Zero-shot Disfluency Detection for Indian Languages

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

Abstract

Disfluencies that appear in the transcriptions from automatic speech recognition systems tend to impair the performance of downstream NLP tasks. Disfluency correction models can help alleviate this problem. However, the unavailability of labeled data in low-resource languages impairs progress. We propose using a pretrained multilingual model, finetuned only on English disfluencies, for zero-shot disfluency detection in Indian languages. We present a detailed pipeline to synthetically generate disfluent text and create evaluation datasets for four Indian languages: Bengali, Hindi, Malayalam, and Marathi. Even in the zero-shot setting, we obtain F1 scores of 75 and higher on five disfluency types across all four languages. We also show the utility of synthetically generated disfluencies by evaluating on real disfluent text in Bengali, Hindi, and Marathi. Finetuning the multilingual model on additional synthetic Hindi disfluent text nearly doubles the number of exact matches and yields a 20-point boost in F1 scores when evaluated on real Hindi disfluent text, compared to training with only English disfluent text.

1 Introduction

Disfluencies (e.g., *filled pauses*, *repetitions*, *discourse markers*) are artefacts that are inherent to spontaneous or conversational speech. Disfluencies typically obey the following surface structure comprising: a *reparandum*, an *interruption point* (+) that marks the end of the reparandum, an *interregnum*, and finally the *repair* (Shriberg, 1994). The *reparandum* consists of one or more words that are not intended by the speaker and will be replaced or ignored. The *interregnum* consists of an editing term indicating that the *reparandum* will be edited, or it may be empty, or it can contain *fillers*, *discourse markers*, etc. The *repair* section reflects the fluent part of the utterance. Words from the *reparandum* are repeated or corrected in the *repair* section, or a new chain of thought is started in case of a *false start*.

Consider the following example that illustrates two disfluency types:

{well} [i think + {you know} i think] the idea will work

The words highlighted in red and green refer to *discourse marker* and *repetition* disfluency types, respectively. The part in blue is the fluent version of the original sentence. The example also follows the standard annotation scheme:

$$[reparandum + \{interregnum\}\ repair]$$

Disfluencies in automatically transcribed text pose a major challenge for downstream NLP tasks such as machine translation, summarization, etc. (Rao et al., 2007; Wang et al., 2010). Disfluency detection/correction is often used as a preprocessing step for NLP, where the goal is to identify/remove the disfluent words (Shriberg et al., 1992). While disfluency correction has been extensively studied for English (Honal and Schultz, 2003; Zayats et al., 2014), it has received far less attention in other languages. This is largely due to the lack of labeled data for other languages.

In this work, our main objective is to build disfluency detection models for four Indian languages — Bengali, Hindi, Malayalam, and Marathi — in the zero-shot setting with no access to labeled disfluent data in these languages. To the best of our knowledge, this is the very first study of disfluency detection across multiple Indian languages and also the very first to investigate the ability of large pretrained models to do zero-shot disfluency detection.

We specify a rule-based procedure to generate disfluencies starting from fluent sentences. It is worth noting that the synthetically generated disfluent data might not completely reflect real world disfluencies. Nevertheless, we find the synthetic data to be useful in improving disfluency detection

Proceedings of the 29th International Conference on Computational Linguistics, pages 4442–4454
October 12–17, 2022.

for low-resource Indian languages. Also, we can create near-real disfluent data by manually editing the synthetic data, which will take significantly less time than annotating from scratch. Our rule-based pipeline is targeted at Indian languages, and the same set of rules is applied to sentences in four Indian languages — Bengali, Hindi, Malayalam, and Marathi. From within these synthetic datasets, native speakers of the respective languages manually identified disfluent sentences that seemed natural. This resulted in manually-verified evaluation datasets for all the four languages. We also constructed evaluation datasets for Bengali, Hindi, and Marathi with real[1] disfluent sentences, transcribed and extracted from conversational speech in interviews. Using these datasets as evaluation benchmarks and inspired by prior work on cross-lingual zero-shot transfer using large pretrained multilingual models (Pires et al., 2019; Hu et al., 2020; Khanuja et al., 2021), we investigated the effectiveness of a large pretrained multilingual model MuRIL (Khanuja et al., 2021) on the task of zero-shot disfluency detection. MuRIL is a multilingual transformer-based model that is pretrained on large amounts of text in a number of different Indian languages. We finetuned MuRIL using labeled disfluent sentences in English (Godfrey et al., 1992) (and synthetic disfluent text) and evaluated disfluency correction for all four Indian languages in the zero-shot setting.[2]

Four MuRIL-based disfluency detection models were trained, viz. those using (1) only real English disfluent data, (2) both real English and synthetically generated Bengali disfluent data, (3) both real English and synthetically generated Hindi disfluent data, and (4) both real English and synthetically generated Marathi disfluent data. Model (2) significantly improves disfluency detection on the real Bengali disfluent data compared to model (1). We also observe similar results for the other two languages. This validates our claim that our synthetically generated data is effective in capturing (some subset of) the kinds of disfluencies that are encountered in real conversational data.

The main ideas in this work can be summarized as follows:

- We outline a common rule-based procedure that allows us to synthesize disfluencies for four Indian languages: Bengali, Hindi, Malayalam, and Marathi.
- We construct manually-verified evaluation datasets for all four Indian languages, starting from synthetically generated data. The Bengali, Hindi, Malayalam, and Marathi test sets contain 500, 575, 575, and 420 sentences, respectively.
- We also annotate real labeled disfluent datasets in Bengali, Hindi, and Marathi, containing 300, 150, and 250 sentences, respectively. These sentences were transcribed and extracted from real conversational speech. We note that this annotation process is substantially more tedious than identifying natural disfluencies starting from our synthetic data.
- We finetune a pretrained multilingual model, MuRIL, on labeled disfluent data in English and show its effectiveness at zero-shot disfluency detection for all four Indian language datasets.
- We show the utility of our synthetic disfluency generation pipeline by comparing performance of a model finetuned only on real English disfluent data versus a model finetuned on both real English and synthetically generated disfluent data of one Indian language.
- We present a detailed breakdown of performance across various disfluency types, show qualitative analyses of our model predictions and highlight some interesting aspects related to disfluencies in Indian languages (e.g., *reduplication*).

