DACL: Disfluency Augmented Curriculum Learning for Fluent Text Generation

Rohan Chaudhury, Maria Teleki, Xiangjue Dong, James Caverlee
Texas A&M University
College Station, TX, USA
{rohan.chaudhury, mariateleki, xj.dong, caverlee}@tamu.edu

Abstract

Voice-driven software systems are in abundance. However, language models that power these systems are traditionally trained on fluent, written text corpora. Hence there can be a misalignment between the inherent *disfluency* of transcribed spoken content and the *fluency* of the written training data. Furthermore, gold-standard disfluency annotations of various complexities for incremental training can be expensive to collect. So, we propose in this paper a **D**isfluency **A**ugmented **C**urriculum **L**earning (**DACL**) approach to tackle the complex structure of disfluent sentences and generate fluent texts from them, by using Curriculum Learning (CL) coupled with our synthetically augmented disfluent texts of various levels. DACL harnesses the tiered structure of our generated synthetic disfluent data using CL, by training the model on basic samples (i.e. more fluent) first before training it on more complex samples (i.e. more disfluent). In contrast to the random data exposure paradigm, DACL focuses on a simple-to-complex learning process. We comprehensively evaluate DACL on Switchboard Penn Treebank-3 and compare it to the state-of-the-art disfluency removal models. Our model surpasses existing techniques in word-based precision (by up to 1%) and has shown favorable recall and F1 scores.

Keywords: Disfluency, Fluent Text Generation, Curriculum Learning

1. Introduction

"Hey Siri, uh I mean, hey Alexa, what is the weather like today?" – any breaks in the regular flow of speech, such as false starts, corrections, repeats, and filled pauses, are referred to as *disfluencies*. Disfluencies are common in everyday speech (Shriberg, 1994). Disfluent speech has three distinct parts: the *reparandum,* the *interregnum,* and the *repair* (Shriberg, 1994). In Figure 1, we show an example from and Jamshid Lou and Johnson (2020b) from the Switchboard Corpus (Mitchell et al., 1999; Godfrey and Holliman, 1997). The *reparandum* in this instance – "The first kind of invasion" – is replaced by the *repair* part "the first type of privacy". The interregnum "uh I mean" consists of a *filled pause* "uh" and a *discourse marker* "I mean". Fluent speech is obtained from the disfluent speech by removing the *reparandum* and *interregnum* (Shriberg, 1994; Jamshid Lou and Johnson, 2020b).

It is important to remove disfluencies from transcribed speech. Jones et al. (2003) find that humans have trouble reading and parsing Automated Speech Recognition (ASR) outputs that contain disfluencies and lack punctuation. Also, data which does not contain disfluencies is beneficial for performance on downstream tasks for conversational systems, summarization, and machine translation systems (Rao et al., 2007; Cho et al., 2014; Wang et al., 2010; Hassan et al., 2014; Teleki et al., 2024).

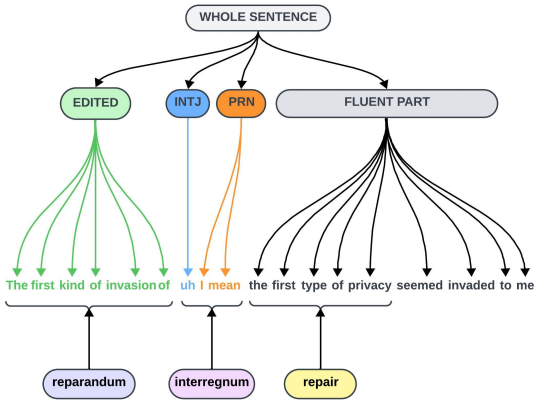

Figure 1: An example used by Jamshid Lou and Johnson (2020b) from the Switchboard corpus, annotated in the Shriberg (1994) disfluency structure.

Switchboard[1] (Mitchell et al., 1999; Godfrey and Holliman, 1997) is a commonly used disfluency corpus for the disfluency removal task, however, it does not reflect various levels of disfluency in a systematic way. Thus, we propose **DACL**, a **D**isfluency **A**ugmented **C**urriculum **L**earning method to generate fluent text directly from disfluent text. Our method consists of two

[1] In the Switchboard treebank corpus (Mitchell et al., 1999; Godfrey and Holliman, 1997), the reparanda, filled pauses, and discourse markers are made of the nodes labeled EDITED, INTJ, and PRN, respectively as shown in Figure 1 (Jamshid Lou and Johnson, 2020b).

modules – disfluency augmentation (DA) and curriculum learning (CL). First, in DA, we use a synthetic disfluency amplification process that produces progressive versions of the original text with varying degrees of disfluency to simulate the wide spectrum of disfluent speech (as some speakers may be more disfluent than others, for example, in the case of speakers with a fluency disorder) (American Speech-Language-Hearing Association, 2023). Next, we pair the tiered structure of our synthetically augmented disfluent texts (DA) with CL – a process that dictates the order of training samples in machine learning (Bengio et al., 2009). This aligns with our synthetic disfluency augmentation process to build a curriculum for our model, as DACL uses the tiered structure of our generated synthetic disfluent data: basic samples (i.e. more fluent) are fed to the model before more complex samples (i.e. more disfluent). This contrasts a random data exposure paradigm, focusing instead gradually increasing the "difficulty" of our data. This incrementally increases the ability of DACL to identify which phrases are disfluencies in subsequent stages in a controlled manner, leading to higher precision. Higher precision ensures that the model is careful when identifying phrases as disfluencies and minimizes false positives. This lowers the risk that valid sections of the speech will be incorrectly eliminated which can lead to information loss, resulting in poor-quality transcriptions with relevant parts of the sentences missing.

