Towards Fluent Translations from Disfluent Speech

Elizabeth Salesky[1], *Susanne Burger*[1], *Jan Niehues*[2], *and Alex Waibel*[1,2]

[1]Carnegie Mellon University, Pittsburgh PA, U.S.A.
[2]Karlsruhe Institute of Technology, Karlsruhe, Germany

Abstract

When translating speech, special consideration for conversational speech phenomena such as disfluencies is necessary. Most machine translation training data consists of well-formed written texts, causing issues when translating spontaneous speech. Previous work has introduced an intermediate step between speech recognition (ASR) and machine translation (MT) to remove disfluencies, making the data better-matched to typical translation text and significantly improving performance. However, with the rise of end-to-end speech translation systems, this intermediate step must be incorporated into the sequence-to-sequence architecture. Further, though translated speech datasets exist, they are typically news or rehearsed speech without many disfluencies (e.g. TED), or the disfluencies are translated into the references (e.g. Fisher). To generate clean translations from disfluent speech, cleaned references are necessary for evaluation. We introduce a corpus of cleaned target data for the Fisher Spanish-English dataset for this task. We compare how different architectures handle disfluencies and provide a baseline for removing disfluencies in end-to-end translation.

1 Introduction

Spoken language translation applications suffer due to disfluencies in spontaneous speech. In conversational speech, speakers often use disfluencies such as filler words, repetitions, false starts, and corrections. These speech phenomena interfere with recognition and translation steps. In this work, we use disfluent conversational speech from the Fisher Spanish dataset[1], which has been translated to English [1, 2], with the disfluencies faithfully translated.

Machine translation systems are typically trained using well-structured and cleanly written text. The mismatch between clean training data and test data with speech phenomena causes a drop in performance. Systems to detect and remove disfluencies from input speech, creating cleaner source data for MT, have been shown to greatly improve the performance of spoken language translation systems, even on broadcast news and TED talks where these phenomena are less common [3, 4, 5, 6, 7].

Before end-to-end models, different datasets could be used to train the speech recognition and translation components of a speech translation system. Where aligned speech and translations exist, the data is usually clean speech→clean text, as in news data or TED talks, or disfluent speech→disfluent translations, as in Fisher or meeting data, where the disfluencies have been explicitly included in the references for completeness, but are not labeled. Some corpora with labeled disfluencies exist; these labeled sections can be removed to create clean target text. However, only parts of these corpora have been translated and/or released [4, 8]. Using disfluent/disfluent parallel data, we cannot score generated translations with disfluencies removed; we need cleaned reference translations. We used MTurk to create clean target data for the Fisher Spanish-English data. This data will be released with the final paper.

Disfluency recognition and removal has previously been performed as an intermediate step between speech recognition (ASR) and machine translation (MT), to make transcripts more similar to typical machine translation data. With the rise of end-to-end sequence-to-sequence speech translation systems [9], disfluency removal would need to be incorporated into the model instead of handled as a separate step. Further, implicit handling in the model architecture may promote the ability to recognize disfluencies and corrections specific to current data, outside of a set of handcrafted labels. We hope this data will promote further research in this area.

2 Data

For our experiments, we use the Fisher Spanish dataset[2], composed of telephone conversations between mostly native Spanish speakers. The corpus consists of 819 transcribed conversations on provided topics between strangers, yielding ∼160 hours of speech and 150k utterances. The transcripts were translated and released by JHU[3], with one reference for the training data and four each for the dev and test sets.

This data is conversational and disfluent. Disfluencies can be filler words and hesitations, discourse markers (*you know*, *well*, *mm*), repetitions, corrections and false starts, among others. The reference translations maintain and translate where possible the disfluencies in the Spanish source. Examples of certain types of disfluencies shown in both source and target are below in Table 1.

We note that there can be many different and often over-

[1]LDC2010S01 and LDC2010T04

[2]LDC2010S01 and LDC2010T04

[3]joshua-decoder.org/fisher-callhome-corpus

Hesitation	eh, eh, eh, um, yo pienso que es así. uh, uh, uh, um, i think it's like that.
Repetition	Y, y no cree que, que, que, And, and I don't believe that, that, that
Correction	no, no puede, no puedo irme para ... no, it cannot, I cannot go there ...
False start	porque qué va, mja ya te acuerda que ... because what is, mhm do you recall now that ...

