Improving Spoken Semantic Parsing using Unpaired Text from Textual Corpora and Large Language Model Prompting

Anonymous ACL submission

Abstract

Spoken semantic parsing (SSP) involves generating machine-comprehensible parses from input speech. Training robust models for existing application domains represented in training data or extending to new domains requires corresponding triplets of speech-transcript-semantic parse data, which is expensive to obtain. In this paper, we address this challenge by examining methods that can use or generate transcript-semantic parse data (unpaired text) without corresponding speech. First, when unpaired text is drawn from existing textual corpora, we compare Joint Audio Text (JAT) and Text-to-Speech (TTS) as ways to use unpaired text to generate speech representations. Experiments on the STOP dataset show that unpaired text from existing and new domains improves performance by 2% and 30% in absolute Exact Match (EM) respectively.

Second, when unpaired text is not available from existing textual corpora, Large Language Models (LLMs) can be prompted to generate unpaired text for existing and new domains, and JAT or TTS can be used with the generated unpaired text to improve SSP. Prior work has mostly focused on using LLMs to generate synthetic data for classification tasks. In this paper, we introduce multiple prompting strategies to obtain synthetic data in existing and new domains based on intent classes, intent-slot combinations and example transcripts and parses. Experiments show that using synthetic parse data with JAT for existing domains can improve SSP performance on STOP by 1.4 % absolute EM. Using synthetic parse data with TTS for a new held-out domain improves EM on STOP for the held out domain by 2.6% absolute.

1 Introduction

Spoken Language Understanding (SLU) is essential for many real-world applications today including conversational agents and virtual assistants. Spoken Semantic Parsing (SSP) is the SLU task that involves transforming a recording to a machine-comprehensible parse tree (Wang et al., 2023a). End-to-end models (Arora et al., 2023) operate directly on speech while cascade models (Futami et al., 2023) generate a semantic parse based on the speech transcript. Two-pass deliberation models (Le et al., 2022) combine the best of both worlds, by using first-pass transcripts and speech embeddings to perform spoken semantic parsing within a second pass. However, training such models with supervision requires matched triplets of speech, transcript, and semantic parse. Annotating these triplets is expensive, which limits the size of training data, and consequently model performance.

The need for matched data can be alleviated by developing methods that can use text-only unpaired data. Text data (transcript-semantic parse) is more easily obtained than speech – either from existing textual corpora or by prompting Large Language Models (LLMs), and training models with a small amount of paired speech-text data and a large amount of unpaired text is useful. It is non-trivial to incorporate text-only data into end-to-end models because model outputs cannot be obtained without speech inputs. Prior work has explored the use of text data for speech recognition (Wang et al., 2020a; Toshniwal et al., 2018; Hori et al., 2019). External language models trained on text can be used to interpolate token prediction probabilities (Meng et al., 2022), but require additional memory, making them unsuitable for on-device applications. Coordinated learning methods (Chen et al., 2022; Sainath et al., 2023) project speech and text to a shared embedding space for speech recognition, but such models require significant amounts of paired speech-text data to learn robust mappings. The final class of work generates speech representations for unpaired speech - Joint Audio Text (JAT) (Kim et al., 2022) uses mean speech embeddings from paired data to represent unpaired

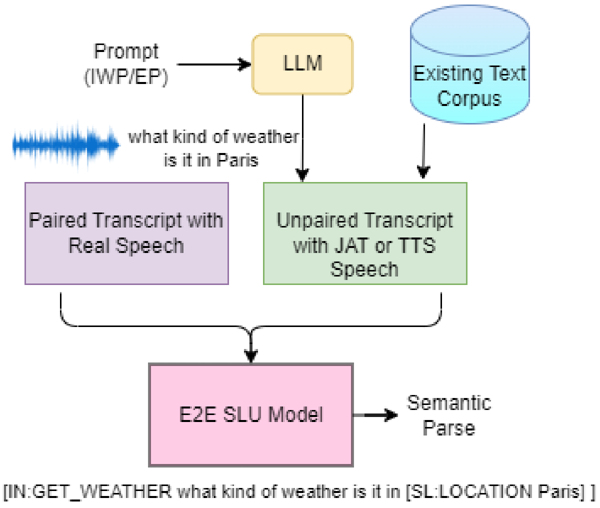

Figure 1: This paper: We describe ways to unpaired text to train deliberation models, where unpaired data can be obtained from LLMs or existing textual corpora. We use JAT or TTS to obtain speech representations of unpaired data

text. This is computationally inexpensive, but the speech embeddings do not contain information embedded in real speech. In contrast, synthetic speech from Text-to-speech (TTS) models (Wang et al., 2020a) produce informative speech representations, but they can be expensive to compute.