Our CL phase has 6 training stages on the augmented datasets with increasing disfluencies. Then, we have a final fine-tuning phase on the target dataset so that the model can learn about any new disfluencies in the target dataset that are not included in our augmentations. We make the code available at https://github.com/Rohan-Chaudhury/Generating-Fluent-Text-through-Curriculum-Learning-And-Disfluency-Augmentation.

Our contributions are:

- We propose a new training methodology – DACL – to train a sequence-to-sequence generation model that generates fluent text directly from disfluent text.
- We evaluate the performance of DACL against the Switchboard Penn Treebank-3 test set and compare it to the leading parsing-based and translation-based disfluency removal models. DACL obtains state-of-the-art results in word-based precision scores, and also obtains favorable word-based recall and F1 scores.

2. Related Work

2.1. Modeling Approaches

There are two main types of modeling approaches for removing disfluencies from the text: *parsing-based models* and *translation-based models*. Most of these models use labeling, wherein they categorize each word as either *fluent* or *disfluent*, to generate fluent text from disfluent text.

Parsing-Based Models. Recent parsing-based models tend to all follow a similar pipeline, introduced in Kitaev and Klein (2018). Initially, the input text is segmented into spans. Every span from index i to index j, where $i \leq j$, is then processed using a language model to extract embeddings. Next, a classifier is applied to the embeddings to identify the suitable label for each span in the parse tree, and the optimal parse tree is constructed. Within the set of parse tree labels, some labels are designated as representing *disfluency*, and the respective words or spans are removed to form the parallel fluent span.

In recent studies, a variety of embedding models have been employed: Jamshid Lou et al. (2018) proposes a convolutional neural network (CNN) to obtain embeddings, while in a different study, Jamshid Lou et al. (2019) apply ELMo (Peters et al., 2018) to procure embeddings. On a similar note, Sarzynska-Wawer et al. (2021), and Jamshid Lou and Johnson (2020b) made use of BERT (Devlin et al., 2019) to derive embeddings that were subsequently used as input for the labeling classifier. As noted in Wang et al. (2017), these models are powerful in their ability to process information related to disfluencies in the context of the entire span, but they suffer from the computational burden of needing to carry out the parsing task as well.

Translation-Based Models. Several papers have investigated translation-based models for disfluency detection and elimination. Chen et al. (2022) introduces a streaming BERT-based sequence tagging model along with a novel training objective, capable of real-time disfluency detection while maintaining a balance between accuracy and latency. Wang et al. (2023) propose a multi-scale self-attention mechanism (MSAT) for disfluency detection, coupled with contrastive learning (CL) loss to retain a rough copy and sustain semantic consistency. Tian et al. (2022) approach disfluency detection as a sequence labeling problem, employing Bi-LSTM and attention mechanisms to cater to long-distance dependencies. Jamshid Lou and Johnson (2020a) explore end-to-end speech recognition and disfluency removal, with the aim of training an ASR model to independently generate fluent transcripts, without relying on a separate disfluency detection model.

2.2. Disfluency Augmentation

In this paper, we introduce additional disfluencies into the text for our curriculum learning process. In a similar spirit, Wang et al. (2020a) use disfluency augmentation as a pretraining step to self-train their translation-based model. Additionally, they explore two distinct pretraining steps: labeling disfluencies at the word level and classifying grammar at the sentence level. They suggest three disfluency transformations: repetition, insertions, and deletions. Similarly, Passali et al. (2022) synthetically creates disfluencies, either as repetitions, replacements, or restarts.

2.3. Curriculum Learning

CL is a method where a machine learning model is trained on simpler data before moving on to more complex data, mimicking the way a school curriculum is structured. This strategy has proven effective in different settings like computer vision and natural language processing (Soviany et al., 2022; Wang et al., 2021), enhancing the model's generalization ability and rate of convergence (Shi and Peng, 2023; Wang et al., 2021). Utilizing reinforcement learning to determine the level of ease or challenge of samples in a dataset is another method being examined in machine learning (Wang et al., 2021).

CL has been used for various speech applications. Zhu et al. (2021) use CL for ASR tasks for converting speech input to text output. Lang and Wang (2019) apply CL to two spoken language understanding (SLU) endeavors: named entity recognition (NER) and program sequencing.

Additionally, CL has been found to be very successful, particularly in Neural Machine Translation (NMT) domains (Zhang et al., 2019; Platanios et al., 2019). Models based on CL are capable of faster training and generating better results in comparison to those reliant on unstructured, stochastic sampling (Penha and Hauff, 2020; Zhang et al., 2019; Platanios et al., 2019).

3. Methods

We propose **DACL**: **D**isfluency **A**ugmented **C**urriculum **L**earning to remove disfluencies and generate fluent text. We perform synthetic disfluency injection into the original text to varying degrees and then apply curriculum learning to the increasing levels of disfluent data in multiple stages. The model learns to precisely identify disfluent patterns from fluent phrases during DACL and thus is able to better generate fluent text from disfluent text.

3.1. Synthetic Disfluency Augmentation

Here we introduce the first phase of DACL, where we create three synthetic disfluency augmentation processes that gradually *decrease the fluency* of the original sentence by *increasing the number of disfluencies* in the sentence, as shown in Figure 2.

