

Adversarial Training for Low-Resource Disfluency Correction

Vineet Bhat, Preethi Jyothi, Pushpak Bhattacharyya

Indian Institute of Technology Bombay, India

vineetbhat2104@gmail.com, pjyothi@cse.iitb.ac.in, pb@cse.iitb.ac.in

Abstract

Disfluencies commonly occur in conversational speech. Speech with disfluencies can result in noisy Automatic Speech Recognition (ASR) transcripts, which affects downstream tasks like machine translation. In this paper, we propose an adversarially-trained sequence-tagging model for Disfluency Correction (DC) that utilizes a small amount of labeled real disfluent data in conjunction with a large amount of unlabeled data. We show the benefit of our proposed technique, which crucially depends on synthetically generated disfluent data, by evaluating it for DC in three Indian languages- *Bengali*, *Hindi*, and *Marathi* (all from the Indo-Aryan family). Our technique also performs well in removing *stuttering disfluencies* in ASR transcripts introduced by speech impairments. We achieve an average 6.15 points improvement in F1-score over competitive baselines across all three languages mentioned. To the best of our knowledge, we are the first to utilize adversarial training for DC and use it to correct stuttering disfluencies in English, establishing a new benchmark for this task.

1 Introduction

Disfluencies are words that are part of spoken utterances but do not add meaning to the sentence. Disfluency Correction (DC) is an essential pre-processing step to clean disfluent sentences before passing the text through downstream tasks like machine translation (Rao et al., 2007; Wang et al., 2010). Disfluencies can be introduced in utterances due to two main reasons: the conversational nature of speech and/or speech impairments such as stuttering. In real-life conversations, humans frequently deviate from their speech plan, which can introduce disfluencies in a sentence (Dell et al., 1997). Stuttering speech consists of involuntary repetitions or prolongations of syllables which disturbs the fluency of speech.

Conversational disfluencies occur once every 17 words (Bortfeld et al., 2001) whereas a 2017 US

study¹ shows that roughly 1% of the population stutters and predominantly consists of children. One out of every four children continues to suffer from this disorder lifelong. When such speech passes through an ASR system, readability of the generated transcript deteriorates due to the presence of disfluencies in speech (Jones et al., 2003).

Shriberg (1994) defines the surface structure of disfluent utterances as a combination of reparandum, interregnum and repair. The reparandum consists of the words incorrectly uttered by the speaker that needs correction or complete removal. The interregnum acknowledges that the previous utterance may not be correct, while repair contains the words spoken to correct earlier errors.

Type	Example
Conversational	Well, you know, this is a good plan.
Stuttering	Um it was quite fu funny

Table 1: Examples and surface structure of disfluent utterances in conversational speech and stuttering. Red - Reparandum, Blue - Interregnum, Orange - Repair

Data in DC is limited because of the time and resources needed to annotate data for training (Appendix ??). Through this work², we provide a method to create high-quality DC systems in low resource settings. Our main contributions are:

1. Improving the state-of-the-art in DC in Indian languages like Bengali, Hindi and Marathi by 9.19, 5.85 and 3.40 points in F1 scores, respectively, using a deep learning framework with adversarial training on real, synthetic and unlabeled data.
2. Creating an open-source stuttering English DC corpus comprising 250 parallel sentences
3. Demonstrating that our adversarial DC model can be used for textual stuttering correction

¹<https://www.nidcd.nih.gov/health/stuttering>

²<https://github.com/vineet2104/AdversarialTrainingForDisfluencyCorrection>

with high accuracy (87.68 F1 score)

