# Machine Translation of Multi-party Meetings: Segmentation and Disfluency Removal Strategies

*Eunah Cho, Jan Niehues, Alex Waibel*

International Center for Advanced Communication Technologies - InterACT
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology, Germany
{eunah.cho|jan.niehues|alex.waibel}@kit.edu

## Abstract

Translating meetings presents a challenge since multi-speaker speech shows a variety of disfluencies. In this paper we investigate the importance of transforming speech into well-written input prior to translating multi-party meetings. We first analyze the characteristics of this data and establish oracle scores. Sentence segmentation and punctuation are performed using a language model, turn information, or a monolingual translation system. Disfluencies are removed by a CRF model trained on in-domain and out-of-domain data. For comparison, we build a combined CRF model for punctuation insertion and disfluency removal. By applying these models, multi-party meetings are transformed into fluent input for machine translation. We evaluate the models with regard to translation performance and are able to achieve an improvement of 2.1 to 4.9 BLEU points depending on the availability of turn information.

## 1. Introduction

Machine translation (MT) of spontaneous speech has recently drawn a great deal of interest. For instance, the importance of sentence segmentation, punctuation insertion and disfluency removal for translating monologue data, such as lectures, has been researched extensively. In addition, there have been research efforts investigating MT of two-party speech, such as telephone calls. However, automatic translation of multispeaker speech remains yet underexplored.

In our globalized world, teams of different parts of the world are increasingly working together. Internal team meetings held in one language need to be translated into another language in order to make the discussions available to all involved parties. Human translation is time-consuming and costly, so MT can be a supportive tool to overcome this challenge. State-of-the-art MT systems, however, are not designed for such conversational speech, especially when multiple speakers are involved. Since conventional MT systems are built using written texts, their performance drops when they are applied to such a different domain. We therefore propose an approach to transform multi-party meetings so they are closer in style to the training data of the MT system.

Natural language processing (NLP) of multispeaker speech presents unique research challenges. Speech disfluencies should be removed, while the punctuation marks and sentence boundaries need to be inserted.

Spontaneous speech contains a large number of disfluencies, such as hesitations as well as repetitions, either exactly or vaguely the same, and speech fragments. In addition to these disfluencies, speakers may interrupt each other. Due to such interruptions, aborted speech fragments occur very often in multispeaker speech. Therefore, it is one of our main goals to model such disfluencies which can better fit the domain. One of the difficulties of disfluency detection, however, is data sparsity, since speech disfluencies are usually modelled using disfluency-annotated data. Thus it is necessary to explore how to improve the performance given the limited quantity of data as well as evaluate how important the domain is for the given task.

Since the output of an automatic speech recognition (ASR) system is a stream of word tokens, without punctuation or segmentation information, it is necessary to properly segment and punctuate the ASR output for translation.

In this work, we present various approaches to reformulate multispeaker speech prior to MT, through segmentation, punctuation insertion and disfluency removal. In order to explore the importance of domain in this task, we train disfluency removal models on in-domain and out-of-domain data and compare the results. Every experiment is conducted in two conditions whether turn information is available or not. Once the disfluencies of the meeting data are removed and punctuation marks are inserted, the data goes through our English to French MT system. For comparison, oracle experiments results and a baseline system are shown.

## 2. Related Work

There has been extensive effort on disfluency removal on telephone speech, or Switchboard data [1]. In [2], Johnson et al. combined the noisy channel approach with a tree adjoining grammar for modeling speech disfluencies. In the noisy channel model, it is assumed that fluent text goes through a channel which adds disfluencies. Disfluency removal on the same data is modeled using a conditional random fields (CRF) model in [3], using language and lexical model, and parser information as features.

In [4], segmentation and disfluency removal issue in meeting data is handled in the scope of ASR. Baron et al. explored sentence boundary and disfluency detection in meetings using prosodic and lexical cues. For multi-party meeting data they used data collected as part of the ICSI Meeting Recording Project [5]. Sentence boundary detection is treated as a sequence classification problem, where each word boundary is labeled as either a sentence boundary, a disfluency interruption point, or a clean word transition. Therefore, disfluency is viewed from a different perspective, as an interruption point, where once it occurs a new segment boundary is added. Baron et al. find that combining prosodic and word-based classifier information yields the best results for the given task.