For each of our disfluency transformation types (repeats, interjections, and false starts), we control the level of these synthetic disfluencies using a parameter N (Wang et al., 2020a; Passali et al., 2022). We apply each of these transformations on each of the pre-training datasets (the Spotify Podcast Dataset and WikiSplit) from $N = 0$ to $N = 10$. A value of $N = 0$ indicates that no synthetic disfluency has been added to the dataset, whereas a value of $N = 10$ signifies that the highest level of disfluency has been introduced. Then the three transformations are:

1. Repeats: Here, we increase the number of repetitions in the input text. We divide the input text into random-length substrings, by drawing samples from $X \sim Normal(\mu = 10, \sigma = 1)$ to determine the length of each consecutive substring. At the end of each substring, the last word in the substring is appended N times, and punctuation and capitalization are modified accordingly within the substring.

2. Interjections: Here, we increase the number of interjections in the input text. We divide the input text into random-length substrings, by drawing samples from $X \sim Normal(\mu = 10, \sigma = 1)$ to determine the length of each consecutive substring. At the end of each substring, N uniformly randomly selected interjections from the list ["uh", "um", "well", "like", "so", "okay", "you know", "I mean"] are appended, and punctuation and capitalization are modified accordingly within the substring.

3. False Starts: Here, we increase the number of false starts in the input text. Out of the sentences with word length ≥ 4, we non-uniformly sample sentences with 80% probability to get a false start repeated N times in it. False starts are considered as the first two words of the sentence being repeated N times consecutively.

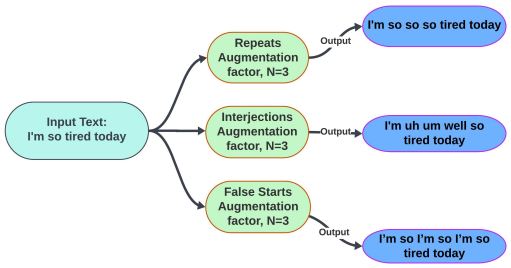

Figure 2: An example of the three different synthetic disfluency augmentation processes (repeats, interjections, and false starts).

3.2. Disfluency Augmented Curriculum Learning

Through disfluency augmentation, we create progressive versions of the original text with varying degrees of disfluency. This tiered structure of our augmented disfluent sentences naturally pairs with the CL process of DACL. The model's capability to better identify the disfluencies in a sentence increases with each subsequent stage of the CL phase. At each stage of the CL phase, we frame the problem as a *sequence-to-sequence* generation task, where the model is presented with a disfluent sentence and aims to produce a fluent one in return. The training stages during this phase are conducted without changing most of the model hyperparameters except the sequence length of the model. We use T5-base (Raffel et al., 2020)[2] as the backbone generative model for all our experiments. To guide the generative model for our disfluency removal and fluent sentence generation task, we add *"Remove text disfluency: "* at the beginning and *" [END]"* at the end of the input sentences and add *" [END]"* at the end of the output sentences.

There are 6 stages in DACL:

Stage 1. In the first stage of DACL, we feed the non-augmented sentences ($N = 0$) into the generative model and ask it to generate these sentences exactly as they are. This stage enables the model to recognize that if the provided sentences are free from disfluencies, they should be returned unaltered. Consequently, this helps prevent the model from returning unintended random outputs or empty strings, as witnessed in some of our ablation studies.

Stage 2. From the second stage, we begin introducing some synthetic disfluencies in our input texts. We start with the *repeats* (word repetitions) dataset, as it's relatively more straightforward compared to the two other datasets, interjections and false-starts. Our input texts are the *repeats* dataset with $N = 1$ and the outputs are the original non-augmented ($N = 0$) dataset. We train the best model obtained from the previous stage (with the least validation loss) on the training set of these pairs of inputs and outputs.

Stages 3-6. Similarly, in the third, fourth, fifth, and sixth stages, our input texts are the *repeats* dataset with $N = 5$, *repeats* dataset with $N = 10$, *interjections* dataset with $N = 10$, and *false starts* dataset with $N = 10$ respectively. The outputs for all the stages are set as the original non-augmented ($N = 0$) texts. We train the model from the previous stages on the training set of the respective pairs of inputs and outputs for the corresponding stages.

[2] https://huggingface.co/t5-base

Table 1: Statistics of the 3 datasets post data cleaning.

Dataset	Train	Dev	Test
Spotify (all augmentations)	714	153	153
WikiSplit (all augmentations)	2,970	637	637
Switchboard	21,849	1,884	2,448

After the sixth stage, we take the model with the lowest validation loss, **DACL-best**, and proceed to fine-tune it on the target dataset.

3.3. Fine-Tuning for Evaluation

The **DACL-Best** now understands the basics of what is a disfluency and can precisely remove disfluencies that we have shown through our synthetic augmentations. However, naturally disfluent datasets like Switchboard Treebank-3 can have more kinds of disfluencies that we have not engineered in our augmentations. To tackle these new types of disfluencies, we fine-tune **DACL-best** – the best checkpoint from DACL – on the training set of the target dataset, Switchboard Treebank-3. (Mitchell et al., 1999).

