# Survey: Exploring Disfluencies for Speech-to-Speech Machine Translation

**Rohit Kundu, Preethi Jyothi** and **Pushpak Bhattacharyya**
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
{rkundu, pjyothi, pb}@cse.iitb.ac.in

## Abstract

Disfluencies that appear in the transcriptions from automatic speech recognition systems tend to impair the performance of downstream NLP tasks like machine translation. Disfluency removal models can help alleviate this problem. However, the unavailability of labeled data in low-resource languages impairs progress. In this survey paper, we clearly state the problem in hand and the motivation behind doing disfluency removal. Next, we discuss speech-to-speech machine translation to understand where disfluency detection fits. Then, we cover the disfluency phenomenon in great detail. We also describe the prior works to solve the disfluency detection problem.

## 1 Problem Statement

Spontaneous speech contains many irregularities, e.g., disfluency, which are not evident in read speech. Due to the presence of irregularities in speech, the transcriptions from Automatic Speech Recognition systems may also retain the irregularities, which makes the downstream tasks difficult. Machine Translation is not an exception to that. Our problem statement is to explore the disfluencies and develop models to remove disfluencies from Automatic Speech Recognition transcriptions and prepare them for Machine Translation.

## 2 Motivation

Translation of speech in one language to speech or text in another language sees a huge range of applications, e.g., movie subtitling, movie dubbing, conversing with foreign-language speakers. Previously, people had to confine themselves to their native languages only. Nevertheless, these developments have reduced the language barrier and allowed people from different places to share their ideas.

Many of the current systems are able to handle fluent speeches well, but they suffer in the case of spontaneous speeches. Spontaneous speech contains many irregularities, e.g., disfluency, which are not evident otherwise. Even if Automatic Speech Recognition (ASR) does its task of transcribing each of the uttered words correctly, the transcriptions may reflect the irregularities. Downstream NLP tasks such as parsing, machine translation, and summarization are usually trained on well-formed sentences. Hence, disfluencies in automatically transcribed text pose a significant challenge for downstream NLP tasks (Rao et al., 2007; Wang et al., 2010). Disfluency detection/correction is often used as a preprocessing step for NLP, where the goal is to identify/remove the disfluent words (Shriberg et al., 1992). While disfluency detection has been extensively studied for English (Honal, 2003; Zayats et al., 2014), it has received far less attention in other languages. This is largely due to the lack of labeled data for other languages. This motivates us to investigate disfluencies and their removal mechanism in both high-resource and low-resource settings.

## 3 Speech-to-Speech Machine Translation

Speech-to-Text translation is the task of translating speech in one language to the text in another language. There are two ways to build speech-to-speech Machine Translation models, viz. (1) using direct end-to-end models (Jia et al., 2019b; Lee et al., 2022), (2) using cascaded models. Cascaded models convert the translated text (from source speech) into the speech in the target language using a text-to-speech (TTS) model (Wang et al., 2017). Now, we will discuss the two approaches for the speech-to-text translation (ST) task — Cas-

caded models and end-to-end models (Sperber and Paulik, 2020).

Our main goal is to find the best translation $T^* \in T$ ($T$ denote candidate translations from MT hypothesis space) from the input speech features X. $S \in \mathcal{H}$ denotes a transcription from the ASR hypothesis space. Equation 1 clearly shows our goal.

### 3.1 Cascaded Model

We can arrive at Equation 2 by marginalizing over all the transcription $S \in \mathcal{H}$. Then, we use the chain rule to get Equation 3. Conditional independence assumption of input ($S$) and output ($T$), given the transcript ($X$) leads to equation 4. In Equation 5, we consider $\mathcal{H}'$ contains only 1 hypothesis, the 1-best ASR output. Equation 5 justifies the decomposition of cascaded models into MT and ASR models.