There are two cases where additional textual data may be acquired for semantic parsing – (a) to improve models on existing domains (ED) and (b) to support new domains (ND). In this paper, we compare JAT and TTS for SSP when unpaired text data is drawn from these two setups - ED and ND.

When unpaired text is not available from existing corpora, we propose to prompt Large Language Models (LLMs) (Ouyang et al., 2022; Touvron et al., 2023a,b) to generate textual data for SSP. LLMs have been used in prior work to generate synthetic data for text classification using approaches such as Self-Instruct (Wang et al., 2023b), Attr-Prompt (Yu et al., 2023a), ZeroGen (Ye et al., 2022), and more recently use in-context learning with seed samples (Yu et al., 2023b). Semantic parsing requires sequence labeling, i.e., (a) it requires the correct identification of identification of the number and identity of intent and slot tags, and (b) correct placement of entity and slot tags to form the right parse tree, all while not inserting unrelated or unseen intent and slot tags. Therefore, it is more complex to generate useful and diverse data for semantic parsing compared to other classification tasks.

Prior work (Tran and Tan, 2020) has proposed the use of template-based masked training of BART to produce additional variants for masked words, however this limits the potential lexical diversity of the generated data, and requires significant amount of labeled data, which may not be available for the ND setting. Since LLMs can learn in-context and generalize better under few-shot settings, they consequently need fewer exemplars to generate diverse and high quality synthetic data for semantic parsing. In this paper, we address the task of generating synthetic text data for semantic parsing by using different prompting approaches with Llama 2.

For the ED setup, it is sufficient to generate transcripts (similar utterances) since semantic parses can be obtained from transcripts using pre-trained semantic parsers. We describe two prompting methods: (a) intent-word-based prompting (IWP), where the LLM produces transcripts corresponding to a particular intent class and containing words that co-occur with the intent, and (b) exemplar-based prompting (EP), where it generates transcripts that are similar to provided examples. We generate pseudo-labels for the generated utterances using a pre-trained RoBERTa (Liu et al., 2020) model and train SSP models using JAT. We find that EP is simpler but IWP generates the desired intent more often. Using data from both methods improves the Exact Match (EM) on STOP data by 1.4 points absolute.

For the ND setup, pre-trained models for pseudo-labeling are unavailable for the new domain(s), and hence LLMs are used to generate the seqlogical form (containing the transcript with intent and slot tags annotated) of semantic parses directly. The transcript is then inferred from the seqlogical form of the semantic parse. Exemplar-based prompting (EP) is used with 3 real examples for every possible intent-slot combination to generate large-scale data. We find that the generated data improves EM by 2.3 points absolute over a baseline that uses only 3 examples per combination.

In summary, this paper makes the following contributions:

1. Extends JAT, previously used for ASR, to end-to-end spoken semantic parsing, and compares JAT with TTS for textual data from existing domains and new domains.

2. Develops prompting strategies to generate textual transcripts and semantic parses in existing and new domains using LLMs.

3. Demonstrates that LLM-generated textual data can be used in conjunction with JAT and TTS to improve spoken semantic parsing.

2 Deliberation Model for SLU

Deliberation-based SLU models (Le et al., 2022; Kim et al., 2023) are two-pass models that predict an ASR transcript in the first pass. Using the first pass transcript and audio, it then generates the semantic parse in the second pass. In contrast to cascade models that utilize separately trained Automatic Speech Recognition (ASR) and SLU components, a deliberation model optimizes both ASR and SLU components jointly. To achieve on-device streaming functionality, the first pass ASR component is implemented using the Recurrent Neural Network Transducer (RNNT) (Graves, 2012; Kim et al., 2021; Liu et al., 2021).