Post-Processing. After obtaining the output from our models, we take the longest common subsequence (LCS) between the *predicted sentence* and the *disfluent input sentence*. LCS is the longest series of words that appear in both texts in the same order, however, they do not have to be contiguous. The model tends to fix grammatical errors in sentences, which is a positive aspect of our method. However, since the word-level precision, recall, and F1 score depend on the ordering and position of the tokens in the sentence, the addition of such extra words tends to slightly impact the accuracy of our calculations. Hence we remove such extra words by doing an LCS operation with the *input disfluent sentences* which are also devoid of such fixes.

4. Experimental Setup

We design experiments to evaluate the effectiveness of our proposed DACL approach and answer two research questions: **RQ1:** How will DACL perform with CL on in-domain datasets? **RQ2:** How will DACL perform with CL on out-of-domain datasets?

4.1. Datasets

For DACL, we utilize two datasets: The Spotify Podcast Dataset (Clifton et al., 2020), and The WikiSplit Dataset (Botha et al., 2018), and then do fine-tuning and evaluation on the Switchboard Dataset (Mitchell et al., 1999; Godfrey and Holliman, 1997). We perform synthetic disfluency aug-

mentation on these datasets and then study the impact of CL over each of these datasets.

Spotify Podcast Dataset. The Spotify Podcast Dataset[3] is comprised of over 100,000 podcasts which contain "a rich variety of genres, subject matter, speaking styles, and structural formats" (Clifton et al., 2020). The podcasts were transcribed using Google's automatic speech recognition (ASR) tool (Google Cloud), which filters out some basic disfluencies. For our experiments, we perform synthetic disfluency transformations on the *test set* of the Spotify Podcast Dataset, which contains 1,028 podcasts; however, we only select podcasts that have speech in their first minute of duration, which cuts the dataset down to 1,020 podcast transcripts. We observe that in the first minute of podcasts, people are generally fluent, sometimes due to scripted beginnings where hosts read off a standard intro for each podcast, which results in a low number of natural disfluencies in the transcripts.

WikiSplit Dataset. The WikiSplit Dataset[4] is "[o]ne million English sentences, each split into two sentences that together preserve the original meaning, extracted from Wikipedia edits" (Botha et al., 2018). We use a small subset of the dataset for our experiments that contains 5,000 sentences.[5] These sentences are written texts and completely devoid of disfluencies.

Switchboard Treebank-3. After performing DACL using the pre-training datasets, we further fine-tune the model on the Switchboard dataset,[6] and evaluate it on the Switchboard Treebank-3[7] test set (Mitchell et al., 1999; Godfrey and Holliman, 1997). This allows us to directly compare our results to previous work (Jamshid Lou and Johnson, 2020b; Bach and Huang, 2019; Wang et al., 2018; Zayats et al., 2016).

The Switchboard dataset is a transcribed and syntactically annotated dataset comprised of phone calls between randomly paired participants on selected topics (Godfrey and Holliman, 1997). The disfluencies are mainly made up of the nodes labeled EDITED, PRN, and INTJ in the Switchboard Treebank-3 dataset (Jamshid Lou and Johnson, 2020b). We therefore derive the *fluent* and *disfluent* versions of the text from these annotations.

[3]Available at: https://podcastsdataset.byspotify.com/

[4]Available at: https://github.com/google-research-datasets/wiki-split

[5]We use the *test set*.

[6]Following Charniak and Johnson (2001) and Jamshid Lou and Johnson (2020b), we divided the Switchboard into training, dev, and test sets. Specifically, the training data includes the sw[23]*.mrg files, the dev data comprises the sw4[5-9]*.mrg files, and the test data encompasses the sw4[0-1]*.mrg files.

[7]Available at: https://catalog.ldc.upenn.edu/LDC99T42

4.2. Data Pre-Processing

For the Spotify Podcast dataset and the WikiSplit dataset, we eliminate extra spaces from the sentences. For the Switchboard dataset, we create disfluent and fluent versions. According to Jamshid Lou and Johnson (2020b), within the tree-annotation structure of the Switchboard dataset, there are mainly 3 types of nodes that are disfluent. As demonstrated by Jamshid Lou and Johnson (2020b), Figure 1 illustrates one such parse tree. The process to create the fluent and disfluent version follows:

1. Disfluent: This is the original transcript that is obtained by recursively going through the Switchboard parse trees and collecting their leaves, following the method outlined by Jamshid Lou and Johnson (2020b).

2. Fluent: These are the original transcripts with nodes labeled *EDITED, INTJ*, and *PRN* filtered out. The resulting sentences are devoid of disfluencies.

Next, following Jamshid Lou and Johnson (2020b); Johnson and Charniak (2004), all partial words[8] and punctuations have been omitted from the disfluent and fluent sentences, as they are not found in realistic ASR applications.

Additionally, we split each of the transcripts after every 4 speaker turns to ensure that all sentences in the dataset, when tokenized by the T5-base Tokenizer, are under 512 tokens in size. To ensure each sentence comprises at least 20 characters, subsequent sentences are recursively joined together if any sentence falls short of the 20-character threshold. This is carried out to ensure that neither disfluent nor fluent sentences contain any blank entries after splitting them based on speaker turns.

Table 1 presents the number of sentences of each of the datasets in each of the three data splits following the data cleaning process.

4.3. Model and Implementation Details

We utilize T5-base[9] (Raffel et al., 2020) as our backbone model, which is an encoder-decoder model with 220 million parameters. It is a medium-sized model compared to recent Large Language Models (i.e., LLaMA 2[10] and GPT-4[11]). Fine-tuning T5 on a task-specific dataset allows it to specialize in disfluency correction, drawing on the generic language knowledge from pretraining to improve its disfluency correction capacity. All experiments are run on NVIDIA RTX A5000 24GB

[8]Words ending in "-" or words tagged as "XX"

[9]https://huggingface.co/t5-base

[10]LLaMA 2 has 7 billion to 70 billion parameters (Touvron et al., 2023).