2 Related Work

There are three main categories of approaches for disfluency detection. They are based on (1) sequence tagging, (2) parsing, and (3) a noisy channel model (Kundu et al., 2022).

Sequence tagging based approaches use classification techniques to label individual words (Liu et al., 2006; Ostendorf and Hahn, 2013; Zayats et al., 2014; Ferguson et al., 2015; Hough and Schlangen, 2015; Zayats et al., 2016; Wang et al.,

[1] We use the term *real* throughout the paper to contrast the manually-edited synthetic datasets with the (real) datasets containing disfluencies annotated from conversations.

[2] Our code and datasets can be found at `https://github.com/RKKUNDU/zero-shot-disfluency-detection`.

2018). Parsing-based approaches detect disfluencies along with identifying the syntactic structure of the sentence (Rasooli and Tetreault, 2013; Honnibal and Johnson, 2014; Wu et al., 2015; Yoshikawa et al., 2016; Jamshid Lou and Johnson, 2020). The main idea behind a noisy channel model of disfluency is that we assume there is a fluent source sentence X to which some noise has been added, resulting in a disfluent sentence Y. The goal is to find the most likely fluent sentence given Y (Johnson and Charniak, 2004; Zwarts and Johnson, 2011; Jamshid Lou and Johnson, 2017).

Prior works (Hu et al., 2020; Khanuja et al., 2021) have used pretrained multilingual models for many zero-shot NLP tasks such as Named Entity Recognition (NER), Part of Speech (POS) tagging, Question Answering (QA), etc. However, this is the first work to attempt disfluency detection in a zero-shot setting and the very first work to study disfluency detection for multiple Indian languages.

Our work on synthetic disfluency data generation has parallels to the recent work of Passali et al. (2022) where they focus on an artificial disfluency generation algorithm. They focus broadly on *Repetitions*, *Replacements*, and *Restarts* and only focus on English. Saini et al. (2020) is another prior work that has looked into inducing disfluencies in English fluent text. Our disfluencies are much more fine-grained in construction (e.g., pronoun corrections, missing syllables, etc.) compared to prior work and apply to Indian languages.

3 Generating Synthetic Disfluencies

We focus on four major disfluency types as listed in Honal and Schultz (2003), i.e., *Fillers*, *Repetitions*[3], *Corrections*, and *False Starts*. We specify a total of nine rules across the four disfluency types to introduce disfluencies in fluent sentences. We show examples of Bengali disfluent sentences in Appendix A for all the disfluency types. Apart from what we describe in this section, there are some more fine-grained details governing how and where various disfluencies are introduced in a fluent sentence; these details are specified in our released codebase.

[3]In the literature, *Repetitions* and *Corrections* disfluency categories are considered as a single category

3.1 Fillers

We loosely use the term *Fillers* to denote *editing terms*, *discourse markers*, *filled pauses* and *interjections*. *Editing terms* are used to explicitly indicate that the previously uttered word(s) were not intended. *Discourse markers* help in beginning or keeping a turn (e.g., well) or merely serve as a form of acknowledgment (e.g., yeah). *Filled pauses* are non-lexicalized sounds without any semantic content. *Interjections* are defined as non-lexicalized sounds indicating affirmation or negation.

We simply introduce frequent *filler* phrases at randomly chosen positions. We choose frequent *filler* phrases after carefully observing conversations. We assume that there will be at most 3 *fillers* in a sentence and uniformly choose a number between 1 and 3. Thereafter, with uniform probability, we pick the location in the sentence at which the next *filler* will be inserted and also choose the *filler* phrase to be inserted with uniform probability from a pool of *filler* phrases.[4]

Speakers might tend to use *fillers* before long words. For words with 12 or more characters, we first choose a *filler* phrase with uniform probability and then place it before the long word.

3.2 Repetitions

Repetition is defined as the phenomenon of speakers repeating a word or phrase.

Word Repetition. For this rule, we pick a word uniformly at random and repeat it.

Phrase Repetition. In this rule, we repeat a phrase[5] containing 2 to 5 words. We first randomly pick a length from $[2, 3, 4, 5]$ using a weighted distribution of $[0.4, 0.3, 0.2, 0.1]$.[6] Then, we pick a phrase of the chosen length uniformly at random and repeat it.

Pronoun Repetition. We find pronouns to be commonly repeated in Indian languages. First, we accumulate a list of pronouns for each language. If any word in the fluent sentence appears in the pronoun list, then we repeat the pronoun with a predetermined probability.[7]

[4]These *filler* phrases are separately listed for all four Indian languages after consulting native speakers.

[5]Here, we mean an n-gram of consecutive words regardless of their real phrasal structure.

[6]This distribution was chosen only to signify that phrases of shorter length are more frequent than phrases of longer length.

[7]More details about the probability with which a pronoun is chosen to be repeated is specified in our code.

3.3 Corrections

Corrections involve substitutions, deletions, or insertions of words from the *reparandum* section. *Corrections* may include the *interregnum*.

Partial Word. For this rule, we introduce *partial words* before long words with 12 or more characters. Firstly, we find the orthographic syllables[8] of a long word using the Indic NLP Library[9] (Kunchukuttan, 2020). Thereafter, we create the *partial word* by joining the first n syllables where n comes from a weighted distribution in which probability $P(n)$ is proportional to $\frac{1}{n}$.

Missing Syllables. For this rule, preceding a long word of 12 or more characters, we insert the same word but with one or more syllables missing. We first find the orthographic syllables of the long word. Then, we remove n contiguous syllables from the word (where n is sampled from a weighted distribution similar to what we used for *phrase repetition)* and add this reduced form of the word prior to the original long word.

Pronoun Correction. In this rule, a pronoun gets explicitly corrected. From the pronoun lists mentioned in Section 3.2, we create groups of similar types of pronouns (e.g., all first person pronouns are in one group). For each pronoun in the fluent sentence, we find its group and pick a different pronoun from the group to serve as its correction. We also (optionally) insert a frequent filler phrase before using the correct pronoun.