To maintain transcription accuracy, the ASR component of our deliberation model is trained independently and kept frozen. Our deliberation-based SLU model comprises two primary modules: (1) Fusion, and (2) Decoder. The fusion module combines intermediate audio and text embeddings from the first pass RNNT encoder and predictor respectively. Using Multi-Head Attention (Vaswani et al., 2017), the fusion module generates a combined representation that is used by the transformer-based decoder module to predict the target semantic parse sequence.

3 Speech Representations for Unpaired Text

3.1 Joint Audio-Text Training (JAT)

Joint Audio-Text training (JAT) (Kim et al., 2022) is a recent approach for leveraging unpaired text-only data to improve ASR (Kim et al., 2022; Sainath et al., 2023, 2020; Wang et al., 2020b). Unlike shallow fusion that considers token distributions from an external neural network language model (NNLM), JAT does not require additional model parameters or latency, making it suitable for on-device streaming ASR. The core idea behind JAT is that speech representations for unpaired text can be generated by simply using average speech embeddings computed over available paired speech/text data. In this paper, we use the JAT approach to train our Spoken Language Understanding (SLU) models to enable training with both "speech-text-semantic parse" and "text-semantic parse" datasets.

3.2 Speech Synthesis with Voicebox

Voicebox(Le et al., 2023) is a state-of-the-art non-autoregressive speech generation model based on Flow Matching (Lipman et al., 2022). We generate representations for unpaired text by extracting speech features from synthesized speech. Synthetic speech can be obtained by using Voicebox in TTS mode, i.e. where audio is generated by conditioning on input text. Different from (Le et al., 2023), the Voicebox model we use represents input text as graphemes rather than phonemes. To generate audio, we first sample unit durations for each grapheme in the input text using a flow-matching-based duration model and then upsample the grapheme sequence using the unit duration information. This information is used as conditioning to generate the spectrogram using the audio model. Finally, we used a HiFi-GAN (Kong et al., 2020) vocoder to convert the spectrograms into time-domain signals.

4 Generating Textual Data with LLama 2.0

LLama 2.0 (Touvron et al., 2023b) is a public open-source large language model trained on large volumes of publicly available data and code with context as large as 4096. In this paper, we use the 13B parameter chat model.

4.1 Generating Textual Data for Existing Domains

In the ED setup, we propose to use LLMs to generate transcripts. Corresponding semantic parses are obtained using a pseudo-labeling textual semantic parse model trained on existing paired data. The semantic parse model here takes transcripts as inputs and produces pseudo-label semantic parses as output. Transcripts can be generated using one of two prompting strategies, i.e., intent-word-based or exemplar-based.

4.1.1 Intent Word-based prompting (IWP)

The goal of IWP is to generate transcripts that may be classified under a certain intent, optionally containing "intent words". Intent words are the words from semantic parses that occur most frequently with given intents after removing stop-words. An example is shown in Figure 2. The 40 words that co-occur most frequently with every intent in the STOP data are used as intent words. 40 examples are generated for every intent and intent-word combination. Though IWP produces good synthetic data, it is limited by the fact that words that co-occur less frequently with the intent are less related to the intent. Such examples produced with less relevant intent words may not be classified under the desired intent class. This also limits the amount of synthetic data that can be generated since the LLM cannot generate many unique examples using a small number of intent-intent word combinations.

4.1.2 Exemplar-based Prompting (EP)

Since LLMs are strong in-context learners (Wei et al., 2022), an alternative approach is to prompt LLMs to generate transcripts based on examples. For every intent-slot combination, we provide up to 4 random example transcripts and ask the model to generate 60 more transcripts that are similar but have diverse sentence structures. An example prompt is shown in Fig 3. Though the resulting transcripts may not always correspond to the intent classes from which the examples are drawn, this method enables us to generate larger volumes of data without duplication.

4.1.3 Semantic Parse generation and Quality Assessment

Transcripts generated by LLMs are first normalized – written text is converted to spoken form, punctuation except apostrophes are removed and text is transformed into lower case. Semantic parse pseudo-labels are obtained from these normalized transcripts using a strong RoBERTa-based semantic parser trained on STOP (EM=86.8). To assess data quality, we compare the intent in the obtained pseudo-labels to the intent in the prompt for IWP or the intent of the provided examples for EP. Intent Match Accuracy (IMA) is defined as the percentage of times the intent of the pseudo-label matches the desired intent of the prompt.