Table 1: Examples of disfluencies in Fisher Spanish-English, in the Spanish transcripts and English reference translations

lapping types of disfluencies in a single utterance, as in this example from the Spanish data, 'también tengo um eh estoy tomando una una clase ...' which contains filler words (*um, eh*), a correction (*tengo* → *estoy tomando*), and repetition (*una, una*) with overlapping scope. Additionally, context affects whether some word types are disfluent (*so, oh,* ...), and so removing them in all cases will affect meaning. Disfluency removal is a more complex problem than merely recognizing disfluencies from a compiled list.

2.1 Mechanical Turk

To create clean 'copy-edited' reference translations, we crowd-sourced the task on Amazon Mechanical Turk. Turkers were presented with the original English translations in context and asked to remove certain types of disfluencies while maintaining meaning. We specify here filler words, repetitions, corrections, false starts, with examples of each. Where an utterance was deemed to have only disfluencies, Turkers were instructed to enter 'None' for no content and specify why in the comments. Turkers were paid a competitive rate equating to a U.S. hourly minimum wage. Each 'Human Intelligence Task' (HIT) was limited to 5 utterances to not overwhelm Turkers. The first utterance was a control, allowing Turkers to familiarize themselves with the task, the results of which are not included in our data. Utterances with only 1 token were not crowd-sourced but labeled manually by us.

We required that Turkers had an approval rate of 95%. We had 1250 unique workers complete 26,270 HITS. The ratio of approved to rejected HITs was 25:1. HITs were rejected if answered implausibly fast, the answers were incomplete, or they included clearly unrelated content. Each utterance in the dev and test sets was cleaned by at least 2 Turkers to reduce variability, and for the larger training set, one Turker.

2.2 Original vs Cleaned References

The cleaned references contain on average 3 fewer tokens per sentence, reducing the average sentence length from 11.3 to 8.2. Most utterances contained at least one disfluency; only 35% of utterances were unchanged by Turkers. However, 16,829 sentences or 10.5% of all utterances were marked only disfluencies. These ranged from single token utterances ('Mhm') to potentially several ('Hmm mm hmm mm we'). They were typically very short (fewer than three tokens) and contained only filler words or false starts; in context, most can be viewed as backchanneling. Backchanneling, or verbal cues to indicate attention, can sometimes convey information or meaningful reactions ('oh?'), and other times, may be classified as disfluencies. To determine which, it is important to view the utterance in context. In most cases, Turkers reduced these utterances to 'None'. Below we discuss annotator agreement, and sources of least disagreement.

Below is an example of the types of changes made by Turkers. We show the original Spanish transcript and the disfluent English translation, along with the generated clean reference. We use NIST's sclite tool [10] to evaluate changes made by Turkers in terms of insertions (I), deletions (D), and substitutions (S).

SRC	Y,	bueno,	y	que,	aunque	no	se	ve	
REF	and,	well,	and	that,	even	though	you	don't	see him
CLEAN	***	***	and	***	even	though	you	don't	see him
Eval:	D	D		D					

Example of generated cleaned references (CLEAN) with original Spanish source (SRC) and disfluent English target (REF).

While many disfluencies can be removed through deletions, false starts and corrections can often lead to insertions, substitutions, or reorderings in the cleaned text. Table 2 shows the percentage breakdowns of insertions, deletions, and substitutions made by Turkers in cleaning each of the datasets.

Dataset	Insertions	Deletions	Substitutions
train	0.6%	25.8%	2.8%
dev	1.5%	31.7%	5.2%
dev2	1.0%	33.2%	4.5%
test	1.2%	31.4%	4.5%

Table 2: Percentages of token insertions, deletions, and substitutions made by Turkers in generating the cleaned reference translations.

To verify the content changed by Turkers, we first look at the agreement between Turkers. For the `dev`, `dev2`, and `test` datasets, we collected annotations for each utterance from two different Turkers to measure consistency. We use the two collected annotations for each utterance to make two clean reference translations. We can look at the BLEU (n-gram precision) [11] between the two references as a measure of annotator agreement, shown in Table 3. The typical preprocessing scheme for this dataset is lowercased and with all punctuation removed [1, 2, 9]; to provide a fair comparison, we remove punctuation and case to test annotator agreement. We show the average BLEU against a single reference,

as a multi-reference score would not be a fair comparison here. We find a high level of agreement between MTurk annotators, suggesting this is a task that can be crowd-sourced. For context, we additionally show the average BLEU score between the pairs of the four original references for each of these datasets, as well as the BLEU score of the two clean references against the original disfluent translations.

