Survey: Exploring Disfluencies for Speech To Text Machine Translation

Nikhil Saini, Preethi Jyothi and Pushpak Bhattacharyya
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Mumbai, India
{nikhilra, pjyothi, pb}@cse.iitb.ac.in

Abstract

Spoken language is different from the written language in its style and structure. Disfluencies that appear in transcriptions from speech recognition systems generally hamper the performance of downstream NLP tasks. Thus, a disfluency correction system that converts disfluent to fluent text is of great value. This survey paper talks about disfluencies present in speech and its transcriptions. Later, we describe methodologies to correct disfluencies present in the transcriptions of a spoken utterance via various approaches viz, a) style transfer for disfluency correction b) transfer learning and language model pretraining. We observe that disfluency inherent speech phenomenon and its correction is crucial for downstream NLP tasks.

1 Introduction

Natural Language and Speech Processing strives to build machines that understand, respond and generate text and voice data in the same way humans do. NLP and speech come under the umbrella of Artificial Intelligence, which is a branch of computer science. NLP and Speech processing has come a long way from rule-based systems to traditional statistical systems to machine learning and deep learning based systems. The NLP and speech systems enable machines to understand the whole meaning of text or speech and the intent and sentiment of the writer or speaker.

Natural language and speech processing is the driving force behind several computer programs like those which translate text from one language to another, respond to spoken commands from users, correct spelling, grammar and prompts suggestions on keyboards, recommends movies and shows on streaming websites, recommends products in e-shopping websites, speech to text dictation systems, chat-bots, search engines, fitness apps, sleep monitoring, spam detection in email, and many more.

In Natural Language Processing (NLP), it becomes more and more critical to deal with spontaneous speech, such as dialogs between two people or even multi-party meetings. The goal of this processing can be translation, text summarization, spoken language translation, real-time audio dubbing or subtitle generation, or simply the archiving of a dialog or a meeting in a written form.

Disfluencies are disruptions to the regular flow of speech, typically occurring in conversational speech. They include filler pauses such as *uh* and *um*, word repetitions, irregular elongations, discourse markers, conjunctions, and restarts. For example, the disfluent sentence "well we're actually uh we're getting ready" has its fluent form as, "we're getting ready". Here, the words highlighted in green, blue and red refer to discourse, filler and restart disfluencies, respectively.

Disfluencies in the text can alter its syntactic and semantic structure, thereby adversely affecting the performance of downstream NLP tasks such as information extraction, summarization, translation, and parsing (Charniak and Johnson, 2001; Johnson and Charniak, 2004). These tasks also employ pre-trained language models that are typically trained to expect fluent text. This motivates the need for disfluency correction systems that convert disfluent to fluent text. Prior work has predominantly focused on the problem of disfluency detection (Zayats et al., 2016; Wang et al., 2018; Dong et al., 2019). The effect is profound for pre-trained language models (Devlin et al., 2019; Edunov et al., 2018) that are typically trained to expect fluent language. Various systems such as System User Interfaces and speech-to-speech translations systems suffer due to disfluencies. Additionally, it is crucial to model disfluencies for different higher-level natural language processing tasks such as information extraction, summarization, parsing from transcribed textual inputs. In the tasks of parsing and machine translation (Rao et al., 2007), it has been observed that disfluencies adversely affect performance. Most of the existing NLP tools, such as pre-trained language models (Devlin et al., 2019) and translators (Edunov et al., 2018) are developed for well-formed fluent text without considerations of disfluency. Therefore, in spite of their very high accuracy on fluent text, utilizing them for solutions on disfluent (transcribed from spoken) text is relatively less accurate. For example, to predict sentiment in customer care scenario, we could potentially use pre-trained language models and sentence classifiers, if we could make the transcribed text nearly fluent.