2 Related work

Approaches in DC can be categorized into noisy channel-based, parsing-based, and sequence tagging-based approaches. Noisy channel-based approaches rely on the following principle: a disfluent sentence Y can be obtained from a fluent sentence X by adding some noise. These models try to predict the fluent sentence X given the disfluent sentence Y (Honal and Schultz, 2004; Jamshid Lou and Johnson, 2017; Johnson and Charniak, 2004). Parsing-based approaches jointly predict the syntactic structure of the disfluent sentence along with its disfluent elements (Honnibal and Johnson, 2014; Jamshid Lou and Johnson, 2020; Rasooli and Tetreault, 2013; Wu et al., 2015; Yoshikawa et al., 2016). Sequence tagging-based approaches work on the following hypothesis: every word in a disfluent sentence can be marked as fluent/disfluent. These methods work best for shorter utterances and perform optimally for real-life conversational DC (Hough and Schlangen, 2015; Ostendorf and Hahn, 2013; Zayats et al., 2016). Moreover, sequence-tagging based methods require far less labeled data to perform well, compared to the other two methods. Our approach to DC focuses on treating it as a sequence tagging problem rather than a machine translation task. The objective is to accurately classify each word as either disfluent or fluent, and create fluent sentences by retaining only the fluent words. The lack of labeled data for DC in low-resource languages has prompted the use of semi-supervised methods and self-supervised techniques (Wang et al., 2018; Wang et al., 2021). DC has also been studied as a component in speech translation systems, and thus its effect has been analyzed in improving the accuracies of machine translation models (Rao et al., 2007; Wang et al., 2010). Synthetic data generation for DC has also received attention recently. These methods infuse disfluent elements in fluent sentences to create parallel data for training (Passali et al., 2022; Saini et al., 2020). Our work is an extension of Kundu et al. (2022), which creates the first dataset for DC in Bengali, Hindi and Marathi. We use this dataset to train our adversarial model to improve over the state-of-the-art in these languages. To the best of our knowledge, we are the first to model DC to correct stuttering ASR transcripts.

3 Types of Disfluencies

There are six broad types of disfluencies encountered in real life - Filled Pause, Interjection, Discourse Marker, Repetition or Correction, False Start and Edit. Although these are common in conversational speech, stuttering speech consists mainly of Filled Pauses and Repetitions. This section describes each type of disfluency and gives some examples in English.

1. **Filled Pauses** consist of utterances that have no semantic meaning.
Example - What about the **uh** event?
2. **Interjections** are similar to filled pauses, but their inclusion in sentences indicates affirmation or negation.
Example - **Ugh**, what a day it has been!
3. **Discourse Markers** help the speaker begin a conversation or keep turn while speaking. These words do not add semantic meaning to the sentence.
Example - **Well**, we are going to the event.
4. **Repetition or Correction** covers the repetition of certain words in the sentence and correcting words that were incorrectly uttered.
Example - If I **can't** don't go to the event today, it is not going to look good.
5. **False Start** occurs when previous chain of thought is abandoned, and new idea is begun.
Example - **Mondays dont work for me**, how about Tuesday?
6. **Edit** refers to the set of words that are uttered to correct previous statements.
Example - We need **three tickets**, **I'm sorry**, four tickets for the flight to California.

4 Architecture

The lack of labeled data for DC is a significant hurdle to developing state-of-the-art DC systems for low-resource languages. Passali et al. (2022), Saini et al. (2020) and Kundu et al. (2022) introduced data augmentation by synthesizing disfluencies in fluent sentences to generate parallel data. In this work, we propose a deep learning architecture that uses adversarial training to improve a BERT-based model's token classification accuracy of whether a token is disfluent or not. Our proposed architecture uses real, synthetic and unlabeled data to improve classification performance.

Our model, Seq-GAN-BERT, is inspired by Croce et al. (2020), who first used a similar model

for sentence classification. It consists of three modules: a BERT-based encoder (Devlin et al., 2019), discriminator and generator. The encoder converts the input sequence $X = (X_1, X_2, \dots, X_n)$ into encoded vector representations (H_{real}). Simultaneously, the generator creates fake representations (H_{fake}) from Gaussian random noise (Z), mimicking the real data that passes through the encoder. The discriminator aims to solve a two-pronged objective: i) predicting every word in the sentence to be disfluent or fluent and ii) determining whether the input from the generator comes from real or fake data.

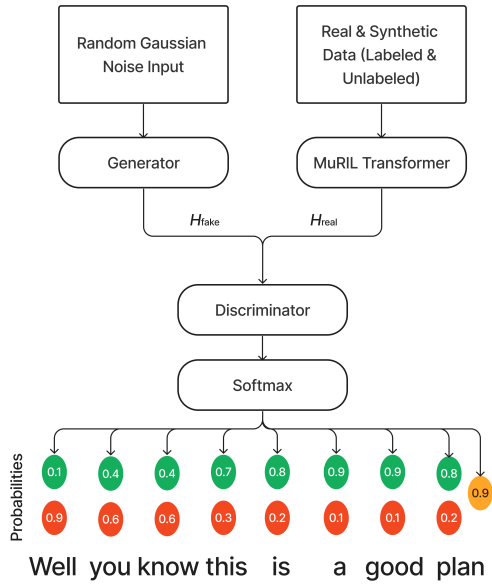


Figure 1: Architecture of the Seq-GAN-BERT model. Green nodes denote fluent class probabilities, red nodes denote disfluent class probabilities and the orange node shows the probability of classifying a sample as real (1) or fake (0).