$$T^* = \arg\max_{T \in \mathcal{T}} P(T|X) \tag{1}$$

$$= \arg\max_{T \in \mathcal{T}} \sum_{S \in \mathcal{H}} P(T, S|X) \tag{2}$$

$$= \arg\max_{T \in \mathcal{T}} \sum_{S \in \mathcal{H}} P(T|S, X)P(S|X) \tag{3}$$

$$\approx \arg\max_{T \in \mathcal{T}} \sum_{S \in \mathcal{H}} P_{MT}(T|S, X)P_{ASR}(S|X) \tag{4}$$

$$\approx \arg\max_{T \in \mathcal{T}} \sum_{S \in \mathcal{H}'} P_{MT}(T|S, X)P_{ASR}(S|X) \tag{5}$$

Cascaded models use a pipeline of ASR and MT for speech-to-text MT. Firstly, the ASR model (Schneider et al., 2019; Baevski et al., 2020) generates the transcription in the source language. Then, the MT model (Bahdanau et al., 2015; Vaswani et al., 2017) translates the transcription to the target language.

#### Limitations

**Erroneous Early Decisions:** Cascaded models suffer from the well-known problem of *error propagation* due to considering an erroneous 1-best ASR output. A possible solution to this problem is to increase the hypothesis space $\mathcal{H}'$ in Equation 5.

**Information Loss:** Cascaded models lose useful information due to the conditional independence assumption in Equation 4. *Prosody* not only helps in disambiguation but also conveys useful information about the speaker, but the transcription from the ASR model is unable to capture that. Hence, the MT model is unaware of the *prosody* (since the MT model only considers the ASR transcription).

### 3.2 End-to-end Model

End-to-end models (Bansal et al., 2017; Weiss et al., 2017; Jia et al., 2019a; Liu et al., 2019) directly generate text in the target language from the source speech. These models have to learn the complex mapping of source speech utterances to the translated text. Due to this fact, these models need a huge amount of data to work well. Also, the end-to-end corpora (speech utterances and translated text pair) is harder to get than ASR or MT data.

## 4 Disfluency

Spontaneous speech, like the conversation between multiple people, may contain irregularities. One of the irregularities is *disfluency*. In contrast to texts containing more formal language, such as articles in a newspaper or broadcast news texts, spontaneous speech includes a huge number of sentences that are not fluent. The elements that make a sentence not fluent are referred to as "disfluencies". Speakers often use filler words, repeat fluent phrases, suddenly change the content of speech, and make corrections to their statements. These are some common disfluencies. Also, speakers use words like "yeah", "well", "you know", "alright", etc., which do not contribute to the semantic content of the text but are only used to fill pauses or start a turn, are considered to be disfluent. We give an example of a disfluent sentence. Disfluent parts are highlighted.
Example: **Well, this is** this is **you know** a good plan.

### 4.1 Types of Disfluencies

Here, we assume that the disfluencies occur at the sentence level and do not span over multiple sentences. The annotation of disfluencies varies slightly from corpus to corpus. We will discuss the classification borrowed from Honal (2003). The complexity of disfluencies ranges from simple types to complex types. We will start with simple types. Filled pauses like "uh", "um", "ah" and discourse markers like

"well", "yeah", "alright", "you know", "okay" are considered as simpler disfluencies. Many times, words like "yeah", "okay" are marked as filled pauses. Interjection includes words like "oops", "ugh", "uh-huh". In repetition or correction, the phrase which is abandoned is repeated with only slight or no changes in the syntactical structure. On the other hand, if a completely different syntactical structure with a different semantic is started after the abandoned phrase, then it is a false start. Edit occurs to indicate that the words which just previously have been said are not intended. We present the types of disfluencies in a tabular format with examples in Table 1.

### 4.2 Surface Structure of Disfluencies

Now, we demonstrate the structure of disfluencies, which gives a common pattern (Shriberg, 1994). A disfluency can be divided into three sections: *reparandum* (followed by *interruption point*), then *interregnum*, then *repair*. We show an example in Figure 1. It is important to note that none of the sections is mandatory to be present. But obviously, a disfluent sentence would contain at least one section. *Reparandum* contains those words which are originally not intended to be in the utterance. Thus this section consists of one or more words that will be repeated or corrected (in case of Repetition or Correction) or abandoned completely (in case of a False Start). Then comes the *interruption point* which marks the end of the *reparandum*. It is not connected with any kind of pause or audible phenomenon. It is followed by the *interregnum*. This part consists of an editing term, or a non-lexicalized filler pause like "uh", "um" or discourse markers like "well", "you know" or interjections or simply an empty pause, i.e., a short moment of silence. The last part is *repair*. Words from the *reparandum* are finally corrected or repeated (in case of Repetition or Correction), or a completely new sentence is started (in case of False Start) in the *repair* section. Figure 2 shows an example of disfluency without *interregnum*. It is an example of Correction, i.e., "we will" is corrected as "we can". Figure 3 shows an example of disfluency which does not have *reparandum* or *repair*. This example contains the discourse marker "well" in the *interregnum*.

