Improving Long-form Speech Translation through Segmentation with Large Language Models and Finite State Decoding Constraints

Anonymous ACL submission

Abstract

One challenge in spoken language translation is that plenty of spoken content is long-form, but short units are necessary for obtaining high-quality translations. To address this mismatch, we adapt large language models (LLM) to split long ASR transcripts into segments that can be independently translated so as to maximize the overall translation quality. To combat the tendency of hallucination by LLMs, we incorporate finite-state constraints during decoding to eliminate invalid outputs. We discover that LLMs are adaptable to transcripts containing ASR errors through prompt-tuning or fine-tuning. In comparison to a state-of-the-art automatic punctuation baseline, our best LLM improves the average BLEU for English-German, English-Spanish, and English-Arabic TED talk translation in 9 test sets by 2.9 points, just by improving segmentation.

1 Introduction

With the proliferation of long-form audiovisual content online, its translation and captioning becomes paramount for accessibility. Cascade models remain the dominant approach for speech translation (Arivazhagan et al., 2020; Li et al., 2021), decomposing the problem into automatic speech recognition (ASR), post-processing of the transcript, and machine translation (MT).

The cascade's MT component typically operates on sentence-like units, with each sentence translated independently of the others. When asked to translate long passages, models regularly fail or degenerate (Cho et al., 2014; Pouget-Abadie et al., 2014; Koehn and Knowles, 2017). This differs considerably from the expectations for automatic speech recognition models (e.g. Graves, 2012) that can process inputs of unbounded lengths. MT models must either be able to cope with potentially long, multi-sentence inputs or, alternatively, they must be able to determine cutpoints at which the transcript can be segmented into compact, independently translatable units. This work introduces a new, effective approach for the latter.

While numerous text segmentation techniques have been proposed to improve spoken language translation (Section 6), the problem remains hard and unsolved. Indeed, Li et al. (Li et al., 2021) demonstrate that poor sentence segmentation degrades performance almost twice as much as transcript lexical errors.

We cast sentence segmentation as a sequence-to-sequence task, rather than a traditional structured prediction task that tags sentence-final tokens. While this lets us leverage large language models, such models' outputs can be ill-formed. Even by using additional data for fine-tuning, residual adapters (Tomanek et al., 2021; Chronopoulou et al., 2022), or future discriminators (Yang and Klein, 2021), simple syntactic constraints can be difficult to enforce. Moreover, all three require modifying the model or storing additional learned parameters. In light of these concerns, we introduce a simple, flexible, and modular approach to generating well-formed task-specific strings at inference time without any additional training. We compactly express constraints on the output format as finite-state machines, then efficiently enforce these via composition. While the approach is simple, it remains unexplored for large language models, and it yields automatic gains on downstream performance, advancing the state of the art for speech translation and thereby applicable to existing systems. Moreover, the approach is sufficiently general that it can be applied to other domains in a *plug-and-play* manner.

We benchmark our approach as a component in a speech translation cascade. Experiments in three language pairs indicate that our approach outperforms a baseline cascade system that predicts punctuation marks before inferring sentence boundaries, and a strong neural structured predic-

tion model. Overall, we improve the BLEU score on the IWSLT test sets by 2.9 points, closing 3/4 of the gap between the previous best and the oracle system. Our contributions are three-fold:

- We propose a novel LLM-based approach for long-form speech translation, which can be applied to any ASR-MT speech translation cascade system and yield a significant increase in translation quality.
- To the best of our knowledge, we are the first to investigate the use of finite state decoding constraints in combination with LLMs to produce consistent improvements.
- We report additional small but consistent improvements by prompt-tuning or fine-tuning LLMs on ASR transcripts containing lexical and grammatical errors.

2 Windowing Approach

One major challenge in modeling and inference of long-form transcript segmentation is that the input sequences can be very long. For example, a TED talk can contain more than one thousand words (Li et al., 2021). We take a divide-and-conquer approach that operationalizes two straightforward principles in modeling. First, words on the left and right are both useful for deciding if a sentence delimiter should be present at the current word position. Second, distant words are less useful than nearby words. From these two principles, we design a top-level sliding window algorithm to balance the need for bidirectional modeling and efficiency of computation. We divide the passage into windows at both training and test time, with a small context window on each side to inform decisions at window edges (Figure 1). With this top-level inference algorithm, the sequence-to-sequence machine learning problem is now reduced to the window-level. The problem is now to predict a sequence of segmentation decisions $\mathbf{y} = y_1, \ldots, y_w$ for each text *window* of size at most w tokens: $\mathbf{x} = x_1, \ldots, x_w$.

3 Modeling Approaches

A classic approach to discriminative sequence modeling is the conditional random field (CRF) (Lafferty et al., 2001; Liu et al., 2005). This conditional graphical model allows incorporating arbitrary features of the transcript, including linguistic variables and word embeddings.