Synonym Correction. In this rule, we introduce a synonym of the word before the actual word, obtained using IndoWordNet[10] (Bhattacharyya, 2010).

3.4 False Start

For the *False Start* disfluency, a sentence is aborted before it is completed, and a new idea or line of thought is introduced. To create *false starts*, we first randomly pick two different fluent sentences. Then, we split the first fluent sentence from a random position and we concatenate the first part of the split with the second fluent sentence.

[8]Orthographic syllable is a sequence of one or more consonants followed by a vowel.

[9]https://github.com/anoopkunchukuttan/indic_nlp_library

[10]https://www.cfilt.iitb.ac.in/indowordnet/

4 Dataset Details

4.1 Disfluency Datasets for Indian Languages

Real Disfluent Data. We create *real disfluent datasets* in Bengali, Hindi, and Marathi by transcribing and annotating real disfluencies from conversations in the respective languages. For this purpose, we used publicly available *Interviews* in Bengali[11], Hindi[12], and Marathi[13] from YouTube[14]. From these videos, we constructed three datasets containing 300, 150, and 250 disfluent and fluent parallel sentences in Bengali, Hindi, and Marathi, respectively.

Synthetic Disfluent Data. We also induce disfluencies in fluent text using our rule-based algorithm and create evaluation datasets for disfluency detection in Bengali, Hindi, Malayalam and Marathi. We start with fluent monolingual text from the PMIndia corpus[15] (Haddow and Kirefu, 2020). We synthesize disfluent sentences using the rules outlined in Section 3. We ask language specialists in each of the four languages to manually pick sentences from the synthetic dataset that appear like natural disfluencies (and edit the disfluent sentences if needed). We picked utterances such that there is uniform coverage across disfluency types and there is no label imbalance. We used IndicNLP (Kunchukuttan, 2020) for normalization and tokenization, and we removed all punctuation marks. Table 1 shows detailed disfluency type counts for all four datasets. The test sets for Bengali, Hindi, Malayalam, and Marathi contain 500, 575, 575, and 420 sentences, respectively.

Each of the test sets is grouped into five categories: *fillers*, *repetitions*, *corrections*, *false starts* and fluent sentences. Fluent sentences are included as a control set to check whether the model is incorrectly detecting disfluencies in fluent sentences. We also include fluent sentences with *reduplications* which are a special category in Indian languages as mentioned below.

Reduplication. *Reduplication* is the act of repeating all or part of a word for emphasis or to

[11]https://www.youtube.com/c/WBCSMadeEasyTM

[12]https://www.youtube.com/NeeleshMisraChannel

[13]https://www.youtube.com/c/GopalDarji

[14]We obtained explicit permission from the Bengali and Marathi content creators to use the data for research. The Hindi content creator allows *fair use* of specific videos for research.

[15]https://data.statmt.org/pmindia/

Type	Bn	Hi	Ml	Mr
Filler (3.1)	50	100	100	70
Word Repetition (3.2)	42	50	50	35
Phrase Repetition (3.2)	42	50	50	35
Pronoun Repetition (3.2)	41	50	50	35
Partial Word (3.3)	66	50	50	35
Missing Syllables (3.3)	34	50	50	35
Pronoun Correction (3.3)	66	50	50	35
Synonym Correction (3.3)	34	50	50	35
False Start (3.4)	50	50	50	35
Fluent Sentences with Redpl	25	25	25	20
Normal Fluent Sentences	50	50	50	50
Total	500	575	575	420

Table 1: Synthetic Dataset Statistics: Number of sentences of each disfluency type. Redpl: Reduplication, Bn: Bengali, Hi: Hindi, Ml: Malayalam, Mr: Marathi.

convey a meaning. It is widely used in Indian languages; a few examples of reduplication in Hindi are shown in Table 2 (Montaut, 2009). In the context of disfluencies, we note that *reduplications* could be mistaken for a *repetition* disfluency type. *Reduplications* are intentional repetitions which are grammatically correct and should not be flagged as disfluencies. To check for this, we include fluent sentences with *reduplication* in our test set.

4.2 English Disfluency Data

We also present results for English to check how well MuRIL performs when compared with previously published results. Switchboard[16] (Godfrey et al., 1992) in English is the most commonly used dataset for disfluency detection. Following the experimental settings in Wang et al. (2021), we split the Switchboard corpus such that the dev set consists of all sw_04[5-9]*.utt files, the test set consists of all sw_04[0-1]*.utt files, and the training set consists of all the remaining files. We do not include sentences without disfluencies in the training data, but do so in the dev, test set. Following Honnibal and Johnson (2014), we lowercase the text and remove all punctuation marks.

5 Experimental Setup

In this work, we use MuRIL (Khanuja et al., 2021) which is a BERT model (Devlin et al., 2019) pretrained on 16 Indian languages (including the four we consider) and English. MuRIL is pretrained using two language modeling objectives: Masked Language Modeling and Translation Language Modeling.

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

[17] Orange color denotes *reduplication*

Evaluation Metrics. We test the model on Bengali, Hindi, Malayalam, Marathi and English disfluency detection tasks. Similar to prior work on detecting English disfluencies (Wang et al., 2021), we compute precision, recall, and F1 scores using word-level labels. We also use a more ambitious metric, the *exact match percentage*, where the predicted fluent sentence is compared to the reference fluent sentence and checked for an exact match. We also show BLEU scores between the fluent text predictions and the reference fluent sentences, which are calculated using *sacreBLEU* (Post, 2018).

5.1 Using only English Disfluency Data

We finetune the pretrained MuRIL checkpoint on the English disfluency detection task where the goal is to correctly label each of the tokens as *fluent* or *disfluent*. We use the *muril-base-cased* checkpoint (having 236M parameters) from *HuggingFace*[18] for all our experiments. For each of the subword tokens identified by the MuRIL tokenizer, the model predicts its label as being 0 (*fluent*) or 1 (*disfluent*).

