Generating Fluent Translations from Disfluent Text Without Access to Fluent References: IIT Bombay@IWSLT2020

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

Abstract

Machine translation systems perform reasonably well when the input is well-formed speech or text. Conversational speech is spontaneous and inherently consists of many disfluencies. Producing fluent translations of disfluent source text would typically require parallel disfluent to fluent training data. However, fluent translations of spontaneous speech are an additional resource that is tedious to obtain. This work describes the submission of IIT Bombay to the Conversational Speech Translation challenge at IWSLT 2020. We specifically tackle the problem of disfluency removal in disfluent-to-fluent text-to-text translation assuming no access to fluent references during training. Common patterns of disfluency are extracted from disfluent references and a noise induction model is used to simulate them starting from a clean monolingual corpus. This synthetically constructed dataset is then considered as a proxy for labeled data during training. We also make use of additional fluent text in the target language to help generate fluent translations. This work uses no fluent references during training and beats a baseline model by a margin of 4.21 and 3.11 BLEU points where the baseline uses disfluent and fluent references, respectively.

Index Terms- disfluency removal, machine translation, noise induction, leveraging monolingual data, denoising for disfluency removal.

1 Introduction and Related Work

Spoken language translation often suffers due to the presence of disfluencies. In conversational speech, speakers often use disfluencies such as filler words, repetitions of fillers, repetitions of fluent phrases, false starts, and corrections which do not occur in the text. Standard machine translation and spoken translation systems perform competitively when the input is well-formed text or rehearsed speech as in TED talks or broadcast news (Cho et al., 2014; Wang et al., 2010; Honal and Schultz, 2005; Zayats et al., 2016). With the increasing popularity of end-to-end speech translation systems (Weiss et al., 2017; Bansal et al., 2018), one may not want disfluency removal to be treated as an intermediate step between ASR and MT. It might be more desirable for disfluency removal to be handled within the model itself, or as a separate post-processing step.

To produce fluent translations from disfluent text, one would typically require access to disfluent speech (or text) and its corresponding fluent translations during training. While some corpora with labeled disfluencies exist (Cho et al., 2014; Burger et al., 2002), only subsets have been translated and/or released. (Salesky et al., 2018) introduced a set of fluent references for the Fisher Spanish-English conversational speech corpus (David Graff and Cieri.). This has enabled a new task of end-to-end training and evaluation on fluent references. (Salesky et al., 2019) reports results using a speech-to-text model trained on this corpus using both fluent and disfluent translations. However, fluent translations of disfluent speech or text are a scarce resource. It would be highly desirable to build a system for disfluency removal that does not rely on fluent references.

In this work, we propose a framework for disfluency removal that utilizes a simple noise induction technique for data augmentation using fluent monolingual text in the target language. During denoising, such disfluent text is trained jointly with parallel disfluent-to-disfluent textual translation data, thus simultaneously optimizing the objectives of disfluency removal and translation. This work describes the submission of IIT Bombay to the Conversational Speech Translation challenge at IWSLT 2020 (Ansari et al., 2020). We release code for our

Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT), pages 178–186
July 9-10, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

proposed approach at the following URL.[1]

Section 2 describes the details of the data used and the proposed noise induction technique to leverage monolingual data. Section 3 describes our proposed architecture. We then present experimental details in Section 4, followed by our main results in Section 4.2. Finally, we analyze our model outputs in Section 5 and present our conclusions in Section 6.

2 Data

2.1 Fisher Corpus

For our experiments, we use the Fisher Spanish dataset (David Graff and Cieri.), comprising telephone conversations between mostly native Spanish speakers. The dataset contains speech utterances (disfluent Spanish), their corresponding ASR outputs (disfluent Spanish), and two sets of English translations (both fluent and disfluent) (Salesky et al., 2018; Post et al., 2013). The Fisher dataset has disfluent Spanish ASR output *(text)* which we use as input to our model. Additionally, two sets of English translations *(disfluent & fluent)* are also available in *(text)* form. For training, we only make use of disfluent English sentences. i.e. we train a *text-to-text* model. We explicitly note here that no fluent English reference text was used during training. The corpus consists of 819 transcribed conversations on predetermined topics between strangers, yielding ≈ 160 hours of speech and 150k utterances. We used one reference during training and evaluation with the validation (dev) and test data sets.[2]