2 Disfluency

2.1 Conversational Speech

In contrast to texts which are well-formed like in newspapers, Wikipedia pages, blogs, books, manuscripts, formal letters/documents, etc., conversational/spontaneous speech has a very high degree of freedom and includes a very high number of utterances which are not *fluent/clean*. The elements that make an utterance non-fluent are termed as ***disfluencies***.

Disfluent speech and its disfluent transcriptions possess problems for various downstream NLP tasks. Mainly, all downstream NLP tasks deal with text which is well-formed and formatted. Therefore, it is difficult for such models to incorporate the irregularities present in the speech data in the form of disfluencies. Moreover, since speech is becoming very important looking at the linguistic geography, it is of utmost importance to remove irregularities present in speech utterances so that a clean utterance can be utilized by other NLP applications like Machine Translation, Speech To Speech Translation, Summarization, Question Answering, etc.

The problems pertaining to transcripts of conversational speech can be broadly summarized as (but not limited to):

1. **Presence of disfluent terms/phrases:** Spoken utterances usually contain various disfluent terms in a single utterance, which the speaker didn't intend to speak and must be processed before using in a downstream NLP task.
   Disfluent:
   "well we're actually uh we're getting ready"

2. **Incorrect grammar in the spoken utterance:** Often, speakers do not care much about exact grammar when communicating via speech. This introduces irregularity in the utterance.
   Incorrect Grammar: "i are getting ready"

3. **Incomplete utterances:** Automatic speech recognition systems generate transcriptions by segmenting input speech into fixed slots (say 5 seconds). It leads to creation of utterance that can be the beginning, middle or end of an utterance. Downstream NLP tasks aren't compatible handling incomplete utterances. The related task is known as *sentence boundary detection in asr transcriptions.*
   Incomplete utterance:
   "and i told her to create"

4. **Other errors introduced via ASR system:** ASR systems introduce other errors due to several factors like speaker variabilities (change in voice due to age, illness, tiredness, etc.), spoken language variabilities (pronunciation variation due to dialects and co-articulation), mismatch factors (i.e., mismatch in recording conditions between training and testing data).

2.2 Surface Structure of Disfluencies

In this section, a pattern is described which demonstrates the structure of disfluencies. These patterns are called the surface structure of disfluencies as only characteristics of disfluencies are considered, observable from the text. A disfluency can be divided into three parts: The ***reparandum***, then there is an interruption point, after which comes the ***Interregnum***, followed by ***repair***.

Figure 1 shows a breakdown example. The ***reparandum*** contains those words, which are originally not intended to be in the utterance. Thus it consists of one or more words that will be repeated or corrected ultimately (in case of a repetition/correction) or abandoned completely (in case of a false start). The *interruption point* marks the offset of the reparandum. It is not connected with any pause or audible phenomenon. The ***interregnum*** can consist of an editing term, a non lexicalized pause like *uh* or *uhm* or simply of an empty pause, i.e. a short moment of silence.

In many cases however, the *interregnum* of a disfluency is empty and the *repair* follows directly after the *reparandum*. In the ***repair*** the words from the *reparandum* are finally corrected or repeated (repetition/correction) or a complete new sentence is started (false start). Note that in the latter case, the extension of the repair can not be determined.

The three terms *reparandum, interregnum,* and *repair* can be used to explain repetitions, false starts, and editing terms. The *reparandum* and *interregnum* can be empty in a disfluent sentence. This situation fits the criteria for three different disfluency types, viz., discourse markers, filled pauses and interjections. These three types consists only of interregnum. Figure 2 shows breakdown of *interregnum* being empty and Figure 3 shows the breakdown of *reparandum, repair* being empty.

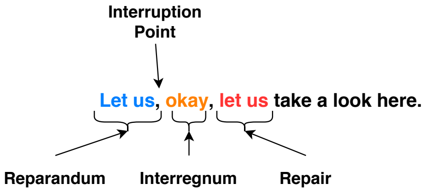

Figure 1: Surface Structure of Disfluency

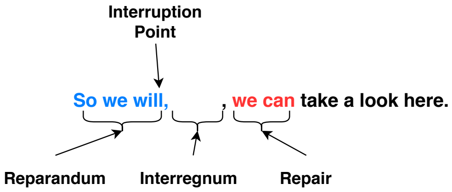

Figure 2: Disfluencies with empty interregnum.