We find that the inter-annotator BLEU score is very high across the cleaned corpus, and considerably higher than the original data's inter-annotator BLEU. This is unsurprising, as in our task, the two Turkers are given the same English sentence and told to edit specific content; unaltered content will be the same between the two Turked references. In the original collection, the four Turkers were given the same Spanish source sentence and independently generated translations, leading to more variability. The original inter-annotator BLEU can serve as a benchmark for our translation systems, as this is the BLEU between human translators on this data.

We also compare the clean translations to the original disfluent translations. Annotator-Original uses the original data as references to score the new clean MTurk data as 'hypotheses', while Original-Annotator does the opposite, scoring the disfluent translations as hypotheses against the cleaned references. We see that scoring disfluent data against clean references has a greater impact on BLEU than the opposite: the Original-Annotator BLEU is much lower, demonstrating the significant impact that disfluent outputs can have when scoring translations of an MT system expecting clean output. We later use these scores as benchmarks for different training data conditions. For all values in Table 3 variance is less than 0.25 BLEU.

Comparison	dev	dev2	test
MTurk Inter-Annotator BLEU	63.04	64.32	64.00
Original Inter-Annotator BLEU	34.81	35.80	33.85
Annotator-Original BLEU	28.45	28.90	28.31
Original-Annotator BLEU	21.00	21.44	20.82

Table 3: Measures of MTurk annotator agreement: Inter-Annotator BLEU between generated MTurk translations, and among the 4 original translations. For comparison, BLEU between the clean references to the original disfluent refs.

We further look at a data sample to verify the content changed is disfluent. We find a set of 268 unique filler words within the original translations after punctuation is removed, 119 ignoring case, in part because they were crowdsourced and different translators used slightly different schemes. Of the tokens deleted by Turkers, 9.5% are filler words. Specifically for dev, we find Turkers disagree on at least one token in 56% of utterances. Of these, some involve context-dependent disfluencies such as backchanneling, or corrections where Turkers re-phrased with minor differences. In most cases, backchanneling was marked as disfluent and reduced to 'None'). Some disagreements involve insertions, either of pronouns (e.g. 'imagined it' $\rightarrow$ 'i imagined it'), or function words to make utterances more grammatical in English where Turkers introduced different tokens. A larger percentage of disagreements involved deletion of transitional words or phrases (sentence-initial 'and') to make utterances more sentence-like. In these cases, Turkers typically removed overlapping spans, with disagreements based on the span of tokens removed.

3 Experiments

Initial work on the original Fisher-Spanish dataset used traditional HMM-GMM ASR systems chained together with phrase-based MT systems using lattices [1, 2]. More recently, it was demonstrated in [9] that end-to-end sequence-to-sequence models perform competitively on this task.

We here focus on translation from the Spanish text transcripts as an initial exploration of the problem of translating directly from noisy speech to clean references without a separate disfluency removal step. We use sequence-to-sequence models and, as a baseline, first demonstrate the efficacy of our models on the original disfluent Fisher Spanish-English task, comparing to the previously reported numbers on the MT subtask [1, 2, 9]. Post et al. [1] and Kumar et al. [2] are both traditional systems, while Weiss et al. [9] is a deep LSTM-based sequence-to-sequence model. We then compare these results with models trained using our collected clean target data. Finally, we look at the mismatched case where we train on disfluent data and evaluate on a cleaned test set; this is a more realistic scenario, as clean training data is difficult to collect, and we cannot expect to have it for each language and use case we encounter. We hope this will spur future work on this lower-resource task.

We compare LSTM-based models, similar to [9], to Transformer [12] models as implemented in OpenNMT [13]. Our LSTM models use a two-layer bidirectional LSTM encoder and two-layer LSTM decoder, 500-dim embeddings, and Luong attention [14]. We follow the default OpenNMT training procedure, optimizing with SGD for 13 epochs using a batch size of 32. Our Transformer models follow the suggested parameters from OpenNMT, with layer size 512, sinusoidal position encodings, dropout of 0.1, label smoothing set to 0.1 [15], and optimizing with adam using the suggested learning rate scheme. We reduce the number of layers to four for our smaller dataset. We batch and normalize by tokens, and compute gradients based on four batches. We experimented with four batch sizes holding other parameters constant $\{548,1096,1644,2192\}$, and determined 1644 is the best for this dataset; all reported numbers use this value. All models use the same preprocessing as previous work on this dataset [1, 2, 9]: lowercasing and removing punctuation.