For disfluency correction, once we have disfluency labels for each subword, we use majority voting to determine whether a word is omitted or not. For a word, if the number of its subwords tagged as disfluent is greater than the number of subwords tagged as fluent, the word is deleted; else, it is retained.

5.2 Using Synthetically Generated Indian Language Data along with English Disfluency Data

We would like to check whether our synthetically generated data in one Indian language (say, Bengali) helps improve performance on real disfluencies in Bengali, Hindi, and Marathi languages.

We generate disfluent sentences having n disfluencies where $n \in \{1, 2, 3, 4, 5\}$. Ignoring false starts, we pick one of the eight disfluency types (listed in Table 1) at random and inject disfluencies in fluent Bengali sentences from the PMIndia corpus. We generated 42500 disfluent Bengali sentences in this way, which is roughly half the number of sentences in the Switchboard corpus.[19] Sim-

[18] https://huggingface.co/google/muril-base-cased

[19] We want to augment the Switchboard data with the synthetic Bengali data, but do not want the synthetic data to dominate the corpus.

Sentence	**Transliteration**	**Gloss**	**Translation**	**Comment**
तुम कहा कहा[17] गए	tuma kahaa kahaa gae	you where where went	where did you go	Reduplication of interrogative pronoun. Here the questioner expects a list of places in response.
खाते खाते मत बोलो	khaate khaate mata bolo	eating eating do_not speak	do not speak while eating	Reduplication of verb
यह लो तुम्हारी चाय. गरम गरम है, पियो	yaha lo tumhaarii caaya. garama garama hai, piyo	this take your tea. hot hot is drink	Take your tea. It is nicely hot, drink it	Reduplication of adjective
बच्चो को एक एक टॉफ़ी दो	bacco ko eka eka taffii do	children to one one toffee give	give a toffee to each child	Reduplication of number.

Table 2: Examples of Reduplication in Hindi. Gloss: word-to-word English translation. "_" in the gloss suggests *fertility* which refers to one word mapping to multiple words in the other language.

ilarly, we construct synthetic data for the other two languages as well.

Next, we finetune the pretrained MuRIL checkpoint on the combined synthetic Bengali data and Switchboard data. The other experimental details are the same as described in Section 5.1. We evaluate this model on real disfluency detection data in Bengali, Hindi, and Marathi. We hypothesize that the performance of this model on the real Bengali disfluency detection dataset will be better than that of the model finetuned only on English data. This would indicate that our synthetically generated data contains disfluencies that mimic the ones seen in real speech.

5.3 Using Only Synthetically Generated Indian Language Data

We also finetune the pretrained MuRIL checkpoint only on the synthetic Bengali/Hindi/Marathi data. We apply the same synthetic data and experimental setup as discussed in Section 5.2.

6 Results & Analysis

This section presents the evaluation results and analyses the quantitative and qualitative performance of our models.

6.1 Performance on Real Disfluent Data

Table 3 shows a comparison of our model finetuned on only English data, and models finetuned on synthetic Bengali/Hindi/Marathi disfluent data (optionally) along with real English disfluent data.

We can see that *MuRIL - En & Syn Bn* (model finetuned using both synthetically generated Bengali data and the Switchboard corpus) outperforms *MuRIL - En* (finetuned only using the Switchboard corpus) by a significant margin of 19% in terms of exact matches, when evaluated on real Bengali disfluencies. Also, *MuRIL - En & Syn Bn* has high precision which leads to an increase of 4.67 F1 scores. Similarly, *MuRIL - En & Syn Hi* model outperforms *MuRIL - En* by a large margin of 19.92 F1 scores, when evaluated on real Hindi disfluencies. Both *MuRIL - En & Syn Bn* and *MuRIL - En & Syn Mr* also outperform *MuRIL - En* by 15.51 and 18.14 F1 scores, respectively. We observe similar trends on the real Marathi evaluation set as well.

All the models that were finetuned with additional synthetic data (irrespective of the language) nearly double the number of exact matches when evaluated on real Hindi/Marathi disfluencies, compared to the model trained with only English disfluent data. Our models trained only on synthetic disfluent data outperform the *MuRIL - En* model. We observe 1.17, 12.33, 17.48 F1 scores improvement over *MuRIL - En* model, when evaluated on real Bengali, Hindi and Marathi evaluation sets, respectively.

These results suggest that using synthetically generated disfluent sentences does enable transfer to real disfluent data and helps validate our synthetic data generation pipeline.

6.2 Disfluency Detection in Indian Languages

Table 4 presents a detailed account of the performance of MuRIL on the four manually-edited synthetic disfluency evaluation sets in Bengali, Hindi, Malayalam, and Marathi. It also provides a breakdown of performance across disfluency types. We report the exact match percentages, the BLEU scores and the F1 scores.

The overall F1 scores for Bengali, Hindi, Malayalam and Marathi are 73.14, 63.82, 67.12 and

Language	Model	Precision	Recall	F1 score	Exact Match %	BLEU
Bengali	MuRIL - En	81.13	65.60	72.54	28.33	80.2
	MuRIL - Syn Bn	91.79	61.58	73.71	43.00	81.2
	MuRIL - En & Syn Bn	**92.90**	**66.06**	**77.21**	**47.33**	**82.6**
	MuRIL - En & Syn Hi	89.90	51.03	65.11	35.67	77.4
	MuRIL - En & Syn Mr	92.72	51.15	65.93	35.33	76.7
Hindi	MuRIL - En	67.81	62.20	64.89	36.67	85.5
	MuRIL - Syn Hi	83.18	72.05	77.22	54.00	89.5
	MuRIL - En & Syn Bn	82.57	78.35	80.40	64.00	91.3
	MuRIL - En & Syn Hi	84.98	**84.65**	**84.81**	**66.00**	**93.7**
	MuRIL - En & Syn Mr	**86.38**	79.92	83.03	**66.00**	91.7
Marathi	MuRIL - En	57.78	54.93	56.32	26.40	83.8
	MuRIL - Syn Mr	**92.25**	61.50	73.80	55.60	88.2
	MuRIL - En & Syn Bn	82.77	68.78	75.13	56.80	88.9
	MuRIL - En & Syn Hi	83.14	68.31	75.00	55.60	89.5
	MuRIL - En & Syn Mr	87.54	**69.25**	**77.33**	**60.80**	**90.0**