While the previous works have focused on enhancing the performance of speech recognition, Peitz et al. [6] compared the translation performance using three different methods to punctuate TED talks. They compare methods depending on when and how the punctuation marks are inserted: prediction in the source language, implicit prediction, and prediction in the target language. They assumed that the proper segments are already available, but punctuation marks are missing therefore should be inserted. Among the three systems, translating from unpunctuated to punctuated text achieves the largest improvements. Later this work is extended in [7] for MT of university lectures, where a monolingual translation system is used for punctuation combined with sentence boundary detection. They prepare the training data by cutting it randomly, so that detection of sentence-like units is possible.

Cho et al. [8] use a monolingual translation system together with CRF-based disfluency removal. Using a CRF model, the disfluency probability of each token is obtained and encoded into word lattices so that potentially disfluent paths can be skipped during decoding.

MT of multi-party meetings was studied in [9], with a particular view towards analyzing the importance of modeling contextual factors. They showed that word sense disambiguation using topic and domain knowledge yields a large improvement on MT performance.

Recently Hassan et al. [10] investigated the impact of segmentation and disfluency removal on translation of telephone speech. They use a CRF model to detect sentence units and a knowledge-based parser for complex disfluency removal.

There are several notable differences between our and previous work. Contrary to many works in disfluency removal and punctuation insertion, our work is expanded to the MT. Our systems are designed for multi-party meetings unlike [7, 10]. We focused on segmentation and disfluency issues in multi-party meetings, while [9] studied the meetings with focus on word sense disambiguation. Additionally, the importance of training the models on out-of-domain data is investigated in our work.

# 3. Task

Before describing the techniques to translate multispeaker speech, the corpus and its characteristics are described. The section is concluded with an overview of the system architecture to detect speech disfluencies and punctuation marks used in this evaluation.

## 3.1. Multi-party meeting data

Our corpus consists of project meetings between project participants with various topics. We use eight sessions, where each meeting involves 5 to 12 different speakers. All meetings are held in English. As in real meeting scenarios, the meeting participants consist of native and non-native English speakers. The eight meeting sessions are transcribed and then disfluencies are manually annotated. We use five of the meetings for training the disfluency removal model and the remaining three for testing. The test data is translated into French in order to evaluate the translation performance.

### *3.1.1. Speech disfluencies*

Disfluencies in the meeting data are annotated manually by human annotators. Previous work on disfluencies [2, 11, 12] categorized the disfluencies into three groups: `filler`, `(rough)copy`, and `non-copy`. `filler` contains filler words as well as discourse markers. Therefore, this class includes words such as *uh*, *you know*, and *well* in some cases. As the class name suggests, `(rough)copy` includes an exact or rough repetition of words or phrases. In spontaneous speech, speakers may repeat what has been already spoken, as stutter or correction. For example, a sentence *There is, there was an advantage* has `(rough)copy` in the phrase *there is*. `non-copy` includes the cases where the speaker aborts previously spoken segments and starts a new segment. It can be rather moderate, so that the newly started fragment still has the same theme as the previously spoken segment. In a more extreme case, however, the speaker may introduce an entirely different topic in the new fragment. For example, in the following sentence from our meeting data: *I don't think it's the, the crucial thing is that we can compile with...*, the part before the comma is annotated as `non-copy`.

After looking into the data, we decided that the disfluency annotations for the multispeaker speech task has an additional category, `interruption`. While the other three categories of disfluency can be used for other tasks such as monologue, the last class `interruption` is devised for this new task. In multispeaker speech, generally there are more than two speakers involved. Therefore, there are many parts of utterances which are interrupted by other speakers. Those segments which are interrupted and therefore could not be finished were classified as `interruption`.