2.1.1 Disfluencies

Disfluencies can be filler words and hesitations, discourse markers *(you know, well, umm)*, phrase repetitions, filler word repetitions, corrections, and false starts, among others. There can be different and often overlapping disfluencies in a single sentence. Fluent words like *so, oh, yes, no, etc.* could either be categorized as fluent or disfluent depending on the context in which they appear. We selected the most commonly occurring filler words in English, namely *hmm, hm, em, eh, uh, um, umm, ah, aha, mm, oh, wow, yes, ok* from the Fisher English corpus. These filler words either occur alone as a single unit or with self-repetitions up to a maximal length of 5 or 6. We extracted the frequencies of each one of them and their repetitions. Table 1 shows the counts for the *aha* token along with its successive repetitions. We repeat this for all filler words and store them in a comma-separated value file.

Filler phrase	Frequency
ah	9572
ah ah	233
ah ah ah	29
ah ah ah ah	5
ah ah ah ah ah	1
ah ah ah ah ah ah	0

Table 1: Filler phrase frequencies in the Fisher English training corpus.

2.2 Parallel Corpus for Translation

We extract monolingual fluent textual data from the news-commentary parallel corpus in Spanish-English from the shared task on machine translation in 2013.[3] The corpus consists of 174,441 parallel sentences. We divide the dataset into two halves. We consider the first half of 87220 sentences to be our fluent monolingual corpus. The other half wasn't used to account for resource constraints. Future work can incorporate the whole corpus for training. This corpus is modified and turned into a parallel disfluent to fluent corpus in the same language i.e. EN-disfluent to EN-fluent. This process is described in more detail in the next section.

2.3 Data Augmentation via Noise Induction

Most disfluent sentences could be loosely thought of as a composition of a fluent part and an additive noise characterizing the underlying disfluency type. We aimed to generate a parallel EN-disfluent to EN-fluent dataset, starting with fluent English text and adding disfluencies that we extract from real disfluent text. We stress here that we do not make use of any parallel disfluent-fluent text to extract patterns of disfluencies; the latter was generated by solely examining disfluent text. Three levels of disfluency induction have been implemented where disfluencies are incrementally added. We have tested with 10%, 30%, 50% disfluency induction. Section 2.3.1, 2.3.2, 2.3.3, 2.3.4 will describe the techniques used to introduce disfluencies within a fluent corpus to create a parallel corpus.

[1] https://github.com/niksarrow/cst

[2] We do not make use of dev2 during training.

[3] https://www.statmt.org/wmt13/translation-task.html

2.3.1 Pronoun Phrase Repetition

The English language has seven pronouns, namely, *"i, we, you, he, she, it, they"*. In the conversational speech, many times an utterance that starts with a pronoun repeats itself. Here is an example:

i am i am fond of paintings ...
it is cold it is cold and windy outside ...

Our algorithm iterates through all 87220 sentences in the English news-commentary corpus and treats every sentence which starts with a pronoun as a candidate. With hyperparameter value $\alpha = 0.1, 0.3, 0.5$, we either select or reject the candidate for disfluency induction. Here, α is the probability of selecting the candidate and $1 - \alpha$ is the rejection probability. If a candidate is selected, we select the length (l) of the phrase starting from the first word (which is the pronoun itself) which will be repeated. The length is uniformly sampled from four length values i.e. $1, 2, 3, 4$. The phrase up to length (l) is repeated in the sentence just after it ends. The following examples show how disfluencies are introduced for two different values of l:

Original fluent sentence: *i was saying that we should go for a movie*
Disfluent sentence ($l = 1$): *i i was saying that we should go for a movie*
Disfluent sentence ($l = 2$): *i was i was saying that we should go for a movie*

2.3.2 Fluent Phrase Repetition

Many disfluencies are just repetitions of meaningful phrases where the speaker intentionally or unintentionally repeats a phrase. We iterate through all the sentences and every sentence with length greater than 5 becomes a candidate with hyperparameter $\alpha = 0.1, 0.3, 0.5$ (as we did with pronoun phrase repetition). We randomly selected a length(l) in the range $[1, 3]$ and carefully selected an index i in the fluent sentence starting from which a phrase of length l is repeated and a disfluent sentence is formed. Here is an example:

Original fluent sentence: *easier to trade and speculate in gold*
Disfluent sentence ($l = 1$**,** $i = 5$): *easier to trade and speculate in in gold*
Disfluent sentence ($l = 2$**,** $i = 3$): *easier to trade and speculate and speculate in gold*

2.3.3 Insertion of filler words/phrases

The filler word/phrase frequency count which is described in Section 2.1.1 and table 1 is used as a guide to introduce them within clean text. We iterate through all sentences in the English corpus and uniformly select a (phrase, frequency) pair such that the frequency is greater than 0. If the sentence becomes a candidate according to the sampling probability $\alpha = 0.1, 0.3, 0.5$ (as we did with pronoun phrase repetition), an index i is uniformly selected from the range $[0, l]$, and the phrase is inserted at index i along with a decrement of one in the frequency of the phrase. As before, l is the length of the candidate. Example:

Before: (filler, frequency) = (*ah ah, 233*)
Original sentence: *the new year is looking grim*
Disfluent sentence ($i = 0$): *ah ah the new year is looking grim*
After: (filler, frequency) = (*ah ah, 232*)

2.3.4 False Start

In disfluent English, an utterance can start with an affirmation (i.e. beginning with a *yes* or *yeah*) and suddenly turn into negation or denial. For example: *yes, no we can't increase the price*. Here, the speaker first uttered *yes* and then shifted to negation with *no*. Similarly, a negative utterance can suddenly shift to an affirmative one.

We iterate through all the sentences in the Fisher English corpus which begins with an affirmation *yes, yeah* or a negation *no, nah* and prepend *yes or no* of length($l = 1 or 2$) to make it a false starting sentence (if the sentence is chosen with a sampling probability of $\alpha = 0.1, 0.3, 0.5$). An example:

Original sentence: *yes the price will go up*
Disfluent sentence ($l = 2$): *no no yes the price will go up*

2.3.5 Other Possible Noise Induction Techniques

Synonym insertion can be done by examining the synset of the language in question (say English), picking a word from a candidate sentence, and attaching its synonym next to it. A denoising step is expected to retain only one of the meanings which is fluent as per the language model. We also introduce singleton utterances which only contain a single filler word and its corresponding fluent version is labeled as *None*. We leave the exploration of more techniques that are relevant to introducing disfluencies within fluent text as future work.

2.4 Data Statistics

Table 2 shows utterance counts from the parallel disfluent Spanish to disfluent English corpus. Apart from this dataset, we also make use of three parallel disfluent-to-fluent English texts which were synthetically created using the techniques described in Section 2.3, corresponding to α values of 0.1, 0.3 and 0.5, respectively. Each of these three parallel datasets contains 87220 sentences each.

	Fisher Data	
Train	138720 (DFLT)	138720 (DFLT)
Validation	3977 (DFLT)	3977 (FLT)
Test	3641 (DFLT)	3641 (FLT)

Table 2: DFLT: Disfluent and FLT: Fluent. Disfluent Spanish source and disfluent English target utterances in training. For validation and test set evaluations, we use fluent translations. Numbers indicate the count of utterances in the train, validation and test sets, respectively.

3 Model

This section describes the proposed architecture for disfluency removal and translation. Since our objective is two-fold, which is disfluency removal and translation, Section 3.2.1 first presents our denoising module which is aimed at achieving the task of disfluency removal and Section 3.2.2 describes how translation is achieved by our model.

3.1 System Architecture

As shown in Figure 1, the proposed system follows a fairly standard encoder-decoder architecture. More concretely, we use a four-layer transformer encoder and another four-layer transformer decoder (Vaswani et al., 2017). There are 8 attention heads in both the encoder and decoder. We pretrain joint token embeddings of 512 dimensionalities on concatenated Fisher Spanish (disfluent) and English (disfluent) and News-commentary English data using fastext (Bojanowski et al., 2016). We use byte-pair encoding (Sennrich et al., 2015) with 50K BPE units to effectively handle out-of-vocabulary words at test time. We share language embeddings in the encoder. We set dropout (Gal and Ghahramani, 2016) and label-smoothing (Szegedy et al., 2016) to 0.3 and 0.1, respectively. In addition to the disfluency induction, we use word-shuffle, word-dropout, and word-blank with probabilities $3, 0.1, 0.2$ (Lample et al., 2017) respectively when training the denoising encoder.