Table 3: Performance on real Bengali, Hindi, and Marathi disfluent data. *MuRIL - En*: Finetuning MuRIL only on Switchboard data, *MuRIL - En & Syn X* (where $X \in \{Bn, Hi, Mr\}$): Finetuning MuRIL on Switchboard data and synthetic disfluency detection data in language X [Bn: Bengali, Hi: Hindi, Mr: Marathi] , *MuRIL - Syn X* (where $X \in \{Bn, Hi, Mr\}$): Finetuning MuRIL only on synthetic disfluent data in language X . We note that the BLEU scores between the original disfluent text and the fluent reference text for Bengali, Hindi, and Marathi are 62.0, 71.9 and 73.2, respectively.

70.22, respectively. Interestingly, the model is able to do a reasonable job of disfluency detection even in the zero-shot setting with no access to labeled disfluent data in the target languages. The BLEU scores between the original disfluent text and the fluent reference for Bengali, Hindi, Malayalam and Marathi are 83.2, 83.4, 78.3, 81.1, respectively. Comparing these scores to the BLEU scores obtained using the finetuned MuRIL (92.5, 90.9, 88.4 and 91.3) clearly shows that the model is effective in removing disfluencies.

6.3 Performance across Disfluency Types

Table 4 shows that our model is doing exceptionally well at detecting *Repetitions* in all the languages and our model shows the best performance in detecting *Phrase Repetitions*. In 64% of the Bengali sentences, our model did not tag a *reduplication* as disfluency, which is correct. This suggests that the model is learning the difference between *word repetition* and *reduplication*. Also, most of the fluent sentences are kept unchanged.

Even without any explicit supervision of the filler words specific to each language, we find that our model is sometimes able to accurately detect the *fillers* based on the context in which they appear. The high F1 score in Marathi for *fillers* (compared to the other three languages) can be attributed to the fact that the *fillers* exhibited a positional bias and mostly appeared at the start of the Marathi test sentences.

Our model performs fairly on detecting *partial words*. In comparison, the model does very well on detecting *missing syllables* and achieves more than 75 F1 scores in all four languages. Despite the complexity of the *pronoun correction* task, which could also involve optional editing terms, our model performs admirably and gets F1 scores of greater than 75 across all languages.

Detecting *synonym correction* correctly is a complex task and our model does not perform too well on this disfluency type. Our model yields the lowest F1 scores across all disfluency types on *false starts*. This is not very surprising because false starts are the hardest of disfluency types to detect (Shriberg, 1994). Sometimes there are ambiguities even in the gold standard utterances containing *false starts*. (Example 7 in Table 8 shows such an ambiguity.) Another reason could be that the Switchboard dataset does not contain many *false start* disfluencies.

6.4 Performance across Languages

From Table 4, we compare the performance of our model across languages for different disfluency types.

Filler detection is done best for Marathi. We observe that the model does not perform well in detecting Hindi *fillers*. Hindi differs from the other languages in that it uses fairly long phrases as *fillers*. For example, "क्या कहते है" means "what to say". Thus, our model might find it challenging to catch these long *filler* phrases. We also observe that the model's capability to detect a *filler*

Type	Bengali			Hindi			Malayalam			Marathi		
	M	B	F1	M	B	F1	M	B	F1	M	B	F1
Filler (3.1)	34.00	86.74	53.66	23.00	82.05	28.45	19.00	78.63	37.24	61.43	90.13	77.24
Word Repetition (3.2)	78.57	98.12	90.53	66.00	96.39	79.67	82.00	98.40	92.91	82.86	97.61	92.50
Phrase Repetition (3.2)	80.95	98.49	96.42	70.00	96.91	93.84	86.00	98.46	96.75	62.86	95.52	91.57
Pronoun Repetition (3.2)	73.17	97.27	86.04	74.00	95.32	84.75	86.00	98.10	92.31	74.29	97.58	88.61
Partial Word (3.3)	45.45	90.31	66.10	38.00	91.00	54.55	52.00	88.59	70.00	42.86	89.88	60.71
Missing Syllables (3.3)	70.59	96.20	83.08	82.00	98.39	89.80	62.00	92.16	76.77	62.86	95.46	82.19
Pronoun Correction (3.3)	59.09	89.49	82.11	54.00	90.00	78.54	60.00	90.08	85.71	40.00	87.11	75.74
Synonym Correction (3.3)	29.41	86.76	48.98	18.00	77.66	54.13	18.00	73.17	45.84	14.29	67.18	41.38
False Start (3.4)	20.00	75.50	37.64	8.00	80.29	27.23	2.00	70.51	15.56	17.14	79.04	35.39
Fluent Sentences with Reduplication	64.00	96.01		64.00	97.31		60.00	94.59		35.00	89.45	
Normal Fluent Sentences	90.00	98.46		98.00	99.89		96.00	99.35		94.00	99.45	
Overall	57.60	92.52	73.14	50.96	90.87	63.82	53.22	88.39	67.12	56.19	91.31	70.22

Table 4: Performance on manually-edited synthetic Bengali, Hindi, Malayalam, and Marathi disfluency datasets. We use *MuRIL - En* model for the evaluation. M: Exact Match Percentage, B: BLEU Score, F1: F1 Score. We note here that the BLEU scores between the original disfluent text and the fluent reference text for Bengali, Hindi, Malayalam and Marathi are 83.2, 83.4, 78.3, 81.1, respectively.

word varies depending on the position of the *filler*. When *hesitations* like “अं”, “आं” are present at the beginning of the sentence, our model always detects them, but not necessarily when they are present in the middle.