4.2 Generating Transcript-Semantic Parse for New Domains

For new domains, paired data and pre-trained models are not available, and therefore, we would need to directly generate pairs of transcript and semantic parse. One way to do this is to generate pairs of semantic parse and corresponding transcript using LLMs directly, however, maintaining consistency across generated parses and transcripts is challenging for current LLMs. Another alternative is to generate only the seqlogical form of the semantic parse from the LLM and infer the transcript from the parse. The seqlogical form of the parse, unlike the decoupled form, comprises all the words in the transcript along with slot and intent tags. Therefore, the transcript can be obtained from the seqlogical parse merely by removing slot and intent tags.

4.2.1 Exemplar-based Prompting

We assume that (a) the intents and slots that must be recognized for the new domain are known, (b) the slots that may occur with every intent, i.e., the intent-slot combinations are known, and (c) some manually annotated examples for every intent-slot combination are known. Using this information, LLMs can be prompted as shown in Figure 4 to produce new seqlogical parses for a given intent-slot combinations. The prompt first describes the steps to generate a valid seqlogical parse and then presents up to 3 examples of seqlogical parses with the desired intent-slot combinations.

4.2.2 Post-processing

The generated seqlogical parses are checked for invalid placement of brackets, and Out of Vocabulary (OOV) intents and slots. OOV intents were fixed by re-prompting the model to replace OOV intents with correct intents and replace any intents other than the first. Any OOV slots are removed while retaining corresponding slot words.

Intent Word based Prompting for Utterance Generation

You are working in an intent-and-slot framework where every utterance can be classified under an intent. Here are some examples of intents and a description of their function:

1. IN:ADD_TIME_TIMER - Creates a new timer
2. IN:GET_ESTIMATED_DEPARTURE - gets estimated departure

Now, we want to classify intents for the weather application. Given the intents IN:GET_WEATHER, generate 40 utterances that are classified under this intent. You may use the word "weather" along with names of people and places to generate 40 utterances.
Your response should have numbered utterances, with one utterance on each line. Make sure not to repeat any responses. Start with 1.

Figure 2: Example Prompt for IWP-based utterance generation

Exemplar based Prompting for Utterance Generation

Generate 60 more sentences that are similar in intent to the following sentences:

1. Is it going to be around 95 in degree Fahrenheit san francisco tomorrow
2. Is it around 72 in degree celsius karachi tonight

Write one sentence per line. Generate statements and questions with different sentence structure.

Figure 3: Example Prompt for EP-based utterance generation

Exemplar based Prompting for Seqlogical Semantic Parse Generation

Each sentence should be enclosed in square brackets [ ]. The first square bracket [should be followed by an intent that is in uppercase letters and begins with IN:, for example, IN:GET_WEATHER. Inside the sentence, you should label some nouns with slots, which are also enclosed in brackets [ ]. Slots are in all uppercase letters and begin with SL:, for example, SL:LOCATION. In each sentence, there can only be 1 intent, but there can be many slots. Here are some examples:

1. [IN:GET_WEATHER what kind of weather is in [SL:LOCATION paris]]
2. [IN:GET_WEATHER what is the temperature at the [SL:LOCATION north pole]]
3. [IN:GET_WEATHER tell me what the weather in [SL:LOCATION central park] is like]

Please generate more examples with the intent IN:GET_WEATHER and any of the slots SL:LOCATION. The sentences should have an intent/slot format like [IN:GET_WEATHER [SL:LOCATION]], but with some other text, like the examples above. Write 30 similar sentences and then stop. Use names of people and places in your examples.

Figure 4: Example Prompt for EP-based generation of seqlogical parses

5 Experimental Setup

5.1 STOP Data, Model and Metrics

Data: STOP [1] (Tomasello et al., 2023) is a public dataset with 100 hours of real speech for spoken semantic parsing. STOP has data for 8 domains - alarm, event, messaging, music, navigation, reminder, timer, and weather. The data contains 28 unique intents and 82 slot types in all. Table 1 summarizes some statistics about the STOP dataset.

[1] STOP was used in accordance with its LICENSE terms

Table 1: Dataset and Partition Statistics - STOP Dataset

Partition	Number of utterances
train	120,903
eval	33,380
test	75,617

Metrics: Exact Match (EM) is used to evaluate all our models. We report EM (No Err) and EM w/ Err, which are the Exact Match accuracies averaged over utterances with no ASR error and averaged over utterances with any ASR error respectively.