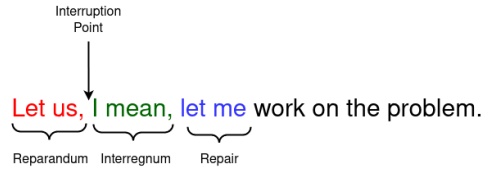


Figure 1: Surface Structure of Disfluency

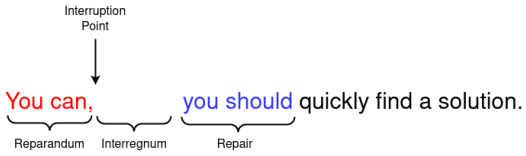


Figure 2: Disfluencies with empty interregnum

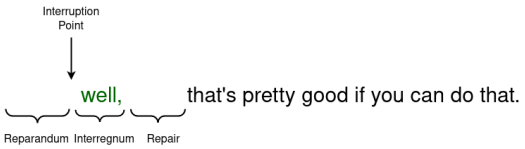


Figure 3: Disfluencies with only interregnum

## 5 Disfluency Detection

In this section, we will discuss various approaches for disfluency detection. It is important to note that the approaches in different subsections might be overlapping.

### 5.1 Sequence Tagging Model

Sequence tagging based approaches use classification techniques to label individual words (Liu et al., 2006; Ostendorf and Hahn, 2013; Zayats et al., 2014; Ferguson et al., 2015; Hough and Schlangen, 2015; Zayats et al., 2016; Wang et al., 2018). We classify each token as fluent/disfluent tags or begin/inside/outside (BIO) tags. Previous works have used Hidden Markov Models (HMM) (Liu et al., 2006), Conditional Random Field (CRF) (Liu et al., 2006; Georgila et al., 2010; Ostendorf and Hahn, 2013; Zayats et al., 2014), Max-Margin Markov Networks ($M^3N$) (Qian and Liu, 2013), Semi-Markov CRF model (Ferguson et al., 2015), Recurrent Neural Network (RNN) (Hough and Schlangen, 2015), bidirectional Long Short-Term Memory (Bi-LSTM) (Zayats et al., 2016), Bi-LSTM with attention mechanism (Wang et al., 2016), Transformer (Wang et al., 2020a) etc. as sequence tagger.

| Type | Description | Constituents | Examples |
|---|---|---|---|
| Filled Pause | Non lexicalized sounds with no semantic content. | uh, um, ah, etc | but **uh** we have to go through the same thing. |
| Interjection | A restricted group of non lexicalized sounds indicating affirmation or negation. | uh-huh, mhm, uh-uh, ugh, uh-oh, oops etc. | **Oops,** I did not know that you would get hurt. |
| Discourse Marker | Words that are related to the structure of the discourse in so far that they help beginning or keeping a turn or serve as acknowledgment. They do not contribute to the semantic content of the discourse. | well, you know, okay, yeah etc. | **Well,** this is a good plan. |
| Repetition or Correction | Exact repetition or correction of words previously uttered. A correction may involve substitutions, deletions or insertions of words. However, the correction continues with the same idea or train of thought started previously. | | If I **can't** don't know the answer myself, I will find it. |
| False Start | An utterance is aborted and restarted with a new idea or train of thought. | | **We'll never find a day** what about next month? |
| Edit | Phrases of words which occur after that part of a disfluency which is repeated or corrected afterwards or even abandoned completely. They refer explicitly to the words which just previously have been said indicating that they are not intended to belong to the utterance. | | We need two tickets, **I'm sorry**, three tickets for the flight to Boston. |

Table 1: Types of Disfluencies with description and example (Honal, 2003)

### 5.2 Parsing based Model

Parsing-based approaches detect disfluencies along with identifying the syntactic structure of the sentence (Rasooli and Tetreault, 2013; Honnibal and Johnson, 2014; Wu et al., 2015; Yoshikawa et al., 2016; Jamshid Lou and Johnson, 2020b). Jamshid Lou and Johnson (2020b) focus on joint disfluency detection and constituency parsing of transcriptions. In Figure 4, we show an example from the paper. The *reparandum*, *filled pauses* and *discourse markers* are denoted by *EDITED*, *INTJ* and *PRN*, respectively.