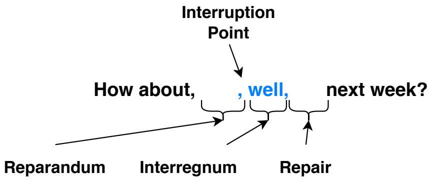

Figure 3: Disfluencies with empty reparandum and empty repair.

3 Types of Disfluencies

This section will describe the different types of disfluencies that can be found in the disfluent text. These disfluency types are present in the switchboard corpus. The annotation of disfluencies can vary slightly from corpus to corpus. Disfluencies can be divided into two sub-groups, viz., simpler and complex disfluencies. Filled pauses like ***oh, uh, um*** and discourse markers like ***yeah, well, okay, you know*** are considered as simpler disfluencies. Sometimes, single word discourse markers like the word ***Yeah*** in the sentence ***"Yeah, we are leaving now."***, is considered as a filled pause. We differentiate between filler words and discourse markers, even in single word occurrences, as this distinction is also present in the annotated switchboard corpus. Now, we will look into complex disfluency types, viz., ***Repetition or Correction, False Start, Edit, Aside***. For the distinction of the categories ***Repetition or Correction*** and ***False Start***, it is important to consider that the phrase which has been abandoned is repeated with only slight or no changes in the syntactical structure. The change can be in the form of Insertion, Deletion, or Substitution of words. The slight or no change identifies it as a ***Repetition or Correction*** disfluency. On the other hand, if a completely different syntactical structure with different semantics is chosen for the repair, the observed disfluency is a false start.

The disfluency classification is important and is used to determine the type of disfluencies one wants to correct in the disfluent text. It also forms the basis for the classifiers one can train to learn the disfluency type domain embeddings. Generally, the approaches do not depend on the type of disfluencies, but making explicit use of the annotated corpus and incorporate the knowledge of specific disfluency types into the models is beneficial. Table 1 describes the different disfluency types, their definitions and examples.

4 Approaches

In this section, we will discuss two approaches to correct disfluencies in disfluent text. The problem statement is: “Correct disfluencies present in transcribed utterances (e.g.noisy ASR output) of conversational speech (e.g. Telephonic conversations, Lectures delivered, etc) by removing the *“disfluent”* part without changing the intended meaning of the speaker.”

4.1 Style Transfer for Disfluency Correction

1. **Architecture**
   Figure 4 clearly shows the two directions of

Disfluency Type	**Description**	**Constituents**	**Example**
Filled Pause	Non lexicalized sounds with no semantic content.	uh, um, ah, etc	We're ***uh*** getting ready.
Interjection	A restricted group of non lexicalized sounds indicating affirmation or negation. An interjection is a part of speech that demonstrates the emotion or feeling of the author.	uh-huh, mhm, mm, uh-uh, nah, oops, yikes, woops, phew, alas, blah, gee, ugh.	1. I dropped my phone again, ***ugh.*** 2. ***Oops,*** I didn't mean it.
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. These are also called linking words.	okay, so, well, you know, etc	1. ***Well,*** this is good. 2. This is, ***you know***, a pretty good report.
Restart 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.	-	1. This ***is is*** a ***bad bad*** situation. 2. Are you you happy?
False Start	An utterance is aborted and restarted with a new idea or train of thought.	-	1. ***We'll never find a day*** what about next month ? 2. ***Yes*** no I'm not coming.
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: Disfluency Types, Description and Examples.

translation. The model obtains latent disfluent and latent fluent utterances from the non-parallel fluent and disfluent sentences, respectively, which are further reconstructed back into fluent and disfluent sentences. A back-translation-based objective is employed, followed by reconstruction for both domains i.e. disfluent and fluent text. For every mini-batch of training, soft translations for a domain are first generated (denoted by $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ in Figure 4), and are subsequently translated back into their original domains to reconstruct the mini-batch of input sentences. The sum of token-level cross-entropy losses between the input and the reconstructed output serves as the reconstruction loss.