The number of tokens of each class of disfluencies and its

Table 1: *Meeting data statistics*

| Class | Training | | Testing | |
|---|---|---|---|---|
| `filler` | 2,666 | 6.9% | 999 | 6.7% |
| `(rough)copy` | 2,232 | 5.8% | 1,017 | 6.8% |
| `non-copy` | 802 | 2.1% | 331 | 2.2% |
| `interruption` | 1,350 | 3.5% | 864 | 5.8% |
| `clean` | 31,507 | 81.7% | 11,660 | 78.4% |
| SUM | 38,557 | 100% | 14,871 | 100% |

proportion are shown in Table 1. The numbers do not include punctuation marks, but only words. Both the training and test data have around a disfluency rate of around 20%, which is much higher than the rate reported in [13], where lecture data has a disfluency rate of roughly 10%. Around 7% of the word tokens in the meeting data are simple disfluencies, or filler words, while the other 11 to 15% are more difficult to detect.

### *3.1.2. Segments*

The training data shown in Table 1 consists of $4.6k$ sentences, while the test data has around $2.1k$ sentences. We found that multi-party meeting data has the characteristic that each segment is rather short. In average, for all meeting data we have, there are around 8 words per segment. This is quite short compared to, for example, lecture data, which has around 15 words per segment [13]. We also compare the number of segments to the training data of our MT system, which is mainly parliamentary proceedings and news text. This data has around 24 words per segment.

Figure 1: *Statistics on number of words in segment*

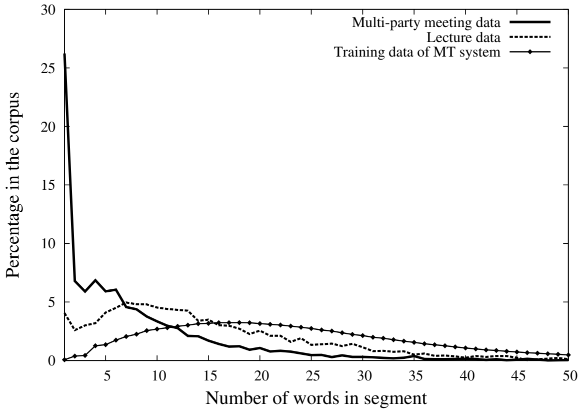

Figure 1 depicts the distribution of segment length for every corpus. In the meeting data short segments are the majority, especially one word segments. There are many segments which only consist of one word, such as *yes* or *okay*. Although some of them are discourse markers and therefore annotated as `filler` disfluency, some of them are also left intact when those tokens are actually used to convey meaning. This is therefore another challenge of detecting disfluencies in meeting data. Another cause of the short segments is that there are also many short segments which are interrupted by another speaker and therefore aborted. The lecture data, which also consists of spoken language, also has higher frequency of shorter segments, compared to the conventional MT training data which has more segments whose length is longer than 15 words.

### *3.1.3. Example*

Table 2 shows an example excerpt from the meeting data. Following the annotation conventions described in [13], `filler` tokens are marked with <>, and `(rough)copy` tokens are marked with +//+. `non-copy` tokens are tagged with −//−, and finally `interruption` are marked with #//#. In this excerpt, the first speaker tried to start a new fragment (starting *what*), then a filler word is occurred (*uh*), and then the fragment is aborted, then yet another fragment is started (*how far*). But this last fragment is interrupted by the next speaker. We can also observe repetition.

Table 2: *Meeting data example with disfluency annotation*

A: I haven't heard anything, so I don't know -/what/-
<uh> #/how far/#
B: I will check for that.
C: Why is the API so hard?
We're waiting for a month now for this.
D: I don't know +/the last/+ the last meeting outcome <uh>
he said he could give us API at the end of the month.
C: Okay.

## 3.2. System architecture

In this work, we chose a work scheme where the output stream from an ASR system passes first through an automatic disfluency detection system. Based on this cleaned-up stream, punctuation and segmentation insertion is performed. Once the disfluencies in the ASR output are removed and punctuation marks are inserted, the cleaned, punctuated data goes through the MT system like normal input data.