### 5.3 Noisy Channel Model

The main idea behind a noisy channel model of disfluency is that we assume there is a fluent source sentence $X$ to which some noise has been added, resulting in a disfluent sentence $Y$. The goal is to find the most likely fluent sentence given $Y$ (Johnson and Charniak, 2004; Zwarts and Johnson, 2011; Jamshid Lou and Johnson, 2017).

### 5.4 Translation Based Models

Dong et al. (2019) treat disfluency detection as a translation task from disfluent sentences to fluent sentences. They adapt a neural machine translation model to achieve that. Addition-

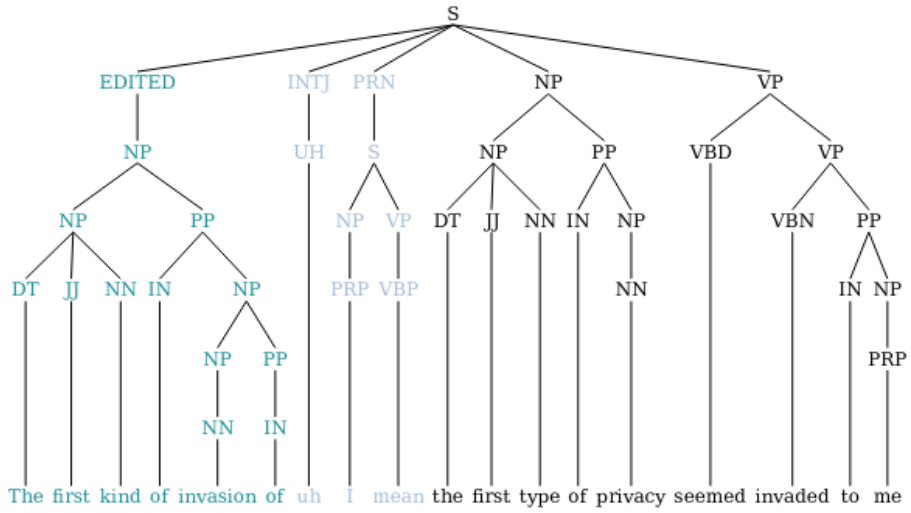


Figure 4: Example of parse tree from the Switchboard corpus (Jamshid Lou and Johnson, 2020b)

ally, they use *constrained decoding*, *denoising autoencoder*. They also take into account a *penalty factor* to reduce wrong deletions (i.e., fluent words tagged as disfluent).

### 5.5 Using Acoustic/Prosodic Cues

Most of the research work on the disfluent text. However, they miss the prosodic cues from the speech. So, some prior works investigated the use of acoustic-prosodic features along with the disfluent text (Ferguson et al., 2015; Zayats and Ostendorf, 2019; Tran et al., 2018)

### 5.6 Data Augmentation

Since the available amount of data for disfluency correction is small, previous research worked on augmenting data. These data augmentation techniques help in overcoming the dearth of gold-standard data. Yang et al. (2020) propose *Planner-Generator* based architecture for generating disfluencies from fluent sentences. The Planner decides where to insert disfluent segments, and the Generator generates appropriate disfluent segments accordingly.

Lee et al. (2020) use auxiliary tasks, namely, Named Entity Recognition (NER) and Part-of-speech Tagging (POS), along with disfluency detection. Since Switchboard disfluency detection data does not have NER tags, they use an off-the-shelf model to annotate silver-standard NER training data.

Passali et al. (2022) generate large-scale disfluency detection data using rules. They focus on *repetition*, *replacement* and *restart* disfluencies.