Our model performs the best on Hindi when it comes to detecting *missing syllables*. F1 scores when detecting *repetitions* and *corrections* are comparable across languages. However, we see a huge gap across languages in the case of *false starts*. For *false starts*, the difference in performance between the Bengali and Malayalam test sets is 22.08 (in terms of F1 scores). This large difference in performance could be attributed to the following reasons:

- The number of words present in *false starts* follows different distributions for Bengali and Malayalam. In Bengali, there are fewer occurrences of *false starts* containing many words, while there are more long *false start* occurrences in Malayalam.
- The model detects Bengali *conjunctions* like “কারণ” (meaning “because”), “কিন্তু” (meaning “but”), “আর” (meaning “and”) etc. as disfluencies which are part of the *false start*.
- At times, the model detects part of the *false start* as being part of some other disfluency type, such as a *repetition* or a *correction*.

6.5 Ablation Study

Table 3 shows that combining synthetic Marathi disfluent data with English disfluent data increases F1 scores by 21 points on real Marathi disfluent data compared to a model trained solely on English data. Via an ablation study, we aim to check which subset of disfluency types in the synthetic dataset is most helpful. We consider seven combinations of disfluency types (including *MuRIL - En*[20] and *MuRIL - En & Syn Mr*[21]) that we think are representative. We generate the same amount of synthetic data for all the combinations using the steps discussed in Section 5.2.

We present the results in Table 5. It is encouraging to see that all the models achieve a precision of nearly 90 when trained on synthetic data whereas *Disf-onlyEng* achieves a precision of only 57.78. In comparison to *Disf-onlyEng*, *Disf-1* (synthetic data containing only *fillers*) has a lower F1 score, which can be attributed to the very low *recall* value. *Disf-1234* improves F1 score by 8.40, which suggests that *repetitions* help. It is interesting to see that *Disf-15678* improves F1 score by 20.09. This implies that *corrections* by themselves are of sig-

[20] *MuRIL - En* and *Disf-onlyEng* are the same. This model is only trained on English disfluent data

[21] *MuRIL - En & Syn Mr* and *Disf-all* are the same. This model is trained on English data, and synthetic Marathi data comprising all the disfluency types

Model Name	Precision	Recall	F1 score	Exact Match %	BLEU
Disf-onlyEng	57.78	54.93	56.32	26.40	83.8
Disf-1	90.97	30.75	45.96	23.60	79.9
Disf-1234	**93.36**	49.53	64.72	45.60	84.8
Disf-15678	89.06	66.90	76.41	56.80	89.4
Disf-137	89.78	47.42	62.06	38.80	83.6
Disf-1357	89.56	62.44	73.58	54.00	88.2
Disf-all	87.54	**69.25**	**77.33**	**60.80**	**90.0**

Table 5: Performance of models trained on English and additional synthetic Marathi disfluent data containing a subset of disfluency types (except *Disf-onlyEng* which is trained only on English data). The numbers in the model name indicate which disfluency types were present in the synthetic data during finetuning (e.g., *Disf-137* was trained on synthetic Marathi disfluent data containing only *filler*, *phrase repetition*, and *pronoun correction*). Mapping of number to disfluency types — 1: *filler*, 2: *word repetition*, 3: *phrase repetition*, 4: *pronoun repetition*, 5: *partial word*, 6: *missing syllables*, 7: *pronoun correction*, 8: *synonym correction*. We note here that *MuRIL - En* is the same as *Disf-onlyEng* and *MuRIL - En & Syn Mr* is the same as *Disf-all*.

nificant help. We also observe that only *phrase repetition* and *pronoun correction* do not help much as *Disf-137* achieves an F1 score of only 62.06. However, when we also add *partial words*, *Disf-1357* overshoots *Disf-137* by a margin of 11.52 F1.

6.6 English Disfluency Detection

In Table 6, we present the performance of our multilingual model MuRIL on the test set of the Switchboard corpus as a sanity check of our model. We achieved an F1 score of 93.62 on the sequence labeling task, whereas the state-of-the-art model in Wang et al. (2021) reported an F1 score of 91.7. (We note that Wang et al. (2021) preprocesses *"uh"*, *"um"*, *"I mean"*, etc. tokens differently; hence, our F1 scores are not directly comparable.)

Precision	Recall	F1 Score	Accuracy
95.22	92.08	93.62	98.00

Table 6: Performance on Switchboard disfluency detection test set.

7 Conclusion

We propose the use of a pretrained multilingual model MuRIL for zero-shot disfluency detection in Indian languages. We evaluate our model on Bengali, English, Hindi, Malayalam, and Marathi disfluency detection tasks. We also show that synthetically generated Bengali/Hindi/Marathi disfluency detection data using simple rules, when combined with real English disfluency data during finetuning, helps improve F1 scores on real Bengali/Hindi/Marathi disfluencies. Our overall results support the claim that it is possible to do cross-lingual transfer of disfluency detection without any labeled data in the target languages. For future work, we intend to evaluate the model on more diverse disfluencies.

Acknowledgements

The authors thank the anonymous reviewers for their constructive suggestions that helped improve the draft. 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.

References

Pushpak Bhattacharyya. 2010. IndoWordNet. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

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

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.

John J. Godfrey, Edward C. Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Proceedings of the 1992 IEEE International Conference on*

Acoustics, Speech and Signal Processing - Volume 1, ICASSP'92, page 517–520, USA. IEEE Computer Society.

Barry Haddow and Faheem Kirefu. 2020. PMIndia–A Collection of Parallel Corpora of Languages of India. *arXiv preprint arXiv:2001.09907*.

Matthias Honal and Tanja Schultz. 2003. Correction of disfluencies in spontaneous speech using a noisy-channel approach. In *Interspeech*. Citeseer.

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

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

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

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 of the Association for Computational Linguistics (ACL-04)*, pages 33–39, Barcelona, Spain.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. MuRIL: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Anoop Kunchukuttan. 2020. The IndicNLP Library. `https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf`.

Rohit Kundu, Preethi Jyothi, and Pushpak Bhattacharyya. 2022. Survey: Exploring disfluencies for speech-to-speech machine translation.

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

Annie Montaut. 2009. Reduplication and echo words in Hindi/Urdu. *Annual review of South Asian languages and linguistics*, pages 21–91.