4.1 Adversarial Training

The discriminator loss comprises two loss terms. The first loss is supervised by the token classification task, while the second loss is defined by the real/fake data identification task. Such adversarial training also allows the model to use unlabeled data during training. For unlabeled samples, only the real/fake data identification task is executed. The generator continuously improves during training and produces fake representations that resemble actual data. The competing tasks of the generator (to create better representations to fool the discriminator) and the discriminator (to perform token

classification for labeled sentences and real/fake identification) compels the MuRIL encoder to generate better representations of input sentences. The resulting high-quality representations allow the discriminator to identify disfluent words with a high accuracy.

5 Task 1: Few Shot DC in Indian Languages

To test our proposed architecture, we train the model on the few-shot DC task for Indian languages. The current state-of-the-art performance in Bengali, Hindi and Marathi DC is obtained by training a large multilingual transformer model using synthetic data created by injecting disfluencies in fluent sentences using rules (Kundu et al., 2022). We train our Seq-GAN-BERT model using the authors’ multilingual real and synthetic data.

5.1 Dataset

Our dataset consists of parallel disfluent-fluent sentences in three Indian languages. We use 300, 150 and 250 real disfluent sentences in Bengali, Hindi and Marathi, respectively and generate 1000 synthetic disfluent sentences in Bengali and 500 synthetic disfluent sentences each in Hindi and Marathi each by infusing disfluent elements in fluent transcriptions using a rule-based approach (Kundu et al., 2022). The synthetic data was created such that the percentage of disfluent words across 3 languages remains constant.

5.2 Text Processing and Training Details

Text pre-processing is performed by removing punctuations, lower-casing and creating word-level tokens for parallel sentences. The Seq-GAN-BERT model uses a combination of labeled and unlabeled data comprising real and synthetically generated disfluent sentences in different languages. We try different combinations of monolingual and multilingual data. Our experiments show that the best model for Bengali uses real and synthetic Bengali sentences as labeled data and disfluent Hindi sentences as unlabeled data. The best model for Hindi uses real and synthetic Hindi sentences as labeled data and disfluent Bengali sentences as unlabeled data. The best model for Marathi uses real and synthetic Marathi sentences as labeled data and disfluent Bengali sentences as unlabeled data. The BERT-based transformer that we use as an encoder is the MuRIL model pretrained on English and many Indian languages (Khanuja et al., 2021).

MuRIL representations for Indian languages are of superior quality compared to other multilingual Transformer-based models like mBERT (Devlin et al., 2019).

5.3 Evaluation

To evaluate our model, we train baselines for DC in zero-shot and few-shot settings. *ZeroShot* is based on Kundu et al. (2022). *FewShot* is based on training MuRIL on all real and synthetic data available in the chosen language, along with labeled data in a related Indian language (for Bengali, either Hindi or Marathi can act as a related Indian language). *FewShotAdv* is the Seq-GAN-BERT model without any unlabeled data. Although models like BiLSTM-CRF have been as alternatives to transformers for sequence tagging, direct fine-tuning often performs better (Ghosh et al., 2022). Performance of DC systems is usually measured with F1 scores (Ferguson et al., 2015; Honnibal and Johnson, 2014; Jamshid Lou and Johnson, 2017). Table 3 shows the comparison of various baselines against our model.

Lang	Model	P	R	F1
Bn	ZeroShot	93.06	62.18	74.55
	FewShot	66.37	68.20	67.27
	FewShotAdv	84.00	78.93	81.39
	Our model	87.57	80.23	83.74
Hi	ZeroShot	85.38	79.41	82.29
	FewShot	82.99	81.33	82.15
	FewShotAdv	88.15	83.14	85.57
	Our model	89.83	86.51	88.14
Mr	ZeroShot	87.39	61.26	72.03
	FewShot	82.00	60.00	69.30
	FewShotAdv	84.21	64.21	72.86
	Our model	85.34	67.58	75.43

Table 2: Comparing the performance of baselines and our model on DC across Bengali (Bn), Hindi (Hi) and Marathi (Mr); ZeroShot - Monolingual supervised training, FewShot - Multilingual supervised training, FewShotAdv - Adversarial training without unlabeled data, Our model - Multilingual adversarial training with unlabeled data; P = Precision, R = Recall

Our model, Seq-GAN-BERT with unlabeled sentences, performs better than the other baselines and establishes a new state-of-the-art for DC in Bengali, Hindi and Marathi. Our model benefits from adversarial training using both unlabeled data and multilingual training. The observed precision and recall scores of these models during testing show that without adversarial training, the model per-

forms with high precision but low recall. However, with adversarial training, the model improves its recall without compromising much on precision. The zero-shot model (without adversarial training) classifies less words as disfluent but at a high accuracy, whereas the few-shot model (with adversarial training) correctly classifies more words as disfluent.