3.1 Structured Prediction Baseline: Bidirectional RNN Model

The limitation of the CRF is in the Markov assumption it makes, considering only the immediately previous word's segmentation decision. Even higher-order CRFs can only consider a fixed-size history within $\mathbf{y}$. Instead, we introduce a neural autoregressive segmenter. It is an encoder-decoder neural network with monotonic hard attention to the bidirectionally encoded input at the current word position, admitting the same rich featurization of $\mathbf{x}$ as the CRF; its likelihood is

$$\begin{aligned} p_\theta(\mathbf{y} \mid \mathbf{x}) &= \textstyle\prod_{t=1}^{w} p_\theta(y_t \mid \mathbf{y}_{<t}, \mathbf{x}) \\ &:= \textstyle\prod_{t=1}^{w} p_\theta(y_t \mid \mathbf{y}_{<t}, \mathbf{BiRNN}(\mathbf{x})_t) \end{aligned}$$

where p_θ is parameterized by a recurrent neural network followed by a linear projection layer and a softmax layer to obtain a locally normalized distribution. Exact inference here is intractable (unlike a CRF); we approximate it with beam search. This model as well as a QRNN (Bradbury et al., 2017) based automatic punctuation model will serve as baseline models.

3.2 Large Language Models for Segmentation

More recently, the paradigm of pre-training followed by fine-tuning or few-shot learning has achieved great successes across many NLP tasks. The pre-training task is typically a variant of a language model (Brown et al., 2020; Chowdhery et al., 2022) or an autoencoder (Raffel et al., 2020) where a corrupted version of a sentence is mapped to its uncorrupted counterpart. We can encode segmentation as such a task: reproducing the input with inserted sentence delimiters. Concretely, we encode $\mathbf{y}$ as $z_1, \ldots, z_w$ where $z_t = \text{Concat}(d_t, x_t)$ and $d_t \in \{\epsilon, \blacksquare\}$. For example, we feed `i am hungry i am sleepy` to the model, and it produces the sentence-delimited string `i am hungry` ■ `i am sleepy`. We use the publicly available T5 (Text-to-Text Transfer Transformer) model (Raffel et al., 2020) and the GPT-style (Brown et al., 2020) PaLM model (Chowdhery et al., 2022) as the foundation for our text-based segmenters.

3.2.1 Prompting and Fine-tuning

Training examples for this task look like the input output pairs in Figure 2. In fine-tuning, we

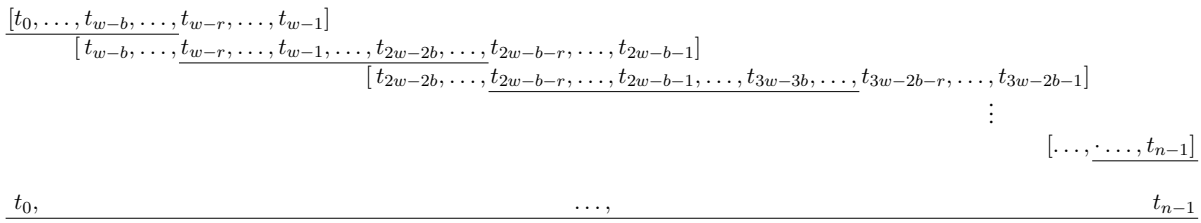

Figure 1: Processing overlapping windows instead of entire transcript passages. w is the window size used in both training and inference. b is the total context window size. r ($\leq b$) is the right context window size. The underlines below the windows indicate which local segmentation decisions are taken as global decisions. Portions not underlined (i.e., the context window) are still provided to the segmentation model to inform segmentation of underlined portions.

>>> Segment a sequence of words into sentences separated by the delimiter **<SENT>**

Input: well first of all thank you so much that is a beautiful compliment i do think he is the best interviewer alive
Segmented Output: well first of all thank you so much **<SENT>** that is a beautiful compliment **<SENT>** i do think he is the best interviewer alive

Input: i remember when my dad had to leave our home in scranton pennsylvania to find work i grew up in a family where if the price of food went up you felt it that's why one of the first things i did as president was fight to pass the american rescue plan
Segmented Output: i remember when my dad had to leave our home in scranton pennsylvania to find work **<SENT>** i grew up in a family where if the price of food went up you felt it **<SENT>** that's why one of the first things i did as president was fight to pass the american rescue plan

Input: we're done talking about infrastructure weeks we're going to have an infrastructure decade
Segmented Output: we're done talking about infrastructure weeks **<SENT>** we're going to have an infrastructure decade

---

Input: it is going to rain today remember to bring an umbrella
Segmented Output:

Figure 2: Prompting PaLM to segment a text window (red) based on three examples.

update the full set of parameters for a given model on such examples to minimize the cross entropy on the output. For T5 models, the input sequence will be fed to the encoder, and the output sequence will be fed to the decoder through teacher forcing. For PaLM models, the input sequence and the output sequence are concatenated and fed to the decoder with an optional prompt as the prefix. For decoder-only PaLM models, a text prompt like the one in Figure 2 or a fine-tuned soft prompt (Lester et al., 2021) in the embedding space prompts the decoder to enter the state for the segmentation task. When we fine-tune PaLM, the entire model is updated for this task so that no prompting is necessary.

