RETRO NLU: Retrieval Augmented Task-Oriented Semantic Parsing

Anonymous ACL submission

Abstract

While large pre-trained language models accumulate a lot of knowledge in their parameters, it has been demonstrated that augmenting it with non-parametric retrieval-based memory has a number of benefits ranging from improved accuracy to data efficiency for knowledge-focused tasks such as question answering. In this work, we apply retrieval-based modeling ideas to the challenging complex task of multi-domain task-oriented semantic parsing for conversational assistants. Our technique, RETRONLU, extends a sequence-to-sequence model architecture with a retrieval component, which is used to retrieve existing similar samples and present them as an additional context to the model. In particular, we analyze two settings, where we augment an input with (a) retrieved nearest neighbor utterances (utterance-nn), and (b) ground-truth semantic parses of nearest neighbor utterances (semparse-nn). Our technique outperforms the baseline method by \( 1.5\% \) absolute macro-F1, especially at the low resource setting, matching the baseline model accuracy with only 40% of the complete data. Furthermore, we analyse the quality, model sensitivity, and performance of the nearest neighbor retrieval component’s for semantic parses of varied utterance complexity.

1 Introduction

Roberts et al. (2020) demonstrated that neural language models quite effectively store factual knowledge in their parameters without any external information source. However, such implicit knowledge is hard to update, i.e. remove certain information (Bourtoule et al., 2019), change or add new data and labels. Additionally, parametric knowledge may perform worse for less frequent facts, which don’t appear often in the training set, and “hallucinate” responses. On the other hand, memory-augmented models (Sukhbaatar et al., 2015) decouple knowledge source and task-specific “business logic”, which allows updating memory index directly without model retraining. Recent studies showed their potential for knowledge-intensive NLP tasks, such as question answering (Khandelwal et al., 2020b; Lewis et al., 2020c).

In this work, we explore RETRONLU: retrieval-based modeling approach for task-oriented semantic parsing problem, where explicit memory provides examples of semantic parses, which model needs to learn to transfer to a given input utterance. An example semantic parse for task-oriented dialog utterance and its corresponding hierarchical representation are presented in Figure 1.

![Figure 1: An intent-slot based compositional semantic parsing example(coupled) from TOPv2 (Chen et al., 2020).](image)

In this paper we are focusing on the following questions: Data Efficiency: Can retrieval based on non-parametric external knowledge alleviate reliance on parametric knowledge typically acquired via supervised training on large labeled datasets? We examine how different training settings, depending on the amount of supervision data available, impact model prediction, i.e. fully supervised vs. limited supervised training. Semi-supervised Setting: Can we enhance models by using abundant and inexpensive unlabeled external non-parametric knowledge rather than structurally labeled knowledge? We examine the effect of uti-

---

1 Parametric knowledge is information stored in model parameters. Non-parametric knowledge refers to external data sources that the model uses to infer.
We examine whether ex-
hierarchical) examples? 
 Robustness to Noise: Can a model opt to employ parametric knowledge rather than non-parametric knowledge in a resilient manner, for example, when the non-parametric information is unreliable? We examine the model’s resilience and its reliance on non-parametric external information. External knowledge is not always precisely labeled and reliable for all examples/utterances. Utterance Complexity: Is non-parametric external knowledge addition effective for both uncommon and complex structured (hierarchical) examples? We examine whether external knowledge addition is more beneficial in certain cases than others, or if it supports accurate predictions for all situations equally. It would be fascinating to investigate if external information could also help enhance difficult and complex examples/utterances. Finally, we examine the upper limit on the utility of external information. We examine structural redundancy concerns in nearest neighbor retrieval. Knowledge Efficiency: Is it beneficial to continue adding external information, or are there certain boundaries and challenges? Our contribution are as follows:

1. We demonstrate that combining parametric and non-parametric knowledge enhance model performance on the complex structured task of task-oriented semantic parsing.

2. We illustrate the effectiveness of our approach in a critical situation of learning with sparse labeled data (i.e. limited parametric knowledge).

3. We establish the efficacy of retrieval-based method in semi-supervised settings, where model’s input is supplemented with unannotated instances.

4. By comparing predictions on clean vs. noisy neighbours, we establish the model’s resilience to external non-parametric knowledge quality.

5. Finally, we examine performance gains with inputs of varying complexity: semantic structure composition and it’s frequency (i.e. frequent/rare).