3.1.1 Shared Encoder

Our system makes use of two encoders with three out of four layers shared by the two input languages. This is inspired from (Artetxe et al., 2017; Lample et al., 2017). The first three layers are shared across both tasks i.e. denoising from disfluent English to fluent English and translating from Spanish to English. The fourth layer is language-dependent to allow the encoder to learn language-specific information. The shared layers of the encoder encourage the output representations of Spanish and English to use a common subspace shared across both languages, which is further transformed into fluent English using a common decoder. In this manner, our model jointly achieves disfluency removal along with translation.

3.1.2 Fixed Language Embeddings in the Encoder

While many machine translation systems randomly initialize their embeddings and update them during training, we use pretrained sub-word level embeddings and keep them fixed during training (Artetxe et al., 2017). We share the vocabulary for Spanish and English as their alphabet size is 27 and 26 respectively. The additional letter in Spanish is a *ñ* to indicate the palatal nasal; the remaining letters are the same as in English.

3.2 Training

The encoder & decoder are trained using Adam (Kingma and Ba, 2015) with a learning rate of 0.001 and a mini-batch size of 32. The training alternates evenly between denoising and translation procedures.

3.2.1 Denoising

C(x) is a randomly sampled noisy version of sentence x similar to the noise model by (Lample et al., 2017). The denoising done here is a mixture of standard denoising as done in (Lample et al., 2017) and a supervised training step using the parallel disfluent-fluent English text that we created using the techniques described in Section 2.3. The loss for the latter training phase is the sum of cross-entropy losses between pred_{EN} and its fluent counterpart.

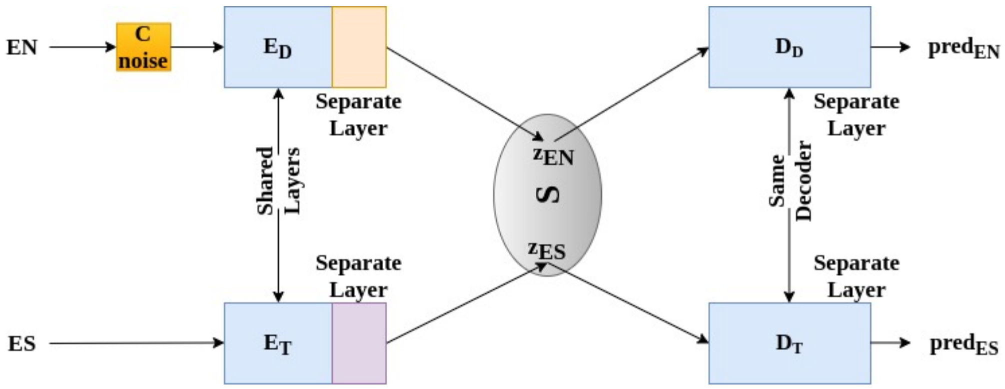


Figure 1: Illustration of proposed architecture. EN: noise-augmented input in English language (text), ES: disfluent Spanish input (text), E_D: Denoising encoder for English language, D_D: Denoising decoder for English language, E_T: Translation encoder whose input is Spanish, D_T: Translation decoder whose output is English, pred_{EN}: fluent output of denoising decoder, pred_{EN}: Translated output of translation decoder in English language. C_{noise} is the noise model used in (Lample et al., 2017) i.e. word-dropout, word-shuffle, S: Shared latent space, z_{EN} and z_{ES}: latent representation of top and bottom encoders, respectively.

3.2.2 Translation

The translation is done using the parallel Fisher dataset, where disfluent Spanish is used as input and disfluent English is generated as output. The sum of token level cross-entropy is used as the loss function between pred_{EN} (predicted English output) and reference disfluent English.