Table 4 shows our results on the original disfluent data. We provide both single and multi-reference scores: Fisher has four reference translations for dev, dev2, and test, which boosts scores considerably as hypotheses can match in any of the references. We do not have four references for

System	dev		dev2		test	
	1R	4R	1R	4R	1R	4R
LSTM	35.2	61.9	36.3	62.8	33.3	60.4
Transformer	32.1	57.0	32.7	58.1	30.6	55.4
Post et al. [1]	–	–	–	–	–	58.7
Kumar et al. [2]	–	–	–	65.4	–	62.9
Weiss et al. [9]	–	58.7	–	59.9	–	57.9

Table 4: BLEU score using **original disfluent references**. Comparing average single reference score (1R) vs multi-reference score using all four references (4R).

our clean data, so the single reference scores provide a better basis for comparison to the clean target task. We show both of our models perform competitively, approaching or exceeding previous best results. Further, our single reference scores approach the inner-annotator BLEU between the four human-generated references, shown in Table 3. For `test`, our LSTM model has a BLEU of 33.3 on a single reference, as compared to 33.8 between the four human translators. The LSTM model is consistently slightly better than the Transformer model. We note though that the Transformer is quite sensitive; it is possible with other parameters, it would perform better.

System	dev		dev2		test	
	1R	2R	1R	2R	1R	2R
LSTM	28.18	34.07	28.87	35.44	27.96	33.84
Transformer	26.20	32.16	27.27	33.87	26.31	31.89

Table 5: BLEU score using **new cleaned references** to train and evaluate. Comparing average single reference score (1R) vs multi-reference score using both generated references (2R).

Turning to the task of generating clean translations, we now make use of our clean target data to train. Table 5 shows our results using this new data. BLEU scores go down on the clean task; a main contributor is filler words, which previously may have been overgenerated and provided partial n-gram matches and have now been removed. For example, removing only our list of 119 filler words from the original references and scoring our disfluent LSTM model's with a single reference drops the score on `test` from 33.8 to 18.40. In this dataset, filler words are quite one-to-one and easy to generate (see Table 1). Our systems improve on the mismatched condition, learning not to generate some disfluencies: we see scores on average 5.5 BLEU higher than the Original-Annotator scores in Table 3, which scores the disfluent target data against our new clean references, and can be seen as a lower bound. Further, the original Spanish-English data is mostly one-to-one and monotonic. With the cleaned targets, the alignment between source and target is not as clear, making the translation task harder. Finally, the utterances, which were quite short to begin with, are now three tokens shorter on average. This means a single mistake has higher consequences for BLEU, which uses 4-gram precision.

Both architectures are able to learn to remove many disfluencies using the clean target data. Figure 1 shows an example from the LSTM model attention where it has clearly learned to place less weight on source disfluencies, generating the fluent translation 'No not yet.' Here the LSTM model has learned to both delete filler words ('mm') and repeated words and phrases ('no no', 'no todavía').

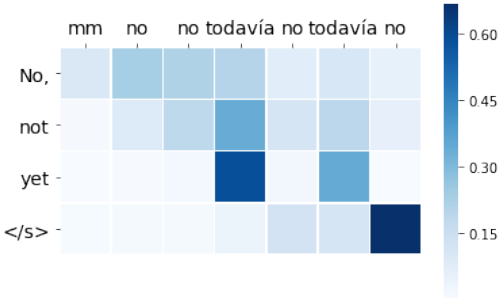

Figure 1: LSTM attention: with cleaned target data, learns to place less weight on source disfluencies

Table 6 shows an example of the different translations generated by the LSTM and Transformer models.

SRC	también tengo um eh estoy tomando una clase ...
REF	i also have um eh im taking a marketing class ...
CLEAN	im taking a marketing class ...
LSTM	im taking a class of marketing
Transformer	i also have a class of marketing classes

Table 6: Example outputs training with clean target data

While we make use of cleaned target references here, these references are expensive and time-consuming to create, even making use of crowdsourcing. We cannot expect to have this resource to train on for every language and use case, though we may have smaller datasets available for dev and test. Simulating this more likely scenario, Table 7 shows our results training on the original parallel disfluent data, and evaluating on cleaned dev and test data. As expected, we do similarly to the Annotator-Original scores in Table 3. Using clean data, the entropy after each word is much lower, leading to lower perplexities and clearer, more concise translations. With the disfluent data, the model is less able to transition between meaningful tokens: filler words and clause restarts are able to appear in many places, causing the model to stutter. It not only generates filler words and repetitions, but loses general coherence, ex) *'I would tell you, I mean, it's more, it's easier, no, I mean'*. In general, disfluent models overgenerate, producing utterances $1.25\times$ longer than our clean models.

