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
esalesky@cs.cmu.edu

Data

We use the Fisher Spanish-English dataset, composed of telephone conversations between mostly native Spanish speakers. The corpus consists of ∼160 hours of speech and 138k utterances.
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. English reference translations faithfully translate disfluencies in the source speech.

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

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

To train and evaluate 'fluent' output, we collected clean 'copy-edited' reference translations crowd-sourced on Mechanical Turk. Where utterances had only disfluencies, Turkers marked 'No fluent content.'

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					

Table 2: Example of generated cleaned references (CLEAN) with original Spanish source (SRC) and disfluent English target (REF).

Turkers were allowed to reorder the data:

y entonces am es entonces la universidad donde yo estoy es university of pennsylvania
and so um and so the university where i am it's the university of pennsylvania
→ *i am at the university of pennsylvania*

Most common utterances in dataset are 1-2 token backchanneling:

'yes', 'aha', 'mm', 'hmm', 'sure', 'oh', 'ah', 'mhm', 'yeah', 'yes yes', 'right', 'uh huh', 'hello', 'exactly', 'no', 'okay', 'uh uh', 'hm mm', 'oh yes', 'um'

16,829 utterances or 10.5% of all utterances marked only disfluencies; These could be up to several tokens: 'Mhm' vs. 'Hmm mm hmm mm we'

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

We compare the two sets human-generated reference translations and use the scores as benchmarks for different training data conditions. For all values in Table 4 variance is less than 0.25 BLEU.

- Annotator-Original uses the original references to score the new clean MTurk references as 'hypotheses'
- Original-Annotator scores disfluent references as hypotheses against the cleaned references. Can be seen as a **lower bound** for models generating clean output.

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 4: 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.

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.

Abstract

- Generating clean text from noisy, disfluent speech may be desired e.g. for simultaneous applications where this will increase the clarity and usability off the application
- Previous work on disfluency removal in speech translation (SLT) uses an intermediate step between speech recognition (ASR) and machine translation (MT) to make ASR output better-matched to clean MT training data
- To do disfluency removal in **end-to-end** speech translation systems, need to incorporate this step into training or handle as a post-processing step
- To incorporate into training requires parallel disfluent speech and clean text
- We collected a corpus of cleaned target data for the Fisher Spanish-English dataset for **training** and **evaluation**
- We present baseline results with text-only models, comparing 2 different architectures (LSTM and Transformer)

Output

Clean data has lower perplexities and clearer, more concise translations. Models trained with the disfluent data have difficulty in search: generating content after over-represented disfluencies can cause the model to 'stutter' during generation.
ex) *'I would tell you, I mean, it's more, it's easier, no, I mean'.*
Disfluent models overgenerate, producing 1.25× longer utterances. Table 5 shows an example of the generated translations by model.

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 5: Example outputs training with clean target data

Transformers' decoder self-attention enables attending to all previous decoder timestamps. Could this help it better generate clean, fluent text when disfluencies depend on other generated context (corrections, repetitions, etc)?
Though Transformer scores were generally lower, the Transformer model generates 66% fewer repetitions. However, both architectures learn to remove repeated words quite well: original `test` references have 657 repeated tokens, the LSTM model generates only 67, and the Transformer generates only 44.

Figure 1 shows an example where the LSTM model attention has learned to downweight source disfluencies, to generate the fluent translation 'No not yet.' The LSTM model successfully deletes both filler words ('mm') and repetitions ('no no', 'no todavía').

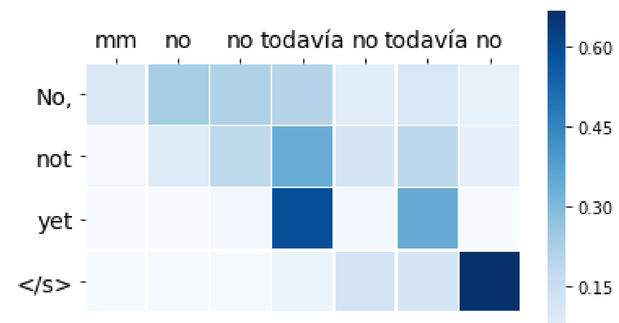

Figure 1: LSTM attention: with cleaned target data, learns to place less weight on source disfluencies

Contact Information

• **Email:** esalesky@cs.cmu.edu

References

[1] M. Post, *et al.*, "Improved speech-to-text translation with the Fisher and Callhome Spanish–English speech translation corpus," in *IWSLT*, December 2013.
[2] G. Kumar, *et al.*, "Some insights from translating conversational telephone speech," in *ICASSP*, 2014.
[3] R. J. Weiss, *et al.*, "Sequence-to-sequence models can directly transcribe foreign speech," *arXiv preprint arXiv:1703.08581*, 2017.

Models

We compare **LSTM** models to **Transformer** models, as implemented in OpenNMT.

- Our **LSTM** models use a 2-layer bidirectional LSTM encoder and 2-layer LSTM decoder, 500-dim embeddings, and Luong attention. 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 WMT parameters from OpenNMT: layer size 512, sinusoidal position encodings, dropout of 0.1, label smoothing set to 0.1, and optimize with adam using the suggested learning rate scheme. We reduce the number of layers to 4 for our smaller dataset. We batch and normalize by tokens, and compute gradients based on 4 batches. We tried 4 batch sizes holding other parameters constant {548,1096,1644,2192}, and determined 1644 was best; all reported numbers use this value.

All models use the same preprocessing as previous work on this dataset: lowercasing and removing punctuation [1, 2, 3]

Results

We compare results on the **original** references to previous work to show our models are competitive, before turning to our target task.

Previous work reports 4-reference BLEU. We report both 4-reference and average 1-reference scores to contextualize results on our new data.

- Post et al. [1] and Kumar et al. [2] are phrase-based models implemented in Joshua.
- Weiss et al. [3] is a neural LSTM-based model

	dev		dev2		test	
System	**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. [3]	–	58.7	–	59.9	–	57.9

Table 6: BLEU score using **original disfluent references**. Comparing average single reference score (1R) vs multi-reference score using all four references (4R).

BLEU scores go down on the clean task; most frequent vocab removed. **ex)** post-processing our disfluent model output by removing filler words drops the 1R `test` BLEU from 33.8 to 18.40.

Scores improve relative to Original-Annotator scores in Table 4 by avg. 5.5 BLEU; this scores the disfluent target data against our new clean references, and can be seen as a lower bound.

Challenges:

- Filler words are the most common vocab and are easy to translate.
- The original Spanish-English data is mostly one-to-one and monotonic. Clean targets create more challenging alignments.
- Utterances are even shorter: down from 11.3 to 8.2 tokens on avg. Single mistake has higher consequences for BLEU.

	dev		dev2		test	
System	**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 7: 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).

Most languages and datasets will not have cleaned training data. Expectedly, mismatched condition performs similar to Annotator-Original scores in Table 4. Provides a baseline for future work to reduce training data requirements, e.g. through pre-training or LM multi-tasking.

	dev		dev2		test	
System	**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 8: 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).