Overall, we demonstrate that retrieval enhanced method can improve performance on complicated structured prediction tasks like task oriented semantic parsing without extra supervision. Furthermore, the augmentation approach is data efficient and performs well in low resource settings with limited label data. The dataset, and associated scripts, will be available at anonymous_for_submission.

2 Proposed Approach

Our proposed approach has four main steps (1) **index building**, by embedding training examples and computing cosine similarity; (2) **retrieval**, given an example utterance, retrieve nearest neighbor utterances from the index. (3) **augmentation**, append the nearest-neighbor utterance ground truth semantic parse (semparse-nn) or utterance itself (utterance-nn) to the original input using a special separator token (such as ‘|’), and (4) **semantic parsing**, use the retrieval-augmented input with output ground truth for parsing model training.

- **Indexing**: To build an index we use a pre-trained BART model to get training utterance embeddings. More specifically, we get sentence embedding for all the training utterances. These sentence embeddings are obtained as average of token embeddings from last model layers of the BART models.
- **Retrieval**: Next, given a new input (training or test row), we obtain embeddings by running it through same pre-trained BART, and then query the index with it to retrieve nearest neighbors text and their ground truth semantic parses based on cosine similarity. For training data, we exclude an example itself from the retrieved list. For example, for input utterance “please add 20 minutes on the lasagna timer”, we retrieve the nearest neighbour “add ten minutes to the oven timer” along with the semantic parse as “[in: add_time _ timer add] [sl: date _ time ten minutes] to the [sl: timer _ name oven] [sl: method _ timer _ timer]”.
- **Augmentation**: Once we got a list of nearest neighbors, we can append either utterance text or semantic parse to the input, following the left to right order. The closest neighbor appears to the immediate left of the input example utterance. One can also directly append the nearest neighbor utterance rather than the semparse, refer as utterance-nn.

Overall, we followed GPT-3 and other generation model, where task examples are pre-pended to the input. Hence, utterance is always nearest to the decoder followed by the first nearest neighbour in order.

---


3 We followed GPT-3 and other generation model, where task examples are pre-pended to the input.
Figure 2: High level flowchart for retrieval augmented semantic parsing (RETRONLU) approach.

Representation would be “[in: add_ time_ timer add [sl: date_ time ten minutes ] to the [sl: method_ timer _ oven] [sl: method_ timer _ lasagna timer ]] | please add 20 minutes on the lasagna timer” for semparse-nn, and “add ten minutes to the oven timer | please add 20 minutes on the lasagna timer” for utterance-nn. Here, the token ‘|’ act as a separator between the input utterance and the neighbour's.

Semantic Parsing: The final step is to train a sequence-to-sequence model such as LSTM or Transformer. We fine-tune a BART model with copy mechanism (Aghajanyan et al., 2020), which incorporates benefits of pre-trained language model (BART) and sequence copy mechanism (copy-ptr), and most importantly obtain state-of-the-art results on the TOPv2 (Chen et al., 2020), a challenging task oriented semantic parsing dataset with hierarchical compositional instances. The retrieval augmented example is an input to the encoder and the corresponding ground-truth semantic parse as the labeled decoded sequence. At test time, we simply pass the augmented input to the trained RETRONLU model, and take it’s output as the predicted semantic parse for the input utterance.

3 Experiment and Analysis

Our experiments examines how our knowledge retrieval-based augmentation technique impacts model performance indicators such as accuracy and data efficiency. We study the following questions:

RQ1. Can today’s pre-trained models leverage non-parametric information in manner as described in §2 to enhance task-oriented semantic parsing?

RQ2. If only part of the dataset has semantic parses, i.e. limited supervision setting, can augmentation with unannotated instances (utterance_nn) enhance semantic parsing accuracy?

RQ3. How efficient is a retrieval-augmented model in terms of data? Is it more accurate even with less training data than the baseline seq2seq model?

RQ4. Does non-parametric memory benefit instances equally, e.g., do we notice greater benefits for (a) more complex (i.e. compositional) or (b) less frequent semantic frames (i.e. tail over head)?

RQ5. (a.) Does augmentation more nearest neighbors benefits? (b.) How sensitive is the model to retrieval noise? Can the model predict right intent/slots for low-quality retrieve instances?

Our experiments are designed to demonstrate how non-parametric external information can be beneficial to a parametric model and to undertake an in-depth assessment of the impact.