Model Configuration: For the ASR module, we use RNNT with 3 layers of conformer in the encoder, 1 layer of LSTM in the predictor, and 1 linear layer in the joiner. For the deliberation model, we use attention in the Fusion module, 2 transformer encoder layers in the Pooling module, and a transformer decoder layer with a pointer-generator in the Decoder module (Kim et al., 2023). Models are optimized with Adam (Kingma and Ba, 2015), having a peak learning rate of 8e-3.

Voicebox TTS Model: We use a Voicebox model trained on approximately 14k hours of manually transcribed data that comprises a diverse range of speakers, accents, topics, and acoustic conditions. The audio model has 12 transformer layers (Vaswani et al., 2017) containing 16 attention heads, convolutional positional embeddings (Baevski et al., 2020) and ALiBi self-attention bias (Press et al., 2021). Graphemes are embedded into 80-d features and concatenated with the 80-d log-mel features. The duration model has 8 transformer layers with 8 heads, and graphemes are embedded into 40-d features. Training hyper-parameters are similar to the setup described in (Le et al., 2023).

Computational Cost : Our experiments were performed on a single node with 8 V100-32 GB GPUs on the cluster. Each run took approximately 18 hours for model training. For LLama2 inference, we used 4 x V100-32 GB or 2 x A100-40GB with model parallelism and fp32 precision. For Voicebox inference, we used 1X V100-32 GB GPUs over 40 parallel processes to speed up speech synthesis.

5.2 Setup: Textual Data from Text Corpora

For experiments where we assume textual data is available, we split the STOP datasets into two parts. We perform two experiments – one using the first and second splits as paired and unpaired data respectively and the other using the second and first splits as paired and unpaired data respectively. The average performance across these 2 experiments is reported in each case. In the ED setup, equal amounts of data from every domain are present in the two splits. For the ND setup, STOP is split by domain, where one split contains all training data from 4 domains(messaging, reminder, time, and weather), while the other split contains training data from the other 4 domains (alarm, event, music, and navigation). Both splits are designed to ensure that they have a nearly equal number of utterances.

5.3 Setup: Textual Data from LLMs

When unpaired data is not available, we use Llama 2.0 to generate examples for the ED and ND setups. For the ED setup, LLama 2.0 is used to generate utterances. We then use a pre-trained 12-layer RoBERTa model trained on STOP to generate pseudo-labels for the generated utterances. We augment STOP with the generated LLama 2.0 transcript-semantic parse. JAT is used to represent LLama 2 text.

For the ND setup, LLama 2.0 generated data is not suitable as a real test set since it does not have matching real speech. Therefore, we choose to partition the existing STOP data into 7 seen domains and 1 new domain - weather. We use exemplar-based prompting to generate transcript-semantic parse pairs for weather. For this, real examples of transcript-semantic parse from STOP are used. We use TTS to generate equivalent speech representations for the generated data. We compare the performance on the weather domain for models trained on (a) 7 domains of STOP, (b) 7 domains of STOP with examples for the weather (with TTS for examples and real speech for 7 domains), (c) 7 domains of STOP with examples and Llama 2.0 generated data, and (d) the topline that uses 7 domains of STOP with real data and TTS.

6 Experimental Results and Discussion

6.1 When textual data is available

Table 2 compares the performance of different models for the ED and ND settings where unpaired text is drawn from existing domains and new domains respectively. Across both ED and ND setups, we find that the use of unpaired text improves EM scores.

For the ED setup, we find that JAT and TTS achieve similar Exact Match scores. Since JAT is comparable in performance to TTS and relatively inexpensive compared to complex TTS models like Voicebox, JAT is optimal for the ED setup. TTS model training depends on the specific model, but in our case Voicebox training takes 3 days on 8 GPUs, and inference to produce synthetic speech takes 3 hours on 40 parallel GPU inference jobs. In comparison, JAT data preparation involves using mean speech embeddings, which takes 1 hour on 40 CPUs for the STOP training, evaluation and test data. Therefore, JAT indeed takes little time in comparison to TTS.