3.2.2 Decoding Constraints

A deficiency of generation with LLM is that the output might not only fail to correctly segment the passage; it might not even contain the same tokens as the passage. We shall say that an output is *well-formed* if it contains the same token sequence as the input, with zero or one sentence delimiters before each token. While the rich parameterization of such large Transformer models may learn the inherent structure of the output, we provide two solutions to enforce well-formedness.

Levenshtein Alignment for Post-processing The generation models' ability to produce arbitrary outputs may be seen as a strength: the model could correct transcription errors and remove disfluencies, if so trained. Therefore, we can let the model generate freely without enforcing structural constraints, then enforce well-formedness post-hoc. Kumar and Byrne (Kumar and Byrne, 2002) describe a WFST for *Levenshtein alignment* between two strings. We use it to align the generated string with $\mathbf{x}$. We then project segment boundaries across alignment links from the generated string onto $\mathbf{x}$ to determine $\mathbf{y}$. In this way, annotations can be salvaged when LLM does not precisely recreate the input.

Finite State Constraints in Decoding A natural strategy to force well-formed outputs is *constrained decoding* (e.g. Zhang et al., 2019). In it, we compose the input FSA $\mathbf{x}$ and a special FST $\mathcal{T}$ encoding all possible segmentation decisions, then project the FST to the output tape to obtain a determinized FSA for the output space. The FST $\mathbf{x} \circ \mathcal{T}$ is shown in Figure 3.

4 Experiments

We evaluate our proposed method for using large language models for long-form speech translation with three sets of experiments: (1) analysis of hyperparameters, (2) comparison to competing methods, and (3) robustness to speech recognition errors. In each case, we are concerned with translation quality as measured by BLEU. We

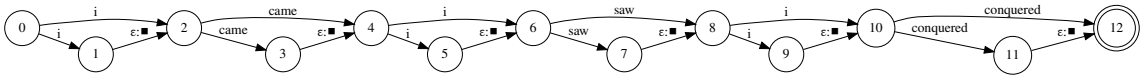

Figure 3: FST representing all possible segmentations for the transcript "i came i saw i conquered".

also assess the LLM output directly by qualitative analysis, well-formedness percentage, and (for diagnostic purposes, following Goldwater et al., 2009) segmentation F_1 score against the sentence-segmented reference.

Our experiments are carried out on the IWSLT spoken language translation data sets, subjected to the same pre-processing as described in Li et al., 2021. We use the 2014 data for dev and 2015 and 2018 for test. The fourteen reference transcripts in our dev set range from 861 to 1234 words; by contrast, the median length of a sentence in written English is close to 17 words (Kučera and Francis, 1970). We use the publicly available Speech-to-Text Google API[1] to generate ASR transcripts. We remove the automatically predicted punctuation and lowercase the ASR transcripts and use English-{German,Spanish,Arabic} machine translation models trained with the same preprocessing on the source side as Li et al., 2021. The MT model is a Transformer with a model dimension of 1024, hidden size of 8192, 16 attention heads, 6 encoder layers, and 8 decoder layers. We decode with a beam size of 4. In our experiments, the three MT model instances and the ASR model (and thereby its transcripts) are fixed while we vary the sentence segmentation policies.

4.1 Context Window Size

In Section 2, we introduced the top-level sliding window inference algorithm above all modeling choices. To compare different models fairly, we fix the hyperparameters $(w, b, r) = (40, 10, 5)$ for the algorithm throughout the experiments. This choice is guided by a linear search over the window lengths w in the range of $[20, 100]$. The overlapping buffer size for both ends is set to 5 heuristically. According to Figure 4, translation quality degrades slightly as window size approaches 20. But very large windows do not appear to be beneficial. The observation validates our guiding principles of the sliding window algorithm.

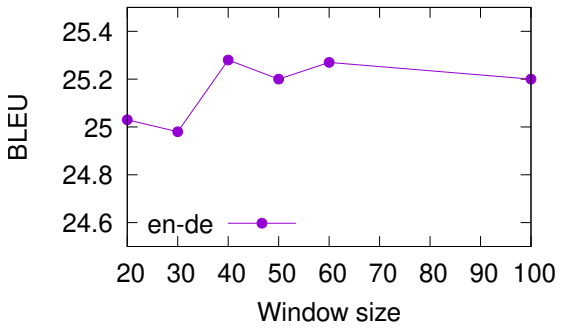

Figure 4: BLEU for English-German as context window size for segmentation increases. Each dot represents a T5 segmentation model trained with the same window size for inference time.