[11]GPT-4 has 1.7 trillion parameters (OpenAI).

Table 2: DACL with the Spotify Dataset and Evaluation on the Switchboard Dataset

Method	Word-Based			ROUGE		
	P	R	F	1	2	L
Repeats Augmented $[0, 1, 5, 10]$ shuffled (no CL)	26.35	46.99	33.76	73.47	68.43	73.33
Interjections Augmented $[0, 1, 5, 10]$ shuffled (no CL)	27.69	**48.35**	**35.22**	70.33	66.52	70.18
Repeats $0-0$	69.69	0.43	0.85	89.09	83.43	89.10
Repeats $0-0, 1-0$	93.87	8.72	15.96	89.64	83.97	89.65
Repeats $0-0, 1-0, 5-0$	95.72	9.80	17.78	89.77	84.09	89.78
Repeats $0-0, 1-0, 5-0, 10-0$	**95.76**	10.04	18.18	89.79	84.11	89.78
Repeats $0-0, 10-0$	92.09	4.29	8.21	89.32	83.65	89.32
(DACL-Best) Repeats $0-0, 1-0, 5-0, 10-0$, Interjections $10-0$, False Starts $10-0$	94.80	14.74	25.52	**90.14**	**84.62**	**90.13**

Table 3: DACL with the WikiSplit Dataset and Evaluation on Switchboard Dataset

Method	Word-Based			ROUGE		
	P	R	F	1	2	L
Repeats $0-0$	22.84	6.80	10.52	88.75	83.02	88.67
Repeats $0-0, 1-0$	49.68	22.50	30.97	90.01	84.21	89.96
Repeats $0-0, 1-0, 5-0$	53.27	25.68	34.66	90.30	84.48	90.24
(DACL-Best) Repeats $0-0, 1-0, 5-0, 10-0$, Interjections $10-0$, False Starts $10-0$	**71.09**	**68.12**	**69.58**	**93.91**	**90.86**	**93.86**

GPUs and NVIDIA GeForce RTX 2080 Ti.

4.4. Metrics

Word-Level Precision, Recall, and F1 Score. Three word-level evaluation metrics are taken into account, as per prior studies (Ferguson et al., 2015; Zayats et al., 2016; Jamshid Lou and Johnson, 2020b; Bach and Huang, 2019; Wang et al., 2020b). The evaluation code has been adapted from the work of Wang et al. (2020b) to maintain consistency with previous research.

ROUGE Scores. We utilize ROUGE metrics[12] to intermediately evaluate the generated outputs. ROUGE scores compare the N-gram overlaps between the generated text and a reference text to assess the quality of translations.

5. Experimental Results

5.1. RQ1: Results on Spotify Dataset (In-Domain)

The Spotify dataset is in-domain for the Switchboard dataset, as both are transcribed spoken text. We measure the word-based PRF scores and the ROUGE scores on the test set of the Switchboard dataset after executing DACL on the Spotify dataset in Table 2. Then, we fine-tune and evaluate our model on the Switchboard dataset in Table 4. Finally, we compare our model to the state-of-the-art models in Table 6.

[12] https://huggingface.co/spaces/evaluate-metric/rouge

5.1.1. DACL (Table 2)

Starting with the first and second rows, we train the T5-base model using mixed levels of disfluency augmentations with no CL. Specifically, the input augmentation levels are $N = 0, 1, 5,$ and 10, with all corresponding output levels set at 0. They are all combined and *shuffled* to form a larger dataset. Shuffling the input data at various augmentation levels allows us to study the impact of CL, as we remove the curriculum in this ablation. This training is applied to both the *repeat* (first row) and *interjection* (second row) augmented Spotify datasets. We notice in these two rows that the model exhibits low precision on the Switchboard test set. However, it achieves high recall scores. This is because the model frequently categorizes entire sentences as disfluent, removing all the words. Another reason is that the model can sometimes generate entirely random outputs different from the input sequences. On inspecting the raw outputs before postprocessing, we see the model generating random tokens, such as: *TRUE*, *FALSE*, and *END*. The LCS postprocessing step then removes these words and returns null strings–as these words are not originally present in the input disfluent sentences. We conclude that in this case, the models fail to understand the underlying task of clearing disfluencies.

Now, we look at the third through sixth row. Here, we study the impact of progressively increasing the difficulty of the curriculum over input augmentation levels $N = 0, 1, 5,$ and 10. We note that the model's recall score is initially low but increases

Table 4: Fine-Tuning & Evaluation on Switchboard Dataset after DACL on Spotify

Curriculum Learning on Spotify	Fine-tune on SWB?	Word-Based			ROUGE		
		P	R	F	1	2	L
No (T5-base)	N	17.74	49.25	26.08	0.5722	0.5124	0.5696
	Y	93.57	83.66	88.34	0.9752	0.9598	0.9750
DACL-Best	N	94.80	14.74	25.52	0.9015	0.8463	0.9014
	Y, 14 epochs – DACL+FT	**97.10**	84.75	90.50	0.9795	0.9650	0.9793
	Y, Overfitting, 66 epochs – DACL+FT (Overfit)	96.10	**90.25**	**93.08**	**0.9855**	**0.9758**	**0.9854**