Disfluency detection is performed prior to the punctuation and segmentation insertion, because this way punctuation insertion can be trained on much larger data. While disfluency removal can be only trained on disfluency-annotated data, punctuation insertion can be trained on more data. For the disfluency removal model, we use data of two different domains: multi-party meeting and lecture. As the first domain, we train the model using five meeting sessions, which sum up to $38.6k$ annotated words. In order to model the case where we have no in-domain data, we train the second model using lecture data. We use web-based seminar lecture data given in English as well as the annotated English reference translation of the German lecture data shown in [13]. The lecture data sums up to $104k$ annotated words, and shows a moderate level of disfluency.

The punctuation insertion model is not trained using the meeting data, but using the English side of the MT training corpus, which consists of well-segmented, clean text.

Once the models are built, they are applied to the remaining three meeting sessions. The test data consists of $2.1k$ segments with $14.9k$ English words and $11.4k$ French words. After cleaning up the disfluencies manually, the source side contains $11.7k$ English words.

### *3.2.1. Turn information*

For MT of multi-party meetings, turn information can play a big role, since knowing who spoke when can provide basic segmentation. However, turn information is not always available.

In order to compare and study the impact of turn information on our models, we assume two scenarios: in the first scenario turn information is available while in the second one it is not available. With the turn information, basic segment information according to speaker changes is available. Even though this may not be the exact sentence segmentation, it can offer a reasonable baseline for segmentation and punctuation insertion. It can also offer additional features for disfluency detection. As it is possible to know which segment is started by which speaker, we can obtain a cue that the previous segments' last tokens could have been interrupted by the new speaker, given the fact that meetings contain a lot of interruptions.

When the turn information is not available, there is no basic segmentation. Therefore it is required to chunk the stream of ASR output into segments. Different tactics on segmentation and punctuation insertion will be described in Section 5.

## 4. Disfluency Detection

In the disfluency detection model, we start with a sequence of words as input and need to mark parts of the sequence as disfluencies. This problem can intuitively be modeled as a sequence labeling task, where each word is either labeled by one of the disfluency classes (`filler`, `(rough) copy`, `non-copy`, and `interruption`), or by a label representing clean speech. Since sequence labeling is a common problem in NLP, it has been studied intensively. One succesful approach to model these problems is using CRF. As CRFs can represent long-range dependencies in the observations, they have shown good performance in sentence segmentation [14], parts of speech (POS) tagging [15] and shallow parsing [16]. In this work we use the CRF model implemented in the GRMM package [17] to mark the speech disfluencies. The CRF model was trained using L-BFGS, with the default parameters of the toolkit.

### 4.1. In-domain vs. out-of-domain data

In the ideal case, disfluency annotated in-domain data is available for training the CRF model. However, the annotation of speech for different domains can be very time-consuming. As disfluency annotated lecture data [13] is available, we use this data as our out-of-domain training data for the CRF model. As in-domain training data we use the in-house English meeting data. This will show whether the disfluency removal model is portable across different domains.

Compared to the meeting data, lecture data has different characteristics. Although it still provides general speech disfluencies such as repetitions or filler words, lecture data in general contains a moderate level of speech disfluencies compared to the quite noisy meeting data. Especially, unlike meeting data, lecture data does not contain interruptions by other speakers. Therefore, for testing the CRF model using lecture data, we mapped `interruption` onto the `non-copy` class.

As a test data of the CRF model, we use the test data described in Section 3. After potential disfluencies are detected and removed, punctuation and segmentation are inserted into this test set, which is then used as input for MT.

### 4.2. Features

As features for CRF, we use lexical and language model (LM) features inspired by the work in [11]. Lexical features include current and adjacent words/POS tokens, whether the current word is a partial word, and whether words or POS tokens are showing repetitive patterns. LM features include unigram and 4-gram LM scores, and their ratio. In addition to these features, following [12], features obtained from a word representation in vectors and phrase table information are used. Each word is represented as a word vector with 100 dimensions as shown in [18]. Afterwards the vectors are clustered into 100 clusters using the $k$-means algorithm. We use the cluster number of each word as one of the features, as well as the repetitive pattern of the cluster code and adjacent words' cluster codes. For the phrase table information, we use the phrase table which is used for the actual MT of the task and check the potential translations of each word.