3.1 Experimental setup

In this section, we discuss the datasets, pre-processing, and the model used in the experiments.

Datasets. For our experiments, we used the multi-domain complex compositional queries based popular TOPv2 (Chen et al., 2020) dataset for task-oriented semantic parsing. We concentrated our efforts on task-oriented parsing because of the commercial importance of data efficiency requirements in conversational AI assistants dialogues. The TOPv2 dataset contains utterances and their semantic representations for 8 domains: source domains - ‘alarm’, ‘messaging’, ‘music’, ‘navigation’, ‘timer’, ...

4We didn’t seek to modify the architecture which ensure the augmentation methodology is flexible.

5Regardless of augmented neighbors structure the approach remain consistent.
The BART based Seq2Seq-CopyPtr models. For task-oriented semantic parsing, \( \text{R} \) models are used, which enables it to directly copy tokens from the input utterance (or from example semantic parses from nearest neighbors). We use the state-of-the-art BART based models (utterance-nn and semparse-nn).

3.2 Results and Analysis

This section summarizes our findings in relation to the aforementioned research questions.

Full Training Setting. To answer RQ1, we compare performance of original baseline (without-nn) with our retrieval augmented models, i.e. augmenting first neighbour utterance (utterance-nn) and augmenting first neighbour semantic parse (semparse-nn). Table 1 compares the frame accuracy of retrieval augmented (a) top nearest neighbour utterance (utterance-nn), (b) top nearest neighbour ground-truth semantic parse (semparse-nn) with original baseline (without-nn) with model train on complete training data.

Analysis: We observe performance improvements with retrieval-augmented models for most domains compared to the original baseline in both cases. The increase in performance (micro-avg) is more substantial 1.4% with semparse-nn compared to 0.85% with utterance-nn. The improvement in utterance-nn augmentation performance is likely due to memorization-based generalization, as explained earlier by (Khandelwal et al., 2019).

The results shows the retrieval augmented semantic parsing is overall effective. Furthermore, the performance enhancement can be obtained also with unstructured utterance (utterance-nn) as nearest neighbour. The utterance-nn based augmentation is particularly beneficial in semi-supervised scenarios, where we have a large unlabelled dataset.

Limited Training Setting. To answer RQ2, we compare model performance which are trained with limited training data. Figure 3 shows frame accuracy(micro-avg) when we use only 10% to 50% of the training data. The training datasets are created in an incremental setting so that next set include examples from the former set.

---

and ‘event’, and target domains: ‘reminder’ and ‘weather’, designed especially to test the zero-shot setting. For our experiments we chose source domains, which has a good mixture of simple (flat) and complex (compositional) semantic frames. For dataset statistics refer Table 1 in Chen et al. (2020).

Data Processing. To build a retrieval index we used the training split of the dataset. Each utterance was represented by its BART-based embedding and indexed using FAISS library (Johnson et al., 2019).

With FAISS computation cost of updating indexing was kept to bare minimum. The only additional cost will be increase in inference time due to augmented neighbor. To produce augmented examples, we retrieved nearest neighbors for each training and test examples, excluding exact matches. In the augmented examples, we use the special token ‘|’ to separate the nearest neighbors, as well as utterance with the first neighbor.\(^6\) We used only one neighbor for most experiments except when we analyse multiple neighbors effects on performance.

In nearest neighbor augmented input, we followed right to left order, where the actual model input comes last, and its highest ranked neighbor is appended to the left of the utterance, followed by other neighbors in the left based on their ranking.\(^8\) For input data pre-processing, we follow (Chen et al., 2020) procedure, we obtain BPE tokens of all tokens, except ontology tokens (intents and slot labels), which are treated as atomic tokens and appended to the BPE vocabulary. Furthermore, we use the decoupled canonical form of semparse for all our experiments. For decoupling, phrases irrelevant to slot values are removed from semparse, and for canonicality, slots are arranged in alphabetic order (Aghajanyan et al., 2020).

