5.3. Analysis

The detection results for all models are given in Table 10. In total, there are 5,432 speech disfluencies annotated by human annotators, and among them, 3,012 speech disfluencies are detected by the CRF-Extended.

Compared to the case where the obvious filler words are removed, 1,025 more speech disfluencies are detected and removed. Compared to the CRF-Baseline, where the features obtained from the word representations and phrase table information are not used, 103 more disfluencies are detected

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

Table 10: *Results of disfluency detection in tokens*

System	Correct	Wrong
Baseline	0	0
+ no *uh*	1,987	0
CRF-Baseline	2,909	489
CRF-Extended	3,012	552
Oracle	5,432	0

using the CRF-Extended, while also a higher number of tokens are falsely detected.

In order to analyze the difference between the translations produced by CRF-Baseline and CRF-Extended, we score the test set sentence by sentence and rank them according to the difference in BLEU scores. Differences appear in 223 sentences.

One notable difference is that the CRF-Extended system detects a higher number of repetitions. Table 11 shows a sentence from the test set, where a longer phrase of repetition is captured using CRF-Extended. Words which represent a disfluency are marked in bold letters. Both systems can catch the obvious filler word *uh* and the simple repetition *als als*. In addition to this detection, the CRF-Extended system captures the whole disfluency region, in spite of the considerably complicated sentence structure and repetitive patterns. In this sentence the repeated words appear with varying frequencies and with a different distance to the next identical token. In order to detect such disfluencies, the correct phrase boundary needs to be recognized. As a result of this detection, the MT output using the CRF-Extended system is much more fluent than the one using the CRF-Baseline system.

Table 12 shows a sentence from the test set, where the CRF-Extended system does not perform better than the CRF-Baseline system for the given reference. The only disfluency shown in the original sentence *der*, marked with bold letters, is removed using both techniques. The CRF-Extended system additionally detects *einen Umschwung* as a disfluency. However, this deletion harms neither the structure nor meaning of the sentence, as *einen Umschwung* means ‘a turnaround’, or ‘a change’, which conveys practically the same meaning as the next following tokens.

It is an interesting point that using the semantic features we could detect that *einen Umschwung* is semantically closely related with *eine veränderte*, despite their distance in tokens and different syntactic roles in the sentence. This is an example that even though the CRF-Extended output does not match the human-generated annotation in this case, the CRF-Extended still provides a good criteria to detect semantically related words.

The CRF-Extended system also performs better with regard to distinguishing between discourse markers and the normal usages of the words. 59% of difference in correctly classified disfluencies between the CRF-Baseline and CRF-Extended stems from filler words. The rest is achieved from detecting a higher number of correct repetitions.

6. Conclusions

In this paper, we presented a CRF-based disfluency detection technique with extended features from word representations and a phrase table. These features are designed to capture deeper semantic aspects of the tokens. Using the predicted results from the CRF model, we gain around 2 BLEU points on manual transcripts of lectures. From the detailed analysis, we show that usage of the extended features provides a good means to detect semantically related disfluencies. The oracle experiment suggests that the machine translation of spontaneous speech can be improved significantly by detecting more disfluencies correctly.

Table 11: *Syntactically complicated, long phrase with a disfluency captured using CRF-Extended*

Source	Man kann das natürlich sowohl **als Links- als auch als** als Links- als auch als Rechtshänder **uh** verwenden.
Engl. gloss	You can this of course both **as left- as also as** as left- as also as right-handed **uh** use.
CRF-Baseline	Man kann das natürlich sowohl **als Links- als auch** als Links- als auch als Rechtshänder verwenden.
MT output	You can use this, of course, both as a left- as well as on the left- as well as a right-handed.
CRF-Extended	Man kann das natürlich sowohl als Links- als auch als Rechtshänder verwenden.
MT output	You can use this, of course, both as a left- as well as a right-handed.
Reference	You can of course use this as left- as well as also as a right-handed person.