Rocholl et al. (2021) first finetune BERT model (Devlin et al., 2019) on Switchboard dataset. Then, they use the model for predicting disfluency labels of Fisher corpus (Cieri et al., 2004). Next, they use the automatically labeled silver data as an additional source of training data for further finetuning.

### 5.7 Semi-supervised Methods

Prior works (Wang et al., 2018, 2020a; Saini et al., 2021; Wang et al., 2021) have proposed the use of unlabeled data alongside the labeled data for improving the performance of disfluency detection models. Wang et al. (2018) use *denoising auto-encoder*, *multi-task learning*, *weight sharing* and *generative adversarial network* (GAN) to achieve that. They use two partially shared encoders (Transformer based), one totally shared decoder (Transformer based) and a discriminator. They use one encoder for unlabeled data and the other one for labeled data. The discriminator is used to judge whether the output from the encoder is from labeled or unlabeled data. They have two tasks in hand — generating fluent sequence and generating (disfluency) label sequence.

Wang et al. (2020a) construct a large-scale pseudo training data by randomly adding or

removing words from unlabeled data. Then, they propose two self-supervised objectives for pre-training a model with pseudo training data, followed by finetuning the model on a little amount of disfluency detection data. They use the following two tasks for self-supervised learning: (1) a sequence tagging task to detect the newly added noisy words, and (2) a sentence classification task to distinguish the original sentences from the corrupted sentences.

### 5.8 Unsupervised Methods

Some prior works investigated unsupervised disfluency detection, which does not use labeled sentences at all. Wang et al. (2020b) combine self-training and self-supervised learning to achieve that. They first construct two types of large-scale pseudo data, viz. by (1) randomly adding or removing words from fluent sentences and (2) randomly adding words to fluent sentences. They train a *sentence grammaticality judgment model* using pseudo data (1) with a self-supervised learning method, and a weak disfluency correction model (*teacher* model) using pseudo data (2) with self-supervised learning. Then, they generate pseudo labels of ASR outputs using the teacher model. Words tagged as disfluent in the pseudo labels are deleted from the sentences. Then, they pass these sentences through *sentence grammaticality judgment model* to select sentences with high-quality pseudo labels. Next, they train a student model on the selected pseudo-labeled sentences. They iterate the process by using the student model as the teacher model (to generate new pseudo labels) and training a new student model, until performance stops improving.

Saini et al. (2021) use ideas from *style transfer* models, *backtranslation*, and *reconstruction loss* to achieve unsupervised disfluency correction. They use a single encoder and a single decoder to translate in both directions, i.e., disfluent text to fluent text and vice versa. The decoder is additionally conditioned using a *domain embedding* (from a pre-trained CNN model), which conveys the direction of translation.

### 5.9 End-to-end Disfluency Removal and Machine Translation

Saini et al. (2020) integrate both disfluency removal and machine translation into a single model. They experiment with translating disfluent Spanish to fluent English text. They make use of partially *shared encoders* and a fully shared decoder to perform *denoising* (noisy to fluent English) and *translation* (disfluent Spanish to disfluent English) alternatively. They do not use fluent references during training.

### 5.10 End-to-end Speech Recognition and Disfluency Removal

Most of the end-to-end speech recognition research work on transcribing speech without removing the disfluencies. However, there are a few studies that look at end-to-end speech recognition and disfluency removal. These works do away with the need for a disfluency correction post-processing step. Jamshid Lou and Johnson (2020a) show that end-to-end models learn to generate fluent transcriptions from disfluent speech. However, the models slightly lag behind the baseline, which is the pipeline of ASR and disfluency detection models. Mendelev et al. (2021) focus on removing only partial words (e.g., "cal-" in the sentence "open my **cal- cal-** calendar") from speech transcriptions.

### 5.11 End-to-end Speech Translation and Disfluency Removal

Majority of the end-to-end speech translation research work on translating speech in the source language to the text of the target language without removing disfluencies if they exist in the source speech. However, there are a few studies that look at end-to-end conversational speech translation, which entails converting source disfluent speech into target fluent texts. These works get rid of a separate disfluency detection module. Salesky et al. (2019) report results using a speech-to-text model trained on the original disfluent Fisher Spanish-English Spoken Language Translation (SLT) task.