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

Tatiana Passali, Thanassis Mavropoulos, Grigorios Tsoumakas, Georgios Meditskos, and Stefanos Vrochidis. 2022. Lard: Large-scale artificial disfluency generation. *arXiv preprint arXiv:2201.05041*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

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

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

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

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

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. IEEE.

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. In *Proc. Interspeech 2016*, pages 2523–2527.

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

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

A Examples: Synthetic Disfluency Generation Rules

Table 7 presents a brief overview and example of each type of disfluencies covered in Section 3.

[22] Purple color denotes disfluencies

B Analysis: Inconsistency in Predicted Tags of the Subwords of a Word

Since our model works at the subword level, it could be possible that the subwords of a particular word get different tags (*disfluent*/*fluent*) as the prediction. We call this *inconsistency in tagging*.

According to our findings, only 0.02 % of subwords in the Switchboard test set were marked inconsistently. In our Bengali, Hindi, Malayalam, and Marathi test sets, we found inconsistency in just 0.21 %, 0.06 %, 0.38 %, and 0.03 % subwords. These findings indicate that our model is quite likely to consistently predict the tags across the subwords for any word.

C Qualitative Analysis

We analyze the performance of our base model (trained only on English data) on relatively difficult disfluencies, namely, *false starts* and *corrections*. We also analyse one example of a fluent sentence and we present these examples of potential interest in Table 8.

[23] Cyan color denotes that it is unclear whether the words are intended in the sentence

Type	Description	Example	Transliteration	Gloss	Translation
Filler (3.1)	Insert frequent *filler* phrases in a sentence	সমীক্ষায় দেখা যায়, এই মানে[22] তৃতীয় এবং চতুর্থ শ্রেণী থেকেই অধিকাংশ ছাত্রী স্কুলছুট হয়।	samIkShAYa dekhA YAYa, ei mAne tRRitIYa evaM chaturtha shreNI thekei adhikAMsha ChAtrI skulaChuTa haYa	in_survey see can this means third and fourth grade since most girl_student drop_out_of_school is	According to the survey, this means most girl students drop out of school from third and fourth grade.
Word Repetition (3.2)	Repeat a word unnecessarily	এখনও এখনও হয়তো অনেককে বাইরে থেকে জল আনতে যেতে হয়।	ekhanao ekhanao haYato anekake vAire theke jala Anate Yete haYa	still still maybe many outside from water to_bring to_go is	Many may still still have to fetch water from outside.
Phrase Repetition (3.2)	Repeat a phrase unnecessarily	অর্থাৎ, হতাশার আবহেও বাঁচার আশা জাগানোর সামর্থ্য এই সমাজের এই সমাজের রয়েছে।	arthAt, hatAshAra Avaheo vA.NchAra AshA jAgAnora sAmarthYa ei samAjera ei samAjera raYaeChe	in_other_words, despair in_condition to_live hope to_awaken ability this of_society this of_society there_are	In other words, this society this society has the ability to inspire hope to survive in the face of despair.
Pronoun Repetition (3.2)	Repeat a pronoun unnecessarily	আমরা আমরা রোয়াডাবাসীকে এক্ষেত্রে নানাভাবে সাহায্য করতে পারি।	AmarA AmarA roYaAn.DAvAsIke ekShetre nAnAbhAve sAhAYYa karate pAri	We we to_Rwandans in_this_case many_ways help to_do can	We we can help Rwandans in many ways.
Partial Word (3.3)	Use *part of word* before the actual word	এর ফলে, নয়ডা অঞ্চলে আগামী বছরগুলিতে জনসংখ্যাও উল্লেখ উল্লেখযোগ্যভাবে বৃদ্ধি পাবে।	era phale, naYa.DA anchale AgAmI vaCharagulite janasaMkhYAo ullekha ullekhaYogYabhAve vRRiddhi pAve	for_this as_a_result Noida in_region next over_the_years population_too PARTIAL_WORD significantly increase will_get	As a result, the population of the Noida region will also increase signi **significantly** in the coming years.
Missing Syllables (3.3)	Missed a few syllables from the middle of a word; therefore, it is followed by the entire word.	আর্টিফিশিয়াল ইন্টেলিন্সের ইন্টেলিজেন্সের সূত্র ধরেই বহু সমস্যারই সমাধান আমরা করতে পারি	ArTiphishiYaAla inTelinsera inTelijensera sUtra dharei vahu samasYArai samAdhAna AmarA karate pAri	Artificial of_intellince of_intelligence formula taking many problems solution we to_do can	We can solve many problems with the help of Artificial Intellince **Intelligence**.
Pronoun Correction (3.3)	Use an incorrect pronoun, then an optional *edit* phrase, then the proper pronoun.	কচ্ছে এদেরকে না মানে একে ভুঙ্গা বলে।	kachChe ederake nA mAne eke bhu NgA vale	in_Kachchh to_them no mean to_this bhunga called	They no I mean it is called bhunga in Kachchh.
Synonym Correction (3.3)	Use of imprecise synonym before the actual word	কোন্ প্রকল্প কবে শেষ হবে, সেই সময় নির্দেশিত নির্দিষ্ট করে_দেওয়া_হয়েছে।	kon prakalpa kave sheSha have, sei samaYa nirdeshita nirdiShTa kare_deoYaA_haYaeChe	which project when complete will_be, that time directed specified has_been_done	When the project will be completed, the time has been directed specified.
False Start (3.4)	Begin a sentence, then abruptly end it and begin a new sentence.	ইতিমধ্যেই রাজ্যে আজ দেশ উন্নয়নের নতুন উচ্চতা অতিক্রম করছে।	itimadhYei rAjYe Aja desha unnaYanera natuna uchchatA atikrama karaChe	already in_the_state today country of_development new height exceed doing	Already in the state Today, the country is exceeding new heights of development.