Table 5: Fine-tuning & Evaluation on Switchboard Dataset after DACL on WikiSplit

Curriculum Learning on WikiSplit	Fine-tune on SWB?	Word-Based			ROUGE		
		P	R	F	1	2	L
No (T5-base)	N	17.74	49.25	26.08	0.5722	0.5124	0.5696
	Y	93.57	83.66	88.34	0.9752	0.9598	0.9750
DACL-Best	N	71.09	68.12	69.58	0.9391	0.9086	0.9386
	Y	**95.13**	**87.00**	**90.89**	**0.9816**	**0.9691**	**0.9815**

over time during these training and evaluation stages. The precision score remains consistently high and gradually increases. This pattern indicates that the model improves its ability to accurately identify and remove disfluencies without excessive token removal, as these models don't give any empty strings or irrelevant tokens as output when evaluating on the Switchboard test set.

Next, we move on to the seventh row, where we skip intermediate CL steps by directly increasing the input disfluency level from 0 to 10. We note that the model's performance on the Switchboard test set was inferior compared to the model in the sixth row (which had a curriculum that progressively increased in difficulty from 0 to 1 to 5 to 10, rather than a jump in difficulty from 0 to 10), suggesting that training on the intermediate levels (such as 1, 5, etc.) is necessary for the model to identify disfluencies.

Next, in the eighth row, we build on the curriculum of the model from the sixth row as we add in additional disfluency types: *interjections* with $N = 10$ and *false starts* with $N = 10$. This more difficult training regimen yields a model with an overall good score on the Switchboard test set, which we consider as our final checkpoint (named DACL-Best) for the next set of experiments.

Throughout the process, we can see that the ROUGE scores are consistently increasing with the subsequent CL stages and are considerably lower in the non-CL and skip-CL studies. The final DACL-best model has the highest ROUGE scores among all the models in Table 2. This indicates that the CL process is instrumental in increasing the quality of the outputs generated by the model.

5.1.2. Fine-Tuning and Evaluation (Table 4)

We observe in the first row that only using T5-base as is (no CL and no fine-tuning) does not yield good disfluency removal overall, with precision at 17.74 and recall at 49.25; the recall is still relatively high due to the generation of irrelevant output tokens and null strings frequently.

Table 6: Comparison with Previous Work

Method	Word-based		
	P	R	F
DACL+FT	**97.1**	84.7	90.5
DACL+FT (Overfit)	96.1	90.2	93.0
EGBC (Bach and Huang, 2019)	95.9	86.3	90.9
EGBC + residual (Bach and Huang, 2019)	96.1	86.9	91.2
Self-Trained BERT-Based Parser (ensemble) (Jamshid Lou and Johnson, 2020b)	92.5	**97.2**	**94.8**
Self-Trained BERT-Based Parser (single) (Jamshid Lou and Johnson, 2020b)	92.2	96.6	94.3
Noisy BiLSTM (Bach and Huang, 2019)	94.7	89.8	92.2
Weight sharing (Wang et al., 2018)	92.1	90.2	91.1
BiLSTM (Zayats et al., 2016)	91.6	80.3	85.9
Semi-CRF (Zayats et al., 2016)	90.0	81.2	85.4

In the second row, we can observe the impact of directly fine-tuning the T5-base model on the Switchboard dataset. The precision increases to 93.57 and the recall increases to 83.66, however, we observe that the issues with the generation of irrelevant output and null strings still remain here.

In the fourth row, the model achieved the best validation loss in 14 epochs of fine-tuning the DACL-Best model on the Switchboard training set. This approach yielded the highest precision scores observed in our study – 97.10 – and also the highest amongst all the previous approaches on the Switchboard test set as shown in Table 6, denoted as **DACL+FT**.

In the fifth row, we overfit the model (to 66 epochs) on the Switchboard training dataset, denoted in Table 6 as **DACL+FT (Overfit)**. This yielded the highest recall, F1, and ROUGE scores in Table 4 on the Switchboard test set. This model shows the overall best results among all the previous approaches on the Switchboard test set, as shown in **Table 6**. However, there is a decrease in precision to 96.10. This signifies that as the model was overfitting on the training set it started to identify more

Table 7: Examples from Switchboard test set (Mitchell et al., 1999). Blue underlines indicate true positives. Red underlines indicate false negatives. Cyan underlines indicate false positives. **Bold words indicate words that are considered disfluent according to the Switchboard dataset, however, we observe that contextually they are not disfluencies**. **'...'** indicates phrases that were exactly the same in input, output, and reference sentences.

Type	Sentence
Input Text 1	"I mean I've I've I I have ... months of you know reasonably satisfactory use they will accept it in any condition and they will gladly **no questions asked** take it back if for some bizarre reason ... "
Reference Text 1	"I have ... months of reasonably satisfactory use they will accept it in any condition and they will gladly take it back if for some bizarre reason ... "
Generated Text 1	"I have ... months of reasonably satisfactory use they will accept it in any condition and they will gladly **no questions asked** take it back if for some bizarre reason ..."
Input Text 2	"yeah well **I think** I guess nowadays with the uh with the economy the way it is I guess there was a there was a story ..."
Reference Text 2	"nowadays with the economy the way it is I guess there was a story ..."
Generated Text 2	"**I think** nowadays with the economy the way it is I guess there was a story ..."
Input Text 3	"**hi** uh uh I I should say something "
Reference Text 3	"I should say something"
Generated Text 3	"**hi** I should say something"

words and phrases as disfluencies at the cost of precision. However, we should note that the reference texts were human-annotated, thus implying that what might appear disfluent to the annotators might not appear disfluent to others in some contexts. Since the best model judiciously selects its disfluent candidates based on the context of the sentence, it has the highest precision. Such examples where some words that can be argued to not be proper disfluencies based on the context of the sentence are presented in **Table 7**.