Further, the difference between JAT and TTS

Table 2: Comparing JAT and TTS as speech representations for unpaired text from ED and ND. Number of paired and unpaired utterances, and Exact Match (EM) is reported

	Model	#Pair/#Unpair	EM	EM(No Err)	EM w/ Err
ED	Baseline	60.4k / 0	64.25	80.51	24.37
	w/ JAT	60.4k / 60.4k	66.92	83.90	25.25
	w/ TTS	60.4 / 60.4k	**67.05**	**83.88**	**25.80**
ND	Baseline	60.7k / 0	33.28	41.32	13.54
	w/ JAT	60.7k / 60.1k	57.74	73.34	19.50
	w/ TTS	60.7k / 60.1k	**63.95**	**80.70**	**22.88**
	Topline	120.9k / 0	67.67	84.52	26.34

Table 3: Impact of Paired-Unpaired Data Ratio on JAT Performance under the Existing Domain Setting

Paired Data (%)	Unpaired Data (%)	EM-No Err	EM-ASR Error	EM (overall)
0	100	85.48	21.22	66.87
30	70	84.27	24.67	67.01
50	50	84.15	25.5	67.17
70	30	84.24	25.43	67.2
100	0	84.52	26.34	67.67

appears to be primarily on utterances with ASR errors, since synthetic speech representations can be used to reduce the impact of ASR errors on semantic parsing. For the ND setup, we find that though JAT outperforms the baseline, TTS outperforms JAT. This is because new domains may have different entities and domain-specific terms that may be harder to recognize, and TTS provides valid speech representations that can be used to improve predictions based on the first-pass ASR. Figure 5 shows that the amount of unpaired textual data is increased with constant paired data, relative gains increase to a point and saturate.

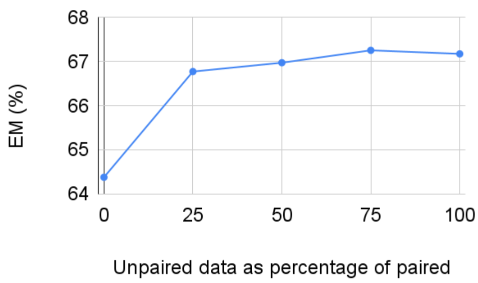

Figure 5: Impact of increasing unpaired text on EM

6.2 Ablation: Does JAT work for different data ratios?

In this experiment, we vary the amount of paired data with speech-transcript-semantic parse and unpaired data with text only to analyze the impact on spoken semantic parsing performance.

From Table 3, we find that JAT works with only a 0.8 % degradation compared to the topline that uses 100% paired data in Exact Match even when no paired speech-text data is used. Therefore, this approach can generalize reasonably to other data ratios apart from the 50-50 ratio used in prior experiments. Further, utilizing more paired data improves performance on the cases when the transcript contains errors when compared to those where the transcript has no errors. This follows as a consequence of the fact that the transcript for unpaired text contains no ASR errors.

6.3 LLama 2.0 Generated Data: ED Setup

Table 4 compares various prompting strategies for generating utterances in the same domain using Llama 2.0. We find that combining LLama-generated data with existing STOP data can improve performance across test examples with and without ASR errors. On further analysis, we find that significant improvements are observed across domains with relatively poor performance in the STOP baseline. Between IWP and EP, we find that EP is slightly better. Since EP is not constrained to generate utterances that may be classified under a given intent, the Intent Match Accuracy (IMA) is lower than that of IWP. Combining the data generated from both these strategies further improves performance over the STOP baseline.

6.4 LLama 2.0 Generated Data: ND Setup

Table 5 compares the performance of baseline models that have no data for weather or 360 examples for weather with models that use LLama 2.0 gen-

Table 4: Assessing the impact of augmenting the training data with LLama 2.0 generated utterances and RoBERTa pseudo-labels.EM is Exact Match Accuracy

Model	#Utts	IMA	EM	EM(No Err)	EM w/ Err
STOP Baseline	160k	-	67.37	84.52	26.34
+ IWP-JAT	230k	**68.87**	68.12	84.96	26.82
+ EP-JAT	218k	64.24	68.21	85.01	**27.04**
+ (IWP+EP)-JAT	298k	67.87	**68.75**	**85.82**	26.86

Table 5: Using TTS to generate speech for LLama 2.0 text when unpaired text is in an unseen new domain

Model	#Utts(Weather)	Weather EM	Overall EM
STOP 7 dom.	0	0	54.61
+ 3 real example-TTS	360	48.18	61.80
+ Exemplar LLama2-TTS	2,910	**50.82**	**62.29**
Topline: STOP Weather-TTS	2,910	63.80	66.33

erated data. Llama 2 generated text can improve performance by over 2 points absolute EM but lags behind the performance of a topline that uses data from STOP.