4 Experiments

4.1 Experimental Setup

We have used lowercased, tokenized, normalized data with all punctuations (except apostrophe) removed. This is the same setting as used by (Salesky et al., 2019) allowing for a comparison with the baseline proposed. The system is evaluated using BLEU[4] and METEOR[5] scoring metrics. BLEU assesses how well predicted translations match a set of reference translations using modified n-gram precision, weighted by a brevity penalty in place of recall to penalize short hypothesis translations without full coverage. In our task of disfluency removal, the generated tokens should contain much of the same content but with certain tokens removed, thereby creating shorter hypotheses. When scoring *fluent output* with *disfluent references*, the difference in BLEU score will come from two sources: shorter n-gram matches, and the brevity penalty. METEOR, on the other hand, is a semantic evaluation metric. It uses the harmonic mean of precision and recall, with more weight assigned to recall. It also takes into account stem, synonym, paraphrase, and exact matches. In our task, semantic meaning should be retained while disfluencies are removed. Similar METEOR scores are expected when scored with fluent references and disfluent references. METEOR will indicate that meaning is maintained, but not assess disfluency removal, while BLEU will indicate whether disfluencies have been removed.

The parallel Fisher data remains constant in all settings. We have tested with three increasing levels of disfluency induction in the synthetic data. This is denoted using three different values, 0.1, 0.3, and 0.5, for the hyperparameter α. We use a batch size of 32 and epoch size 50000. All other hyperparameters are similar to (Lample et al., 2017)'s implementation.

4.2 Results and Discussion

Table 3 compares the baseline BLEU scores of (Salesky et al., 2019) with our implementation. Our proposed model operates in a mismatched setting i.e. training using disfluent-to-disfluent text data, and evaluating on fluent references for the validation and test sets. We show two baseline scores in Table 3. "BaselineD" refers to the use of disfluent reference text during training and "BaselineF" refers to the use of fluent reference text during training. It should be noted that the reported baseline from (Salesky et al., 2019) is a speech-to-text

[4] BLEU scores computed using multi-bleu.pl from the Moses toolkit (Koehn et al., 2007).

[5] METEOR is computed using the script from http://www.cs.cmu.edu/~alavie/METEOR/ (Denkowski and Lavie, 2014).

Model	α	**dev**		**test**	
		1Ref	**2Ref**	**1Ref**	**2Ref**
BaselineD	-	13	16.2	13.5	17.0
BaselineF	-	14.6	18.1	14.6	18.1
Our Impl.D	0.5	**17.27**	**17.54**	17.36	20.47
Our Impl.D	0.3	17.2	17.46	**17.71**	**20.93**
Our Impl.D	0.1	16.96	17.22	17.08	20.15

Table 3: BLEU on development and test set with single vs multiple references. End-to-end model performance evaluated with new fluent references. D: Disfluent reference, F: Fluent reference as used in training. Our implementation is trained using disfluent references only.

Model	α	**dev**		**test**	
		1Ref	**2Ref**	**1Ref**	**2Ref**
BaselineD	-	22.2	23.9	23.1	24.8
BaselineF	-	22.3	24.0	23.1	24.9
Our Impl.D	0.5	24.9	24.7	25.8	27.2
Our Impl.D	0.3	**25.7**	**25.4**	**26.5**	**28.0**
Our Impl.D	0.1	24.9	24.7	25.7	27.1

Table 4: METEOR on development and test set with single vs multiple references. End-to-end model performance evaluated with new fluent references. D: Disfluent reference, F: Fluent reference as used in training. Our implementation is trained using disfluent references only.

model, while our implementation is a text-to-text model. Scores on the development set and test set using both single and multiple references are shown. We demonstrate that our implementation with three levels of disfluency induction and trained only on disfluent references outperforms the baseline score by a margin of 4.21 BLEU when the baseline uses disfluent references and by a margin of 3.11 BLEU even when the baseline system uses fluent references during training.

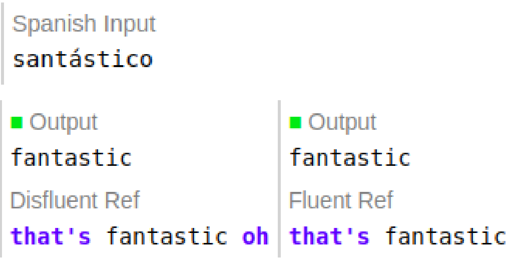
Table 4 shows the METEOR score evaluated on all three disfluency induction levels, using both single and multiple references. When comparing METEOR on single and multiple references of the same setting, the precision is the same up to two decimal digits, while there is a slight drop of 0.01 in recall in 2Ref when compared to 1Ref. The comparable METEOR values indicate that semantic meaning is retained in the output.