Models. For fair comparison with the earlier baseline, we use the state-of-the-art BART based Seq2Seq-CopyPtr model for task-oriented semantic parsing.\(^5\) The BART based Seq2Seq-CopyPtr model initialize both the encoder and decoder with pre-trained BART (Lewis et al., 2020b) model. We choose the BART based Seq2Seq-CopyPtr model for the task because it’s a strong baseline, the performance of the other language model such as RoBERTa without augmentation was inferior (Chen et al., 2020; Aghajanyan et al., 2020).\(^10\) The model is using the copy mechanism (See et al., 2017), which enables it to directly copy tokens from the input utterance (or from example semantic parses from nearest neighbors). We use the same hyper-parameters for model training (c.f. Appendix §D), i.e. baseline (without-nn) and RETRONLU models (utterance-nn and semparse-nn).

---

\(^6\)We use L2 over unit norm BART embedding for indexing.

\(^7\)Using different separator tokens for neighbor-neighbor pair and utterance-neighbor pair didn’t improve performance.

\(^8\)Similar performance is obtained by ordering utterances left to right, followed by their neighbors in index order.

\(^9\)We prefer transformer-based language model over non-transformer models, such as LSTM, because the later does not capture extended context as well as the former.

\(^10\)Our findings, however, we believe, are universal and can be applied to different models, including RoBERTa.

\(^11\)The scores are averaged over three runs with std. of 0.3%
Table 1: Performance of RETRONLU w.r.t original baseline (without-nn) with full training.

<table>
<thead>
<tr>
<th>Domains</th>
<th>without-nn</th>
<th>utterance-nn</th>
<th>semparse-nn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alarm</td>
<td>86.67</td>
<td>87.17</td>
<td>88.57</td>
</tr>
<tr>
<td>Event</td>
<td>83.83</td>
<td>85.03</td>
<td>84.77</td>
</tr>
<tr>
<td>Music</td>
<td>79.80</td>
<td>80.73</td>
<td>80.71</td>
</tr>
<tr>
<td>Timer</td>
<td>81.21</td>
<td>81.75</td>
<td>81.01</td>
</tr>
<tr>
<td>Messaging</td>
<td>93.50</td>
<td>94.52</td>
<td>94.65</td>
</tr>
<tr>
<td>Navigation</td>
<td>82.96</td>
<td>84.16</td>
<td>85.20</td>
</tr>
<tr>
<td>macro-avg</td>
<td>84.43</td>
<td>85.28</td>
<td>85.74</td>
</tr>
<tr>
<td>macro-avg</td>
<td>84.66</td>
<td>85.56</td>
<td>85.82</td>
</tr>
</tbody>
</table>

Analysis: As expected, the performance of all models increases with training set size. Both retrieval augmented models i.e. utterance-nn and semparse-nn outperform the without-nn baseline for all the training sizes. The improvement via augmentation is more substantial with less training data, i.e. 4.24% at 10% data vs 1.30% at 100% data. Furthermore, the semparse-nn augmented model outperforms the original-full train without-nn model with only 40% of the data (i.e RQ3). The results show that the retrieval augmented semantic parsing is more data efficient, i.e. when there is (a) limited labelled training dataset with more unlabelled data for indexing (utterance-nn), and (b) sufficient training data but limited training time (semparse-nn).

Figure 3: Performance of RETRONLU w.r.t original baseline with limited supervised training.

The first case is useful when the ground truth label is missing for utterances due to lack of annotation resources. In such a scenario, one can build the index using large amount of unlabeled utterances and use the index for augmentation. The second case helps us train the model faster, while maintaining all annotated examples in the index. In such a case, one can update the retrieval index only, without re-training the model again and again. This is useful when training on full data is not possible due to limited access to model (black-box), a cap on the computation resources, or for saving training time i.e. industries fast deployment need.

Effect of Utterance Complexity. To answer RQ4(a), we analyse the retrieval augmented model performance improvements (with full training) on simple utterance with only one level in semantic representation (depth-1) vs complex utterance with hierarchical semantic frames (compositional depth-2 and above). Figure 4 shows frame accuracy of without-nn, utterance-nn and semparse-nn model with utterance complexity.

Analysis: As expected, all models perform relatively poorly on complex utterances (79.5%) in comparison to simple utterances (85.5%). Interestingly, both augmentation models equally improve performance on simple queries. And with semantic-frame based augmentation we observe a substantial performance improvement on complex challenging utterances, of 2%, with respect to the original baseline (without-nn). This suggests, that by retrieving nearest neighbors and providing a model with examples of complex parses, the model learns to apply it to a new request. Figure 5 shows precision and recall for intents and slots in retrieved semantic parses. The recall for intent and slot retrieval is 15% lower for complex utterances. Thus, highlighting one reason for a performance gap between simple and complex frames.