### 5.12 Small Models

Rocholl et al. (2021) work on small-sized models for disfluency detection. They have demon-

strated that the model size could be reduced by 99% (the model size is as low as 1.3 MB) and inference latency by 80% while maintaining competitive performance.

## 6 Dataset

Switchboard[1] (Godfrey et al., 1992) in English is the most commonly used dataset for disfluency detection. We use various filters at the time of extracting disfluent data from the Switchboard corpus. We filter out utterances marked as *nonspeech* (x), *indeterminate, interrupted, or contains just a floor holder* (%), incorrect transcription (o@ or +@). We also remove sentences with transcription errors, which are detected by the $*$ mark in the transcription of the utterance. We merge utterance segments if the segments are continued by the same speaker. Then, we remove tags put inside angular brackets like <throat_clearing>, <inhaling> from the transcriptions. We also remove tags put inside curly braces like {very faint}, {water running}. We remove tags put inside square brackets like [Clanking], [child_talking], [Children]. We eliminate ill-formed sentences, i.e., sentences with unbalanced brackets. We also ignore those disfluent sentences whose fluent counterpart has only one word.

Following the experimental settings in Wang et al. (2021), we split the Switchboard corpus such that the dev set consists of all sw_04[5-9]*.utt files, the test set consists of all sw_04[0-1]*.utt files, and the training set consists of all the remaining files. We do not include sentences without disfluencies in the training data, but do so in the dev, test set. Table 2 shows the number of sentences in each split of the dataset.

| Split | Number of Sentences |
|---|---|
| Train | 76673 |
| Validation | 5778 |
| Test | 2690 |

Table 2: Switchboard Dataset Statistics

Following Honnibal and Johnson (2014), most of the prior work lowercased the text and removed all punctuation marks and partial words. There are some works (Wang et al., 2020a,b) which discard the "um" and "uh" tokens and merge "you know" and "i mean" into single tokens.

[1] https://catalog.ldc.upenn.edu/LDC97S62

## 7 Evaluation Metrics

Most of the prior works evaluate disfluency detection models using token-level precision, recall and f1 score. However, a few studies find the bleu score between the generated fluent sentences and the fluent reference sentences.

## 8 Conclusion

In this paper, we looked at how disfluencies hamper the performance of speech-to-speech machine translation models. We discussed the structure of disfluencies. We also described various methods for tackling the disfluency detection problem. We found that some models effectively utilize monolingual data via data augmentation or semi-supervised learning. Even some models are capable of doing disfluency detection in unsupervised settings. We also found that some models integrate disfluency removal along with other modules like MT and ASR. We also saw that a tiny model of size 1.3 MB could achieve competitive performance in disfluency detection.

## References


Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Sameer Bansal, Herman Kamper, Adam Lopez, and Sharon Goldwater. 2017. Towards speech-to-text translation without speech recognition. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 474–479, Valencia, Spain. Association for Computational Linguistics.

Christopher Cieri, David Miller, and Kevin Walker. 2004. The fisher corpus: a resource for the next generations of speech-to-text. In *Proceedings*

*of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Qianqian Dong, Feng Wang, Zhen Yang, Wei Chen, Shuang Xu, and Bo Xu. 2019. Adapting translation models for transcript disfluency detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6351–6358.

James Ferguson, Greg Durrett, and Dan Klein. 2015. Disfluency detection with a semi-Markov model and prosodic features. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 257–262, Denver, Colorado. Association for Computational Linguistics.

Kallirroi Georgila, Ning Wang, and Jonathan Gratch. 2010. Cross-domain speech disfluency detection. In *Proceedings of the SIGDIAL 2010 Conference*, pages 237–240, Tokyo, Japan. Association for Computational Linguistics.

John J. Godfrey, Edward C. Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing - Volume 1*, ICASSP'92, page 517–520, USA. IEEE Computer Society.

Matthias Honal. 2003. Correction of Disfluencies in Spontaneous Speech using a Noisy-Channel Approach.

Matthew Honnibal and Mark Johnson. 2014. Joint incremental disfluency detection and dependency parsing. *Transactions of the Association for Computational Linguistics*, 2:131–142.