On comparing METEOR scores of our implementation with that of both baseline models, we observe that our model retains more semantic meaning than the baseline models. Using a single reference, we obtain an absolute difference of 3.4 and 3.6 METEOR scores on the dev and test sets respectively, between the best baseline system and our proposed model. This shows that while doing disfluency removal, the output also manages to successfully retain semantic meaning.

5 Analysis

In this section, we discuss different types of examples that were generated by our model and how they differ from the disfluent reference and the fluent reference. *Output* is the generated translation from our implementation, *disfluent Ref* and *Fluent Ref* are the disfluent and fluent references, respectively. It should be noted that fluent references were not used during training and are only being shown here for the sake of comparison[6]. Segment Comparison: **Deletion**, **Insertion**, **Shift**.

Figure 2 shows that the filler word *oh* has been omitted in the generated output.



Spanish Input
santástico

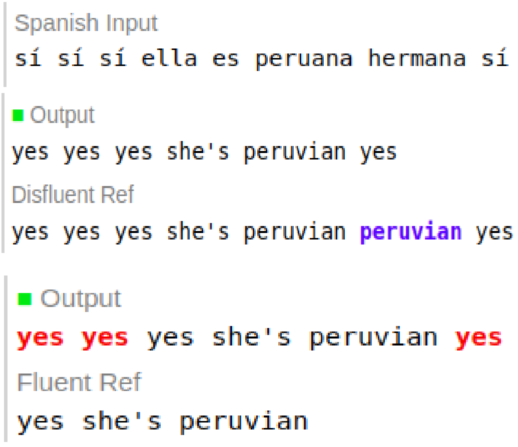
🟩 Output
fantastic
Disfluent Ref
that's fantastic oh

🟩 Output
fantastic
Fluent Ref
that's fantastic

Figure 2: Removing filler words.

[6] The figures used for comparison are created with CharCut (Lardilleux and Lepage, 2017)

In Figure 3, we observe that the repetition of the fluent phrase *peruvian peruvian* is handled correctly, but not the repetition of *yes*.



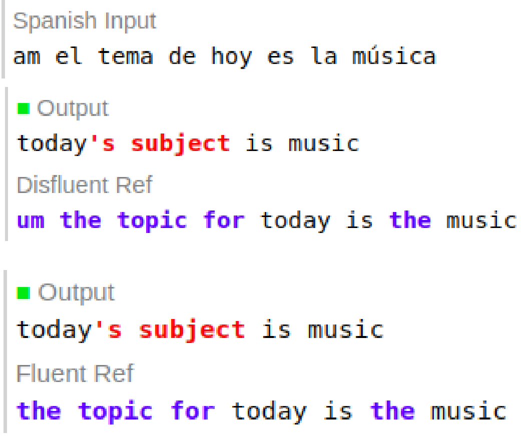
```
Spanish Input
sí sí sí ella es peruana hermana sí

■ Output
yes yes yes she's peruvian yes
Disfluent Ref
yes yes yes she's peruvian peruvian yes

■ Output
yes yes yes she's peruvian yes
Fluent Ref
yes she's peruvian
```

Figure 3: Repetitions (I)

In Figures 4, the output carefully rejects *um*, along with legitimately paraphrasing the sentence as a result of the language model that it has learned from the corpus.



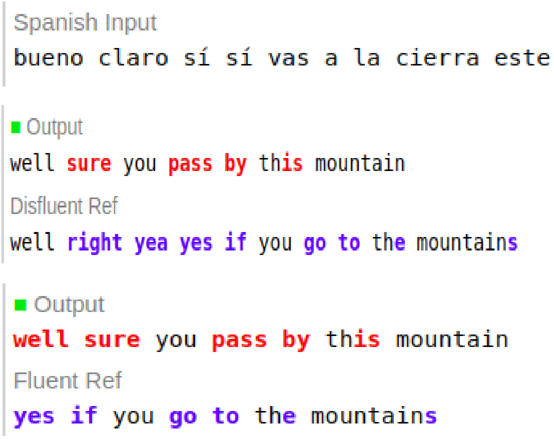
```
Spanish Input
am el tema de hoy es la música

■ Output
today's subject is music
Disfluent Ref
um the topic for today is the music

■ Output
today's subject is music
Fluent Ref
the topic for today is the music
```