Components in a neural model can be shared minimally, completely, or in a controlled fashion. A complete parameter sharing is done, which treats the model as a black box for both translation directions and offers maximum simplicity. Advantages of Parameter Sharing:

- In sequence to sequence tasks, sharing parameters between encoders helps to improve the accuracy when the different sources are related.
- Similarly, when the targets are related, parameter sharing helps to improve the accuracy.
- Parameter sharing allows the model to get benefit from the learning's through the back-propagated loss of different translation directions. Since we are only operating on the English language in the source(disfluent) and target(fluent), it is imperative to utilize the benefit of parameter sharing.

Disadvantages of Parameter Sharing:

- Sometimes, sharing of encoders and decoders leads to burdening the parameters to learn a large representation with limited space.

This bottleneck can be avoided by increasing the layers in both encoders and decoders. The encoders and decoders are shared for both translation directions; disfluent-to-fluent and fluent-to-disfluent. In a sequence to sequence transduction task, the encoder takes an input and generates a representation in the latent space, the decoder then takes it and generates a sequence in the target domain. Since the disfluent and fluent domains in a language share almost all the vocabulary and are in the same language; the components can learn the representations from each other's loss. Moreover, since we are operating in an unsupervised setting; the sharing of parameters forces the encoder to limit the representations of both domains in a common space; thereby allowing the model to mix the knowledge of the two domains.

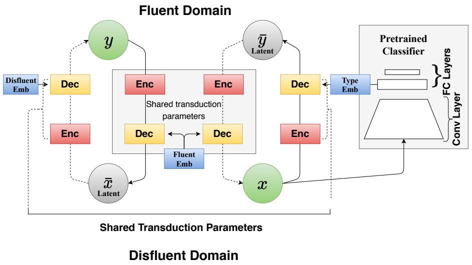

Figure 4: Illustration of Style transfer model modified to use type embedding drawn from a pretrained CNN classifier.

Borrowing from prior work on an unsupervised style transfer model (He et al., 2020), the decoder is conditioned on a domain embedding that specifies the direction of translation. There are two types of embeddings: A vanilla *binary domain embedding* that takes a bit as input to indicate whether the input text is fluent or disfluent and a *classifier-based domain embedding*. The latter is obtained from a trained standalone CNN-based classifier (Kim, 2014) that predicts the disfluency type of a disfluent input sentence. (Here, it is assumed that disfluency type labels are available for the disfluent sentences in our training data.) The penultimate layer from the classifier acts as our classifier embedding, which is further used to condition the decoder. It is hypothesized that additional information about disfluency types via the classifier-based embedding might help guide the process of disfluency correction better.

2. **Choice of Encoder-Decoder Cells:**

    - Bi-LSTM
    - Transformer

3. **Domain Embedding in Transformer:** Figure 5 illustrates the conditioning of the transformer-based decoder. Dimensionality reduced word embedding is concatenated with the domain embedding *DE* at every time-step(t) to form the input for the decoder.

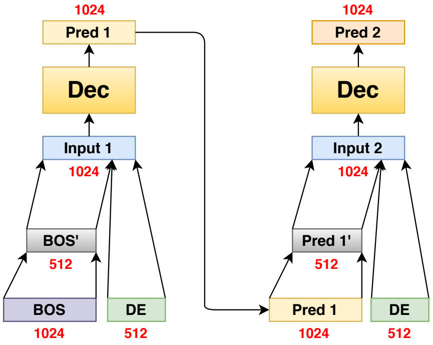

Figure 5: Induction of domain embedding: Demonstration of domain embeddings into transformers' decoder. $Pred(i = 1)$ and $Input(i = 1)$ are decoder's prediction and input to the decoder at i_{th} time-step respectively.