6 Task 2: Stuttering DC in English

We have already shown how our proposed architecture learns better semantic representations for DC using small amounts of manually annotated labeled data. In this section, we present a similar experiment in Stuttering DC (SDC). We define SDC as the task of removing disfluent elements in spoken utterances that are caused by stuttering speech impairment. Since this is the first attempt to model stuttering correction as disfluency removal, we make our version of the existing dataset for stuttering publicly available for research purposes and provide various baseline comparisons. We show that our model generalizes well for this task and is able to remove disfluent elements in stuttering speech.

6.1 Dataset

The UCLASS dataset is created by transcribing audio interviews of 14 anonymous teenagers who stutter and consists of two released versions (Howell et al., 2004). Both versions of this corpus are available for free download and research. We create 250 disfluent-fluent parallel sentences from the available transcripts of such utterances. The dataset is released here³.

6.2 Processing & Training

We follow the same steps as before (section 5.2). Stuttered syllables are represented in the text, separated by a space delimiter and treated as a disfluent term. This gold-standard dataset is split into 150 sentences for training and 100 sentences for testing. The training sentences are used as labeled data for the model and unlabeled data from Switchboard (Godfrey et al., 1992) or Kundu et al. (2022) is used to facilitate multilingual training. Our model performs best when we use synthetic Bengali disfluent sentences as unlabeled data.

³<https://github.com/vineet2104/AdversarialTrainingForDisfluencyCorrection>

Model	P	R	F1
SupervisedGold	89.11	78.08	83.23
SupervisedGoldSWBD	87.34	86.50	86.92
SupervisedGoldLARD	74.58	86.33	80.02
AdversarialSWBD	85.76	84.17	84.96
AdversarialLARD	86.21	84.82	85.51
Our model	87.26	88.10	87.68

Table 3: Comparing baselines and our model for English stuttering DC; SupervisedGold - Supervised training on gold standard dataset, SupervisedGoldSWBD and SupervisedGoldLARD - Supervised training on gold standard dataset and DC data, AdversarialSWBD and AdversarialLARD - Adversarial training without unlabeled data, Our model - Multilingual Adversarial training with unlabeled data; P = Precision, R = Recall

6.3 Evaluation

We use five baselines to evaluate our model’s performance. *SupervisedGold* uses the gold standard data and trains the MuRIL model for token classification. *SupervisedGoldSWBD* and *SupervisedGoldLARD* uses a combination of the gold standard dataset along with 1000 disfluent sentences from the Switchboard corpus and LARD dataset (Passali et al., 2021). *AdversarialSWBD* and *AdversarialLARD* uses the Seq-GAN-BERT to train on a combination of labeled sentences from gold standard corpus and unlabeled sentences from the Switchboard corpus and LARD dataset. Table 4 displays our results averaged over multiple seeds.

Our model outperforms all baselines. Improvement over *AdversarialLARD* shows the benefit of multilingual training. We also used synthetic Hindi or Marathi data while training, but achieved lower scores than the *AdversarialLARD* baseline.

Summary of results: In this paper, we evaluate our proposed architecture for low-resource DC using two tasks: 1) DC in Indian languages and 2) Stuttering DC in English. Our model outperforms competitive baselines across both these tasks establishing a new state-of-the-art for Indian languages DC. The adversarial training in our model improves the representations of a BERT-based encoder for disfluent/fluent classification. We show that multilingual training benefits such tasks as the generator is trained to create better representations of fake data to fool the discriminator.

7 Conclusion

Adversarial training using unlabeled data can benefit disfluency correction when we have limited amounts of labeled data. Our proposed model can

also be used to correct stuttering in ASR transcripts with high accuracy.

Future work lies in integrating speech recognition models like Whisper⁴ or wav2vec 2.0 (Baevski et al., 2020) to create end-to-end speech-driven DC models. It will also be insightful to see how this model transfers to other low-resource languages with different linguistic properties.