6.5 Challenges of using LLMs for generating large-scale data

While large language models can generate useful data based on the prompting strategies employed, there are certain challenges with generating large scale data, i.e., something of the order of few to many thousands of utterances.

LLMs can be reasonably consistent while responding within the current turn, but tend to repeat previously proposed examples after around 40 examples per input prompt, with variance arising from the complexity of different semantic parse structures. Due to input context limits while training, there is a limited number of unique and useful examples that can be elicited for every input prompt. It could be argued that each prompt can be presented multiple times with slight variations to obtain more data. However, LLMs are often not consistent across turns and end up repeating synthetic examples. One solution to this challenge could potentially involve using the "chat" formulation, where previous prompts and responses are part of the hidden states the model can attend to while producing new responses. However, due to memory limits, it is challenging to retain very long contexts in input memory, inhibiting the production of truly large scale data.

In this paper, we attempted to sample multiple times using different temperatures and seeds for every prompt to attempt to scale the obtained data. This remains an interesting problem for future work.

7 Conclusion

We address the high cost of manually labeling speech-transcript-semantic parse data for spoken semantic parsing by enabling models to use text-only data. JAT is preferred for unpaired text in existing domains for its efficiency and gain of 2.5 % EM over a paired data baseline while remaining within 0.1 % EM of the more computationally expensive TTS. For unpaired text in new domains, TTS outperforms JAT by 6 % absolute EM overall, with a gain of 30.6 % EM over a paired baseline. When text data cannot be obtained from existing text corpora, we propose to prompt LLMs to generate transcript-semantic parse pairs. We show that using different prompting strategies, we can generate unpaired text data in relatively large volumes. Using JAT and TTS, we can leverage this LLM-generated data to further improve SSP by 1.4 % EM and 2.6 % EM absolute for existing and new domains.

Limitations

Our work uses the public open-source LLama2 LLM to generate synthetic data due to its open source code, public model weights and deterministic generation behavior. However, prompting behavior is not standard across all LLMs, and though the general structure and strategy behind our prompting can remain the same, specific and small modifications may need to be made for different LLMs.

The STOP dataset, the only public dataset for semantic parsing uses real but read speech, rather than spontaneous speech. Making public data with spontanous speech and experimenting with such will definitely be useful to explore.

Impact and Risks

Our work will enable the development of SLU models for tasks and languages where we have very limited labelled data. We hope that this work also spurs more collaboration across the fields of speech and natural language processing, both of which are needed to make progress in this area.

All the work in this paper was done in such a manner so as to minimize the risk of misuse and bias. Since the approach uses LLama to generate synthetic data, potential risks include the percolation of inherent biases in LLama into models trained on such synthetic data.

References

Siddhant Arora, Hayato Futami, Shih-Lun Wu, Jessica Huynh, Yifan Peng, Yosuke Kashiwagi, Emiru Tsunoo, Brian Yan, and Shinji Watanabe. 2023. A study on the integration of pipeline and e2e slu systems for spoken semantic parsing toward stop quality challenge. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–2. IEEE.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Zhehuai Chen, Yu Zhang, Andrew Rosenberg, Bhuvana Ramabhadran, Pedro J. Moreno, Ankur Bapna, and Heiga Zen. 2022. MAESTRO: Matched Speech Text Representations through Modality Matching. In *Proc. Interspeech 2022*, pages 4093–4097.

Hayato Futami, Jessica Huynh, Siddhant Arora, Shih-Lun Wu, Yosuke Kashiwagi, Yifan Peng, Brian Yan, Emiru Tsunoo, and Shinji Watanabe. 2023. The pipeline system of asr and nlu with mlm-based data augmentation toward stop low-resource challenge. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–2. IEEE.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Takaaki Hori, Ramon Astudillo, Tomoki Hayashi, Yu Zhang, Shinji Watanabe, and Jonathan Le Roux. 2019. Cycle-consistency training for end-to-end speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6271–6275. IEEE.