5.2. RQ2: Results on WikiSplit Dataset (Out-of-Domain)

The WikiSplit dataset is out-of-domain for the Switchboard dataset, as Wikipedia is written text, and Switchboard is transcribed spoken text. We measure the word-based PRF scores and the ROUGE scores on the test set of the Switchboard dataset after executing DACL on the WikiSplit dataset in Table 3. Then, we fine-tune and evaluate our model on the Switchboard dataset in Table 5. Finally, we compare our model to the state-of-the-art models in Table 6.

5.2.1. DACL (Table 3)

Table 3 seems to follow the general trends of Table 2 where the ROUGE and word-level precision, recall, and F1 scores increase with incremental stages of the CL process. The difference is:

(1) The precision scores in Table 3 are lower than the precision scores in Table 2 for the same stages of DACL;

(2) The recall scores in Table 3 are higher than the recall scores in Table 2 for the same stages of DACL.

This difference can be attributed to the fact that WikiSplit is a written dataset whereas Spotify is a spoken transcribed dataset. The presence of existing speech disfluencies (i.e. aside from the generated disfluencies) in the Spotify dataset adds some noise to the entire training process and also instructs the model to better identify disfluencies from the sentences and leave the rest unaltered.

5.2.2. Fine-Tuning and Evaluation (Table 5)

In Table 5, we observe in the fourth row that DACL + fine-tuning yields the best precision and recall scores — as well as the best scores across the table. However, both of these scores (P=95.13, R=87.00) are lower than the best precision and recall scores obtained from Table 4 (P=97.10, R=90.25).

6. Conclusion

Disfluencies are difficult to distinguish from fluent speech. In this work, we propose a curriculum learning-based model that can eliminate disfluencies and provide fluent text with high word-based precision along with robust word-based recall and F1 scores. This would allow the model to be more accurate in distinguishing between disfluent and fluent sentences. We build a synthetic disfluency augmentation approach that produces progressive versions of the original text with various degrees of disfluencies. We combine this with CL to provide a novel training approach for training sequence-to-sequence generation models for producing fluent text from disfluent texts. Our evaluations and ablation studies indicate the efficacy of our approach, and our best model surpasses state-of-the-art methods in word-based

precision and displays good word-based recall and F1 scores on the widely used Switchboard test set. In the future, we plan to study other kinds of disfluencies present in speech. Creating synthetic augmentations with other kinds of disfluencies would allow us to create a more robust model. We also plan to extend this method to other languages and aim to see if the model trained on the English language corpus fares well on other language disfluencies with some amount of fine-tuning.

7. Ethical Considerations

There are a few important ethical considerations that must be made when removing disfluencies from text:
(i) *Authenticity and Accuracy:* Removing or altering speech disfluencies might affect the message's authenticity. Changing the natural form of communication may inadvertently misrepresent what the speaker means or feels. Transparency should be maintained if disfluencies are eliminated, particularly in public or official communications. To prevent deceiving the general public, any changes to the original speech should be made clear.
(ii) *Inclusivity*: Adjusting disfluencies may sideline or stereotype people who naturally display more disfluencies in their speech, like those with speech conditions or non-native speakers. It may enforce a standard of "fluent" speech and disregard diverse communication styles.
(iii) *Bias*: Perceptions of what qualifies as a "disfluency" can be subjective and culturally based. Systems that remove disfluencies may inadvertently sustain linguistic biases, advocating a particular type of speech as more desirable. Additionally, culture and language can impact speech patterns, including disfluencies. Removing these features may accidentally diminish cultural or linguistic identity.

8. Limitations

Our method has a few key limitations to note:
(i) *Preservation of Meaning*: Disfluencies can carry significant meaning and express nuance, hesitation, or emphasis. Automated or manual removal of disfluencies may inadvertently alter the speaker's intended meaning or emotional expression.
(ii) *Accessibility:* Some individuals might find pauses or repeated words helpful to process speech, particularly those with particular cognitive or learning needs.
(iii) *Comprehending Nuances:* Automated systems might lack the insight to understand the reasons for speech disfluencies, and taking them out might remove layers of meaningful communication.

9. Bibliographical References

American Speech-Language-Hearing Association. 2023. Fluency disorders (practice portal). Accessed: 2023-10-19.

Nguyen Bach and Fei Huang. 2019. Noisy bilstm-based models for disfluency detection. In *Interspeech*, pages 4230–4234.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *International Conference on Machine Learning*, pages 41–48.

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

Angelica Chen, Vicky Zayats, Daniel Walker, and Dirk Padfield. 2022. Teaching BERT to wait: Balancing accuracy and latency for streaming disfluency detection. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 827–838.

Eunah Cho, Jan Niehues, and Alex Waibel. 2014. Tight integration of speech disfluency removal into SMT. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 43–47.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

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

Google Cloud. Speech-To-Text.

Hany Hassan, Lee Schwartz, Dilek Hakkani-Tür, and Gokhan Tur. 2014. Segmentation and disfluency removal for conversational speech translation. In *Interspeech*, pages 318–322.