Figure 4: Removing filled pause + paraphrasing (II)

In Figures 5, the disfluency *right yes yes* has been completely removed. Instead of choosing *yes*, it replaced it with *well sure*, but the disfluency has been removed.



```
Spanish Input
bueno claro sí sí vas a la cierra este

■ Output
well sure you pass by this mountain
Disfluent Ref
well right yea yes if you go to the mountains

■ Output
well sure you pass by this mountain
Fluent Ref
yes if you go to the mountains
```

Figure 5: Disfluency removal + paraphrasing (II)

6 Conclusion

In this work, we propose a model for generating fluent translations from disfluent text without any access to fluent references during training. We rely on having access to monolingual fluent text in the target language, which is largely available for most languages. We extract disfluency patterns by examining the disfluent text and inject disfluencies to create a parallel disfluent-to-fluent text corpus in the target language. We compare our results at different levels of disfluency induction and show significant improvements over a competitive baseline.

For future work, we aim at building more sophisticated and rich disfluency induction models. In this work, we focused on the text-to-text setting. We will look at extending this approach to a speech-to-text spoken translation task, with disfluency removal being an auxiliary task and investigate how to meaningfully tie parameters across an audio encoder and a text encoder. Furthermore, we only report standard quantitative metrics like BLEU and METEOR here. More detailed human evaluations may better highlight the benefits and limitations of our approach. We also believe that our proposed approach can be easily applied to other language pairs and hope to verify this as part of future work.

7 Acknowledgements

The authors thank the anonymous reviewers for their helpful feedback and comments. The last two authors gratefully acknowledge support from IBM Research (specifically the IBM AI Horizon Networks-IIT Bombay initiative).

References

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondrej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, and Changhan Wang. 2020. Findings of the IWSLT 2020 Evaluation Campaign. In *Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT 2020)*, Seattle, USA.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *CoRR*, abs/1710.11041.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2018. Low-resource speech-to-text translation. *CoRR*, abs/1803.09164.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Susanne Burger, Victoria MacLaren, and Hua Yu. 2002. The isl meeting corpus: The impact of meeting type on speech style.

Eunah Cho, Sarah Fünfer, Sebastian Stüker, and Alex Waibel. 2014. A corpus of spontaneous speech in lectures: The KIT lecture corpus for spoken language processing and translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1554–1559, Reykjavik, Iceland. European Language Resources Association (ELRA).

Ingrid Cartagena Kevin Walker David Graff, Shudong Huang and Christopher Cieri. Fisher spanish speech (ldc2010s01).

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 1027–1035, Red Hook, NY, USA. Curran Associates Inc.

Matthias Honal and Tanja Schultz. 2005. Automatic disfluency removal on recognized spontaneous speech - rapid adaptation to speaker-dependent disfluencies. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043.

Adrien Lardilleux and Yves Lepage. 2017. CHARCUT: Human-Targeted Character-Based MT Evaluation with Loose Differences. In *Proceedings of IWSLT 2017*, Tokyo, Japan.

Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur. 2013. Improved speech-to-text translation with the fisher and callhome spanish–english speech translation corpus. In *International Workshop on Spoken Language Translation (IWSLT 2013)*.

Elizabeth Salesky, Susanne Burger, Jan Niehues, and Alex Waibel. 2018. Towards fluent translations from disfluent speech. *CoRR*, abs/1811.03189.

Elizabeth Salesky, Matthias Sperber, and Alex Waibel. 2019. Fluent translations from disfluent speech in end-to-end speech translation. *CoRR*, abs/1906.00556.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and ZB Wojna. 2016. Rethinking the inception architecture for computer vision.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

W. Wang, G. Tur, J. Zheng, and N. F. Ayan. 2010. Automatic disfluency removal for improving spoken language translation. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5214–5217.

Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly transcribe foreign speech. *CoRR*, abs/1703.08581.

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