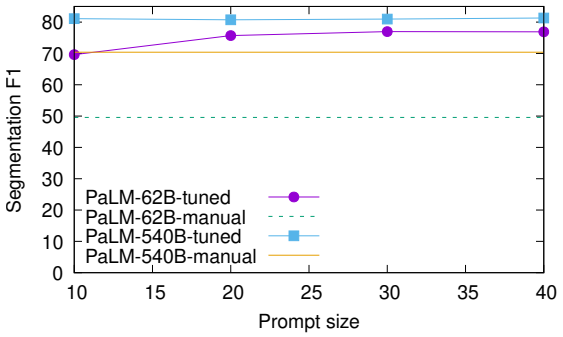

Figure 5: Segmentation F_1 on the dev set as prompt size varies.

4.2 Choice of Prompt

The manual prompt in Figure 2 is the one we selected from a few variants for the decoder-only PaLM models. Instead of exploring the unbounded space of prompts, we resorted to the more principled method of prompt tuning (Lester et al., 2021) to optimize the prompt in the embedding space for the segmentation task. For prompt tuning, the only hyperparameter is the length of the embedding prompt (the embedding size is tied to the corresponding model). In Figure 5, we show that for the PaLM models of 62B and 540B, an embedded prompt as short as 10 can achieve much higher F_1 than our hand-written prompt. But it is also notable that the gap between prompt tuning

[1] https://cloud.google.com/speech-to-text

and manual prompting shrank from 25 percent to 10 percent as the model size increased from 62B to 540B, indicating the increasingly stronger generalization capability of extremely large language models. Based on Figure 5 we select 30 as the size for soft prompts in the main results.

4.3 Effect of Finite State Constraints

We make a contrast between greedy search and beam search, with either the segmentation FST constraint Section 3.2.2 inside the decoder or post-hoc Levenshtein alignment Section 3.2.2 for repairing invalid output. We also vary the model types and model sizes to analyze the impact of constrained decoding in different situations. Table 1 shows that constraints are crucial for smaller models in prompt-tuned scenarios. For example, the rate of output being well-formed is only 14.5% using greedy search for the PaLM 8B model. Even when the model size is increased to 62B, the wellformedness rate is till below 90%. The Levenshtein post-alignment algorithm is effective. But the more general finite state constraint is even more effective. For the 8B model, the improvement in F_1 is 1-2% absolute. For the 62B model, the improvement is nearly 3% absolute. On the other hand, if the cost of fine-tuning is acceptable, LLMs can adapt to this task very well. The fine-tuned T5 base model has a wellformedness rate of 99.4% (the rate is even higher for the T5 11B model: 99.8%). But we shall point out that for the results to be useful to downstream applications, either of the two types of constraints is necessary to completely eliminate hallucinations from LLMs. And the FST constraints are more general and more effective as they affect beam search by rejecting non-wellformed hypotheses during search.

4.4 Main Results: LLMs against Structured Prediction Models

Using the IWSLT TED datasets as preprocessed by Li et al. (2021), we compare LLM models against their approach, two strong custom structured prediction baselines. We also report the performance of an oracle segmenter.

- FIXEDLENGTH – Separates the transcript into disjoint segments with the same number of tokens. While this requires no external segmentation model, the resulting segments are non-sentential (Tsiamas et al., 2022).

MODEL	CONSTRAINT	SEARCH	WELLFORMED	F1
T5 BASE FINE TUNED	UNCONSTRAINED	GREEDY	99.4%	–
		BEAM=4	99.4%	–
	LEVENSHTEIN	GREEDY	100.0%	0.786
		BEAM=4	100.0%	**0.788**
	FST	GREEDY	100.0%	0.786
		BEAM=4	100.0%	**0.788**
PALM 8B PROMPT TUNED	UNCONSTRAINED	GREEDY	14.5%	–
		BEAM=4	52.7%	–
	LEVENSHTEIN	GREEDY	100.0%	0.715
		BEAM=4	100.0%	0.689
	FST	GREEDY	100.0%	0.717
		BEAM=4	100.0%	**0.727**
PALM 62B PROMPT TUNED	UNCONSTRAINED	GREEDY	85.9%	–
		BEAM=4	89.0%	–
	LEVENSHTEIN	GREEDY	100.0%	0.735
		BEAM=4	100.0%	0.737
	FST	GREEDY	100.0%	0.761
		BEAM=4	100.0%	**0.764**

Table 1: Effect of finite state decoding constraints and Levenshtein post alignment on segmentation F_1.