Julian Hough and David Schlangen. 2015. Recurrent neural networks for incremental disfluency detection. In *Proc. Interspeech 2015*, pages 849–853.

Paria Jamshid Lou and Mark Johnson. 2017. Disfluency detection using a noisy channel model and a deep neural language model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 547–553, Vancouver, Canada. Association for Computational Linguistics.

Paria Jamshid Lou and Mark Johnson. 2020a. End-to-end speech recognition and disfluency removal. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2051–2061, Online. Association for Computational Linguistics.

Paria Jamshid Lou and Mark Johnson. 2020b. Improving disfluency detection by self-training a self-attentive model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3754–3763, Online. Association for Computational Linguistics.

Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. 2019a. Leveraging weakly supervised data to improve end-to-end speech-to-text translation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7180–7184. IEEE.

Ye Jia, Ron J Weiss, Fadi Biadsy, Wolfgang Macherey, Melvin Johnson, Zhifeng Chen, and Yonghui Wu. 2019b. Direct speech-to-speech translation with a sequence-to-sequence model. *arXiv preprint arXiv:1904.06037*.

Mark Johnson and Eugene Charniak. 2004. A TAG-based noisy-channel model of speech repairs. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 33–39, Barcelona, Spain.

Ann Lee, Peng-Jen Chen, Changhan Wang, Jiatao Gu, Sravya Popuri, Xutai Ma, Adam Polyak, Yossi Adi, Qing He, Yun Tang, Juan Pino, and Wei-Ning Hsu. 2022. Direct speech-to-speech translation with discrete units. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3327–3339, Dublin, Ireland. Association for Computational Linguistics.

Dongyub Lee, Byeongil Ko, Myeong Cheol Shin, Taesun Whang, Daniel Lee, Eun Hwa Kim, EungGyun Kim, and Jaechoon Jo. 2020. Auxiliary sequence labeling tasks for disfluency detection. *arXiv preprint arXiv:2011.04512*.

Yang Liu, Elizabeth Shriberg, Andreas Stolcke, Dustin Hillard, Mari Ostendorf, and Mary Harper. 2006. Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. *IEEE Transactions on audio, speech, and language processing*, 14(5):1526–1540.

Yuchen Liu, Hao Xiong, Zhongjun He, Jiajun Zhang, Hua Wu, Haifeng Wang, and Chengqing

Zong. 2019. End-to-end speech translation with knowledge distillation. *arXiv preprint arXiv:1904.08075*.

Valentin Mendelev, Tina Raissi, Guglielmo Camporese, and Manuel Giollo. 2021. Improved robustness to disfluencies in rnn-transducer based speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6878–6882. IEEE.

Mari Ostendorf and Sangyun Hahn. 2013. A sequential repetition model for improved disfluency detection. In *Proc. Interspeech 2013*, pages 2624–2628.

Tatiana Passali, Thanassis Mavropoulos, Grigorios Tsoumakas, Georgios Meditskos, and Stefanos Vrochidis. 2022. LARD: Large-scale Artificial Disfluency Generation. *arXiv preprint arXiv:2201.05041*.

Xian Qian and Yang Liu. 2013. Disfluency detection using multi-step stacked learning. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 820–825, Atlanta, Georgia. Association for Computational Linguistics.

Sharath Rao, Ian Lane, and Tanja Schultz. 2007. Improving spoken language translation by automatic disfluency removal: evidence from conversational speech transcripts. In *Proceedings of Machine Translation Summit XI: Papers*, Copenhagen, Denmark.

Mohammad Sadegh Rasooli and Joel Tetreault. 2013. Joint parsing and disfluency detection in linear time. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 124–129, Seattle, Washington, USA. Association for Computational Linguistics.

Johann C Rocholl, Vicky Zayats, Daniel D Walker, Noah B Murad, Aaron Schneider, and Daniel J Liebling. 2021. Disfluency detection with unlabeled data and small bert models. *arXiv preprint arXiv:2104.10769*.

Nikhil Saini, Jyotsana Khatri, Preethi Jyothi, and Pushpak Bhattacharyya. 2020. Generating fluent translations from disfluent text without access to fluent references: IIT Bombay@IWSLT2020. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 178–186, Online. Association for Computational Linguistics.