Though the Transformer here is consistently slightly worse than the LSTM models, we speculate it could more

System	dev		dev2		test	
	1R	2R	1R	2R	1R	2R
LSTM	20.88	26.11	22.03	27.58	20.68	26.01
Transformer	19.50	24.35	21.52	26.48	20.52	25.72

Table 7: **No cleaned training data condition**: BLEU score training on disfluent target data and evaluating on cleaned references. Comparing average single reference score (1R) vs multi-reference score using both generated references (2R).

easily be extended to lower-training data settings. Self-attention within the Transformer models' decoder allow the decoder to attend to all previous decoder timestamps [12]. We hypothesize that this mechanism could help the decoder better learn to generate clean fluent text from clean training data, particularly when disfluencies depend on the generated context (corrections, etc). By attending to previous decoder states, the model may learn not to generate repeated words, for example. Comparing the LSTM and Transformer models trained with the clean target data, we see this borne out: the Transformer model has two-thirds fewer generated repetitions on `test`. However, both architectures learn to remove repeated words quite well: the original `test` references contain 657 repeated tokens, while and the LSTM model generates only 67, and the Transformer 44. We will investigate this claim in future work by pre-training this model on more fluent parallel data, such as TED, to see if pre-training the decoder self-attention enables us to do this task without requiring as much cleaned training data. We hope that this data and our initial results encourage further work on this task, and additionally provide a benchmark to aim for using less clean target data to train, the more common condition in conversational speech translation.

4 Conclusion

Machine translation systems for speech suffer due to conversational speech phenomena, particularly the presence of disfluencies. Removing disfluencies improves performance of downstream translation, as it causes data to better match typically clean training text. However, previous work on removing disfluencies for speech translation have done so as a separate step in between speech recognition and machine translation, which is not possible using end-to-end systems. We release cleaned target data for a parallel speech and text corpus, enabling further work in this area. We compare disfluency handling among two architectures, and present a baseline to implicitly remove disfluencies within the context of end-to-end translation models.

5 References

[1] M. Post, G. Kumar, A. Lopez, D. Karakos, C. Callison-Burch, and S. Khudanpur, "Improved speech-to-text translation with the Fisher and Callhome Spanish–English speech translation corpus," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, December 2013.

[2] G. Kumar, M. Post, D. Povey, and S. Khudanpur, "Some insights from translating conversational telephone speech," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3231–3235.

[3] E. Cho, T.-L. Ha, and A. Waibel, "Crf-based disfluency detection using semantic features for german to english spoken language translation," in *IWSLT*, 2013.

[4] E. Cho, S. Fünfer, S. Stüker, and A. Waibel, "A corpus of spontaneous speech in lectures: The kit lecture corpus for spoken language processing and translation," in *LREC*, 2014, pp. 1554–1559.

[5] W. Wang, G. Tur, J. Zheng, and N. F. Ayan, "Automatic disfluency removal for improving spoken language translation," in *ICASSP*. IEEE, 2010, pp. 5214–5217.

[6] M. Honal and T. Schultz, "Automatic disfluency removal on recognized spontaneous speech-rapid adaptation to speaker-dependent disfluencies," in *ICASSP*, vol. 1. IEEE, 2005, pp. I–969.

[7] V. Zayats, M. Ostendorf, and H. Hajishirzi, "Disfluency detection using a bidirectional lstm," *arXiv preprint arXiv:1604.03209*, 2016.

[8] S. Burger, V. MacLaren, and H. Yu, "The isl meeting corpus: The impact of meeting type on speech style," in *Seventh International Conference on Spoken Language Processing*, 2002.

[9] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly transcribe foreign speech," *arXiv preprint arXiv:1703.08581*, 2017.

[10] J. Fiscus, "Sclite scoring package version 1.5," *US National Institute of Standard Technology (NIST), URL http://www. itl. nist. gov/iaui/894.01/tools*, 1998.

[11] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2002)*. Association for Computational Linguistics, 2002, pp. 311–318.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[13] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "OpenNMT: Open-source toolkit for neural machine translation," in *Proc. ACL*, 2017. [Online]. Available: https://doi.org/10.18653/v1/P17-4012

[14] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 15*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 1412–1421. [Online]. Available: http://aclweb.org/anthology/D15-1166

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.