As mentioned earlier, we assumed two scenarios about turn information availability. In the scenario where the turn information is available, we extracted the word position within the turn. We expect that disfluencies can be more prominent in the initial part of each turn, because many stutters as well as corrections occur within the first several words. In addition, as interruptions between speakers occur at end of each turn, we encoded whether the current token is one of the first or final 5 words of the turn in order to incorporate this information for the training.

The CRF model is trained with a bigram feature, so that first-order dependencies between words with a disfluency can be modeled.

# 5. Segmentation and Punctuation Insertion

After removing disfluencies, the main difference between written text and the disfluency-removed speech is the lack of punctuation marks. In recent work [6], it has been shown that a promising approach to translate unpunctuated text is to automatically insert punctuation marks and segmentation prior to translation. Therefore, we analyzed three different methods to segment and punctuate the multi-party meeting data: simple LM-based segmentation, turn segmentation, and monolingual translation system.

## 5.1. Simple LM-based segmentation

Assuming there is no information about different speakers and their turns available, ASR of such a talk would generate a stream of words. For translation, it is necessary to segment the stream of words. As a baseline system, we segmented based on a hard threshold of word-based LM scores. First we concatenated the test data into a single line without any punctuation marks, in order to mimic the ASR output. We use a 4-gram LM trained on the punctuated English side of the MT training corpus and measure the probability of a final period given the previous words. When the probability exceeded an empirically chosen threshold, we inserted a final period and started a new segment. The output of this baseline system consists of segments where each segment ends with a final period.

## 5.2. Turn segmentation

If we have access to turn information, we can exploit this information in order to obtain a better baseline segmentation. We inserted a final period and began a new segment whenever the speaker changed. Each segment of this system may contain more than one actual sentence, with no further punctuation marks within the segment.

## 5.3. Monolingual translation system

Cho et al. [7] successfully used a monolingual translation system to insert punctuation marks into non-punctuated German lecture data. Following this approach, we built a monolingual translation system from non-punctuated English to punctuated English. While the previous two methods insert only final periods, this system can insert all punctuation marks appeared in the training data. As training data we used the English side of the MT training corpus. This MT training corpus is ideally segmented and contains all punctuation marks, including a final period at the end of each sentence. In order to learn where segment breaks should be inserted, we throw away the segmentation and randomly cut the English side of the data. Aiming to generate data that is similar to the test data, we limit the length of segments to 22 words. The test data goes through the monolingual translation system with a sliding window of 10 words.

For the scenario where turn information is available, we build an additional, slightly different monolingual translation system. When we have the turn information, several segments uttered by a speaker are concatenated. Therefore, in order to make the training data similar to the test data, we concatenated one to three sentences randomly into one sentence. Punctuation marks between sentences are removed, and only a final period is added at the end of each line of the source side data. The target side contains all punctuation marks.

# 6. Experiments

In this section, we briefly describe the MT system we use in our experiments. Oracle experiments and the results are given, followed by results of segmentation and punctuation insertion. The results of disfluency removal are analyzed. Finally, the overview of our system is given in the end.

## 6.1. System description

The translation system is trained on 2.3 million sentences of English-French parallel data including the European Parliament data and the News Commentary corpus. The parallel TED data[1] is used as in-domain data for the MT models. As development data, we use manual transcripts of TED data.

Preprocessing which consists of text normalization and tokenization is applied before the training. In order to build the phrase table, we use the Moses package [19]. Using the SRILM Toolkit [20], a 4-gram language model is trained on 683 million words from the French side of the data. A bilingual language model [21] is used to extend source word context. The POS-based reordering model as described in [22] is applied to address different word orders between English and French. We use Minimum Error Rate Training (MERT) [23] for the optimization in the phrase-based decoder [24]. All scores of translation into French are reported in case-sensitive BLEU scores [25] in this paper. When the sentence boundaries differ from the reference translation, we use the Levenshtein minimum edit distance algorithm [26] to align hypothesis for evaluation.