Figure 4: Performance comparison (micro-avg) of RETRONLU w.r.t original baseline (without-nn) with utterance complexity, i.e. simple and complex.

Effect of Frame Rareness. To answer RQ4(b), we analyze the retrieval augmented model performance improvement (with full training data) with frame rareness, as shown in Figure 6. Rare or uncommon frames are those example utterances whose ground truth semantic parses without slot value tokens appear infrequently in the training set. To analyze this, we divided the test set into five equal sizes i.e., Very Low, Low, Medium, High, and Very High sets, based on the frequency of semantic frame structure. The experiment checks if performance improvement is mainly attributed to frequently repeating frames (frequent frames) or for rarely occurring frames (uncommon frames).

12The precision gap was small 1% (intents) and 4% (slots).
**Analysis:** Figure 6 shows that all models perform worse on rare frames. This is expected as the parametric model gets less data for training on these frames. Furthermore, many of the low-frequency frames are also complex utterances with more than one intent and have more slots too. Moreover, the nearest neighbour will be noisier for less frequent frames. This is evident from the lower values of precision (20% gap) and recall (25% gap) on the intent and slots for nearest neighbor in Figure 7.

However, compared to original baseline (without-nn) the relative performance improvement on rare frames with retrieval augmented model is more substantial, as shown in Figure 8. For example, the relative improvement for **Very Low** frequency frames is 2.37% (utterance-nn) and 4.11% compared to just 1.01% (utterance-nn) and 1.11% for the **Very High** Frequency frames. We suspect this is because of the model’s ability to copy the required intent and slots from nearest neighbors if the parametric knowledge fails to generate it. This shows the retrieval augmented model is even more beneficial for the rare frames. As earlier, semparse-nn outperform utterance-nn.

**Effect of the number of neighbors.** To answer RQ5(a), we compare k = 1, 2, and 3 nearest neighbors for both utterance-nn and semparse-nn setups. The results are reported in Table 2.

![Figure 5: Precision and Recall of intents and slots for semparse-nn nearest neighbour w.r.t to gold semparse.](image)

![Figure 6: Performance of RETRO NLU w.r.t original baseline (without-nn) with varying frame frequency.](image)

![Figure 7: Precision and Recall of intents and slots w.r.t frame frequency for semparse-nn of the RETRO NLU.](image)

![Figure 8: Relative performance improvement of RETRO NLU w.r.t original baseline (without-nn) with varying frame frequency.](image)

**Effect of Retrieval Quality.** To check if our RETRO NLU model is robust to the noise in the retrieved examples (i.e. RQ5(b)), we analyse the effect of quality of retrieval by comparing semantic parsing accuracy of top neighbor augmented models on the test data with (a) the top neighbour with random neighbor from domain other than the
result shows the performance of RETRO-NLU with varying nearest neighbor quality. In addition, we also conducted an experiment in which we added the best possible neighbor based on the gold parse frame structure. The trained model, though this approach was not robust and relies too heavily on coping frames from neighbors, resulting in poor generalization. Our technique, on the other hand, with embedding-based retrieval, is good at generalization and has enhanced the underlying parametric model. Overall, we can conclude that the semparse-nn and utterance-nn model are both quite robust to nearest neighbors quality. We can also conclude that the semparse-nn model was able to capture richer information through additional similar inputs than without-nn. However, to obtain the best performance good quality neighbour is an essential.

Table 3: Intent-slots precision/recall for RETRO-NLU semparse-nn with closest/farthest neighbors.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Average Precision</th>
<th>Average Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Train</td>
<td>81.39</td>
<td>84.84</td>
</tr>
<tr>
<td>Valid</td>
<td>80.46</td>
<td>87.59</td>
</tr>
<tr>
<td>Test</td>
<td>79.09</td>
<td>86.23</td>
</tr>
<tr>
<td>Slot</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Train</td>
<td>75.02</td>
<td>80.05</td>
</tr>
<tr>
<td>Valid</td>
<td>73.40</td>
<td>82.38</td>
</tr>
<tr>
<td>Test</td>
<td>74.59</td>
<td>83.21</td>
</tr>
</tbody>
</table>

Figure 9: Performance of RETRO-NLU with varying nearest neighbor quality on test data.