Table 7: Different types of disfluencies in our synthetic dataset

Disfluent Sentence	Transliteration	Gloss	Translation	Model Output	Comment
তোকে এটা ভাবলে তুই এটা ভাববি যে চার বছর ধরে অল্প অল্প করে কাজ করব	BT: toke eTA bhAvale tui eTA bhAvavi Ye chAra vaChara dhare alpa alpa kare kAja karava	BG: to_you it if_think you it should_think that four year for little little by work will_do	E: If you think about it you would think that I would work little by little for four years	তুই এটা ভাববি যে চার বছর ধরে অল্প করে কাজ করব	Our model is able to detect the *correction* but considers *reduplication* ("অল্প অল্প") as a disfluency. Thus the model removes one "অল্প".
তুই যেটা পড়ছিস তুই যেটা পড়লি সেদিনে সেটাকে সুন্দর করে লেখার চেষ্টা করবি	BT: tui YeTA parChisa tui YeTA parli sedine seTAke sundara kare lekhAra cheShTA karavi	BG: you what reading you what read on_that_day that beautiful by writing try you_should	E: Try to write beautifully what you are reading you read on that day	তুই যেটা পড়লি সেদিনে সেটাকে সুন্দর করে লেখার চেষ্টা করবি	Our model is able to detect the *correction*.
তুমি কি তাহলে ওই কাজটা শেষও হয়ে গেল	BT: tumi ki tAhale oi kAjaTA sheShao haye gela	BG: you what then that work end become gone	E: Are you then that work is over	ওই কাজটা শেষও হয়ে গেল	The sentence starts with a thought ("তুমি কি তাহলে" means "Are you then") and suddenly a new chain of thought is initiated ("ওই কাজটা শেষও হয়ে গেল" means "that work is over"). Our model is able to detect the *false start*.
আসলে আমি না আমি একদমই বুঝে পাচ্ছি না কী করা উচিত	BT: Asale Ami nA Ami ekadamai vujhe pAchChi nA kI karA uchita	BG: Actually I no I absolutely understand getting no what do should	E: Actually I no I have no idea what to do	আসলে আমি একদমই বুঝে পাচ্ছি না কী করা উচিত	The disfluent sentence contains *pronoun repetition* along with an *interregnum*. Our model is able to detect both.
मेरा नहीं मतलब हमारा अतीत अंदरूनी_तौर_पर हमेशा संयमपूर्वक बुना_गया है	HT: merA nahIM matalaba hamArA atIta aMdarUnI_taura_para hameshA saMyamapUrvaka bunA_gayA hai	HG: my no mean our past internally always abstemiously woven is	E: My no I mean our past has always been abstemiously woven inward	हमारा अतीत अंदरूनी तौर पर हमेशा संयमपूर्वक बुना गया है	Our model detects the wrong pronoun along with the *edit* term.
देशातील प्रत्येक शहरात प्रत्येक गावात ही स्वच्छता मोहीम सुरू आहे	MrT: deshAtIla pratyeka shaharAta pratyeka gAvAta hI svachChatA mohIma surU Ahe	MrG: of_the_country each in_city each in_village this cleanliness campaign going_on is	E: This cleaning campaign is going on in every city and every village of the country	देशातील प्रत्येक गावात ही स्वच्छता मोहीम सुरू आहे	We test our model on a fluent sentence. The fluent sentence contains the phrase "प्रत्येक शहरात प्रत्येक गावात" which has two components: "प्रत्येक शहरात" means "in each city" and "प्रत्येक गावात" means "in each village". Our model wrongly assumes that the first component is *corrected* by the second component and hence the model labels the first component as a disfluency.
ওঁরা ছয় মাস ধরে কোনও কোম্পানিতে[23] অ্যাপেল-এর সিইও টিম কুক বলেছেন এই সিদ্ধান্তের পরিণাম সুদূরপ্রসারী হবে	BT: o.NrA Chaya mAsa dhare kono kompAnite aYApela-era siio Tima kuka valeChena ei siddhAntera pariNAma sudUraprasArI have	BG: they six months for any in_company Apple's CEO Tim Cook said this of_decision consequences far-reaching will_be	E: They for six months at a company Apple's CEO Tim Cook said the decision would have far-reaching consequences	ছয় মাস ধরে কোনো কোম্পানিতে অ্যাপেল-এর সিইও টিম কুক বলেছেন এই সিদ্ধান্তের পরিণাম সুদূরপ্রসারী হবে	The disfluent sentence contains a false start "ওঁরা ছয় মাস ধরে কোনো কোম্পানিতে", which makes it ambiguous to find the intended meaning of the utterance. There could be at least three interpretations, which makes the task challenging: • "*At a company* Apple's CEO Tim Cook *has said for six months* that the decision would have far-reaching consequences" • "*At a company* Apple's CEO Tim Cook said the decision would have far-reaching consequences" • "Apple's CEO Tim Cook said the decision would have far-reaching consequences" Our model picks up the first interpretation.
মানে প্রপারলি যদি ইমপ্যাক্টটা আনতে হয় তাহলে এখন থেকে যদি ইমপ্যাক্ট আনতে হয় তাহলে আমাকে সিভিল সার্ভিসে যেতে হবে	BT: maane prapaarali yadi imapyaak.ta.taa aanate haya□ taahale ekhana theke yadi imapyaak.ta aanate haya□ taahale aamaake sibhila saarbhise yete have	BG: mean properly if the_impact to_bring is then now from if impact to_bring is then me civil to_service to_go will_be	E: I mean, if I have to get the impact properly, if I have to get impact from now on, I have to go to the civil service.	প্রপারলি এখন থেকে যদি ইমপ্যাক্ট আনতে হয় তাহলে আমাকে সিভিল সার্ভিসে যেতে হবে	The disfluent sentence has *code-mixing*. English words "properly", "impact", "civil", "service" are present in the Bengali sentence. This disfluent sentence contains *correction* disfluency type. Our model is able to detect all the disfluencies correctly.

Table 8: Qualitative analysis of our model predictions. BT: Bengali Transliteration, BG: Bengali Gloss, HT: Hindi Transliteration, HG: Hindi Gloss, MrT: Marathi Transliteration, MrG: Marathi Gloss, E: English Translation.