## 6.2. Oracle experiments

Table 3 shows the translation performance for oracle punctuation marks and oracle disfluency removal on the multi-party meeting data.

Table 3: *Oracle experiments*

| System | No turns | Turns |
|---|---|---|
| Baseline | 9.53 | 12.93 |
| Oracle segmentation | 13.96 | |
| Oracle punctuation | 15.64 | |
| Oracle disfluency | 12.21 | 15.72 |
| Oracle all | 20.93 | |

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

In the first system, all disfluencies are kept and baseline segmentations are used. As the baseline segments, we use two different segmentation methods. When there is no turn information available, segmentation and final periods are inserted using the simple LM-based method as described in Section 5.1. On the other hand, when we have access to the turn information, a new segment and a final period are inserted whenever the speaker changes as described in Section 5.2. We can observe that using the turn information is very helpful in achieving better performance.

Then we insert oracle segmentation and a final period at the end of segment. When we also inserted all other punctuation marks from the reference transcript, the translation performance is improved up to 15.64 BLEU points even though it still contains all disfluencies. We can observe that nearly 1.7 BLEU points are achieved by inserting all other punctuation marks, on top of we have the ideal reference segmentation and a final period.

In the next experiment, we keep the punctuation and segmentation the same as in the baseline system, but remove all of the manually annotated disfluencies. By doing so, translation performance is improved by around 3 BLEU points compared to the baseline system. Finally, we achieved BLEU score of 20.93 when we have the oracle for both punctuation and disfluency. This is the upper bound of the performance we can get for this test set when we have both perfect segmentation/punctuation and disfluency removal.

As shown by these numbers, the performance can be improved by more than 10 BLEU points if the ideal punctuation and disfluency detection are applied. Therefore, modeling these two problems in a translation system of mutispeaker speech is essential to reach a good translation quality.

### 6.3. Segmentation and punctuation insertion

In this section, we look into the performance of the segmentation and punctuation in a realistic approach (all disfluencies kept) and perfect conditions (remove all disfluencies using the manual annotation).

Table 4: *Punctuation insertion, no turn information*

| System | Keep disf. | Oracle disf. |
|---|---|---|
| Baseline | 9.53 | 12.21 |
| Mono. trans. | 12.44 | 16.34 |
| Oracle punctuation | 15.64 | 20.93 |

Table 4 shows the results under the assumption that no turn information is available. The baseline system has punctuation and segmentation inserted using the simple LM-based method. When punctuation marks are inserted using the monolingual translation system, we achieved an improvement of 3 to 4 BLEU points for both disfluency conditions. This improvement reaches almost half of the difference between the baseline systems and oracle scores. We can also observe that when segmentation and punctuation are improved, the impact of disfluencies increases. There is bigger room of improvement which can be achieved by removing correct disfluencies, when we have better segmentation and punctuation. The same phenomena can be observed in the experiments with turn information, as shown in Table 5.

Table 5: *Punctuation insertion, with turn information*

| System | Keep disf. | Oracle disf. |
|---|---|---|
| Baseline | 12.93 | 15.72 |
| Mono. trans. | 13.25 | 17.71 |
| Oracle punctuation | 15.64 | 20.93 |

We can observe that the baseline scores in this case have already improved a lot over the experiments without turn information. Since the baseline segmentation is already better, the improvements are smaller, but there are still consistent improvements when inserting punctuation marks using the monolingual translation system.

### 6.4. Disfluency removal

This section presents translation performance when we apply the disfluency removal models trained either on in-domain or out-of-domain data. Punctuation and segmentation are inserted not only by the monolingual translation system for the realistic case, but also oracle punctuation is used for comparison.