Paria Jamshid Lou, Peter Anderson, and Mark Johnson. 2018. Disfluency detection using auto-correlational neural networks. In *Conference on Empirical Methods in Natural Language Processing*, pages 4610–4619.

Paria Jamshid Lou and Mark Johnson. 2020a. End-to-end speech recognition and disfluency removal. In *Findings of the Association for Computational Linguistics: Conference on Empirical Methods in Natural Language Processing 2020*, pages 2051–2061.

Paria Jamshid Lou and Mark Johnson. 2020b. Improving disfluency detection by self-training a self-attentive model. In *Association for Computational Linguistics*, pages 3754–3763.

Paria Jamshid Lou, Yufei Wang, and Mark Johnson. 2019. Neural constituency parsing of speech transcripts. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2756–2765.

Mark Johnson and Eugene Charniak. 2004. A tag-based noisy-channel model of speech repairs. In *Association for Computational Linguistics*, pages 33–39.

Douglas A Jones, Florian Wolf, Edward Gibson, Elliott Williams, Evelina Fedorenko, Douglas A Reynolds, and Marc A Zissman. 2003. Measuring the readability of automatic speech-to-text transcripts. In *Interspeech*.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Association for Computational Linguistics*, pages 2676–2686.

Hao Lang and Wen Wang. 2019. Automated curriculum learning for turn-level spoken language understanding with weak supervision. *arXiv preprint arXiv:1906.04291*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics.

OpenAI. GPT-4 Technical Report.

Tatiana Passali, Thanassis Mavropoulos, Grigorios Tsoumakas, Georgios Meditskos, and Stefanos Vrochidis. 2022. LARD: Large-scale artificial disfluency generation. In *Language Resources and Evaluation Conference*, pages 2327–2336. European Language Resources Association.

Gustavo Penha and Claudia Hauff. 2020. Curriculum learning strategies for IR: An empirical study on conversation response ranking. In *Advances in Information Retrieval: European Conference on IR Research*, pages 699–713.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237. Association for Computational Linguistics.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabás Poczós, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1162–1172.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

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*.

Justyna Sarzynska-Wawer, Aleksander Wawer, Aleksandra Pawlak, Julia Szymanowska, Izabela Stefaniak, Michal Jarkiewicz, and Lukasz Okruszek. 2021. Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, 304:114135.

Lin Shi and Bei Peng. 2023. Curriculum learning for relative overgeneralization. In *Adaptive and Learning Agents Workshop at Conference on Autonomous Agents and Multiagent Systems*.

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

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565.

Maria Teleki, Xiangjue Dong, and James Caverlee. 2024. Quantifying the impact of disfluency on spoken content summarization. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*.

Xin Tian, Bei Fang, Juhou He, and Xiuqing He. 2022. Bi-LSTM-attention Based on ACNN Model for Disfluency Detection. In *Journal of*

Physics Conference Series, volume 2303 of *Journal of Physics Conference Series*, page 012018.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Feng Wang, Wei Chen, Zhen Yang, Qianqian Dong, Shuang Xu, and Bo Xu. 2018. Semi-supervised disfluency detection. In *International Conference on Computational Linguistics*, pages 3529–3538.

Peiying Wang, Chaoqun Duan, Meng Chen, and Xiaodong He. 2023. Improving disfluency detection with multi-scale self attention and contrastive learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–5.

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

Shaolei Wang, Wanxiang Che, Yue Zhang, Meishan Zhang, and Ting Liu. 2017. Transition-based disfluency detection using LSTMs. In *Conference on Empirical Methods in Natural Language Processing*, pages 2785–2794.

Shaolei Wang, Zhongyuan Wang, Wanxiang Che, and Ting Liu. 2020b. Combining self-training and self-supervised learning for unsupervised disfluency detection. In *Conference on Empirical Methods in Natural Language Processing*, pages 1813–1822.

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.

Xin Wang, Yudong Chen, and Wenwu Zhu. 2021. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576.

Vicky Zayats, Mari Ostendorf, and Hannaneh Hajishirzi. 2016. Disfluency detection using a bidirectional LSTM. In *Interspeech*.

Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. 2019. Curriculum learning for domain adaptation in neural machine translation. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1903–1915.

Qingqing Zhu, Xiuying Chen, Pengfei Wu, JunFei Liu, and Dongyan Zhao. 2021. Combining curriculum learning and knowledge distillation for dialogue generation. In *Findings of the Association for Computational Linguistics: Conference on Empirical Methods in Natural Language Processing 2021*, pages 1284–1295.

10. Language Resource References

Botha, Jan A and Faruqui, Manaal and Alex, John and Baldridge, Jason and Das, Dipanjan. 2018. *Learning To Split and Rephrase From Wikipedia Edit History*.

Clifton, Ann and Reddy, Sravana and Yu, Yongze and Pappu, Aasish and Rezapour, Rezvaneh and Bonab, Hamed and Eskevich, Maria and Jones, Gareth and Karlgren, Jussi and Carterette, Ben and Jones, Rosie. 2020. *100,000 Podcasts: A Spoken English Document Corpus*. International Committee on Computational Linguistics. [link].

Godfrey, John J and Holliman, Edward. 1997. *Switchboard-1 Release 2*.

Mitchell, Marcus and Santorini, Beatrice and Marcinkiewicz, M and Taylor, Ann. 1999. *Treebank-3 ldc99t42 web download*.