Nikhil Saini, Drumil Trivedi, Shreya Khare, Tejas Dhamecha, Preethi Jyothi, Samarth Bharadwaj, and Pushpak Bhattacharyya. 2021. Disfluency correction using unsupervised and semi-supervised learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3421–3427, Online. Association for Computational Linguistics.

Elizabeth Salesky, Matthias Sperber, and Alexander Waibel. 2019. Fluent translations from disfluent speech in end-to-end speech translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2786–2792, Minneapolis, Minnesota. Association for Computational Linguistics.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised Pre-Training for Speech Recognition. In *Proc. Interspeech 2019*, pages 3465–3469.

Elizabeth Shriberg, John Bear, and John Dowding. 1992. Automatic detection and correction of repairs in human-computer dialog. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Elizabeth Ellen Shriberg. 1994. *Preliminaries to a theory of speech disfluencies*. Ph.D. thesis, Citeseer.

Matthias Sperber and Matthias Paulik. 2020. "Speech Translation and the End-to-End Promise: Taking Stock of Where We Are". In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7409–7421, Online. Association for Computational Linguistics.

Trang Tran, Shubham Toshniwal, Mohit Bansal, Kevin Gimpel, Karen Livescu, and Mari Ostendorf. 2018. Parsing speech: a neural approach to integrating lexical and acoustic-prosodic information. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 69–81, New Orleans, Louisiana. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Feng Wang, Wei Chen, Zhen Yang, Qianqian Dong, Shuang Xu, and Bo Xu. 2018. Semi-supervised disfluency detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3529–3538, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Shaolei Wang, Wangxiang Che, Qi Liu, Pengda Qin, Ting Liu, and William Yang Wang. 2020a. Multi-task self-supervised learning for disfluency detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9193–9200.

Shaolei Wang, Wanxiang Che, and Ting Liu. 2016. A neural attention model for disfluency detection. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 278–287, Osaka, Japan. The COLING 2016 Organizing Committee.

Shaolei Wang, Zhongyuan Wang, Wanxiang Che, and Ting Liu. 2020b. Combining self-training and self-supervised learning for unsupervised disfluency detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1813–1822, Online. Association for Computational Linguistics.

Shaolei Wang, Zhongyuan Wang, Wanxiang Che, Sendong Zhao, and Ting Liu. 2021. Combining self-supervised learning and active learning for disfluency detection. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 21(3).

Wen Wang, Gokhan Tur, Jing Zheng, and Necip Fazil Ayan. 2010. Automatic disfluency removal for improving spoken language translation. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5214–5217. IEEE.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. 2017. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*.

Ron J Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. *arXiv preprint arXiv:1703.08581*.

Shuangzhi Wu, Dongdong Zhang, Ming Zhou, and Tiejun Zhao. 2015. Efficient disfluency detection with transition-based parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 495–503, Beijing, China. Association for Computational Linguistics.

Jingfeng Yang, Diyi Yang, and Zhaoran Ma. 2020. Planning and generating natural and diverse disfluent texts as augmentation for disfluency detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1450–1460, Online. Association for Computational Linguistics.

Masashi Yoshikawa, Hiroyuki Shindo, and Yuji Matsumoto. 2016. Joint transition-based dependency parsing and disfluency detection for automatic speech recognition texts. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1036–1041, Austin, Texas. Association for Computational Linguistics.

Vicky Zayats and Mari Ostendorf. 2019. Giving attention to the unexpected: Using prosody innovations in disfluency detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 86–95, Minneapolis, Minnesota. Association for Computational Linguistics.

Vicky Zayats, Mari Ostendorf, and Hannaneh Hajishirzi. 2016. Disfluency Detection Using a Bidirectional LSTM. In *Proc. Interspeech 2016*, pages 2523–2527.

Victoria Zayats, Mari Ostendorf, and Hannaneh Hajishirzi. 2014. Multi-domain disfluency and repair detection. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Simon Zwarts and Mark Johnson. 2011. The impact of language models and loss functions on repair disfluency detection. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 703–711, Portland, Oregon, USA. Association for Computational Linguistics.