Table 6: *Disfluency removal, no turn information*

| System | Mono. trans. | Oracle punct. |
|---|---|---|
| Keep disfluency | 12.44 | 15.64 |
| CRF in-domain | 14.41 | 17.26 |
| CRF out-of-domain | 14.24 | 16.95 |
| Oracle disfluency | 16.34 | 20.93 |

Table 6 shows the scores under the assumption that there is no turn information available. In the first experiment, we keep all disfluencies. Then we show the scores when we use the disfluency removal model trained only on the in-domain data, multi-party meeting data. These scores are compared with the scores when we use the model trained only on the out-of-domain data, which is lecture data. Finally, we show the scores removing all disfluencies annotated. An interesting point is that using lecture data for training the CRF model yields similar performance to training using the meeting data. Even though using the lecture data is slightly worse than using the meeting data, the difference is minimal.

Our preliminary experiments showed that when we use the in-domain data for training the disfluency removal model, we have around 8 points better F-scores, compared to the case when we train the model using out-of-domain data. However, such differences are not pronounced in terms of BLEU. It shows that the disfluency modeling technique shown in this work can be transfered into a new domain without causing a big loss of performance in MT.

Table 7: *Disfluency removal, with turn information*

| System | Mono. trans. | Oracle punct. |
|---|---|---|
| Keep disfluency | 13.25 | 15.64 |
| CRF in-domain | 15.01 | 17.10 |
| CRF out-of-domain | 14.90 | 17.03 |
| Oracle disfluency | 16.34 | 20.93 |

This result is also observable when the models are trained with turn information, as shown in Table 7. The disfluency removal model trained on meeting data performs only slightly better than the lecture data. In all listed conditions, it is shown that we can improve the translation quality by 1.5 to 2 BLEU points by removing disfluencies.

### 6.5. Combined modeling of punctuation insertion and disfluency removal

As an additional experiment, we model punctuation marks and disfluencies in one model. This yields the advantage that it is not necessary for ASR output to pass through two different steps. We also hope that this experiment can provide the first insight on MT performance when modeling these two in one model for the given task. In this scheme, both the punctuation marks as well as disfluencies are predicted given the potentially disfluency, and unpunctuated ASR output. For modeling we use the same features as for the disfluency removal. Thus, punctuation and disfluencies are trained using the data with speech disfluencies. For the modeling, we use the same CRF tool, but with two decision labels: one with disfluency classes and another one with punctuation marks.

Table 8: *Punctuation insertion and disfluency removal in one model*

| System | No turn | Turn |
|---|---|---|
| Baseline | 9.53 | 12.93 |
| Combined CRF in-domain | 13.92 | 14.45 |
| CRF in-domain + Mono. trans. | 14.41 | 15.01 |
| Combined CRF out-of-domain | 13.99 | 14.58 |
| CRF out-of-domain + Mono. trans. | 14.24 | 14.90 |
| Oracle all | 20.93 | |

Table 8 presents the results of this experiment. When modeling punctuation marks and disfluency removal together in one model, it still provides a big improvement over the baseline, where all disfluencies are kept. Same as in the previous experiments, training the models on in-domain or out-of-domain data does not cause a big performance difference in MT. Comparing the scores of training the models separately for disfluencies and punctuation marks, however, the scores are generally around 0.3 to 0.5 BLEU points worse. The F-score of disfluency removal does not get affected significantly even when we are modeling it along with punctuation marks. However, as the monolingual translation system is trained using much more data, the performance of segmentation and punctuation insertion is affected and therefore degrades the overall performance.

### 6.6. Overview

Finally, Table 9 shows the best scores achieved in this work.

Table 9: *Overview*

| System | No turn | Turn |
|---|---|---|
| Baseline | 9.53 | 12.93 |
| Best system | 14.41 | 15.01 |
| Oracle | 20.93 | |

In our best system we first remove disfluencies using a CRF model trained on the in-domain data, and then insert proper segmentation and punctuation marks using the monolingual translation system. When there is no turn information, we achieve around 4.9 BLEU points of improvement. With turn information, we improve the system by around 2.1 BLEU points.