- ORACLE – Uses punctuation from the reference transcripts to segment. The segmentation is projected onto Levenshtein-aligned words in the noisy ASR transcripts (Section 3.2.2).[2]
- PUNCTUATE – An interpretable two-pass segmentation that first infers punctuation (Soboleva et al., 2021), then uses a fixed set of inference rules to differentiate sentence-terminal punctuation marks from sentence-internal ones as in "St. John" and "The end."
- BIGRU F.T. – On the IWSTL data, fine-tunes a shallow biGRU model (Section 3.1) trained on the C4 data set (Raffel et al., 2020) using the same rules in PUNCTUATE to derive sentence boundaries as supervision. The model has 1 left-to-right GRU layer, 1 right-to-left GRU layer, and 1 GRU layer in the decoder. It uses embeddings of character n-gram projections (Zhang et al., 2019).
- T5-{BASE,11B} – Fine-tunes the base or 11B (xxl) T5 model (Raffel et al., 2020) on the IWSLT data.
- T5-11B-ASR – Fine-tunes the 11B T5 model on the ASR output of IWSLT train and dev set. The sentence boundaries are projected from reference transcripts in the same way as ORACLE.
- PALM-PROMPTTUNED-{62B,540B}{,-ASR} – Prompt-tunes the PaLM model

[2] A true oracle would optimize corpus-level BLEU over all 2^n segmentations, but this is intractable.

Policy	F1 TED 2014	EN-DE 2014	EN-DE 2015	EN-DE 2018	EN-ES 2014	EN-ES 2015	EN-ES 2018	EN-AR 2014	EN-AR 2015	EN-AR 2018	Avg
Baselines and Oracle:											
ORACLE	1.000	26.66	30.24	25.21	40.38	41.72	41.84	15.66	18.18	17.59	29.62
FIXEDLENGTH	0.041	20.82	23.45	19.66	32.76	34.03	34.01	12.64	14.79	13.92	23.66
LI ET AL. (2021)	–	–	27.00	22.00	–	–	–	–	–	–	–
Small Structured Prediction Models:											
PUNCTUATE	–	22.80	26.30	21.60	35.70	36.90	36.70	13.70	15.80	15.40	25.81
BIGRU F.T.	0.697	24.55	28.10	23.14	37.31	39.08	38.64	14.41	16.77	16.19	27.39
LLMs:											
T5-BASE	0.788	25.28	29.14	24.05	38.75	40.23	39.96	14.94	17.32	16.57	28.33
T5-11B	0.821	25.63	**29.63**	**24.27**	**39.16**	40.64	**40.05**	**15.31**	17.60	16.48	**28.66**
T5-11B-ASR	**0.836**	25.71	29.28	24.22	39.11	40.47	40.02	15.24	17.58	16.66	28.59
PALM-PROMPTTUNED-62B	0.764	25.10	28.69	23.92	38.52	40.01	39.22	15.03	17.13	16.58	28.08
PALM-PROMPTTUNED-62B-ASR	0.781	25.15	29.09	23.71	38.69	40.07	39.31	15.13	17.21	16.76	28.17
PALM-FINETUNED-62B	0.820	25.71	29.19	23.97	38.96	40.56	39.74	15.07	17.66	**16.90**	28.51
PALM-FINETUNED-62B-ASR	0.832	**25.84**	29.37	24.13	39.02	40.46	39.89	15.17	**17.80**	16.65	28.61
PALM-PROMPTTUNED-540B	0.816	25.44	29.29	24.23	38.95	**40.70**	39.74	15.03	17.61	16.86	28.49
PALM-PROMPTTUNED-540B-ASR	0.835	25.52	29.37	24.15	39.08	40.67	39.98	15.11	17.64	16.61	28.56

Table 2: Segmentation F1 scores on dev set and BLEU scores on dev and test sets, translating into German, Spanish, and Arabic.

(Chowdhery et al., 2022) on the IWSLT data.

- PALM-FINETUNED-62B{,-ASR} – Fine-tunes the 62B PaLM model.

The current peer-reviewed state of the art for long-form speech translation is Li et al. (2021) on the IWSLT data set for EN-DE. Compared to ORACLE, there is still a large gap of 3 BLEU points which can be closed by improving segmentation alone.

Table 2 lists the complete set of results. BIGRU F.T. already beats Li et al. (2021) by more than 1 BLEU point for EN-DE, proving itself as a strong structured prediction baseline. T5 and PaLM models improve the results furthermore. Within the T5 group, T5-11B improves over T5-BASE by 3% in segmentation F_1 which translates to consistent BLEU score improvement in almost all data sets. Within the PaLM group, the prompt-tuned 540B model is about 5% more accurate than the 62B counterpart. Given the large number of parameters, fine-tuning PaLM models is very expensive. For the completeness of comparison, we include the fine-tune result for PaLM 62B. Its result is on-par with the T5 11B model. This fact indicates that T5's encoder-decoder architecture has an induction bias advantage over the PaLM model's decoder only architecture for this task, from a parameter efficiency point of view. But the strength the the PaLM family lies in its largest member. The 540B model can achieve a result as good as the fully fine-tuned T5 11B or PaLM 62B with a tiny tuned prompt.