8 Acknowledgements

We would like to thank the anonymous reviewers and area chairs for their suggestions to strengthen the paper. This work was done as part of the Bahubhashak Pilot Project on Speech to Speech Machine Translation under the umbrella of National Language Technology Mission of Ministry of Electronics and IT, Govt. of India. We would also like to thank Nikhil Saini for valuable discussions during the course of this project.

9 Limitations

There are two main limitations of our work. Firstly, since there are no known baselines for Indian language DC except Kundu et al. (2022), other architectures might perform better than our model. Our claim that Seq-GAN-BERT tries to maximize the information gained from unlabeled sentences is supported by superior performance over baselines defined in this work and other related models. Secondly, due to the lack of good quality labeled datasets, our test sets contained only 100 sentences. However, we believe that the consistency of our high-performing models across languages and multiple seeded experiments presents a positive sign for DC in low-resource settings.

10 Ethics Statement

The aim of our work was to design an adversarial training-enabled token classification system that is able to correctly remove disfluencies in text. The datasets used in this work are publicly available and we have cited the sources of all the datasets that we have used.

⁴<https://cdn.openai.com/papers/whisper.pdf>

References

- Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. Wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20*, Red Hook, NY, USA. Curran Associates Inc.
- Heather Bortfeld, Silvia D. Leon, Jonathan E. Bloom, Michael F. Schober, and Susan E. Brennan. 2001. Disfluency Rates in Conversation: Effects of Age, Relationship, Topic, Role, and Gender. *Language and Speech*, 44(2):123–147.
- Danilo Croce, Giuseppe Castellucci, and Roberto Basili. 2020. GAN-BERT: Generative adversarial learning for robust text classification with a bunch of labeled examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2114–2119, Online. Association for Computational Linguistics.
- Gary S. Dell, Lisa K. Burger, and William R. Svec. 1997. Language production and serial order: A functional analysis and a model. *Psychological Review*, 104(1):123–147.
- 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.
- 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.
- Sreyan Ghosh, Sonal Kumar, Yaman Kumar Singla, Rajiv Ratn Shah, and Sharma Umesh. 2022. Span classification with structured information for disfluency detection in spoken utterances. In *Interspeech*.
- John J. Godfrey, Edward Holliman, and J. McDaniel. 1992. Switchboard: telephone speech corpus for research and development. *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1:517–520 vol.1.
- Matthias Honal and Tanja Schultz. 2004. 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.
- Peter Howell, Stephen Davis, Jon Bartrip, and Laura Wormald. 2004. Effectiveness of frequency shifted feedback at reducing disfluency for linguistically easy, and difficult, sections of speech (original audio recordings included). *Stammering research : an on-line journal published by the British Stammering Association*, 1(3):309–315.
- 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. 2020. 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.
- Mark Johnson and Eugene Charniak. 2004. A tag-based noisy channel model of speech repairs. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, ACL '04*, page 33–es, USA. Association for Computational Linguistics.
- Douglas Jones, Florian Wolf, Edward Gibson, Elliott Williams, Evelina Fedorenko, Douglas Reynolds, and Marc Zissman. 2003. Measuring the readability of automatic speech-to-text transcripts.
- Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Gali, Vish Subramanian, and Partha Talukdar. 2021. MuriL: Multilingual representations for indian languages.
- Rohit Kundu, Preethi Jyothi, and Pushpak Bhat-tacharyya. 2022. Zero-shot disfluency detection for Indian languages. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4442–4454, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- M. Ostendorf and S. Hahn. 2013. A sequential repetition model for improved disfluency detection. *Proceedings of the Annual Conference of the International Speech Communication Association, INTER-SPEECH*, pages 2624–2628.
- Tatiana Passali, Alexios Gidiotis, Efstathios Chatzikyriakidis, and Grigorios Tsoumakas. 2021. Towards human-centered summarization: A case study on financial news. In *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, pages 21–27, Online. Association for Computational Linguistics.

- Tatiana Passali, Thanassis Mavropoulos, Grigorios Tsoumakas, Georgios Meditskos, and Stefanos Vrochidis. 2022. [LARD: Large-scale artificial disfluency generation](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2327–2336, Marseille, France. European Language Resources Association.
- 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.
- 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.
- Elizabeth Shriberg. 1994. Preliminaries to a theory of speech disfluencies.
- 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, 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.
- 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.
- 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, Mari Ostendorf, and Hannaneh Hajishirzi. 2016. [Disfluency detection using a bidirectional lstm](#). pages 2523–2527.