Table 12: *Semantically related words detected using CRF-Extended*

Source	Die Ausrufung des totalen Kriegs markierte eigentlich *einen Umschwung*, **der** *eine veränderte* Form der Politik.
Engl. gloss	The proclamation of total war marked actually *a turnaround*, **of** *a change* form of politics.
CRF-Baseline	Die Ausrufung des totalen Kriegs markierte eigentlich *einen Umschwung, eine veränderte* Form der Politik.
MT output	The proclamation of the total war was collared actually *a turnaround, a changed* form of politics.
CRF-Extended	Die Ausrufung des totalen Kriegs markierte eigentlich *eine veränderte* Form der Politik.
MT output	The proclamation of the total war was collared actually *a changed* form of politics.
Reference	The call for total war in fact marked *a turnaround*, and *a changed* form of politics.

In future work, we would like to pursue the development of disfluency detection systems which take prosodic features into account in order to apply them to automatic speech recognition output. Furthermore, integrating the disfluency detection system tightly into machine translation systems could improve the performance even more.

7. 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.

8. References

[1] M. Johnson and E. Charniak, "A TAG-based Noisy Channel Model of Speech Repairs," in *ACL*, 2004.

[2] E. Fitzgerald, K. Hall, and F. Jelinek, "Reconstructing False Start Errors in Spontaneous Speech Text," in *EACL*, Athens, Greece, 2009.

[3] W. Wang, G. Tur, J. Zheng, and N. F. Ayan, "Automatic Disfluency Removal for Improving Spoken Language Translation," in *ICASSP*, 2010.

[4] M. Honal and T. Schultz, "Correction of Disfluencies in Spontaneous Speech using a Noisy-Channel Approach," in *Eurospeech*, Geneva, 2003.

[5] S. Rao, I. Lane, and T. Schultz, "Improving Spoken Language Translation by Automatic Disfluecy Removal: Evidence from Conversational Speech Transcripts," in *Machine Translation Summit XI*, 2007.

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

[7] S. Maskey, B. Zhou, and Y. Gao, "A Phrase-Level Machine Translation Approach for Disfluency Detection using Weighted Finite State Tranducers," in *Interspeech*, 2006.

[8] E. Fitzgerald, F. Jelinek, and R. Frank, "What Lies Beneath: Semantic and Syntactic Analysis of Manually Reconstructed Spontaneous Speech," in *ACL*, 2009.

[9] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilitic Models for Segmenting and Labeling Sequence Data," in *ICML*, Massachusetts, USA, 2001.

[10] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper, "Using Conditional Random Fields for Sentence Boundary Detection in Speech," in *ACL*, Ann Arbor, MI, 2005.

[11] F. Sha and F. Pereira, "Shallow Parsing with Conditional Random Fields," in *HLT/NAACL*, 2003.

[12] C. Sutton, "GRMM: A Graphical Models Toolkit," 2006. [Online]. Available: http://mallet.cs.umass.edu

[13] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," in *International Conference on New Methods in Language Processing*, Manchester, UK, 1994.

[14] T. Mikolov, W.-T. Yih, and G. Zweig, "Linguistic Regularities in Continuous Space Word Representations," in *NAACL-HLT*, 2013.

[15] T. Mikolov, M. Karafiat, J. Cernocky, and S. Khudanpur, "Recurrent Neural Network based Language Model," in *Interspeech*, 2010.

[16] L. van der Maaten and G. Hinten, "Visualizing High-Dimensional Data using t-SNE," in *Journal of Machine Learning Research 9*, 2008.

[17] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *WMT*, Edinburgh, UK, 2011.

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

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

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

[21] J. Niehues and M. Kolss, "A POS-Based Model for Long-Range Reorderings in SMT," in *WMT*, Athens, Greece, 2009.

[22] A. Venugopal, A. Zollman, and A. Waibel, "Training and Evaluation Error Minimization Rules for Statistical Machine Translation," in *WPT-05*, Ann Arbor, MI, 2005.

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