4.5 Robustness to Speech Recognition Errors

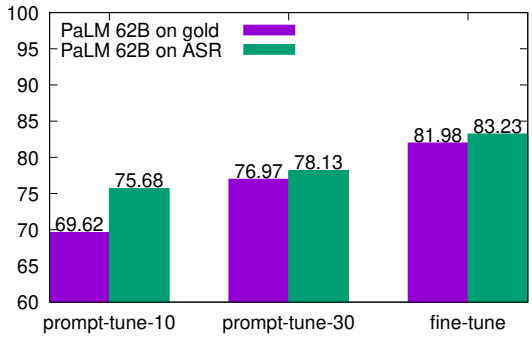

Figure 6: Contrast of segmentation F_1 on the dev set between models trained on gold and ASR transcripts.

One key difference between cascade speech translation and typical document-level translation is that transcription errors can be introduced, which propagate into the translation. When the input to segmentation models contains speech recognition errors, can such models still predict sentence boundaries accurately? The answer is yes, to a certain extent. What we do is switching the tuning data from ground-truth transcripts with punctuation-derived sentence boundaries to ASR transcripts with sentence boundaries projected from their parallel ground-truth transcript counterparts. For example, we will tune the models to predict the segmentation for the passage: *this train leaves at for* ***<SENT>*** *the next train will*

arrive in ten minutes, even though there is a lexical error (*for* versus *four*).

Table 2 shows that training on the ASR transcripts is indeed beneficial. On top of the strong results of the T5 11B model trained on ground-truth transcripts, the ASR version obtains another 1% F_1 improvement. The same is true for the PaLM 62B prompt-tuned and fine-tuned models. Figure 6 shows the relative improvement is consistent across different prompt sizes and fine-tuning methods. But the segmentation accuracy improvement seems to be too small to translate into significant BLEU score improvements.

5 Error Analysis

5.1 Segment Length Histogram Analysis

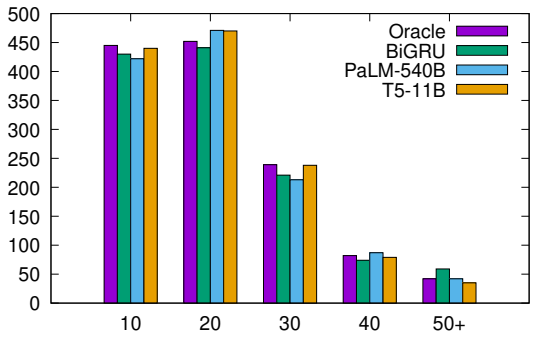

Figure 7: Histograms of segment lengths for ORACLE, BIGRU, PALM PROMPTTUNED 540B ASR, and T5 11B ASR.

To understand the improvements and the remaining errors, we first compare the length distribution of the ORACLE, the small model BIGRU, T5-11B, and PALM-540B. Figure 7 indicates that the more very long (≥ 50) segments a model has, the lower its F_1 and BLEU scores tend to be. Both LLM models were able to reduce the number of very long segments, bringing it closer to the oracle.

5.2 Qualitative Analysis

Table 3 shows examples where the T5-11B-ASR model outperforms competing models. In the first two examples, the LLM model is able to capture the larger context and therefore make the correct prediction. The third example typifies the cases where T5-11B, which is fine-tuned on ground-truth transcripts without ASR errors, tends to make more wrong predictions when the input text is not fluent.

Table 4 lists typical errors the T5-11B-ASR model makes. In the first two, ASR errors make the input difficult to parse. The third one is linguistically ambiguous. In the last one, the model's prediction is actually closer to the ground-truth segmentation than the Levenshein-(mis)aligned ASR transcript.

Overall, LLMs such as T5-11B-ASR made real progress in predictions requiring longer context. However, overcoming ASR errors remains challenging even though fine-tuning on ASR transcripts improved robustness of the models in face of disfluent input.

6 Related Work

Speech translation. While end-to-end systems for spoken language translation have exceeded the performance of cascade models on short sequences (Weiss et al., 2017) even on public data (McCarthy et al., 2020), long-form audio is typically translated with cascades. Previous work uses tagging approaches to separate text into independently translatable units. Segmenting long texts into units suitable for translation has been a recurring topic in MT research (Li et al., 2021; Tien and Thi Minh, 2019; Pouget-Abadie et al., 2014; Doi and Sumita, 2003; Goh and Sumita, 2011). To bridge the gap between ASR and MT, (Li et al., 2021) address long-form spoken language translation. Claiming that segmentation is the bottleneck, they adapt their MT model to work *with* automatic segmentations, however inaccurate they may be.