Suyoun Kim, Ke Li, Lucas Kabela, Rongqing Huang, Jiedan Zhu, Ozlem Kalinli, and Duc Le. 2022. Joint audio/text training for transformer rescorer of streaming speech recognition. *EMNLP*.

Suyoun Kim, Yuan Shangguan, Jay Mahadeokar, Antoine Bruguier, Christian Fuegen, Michael L Seltzer, and Duc Le. 2021. Improved neural language model fusion for streaming recurrent neural network transducer. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7333–7337. IEEE.

Suyoun Kim, Akshat Shrivastava, Duc Le, Ju Lin, Ozlem Kalinli, and Michael L Seltzer. 2023. Modality confidence aware training for robust end-to-end spoken language understanding. *Interspeech*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033.

Duc Le, Akshat Shrivastava, Paden Tomasello, Suyoun Kim, Aleksandr Livshits, Ozlem Kalinli, and Michael L Seltzer. 2022. Deliberation model for on-device spoken language understanding. *Interspeech*.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. 2023. Voicebox: Text-guided multilingual universal speech generation at scale. *arXiv preprint arXiv:2306.15687*.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2022. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.

C. Liu, F. Zhang, D. Le, S. Kim, Y. Saraf, and G. Zweig. 2021. Improving RNN Transducer Based ASR with Auxiliary Tasks. In *Proc. SLT*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Zhong Meng, Yashesh Gaur, Naoyuki Kanda, Jinyu Li, Xie Chen, Yu Wu, and Yifan Gong. 2022. Internal Language Model Adaptation with Text-Only Data for End-to-End Speech Recognition. In *Proc. Interspeech 2022*, pages 2608–2612.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Ofir Press, Noah A Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*.

Tara N Sainath, Ruoming Pang, Ron J Weiss, Yanzhang He, Chung-cheng Chiu, and Trevor Strohman. 2020. An attention-based joint acoustic and text on-device end-to-end model. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7039–7043. IEEE.

Tara N Sainath, Rohit Prabhavalkar, Ankur Bapna, Yu Zhang, Zhouyuan Huo, Zhehuai Chen, Bo Li, Weiran Wang, and Trevor Strohman. 2023. Joist: A joint speech and text streaming model for asr. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 52–59. IEEE.

Paden Tomasello, Akshat Shrivastava, Daniel Lazar, Po-Chun Hsu, Duc Le, Adithya Sagar, Ali Elkahky, Jade Copet, Wei-Ning Hsu, Yossi Adi, et al. 2023. Stop: A dataset for spoken task oriented semantic parsing. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 991–998. IEEE.

Shubham Toshniwal, Anjuli Kannan, Chung-Cheng Chiu, Yonghui Wu, Tara N Sainath, and Karen Livescu. 2018. A comparison of techniques for language model integration in encoder-decoder speech recognition. In *2018 IEEE spoken language technology workshop (SLT)*, pages 369–375. IEEE.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ke Tran and Ming Tan. 2020. Generating synthetic data for task-oriented semantic parsing with hierarchical representations. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, pages 17–21, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Gary Wang, Andrew Rosenberg, Zhehuai Chen, Yu Zhang, Bhuvana Ramabhadran, Yonghui Wu, and Pedro Moreno. 2020a. Improving speech recognition using consistent predictions on synthesized speech. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7029–7033.

Peidong Wang, Tara N Sainath, and Ron J Weiss. 2020b. Multitask training with text data for end-to-end speech recognition. *arXiv preprint arXiv:2010.14318*.

Sid Wang, Akshat Shrivastava, and Sasha Livshits. 2023a. Treepiece: Faster semantic parsing via tree tokenization.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. ZeroGen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023a. Large language model as attributed training data generator: A tale of diversity and bias.

Yue Yu, Yuchen Zhuang, Rongzhi Zhang, Yu Meng, Jiaming Shen, and Chao Zhang. 2023b. ReGen: Zero-shot text classification via training data generation with progressive dense retrieval. In *Findings of the Association for Computational Linguistics: ACL*

2023, pages 11782–11805, Toronto, Canada. Association for Computational Linguistics.

A Example Appendix

This is an appendix.