## 7. Conclusion

In this paper, we showed how machine translation performance is affected when different techniques for segmentation, punctuation insertion and disfluency removal are applied to multispeaker speech. The characteristics and differences of multispeaker speech compared to other data were described. We built two separate disfluency removal systems using in-domain and out-of-domain data and their performances are compared in terms of translation quality. We showed that our disfluency removal technique presented in this work can be transfered to a new domain. Segmentation and punctuation insertion systems are applied after the disfluencies are removed. The best system of disfluency removal and punctuation detection models achieves a gain of 4.9 BLEU points when there is no turn information and 2.1 BLEU points when turn information is available over the baseline. As an additional experiment, a sequence tagging model which models both segmentation, punctuation insertion and disfluency removal is built and the performance is compared to our best automatic systems.

In future work, we would like to explore integrating segmentation, punctuation insertion and disfluency removal systems into end-to-end speech translation systems for real-time evaluation.

## 8. 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.

## 9. References

[1] J. J. Godfrey, E. C. Holliman, and J. McDaniel, “SWITCHBOARD: Telephone Speech Corpus for Research and Development,” in *ICASSP*, San Francisco, CA, USA, 1992.

[2] M. Johnson and E. Charniak, “A TAG-based Noisy Channel Model of Speech Repairs,” in *ACL*, Barcelona, Spain, 2004.

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

[4] D. Baron, E. Shriberg, and A. Stolcke, “Automatic Punctuation and Disfluency Detection in Multi-party Meetings using Prosodic and Lexical Cues,” in *ICSLP*, Denver, CO, USA, 2002.

[5] N. Moran, D. Baron, J. Edwards, D. Ellis, D. Gelbart, A. Janin, T. Pfau, E. Shriberg, and A. Stolcke, “The Meeting Project at ICSI,” in *HLT*, San Diego, CA, USA, 2001.

[6] S. Peitz, M. Freitag, A. Mauser, and H. Ney, “Modeling Punctuation Prediction as Machine Translation,” in *IWSLT*, San Francisco, CA, USA, 2011.

[7] 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.

[8] ——, “Tight Integration of Speech Disfluency Removal into SMT,” in *EACL*, Gothenburg, Sweden, 2014.

[9] Y. Mei and K. Kirchhoff, “Contextual Modeling for Meeting Translation Using Unsupervised Word Sense Disambiguation,” in *Coling*, Beijing, China, 2010.

[10] H. Hassan, L. Schwartz, D. Hakkani-Tür, and G. Tur, “Segmentation and Disfluency Removal for Conversational Speech Translation,” in *Interspeech*, September 2014.

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

[12] E. Cho, T.-L. Ha, and A. Waibel, “CRF-based Disfluency Detection using Semantic Features for German to English Spoken Language Translation,” in *IWSLT*, Heidelberg, Germany, 2013.

[13] E. Cho, S. Fünfer, S. Stüker, and A. Waibel, “A Corpus of Spontaneous Speech in Lectures: The KIT Lecture Corpus for Spoken Language Processing and Translation,” in *LREC*, Reykjavik, Iceland, 2014.

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

[15] J. Lafferty, A. McCallum, and F. Pereira, “Conditional Random Fields: Probabilitic Models for Segmenting and Labeling Sequence Data,” in *ICML*, Williamstown, MA, USA, 2001.

[16] F. Sha and F. Pereira, “Shallow Parsing with Conditional Random Fields,” in *HLT/NAACL*, Edmonton, Canada, 2003.

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

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” in *Workshop at ICLR*, Scottsdale, AZ, USA, 2013.

[19] 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, Demonstration Session*, Prague, Czech Republic, 2007.

[20] A. Stolcke, “SRILM – An Extensible Language Modeling Toolkit.” in *ICSLP*, Denver, CO, USA, 2002.

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

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

[23] A. Venugopal, A. Zollman, and A. Waibel, “Training and Evaluation Error Minimization Rules for Statistical Machine Translation,” in *WPT*, Ann Arbor, MI, USA, 2005.

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

[25] 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.

[26] E. Matusov, G. Leusch, O. Bender, and H. Ney, “Evaluating Machine Translation Output with Automatic Sentence Segmentation,” in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Boulder, Colorado, USA, October 2005.