We are training our models to minimize the loss of source sentence segmentation. The ultimate objective is improving the downstream translation quality. It is interesting to explore reinforcement learning for segmentation (Srinivasan and Dyer, 2021), but the state space is vast for the long-form segmentation problem compared to prior work on RL-based segmentation.

Finally, one may consider additional sources of data or training examples to improve modeling. Using prosodic features when they are available is viable (Tsiamas et al., 2022); however, we show that LLMs close most of the accuracy gap without these. As a contrasting approach, Kumar and Byrne (2002) focus on segmenting an ASR *lattice*, rather than the decoded transcript. Finally, data augmentation (Li et al., 2021; McCarthy et al., 2020) can complement our approach.

Text normalization and segmentation. Mans-

Reference: this great renaissance for ancient egyptian art architecture and religion<SENT>egyptologists have always known the site **ASR**: this great renaissance for ancient egyptian art architecture and religion<SENT>egyptologists have always known the site **B1GRU**: this great renaissance for ancient egyptian art architecture and religion egyptologists have always known the site **T5-11B-ASR**:this great renaissance for ancient egyptian art architecture and religion<SENT>egyptologists have always known the site
Reference: looking for layers of human occupation<SENT>and five meters down underneath a thick layer of mud we found a dense layer of pottery **ASR**: looking for layers of human occupation<SENT>and five meters down underneath a thick layer of mud we found a dense layer of pottery **B1GRU**: looking for layers of human occupation and five meters down underneath a thick layer of mud<SENT>we found a dense layer of pottery **T5-11B-ASR**: looking for layers of human occupation<SENT>and five meters down underneath a thick layer of mud we found a dense layer of pottery
Reference:actually started in 1984 bc at a not-lost-for-long city found from above **ASR**: how actually actually started in 1984 bc at a not lost for long city found from above **T5-11B**: how<SENT>actually actually started in 1984 bc at a not lost for long city found from above **T5-11B-ASR**: how actually actually started in 1984 bc at a not lost for long city found from above

Table 3: Cases where the T5-11B-ASR model is more accurate.

Reference: designers can materialize their ideas directly in 3d and surgeons can practice on virtual organs underneath the screen **ASR**: designers can materialize their ideas directly in 3d sturgeons can practice a virtual audience underneath the screen **T5-11B-ASR**: designers can materialize their idesas directly in 3d<SENT>sturgeons can practice a virtual audience underneath the screen.
Reference: But our two hands still remain outside the screen <SENT> how can you reach inside and interact with the digital information **ASR**: what are two hands still we made outside the screen that <SENT> how can you reach inside and interact with the digital information **T5-11B-ASR**: what are two hands still we made outside the screen that how can you reach inside and interact with the digital information
Reference: this is really what brought me to using satellite imagery<SENT>for trying to map the past i knew that i had to see differently **ASR**: this is really what brought me to using satellite imagery<SENT>for trying to map the past i knew that i had to see differently **T5-11B-ASR**: this is really what brought me to using satellite imagery for trying to map the past<SENT>i knew that i had to see differently
Reference: the equivalent of locating a needle in a haystack blindfolded wearing baseball mitts<SENT>so what we did i **ASR**: the equivalent of locating a needle in a haystack blindfolded wearing baseball<SENT>minutes so what **T5-11B-ASR**: the equivalent of locating a needle in a haystack blindfolded wearing baseball minutes<SENT>so what we did is

Table 4: Cases where the T5-11B-ASR model's prediction is wrong.

field et al. (2019) model text normalization as a sequence-to-sequence problem, using `<self>` tags to bias toward copying, but they place no search constraints to ensure well-formedness. Zhang et al. (2019) is another study that uses finite automata intersected with a neurally generated lattice during decoding.

Wicks and Post (2021) provide a unified solution for segmenting punctuated text in many languages. But ground-truth punctuation is not present in speech recognition output.

Structured prediction as sequence-to-sequence. Vinyals et al. (2015) show that attention-enhanced sequence-to-sequence models can be trained for complex structured prediction tasks such as syntatic parsing. Raffel et al. (2020) takes a step further to model all text-based language problems into a text-to-text format. Paolini et al. (2021) framed many NLP tasks as translation between augmented natural languages.

Constrained decoding. Hokamp and Liu (2017) and Post and Vilar (2018) introduced lexical constraints in neural machine translation beam search. Anderson et al. (2017) formulated lexical constraints as finite state machines. Deutsch et al. (2019) used an active set method to efficiently compose many automata with beam search.

7 Conclusion

We have presented new methods for long-form speech translation by coupling source-side large language models with finite state decoding constraints, allowing large language models to be used for structured prediction with a guarantee for well-formedness in the output space. Finite-state constraints are especially effective when the model is decoder-only, relatively small, or has not been completely fine-tuned (only prompt-tuned, or few-shot-learned) for the structured prediction task. We also observe that even though complete fine-tuning and enlarging model size can reduced the rate of invalid output, models alone are not capable of completely eliminate invalid output.