4.2 Seq2Seq with MASS Pretraining Objective

This is an encoder-decoder model built on Transformer encoder-decoder cells. MASS: Masked Sequence to Sequence (Song et al., 2019) is a novel pretraining method for language generation based tasks. It randomly masks a sequence fragment in the encoder, and then predicts it in the decoder. Figure 6 shows the masked language modeling objective for language generation.

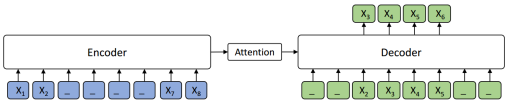

Figure 6: A novel pretraining objective for language generation.

Figure 7 shows a novel pretraining loss for large scale supervised neural machine translation. Masked Language Modeling(MLM) can be seen

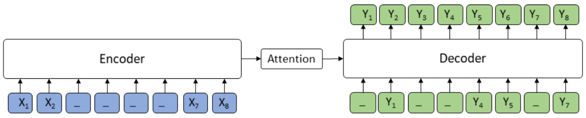
Y_1 Y_2 Y_3 Y_4 Y_5 Y_6 Y_7 Y_8
Encoder
Attention
Decoder
X_1 X_2 _ _ _ _ X_7 X_8
_ Y_1 _ _ Y_4 Y_5 _ Y_7

Figure 7: A novel pretraining loss for supervised learning.

in BERT. BERT is built on Transformer encoder layers. Standard Language Modeling(SLM) GPT-2. GPT-2 is built on Transformer decoder layers. Let the number of words masked/hidden be defined by a parameter *k*. Masked Language Modeling in BERT can be viewed as when *k=1* and Standard Language Modeling in GPT-2 can be viewed as when *k=m* (where m is the length of the output sequence). The model structure of MASS varies between *k=1* and *k=m*.

To pretrain the language model, publicly available clean text corpus in the desired language is used. These sentences do not contain disfluencies and work well as a proxy to a large fluent corpus in the desired language. Following experimental setting can be used to train and evaluate the model:

1. Language Modeling: Pretraining only on Fluent sentences.

2. Language Modeling and Supervised Training: Here, both language modeling and supervised training steps on the respective datasets in each epoch.

3. Supervised Training: on disfluent-to-fluent parallel corpus.

4. Language Modeling and Supervised Training with Pretrained encoder: Reload the encoder from a pretrained model in the same language.

5. Language Modeling and Supervised Training with Pretrained encoder and decoder: Reload both the encoder and decoder from a pretrained model (where the source language is same as the language which is being considered for disfluency correction).

5 Conclusion

This paper presented the problem if disfluency as an inherent phenomenon in conversational speech and its transcriptions. We discussed the definition of disfluency, types of disfluencies and their surface structures. We discussed two broad approaches to correct disfluencies in ASR transcriptions, i) style-transfer based disfluency correction and, ii) disfluency correction using pretraining and language modeling objectives of MASS. We observed that very little research has been done in correction of disfluencies in text and speech, but a lot has been done. However, disfluency correction in languages not limited to English, end-to-end disfluency correction with other downstream NLP tasks like machine translation, speech-to-text translation, etc, is an active and promising area of research.

References

Eugene Charniak and Mark Johnson. 2001. Edit detection and parsing for transcribed speech. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

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.

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*, 33:6351–6358.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. In *International Conference on Learning Representations*.

Mark Johnson and Eugene Charniak. 2004. A tag-based noisy channel model of speech repairs. pages 33–39.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Sharath Rao, Ian Lane, and Tanja Schultz. 2007. Improving spoken language translation by automatic disfluency removal: Evidence from conversational speech transcripts. *Machine Translation Summit XI*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936.

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.

Vicky Zayats, Mari Ostendorf, and Hannaneh Hajishirzi. 2016. Disfluency detection using a bidirectional LSTM. *CoRR*, abs/1604.03209.