Fine-tuning on in-domain ASR transcripts containing recognition errors and disfluency outperforms training on clean transcripts in terms of segmentation accuracy. Our qualitative analysis shows the largest category of remaining errors is ASR errors which make transcripts difficult to parse and segment. The fact that LLMs are capable of adapting to ASR errors to some extent points to future research directions of ASR error recovery within context.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, I Te, Wolfgang Macherey, Pallavi Baljekar, and George Foster. 2020. Re-translation strategies for long form, simultaneous, spoken language translation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7919–7923.

James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. 2017. Quasi-recurrent neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *SSST-8*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling language modeling with pathways.

Alexandra Chronopoulou, Matthew Peters, and Jesse Dodge. 2022. Efficient hierarchical domain adaptation for pretrained language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1336–1351, Seattle, United States. Association for Computational Linguistics.

Daniel Deutsch, Shyam Upadhyay, and Dan Roth. 2019. A general-purpose algorithm for constrained sequential inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 482–492, Hong Kong, China. Association for Computational Linguistics.

Takao Doi and Eiichiro Sumita. 2003. Input sentence splitting and translating. In *Building and Using Parallel Texts*.

Chooi-Ling Goh and Eiichiro Sumita. 2011. Splitting long input sentences for phrase-based statistical machine translation. In *ANLP*.

Sharon Goldwater, Thomas L Griffiths, and Mark Johnson. 2009. A bayesian framework for word segmentation: exploring the effects of context. *Cognition*, 112(1):21–54.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *CoRR*, abs/1211.3711.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *WNGT*.

H. Kučera and W.N. Francis. 1970. *Computational Analysis of Present-day American English*. Brown University Press.

Shankar Kumar and William Byrne. 2002. Risk based lattice cutting for segmental minimum Bayes-risk decoding. In *ICSLP 2002*, pages 373–376.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Daniel Li, Te I, Naveen Arivazhagan, Colin Cherry, and Dirk Padfield. 2021. Sentence boundary augmentation for neural machine translation robustness. In *ICASSP 2021*, pages 7553–7557.

Yang Liu, Andreas Stolcke, Elizabeth Shriberg, and Mary Harper. 2005. Using conditional random fields for sentence boundary detection in speech. In *ACL*.

Courtney Mansfield, Ming Sun, Yuzong Liu, Ankur Gandhe, and Björn Hoffmeister. 2019. Neural text normalization with subword units. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 190–196, Minneapolis, Minnesota. Association for Computational Linguistics.

Arya D. McCarthy, Liezl Puzon, and Juan Pino. 2020. Skinaugment: Auto-encoding speaker conversions for automatic speech translation. In *ICASSP*, pages 7924–7928.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, RISHITA ANUBHAI, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *International Conference on Learning Representations*.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Jean Pouget-Abadie, Dzmitry Bahdanau, Bart van Merriënboer, Kyunghyun Cho, and Yoshua Bengio. 2014. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. In *SSST-8*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Daria Soboleva, Ondrej Skopek, Márius Šajgalík, Victor Cărbune, Felix Weissenberger, Julia Proskurnia, Bogdan Prisacari, Daniel Valcarce, Justin Lu, Rohit Prabhavalkar, et al. 2021. Replacing human audio with synthetic audio for on-device unspoken punctuation prediction. In *ICASSP*, pages 7653–7657.

Srivatsan Srinivasan and Chris Dyer. 2021. Better Chinese sentence segmentation with reinforcement learning. In *Findings of ACL*.

Ha Nguyen Tien and Huyen Nguyen Thi Minh. 2019. Long sentence preprocessing in neural machine translation. In *RIVF*, pages 1–6.

Katrin Tomanek, Vicky Zayats, Dirk Padfield, Kara Vaillancourt, and Fadi Biadsy. 2021. Residual adapters for parameter-efficient ASR adaptation to atypical and accented speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6751–6760, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ioannis Tsiamas, Gerard I Gállego, José AR Fonollosa, and Marta R Costa-jussà. 2022. SHAS: Approaching optimal segmentation for end-to-end speech translation. *arXiv preprint arXiv:2202.04774*.

Oriol Vinyals, Ł ukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. In *Proc. Interspeech 2017*, pages 2625–2629.

Rachel Wicks and Matt Post. 2021. A unified approach to sentence segmentation of punctuated text in many languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3995–4007, Online. Association for Computational Linguistics.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.

Hao Zhang, Richard Sproat, Axel H. Ng, Felix Stahlberg, Xiaochang Peng, Kyle Gorman, and Brian Roark. 2019. Neural models of text normalization for speech applications. *Computational Linguistics*.