4 Comparison with Related Work

Task-oriented Semantic Parsing. Sequence-to-sequence (seq2seq) models have recently achieved state of the art results in semantic parsing (Rong-gali et al., 2020; Gupta et al., 2018), and they also provide a flexible framework for incorporating session-based, complex hierarchical semantic parsing (Sun et al., 2019; Aghajanyan et al., 2020; Cheng et al., 2020; Mehri et al., 2020) and multilingual semantic parsing (Li et al., 2020; Louvan and Magnini, 2020). Architectures, such as T5 and BART (Raffel et al., 2020; Lewis et al., 2020b), with large pre-trained language models pushed the performance even further. Such models are quite capable of storing a lot of knowledge in their parameters (Roberts et al., 2020), and in this work we explore the benefits of additional non-parametric knowledge in a form of nearest neighbor retrieval for the task of semantic parsing. To improve low resource seq2seq parsers Chen et al. (2020) have proposed looking at meta learning methods such as reptile, and Ghoshal et al. (2021) have introduced new fine-tuning methods. Our approach is focused on non-architecture changes to augment...
Incorporating External Knowledge. An idea to help a model by providing an additional information, relevant to the task at hand is not new. This includes both implicit memory tables (Weston et al., 2014; Sukhbaatar et al., 2015), as well as incorporating this knowledge explicitly as an augmentation to the input. Explicit knowledge are incorporated in one of the following two ways (1) suitable model architecture change to incorporate dedicated extended memory space internally i.e. memory network (Bapna and Firat, 2019; Guu et al., 2020; Lewis et al., 2020a; Tran et al., 2020), and (2) appending example specific extra knowledge externally with the input example directly without modifying model architecture (Papernot and McDaniel, 2018; Weston et al., 2018; Lewis et al., 2020c; Tran et al., 2020; Khandelwal et al., 2020a; Fan et al., 2021; Chen et al., 2018; Wang et al., 2019; Neeraja et al., 2021). Retrieval-augmented approaches have been improving language model pre-training as well (Guu et al., 2020; Lewis et al., 2020a; Tran et al., 2020). The idea here is to decouple memorizing factual knowledge and actual language modeling tasks, which can help mitigate hallucinations, and other common problems.

Multiple works like DkNN (Papernot and McDaniel, 2018), RAG (Lewis et al., 2020c), kNN-LM (Tran et al., 2020), kNN-MT (Khandelwal et al., 2020a), and KIF-Transformer (Fan et al., 2021) show that external knowledge is useful for large pre-trained language models, and can help fine-tuning. DkNN shows that nearest neighbour augmented transformer-based neutral network is more robust and interpretable. RAG shows that one can append external knowledge to improve open-domain, cloze-style question answering, and even fact verification task such as FEVER. kNN-LM shows that for cloze task for fact completion, one can combine nearest neighbour predictions with original prediction using appropriate weighting to improve model performance. However, these works mostly study knowledge dependent question answering task, while we are exploring a complex task of structural prediction of semantic frame structures for task-oriented dialog.

Very recently, Pasupat et al. (2021) share similar finding of exemplar augmentation and propose ControllAble Semantic Parser via Exemplar Retrieval (CASPER). In their work, the semantic parser gets relevant exemplars from a retrieval index, augments them with the query, and then generates an output parse using a generative seq2seq model. The exemplars serve as a control mechanism for the generative generative model: by modifying the retrieval index or the construction of the augmented query, one may alter the parser’s behavior. Compare to them, our study focuses more on the influence of augmentation on the performance of the state-of-the-art Copy Transformer BART model for task-oriented semantic parsing. By design, the copy transformer effectively utilizes it’s copy mechanism to get non-parametric information from augmented nearest neighbor semparse/utterances. Additionally, we conduct a detailed investigation of the influence of retrieval quality, utterance and semantic complexity, and the rarity of semantic frames. We anticipate that our findings will shed light on the potential advantages of retrieval enhancing parametric neural networks for the complex structural task of task-oriented semantic parsing.

5 Conclusion and Future Work

We show that task-oriented semantic parsing performance can be enhanced by augmenting neural model-stored parametric information with non-parametric external memory. On the TOPv2 dataset, we demonstrated that adding instances derived from a nearest neighbor index greatly improves the semantic parsing performance of a BART model with copy mechanism. Our RETRONLU model is able to achieve higher accuracy earlier with less training data (limited supervision setting), which allows maintaining a large index with annotated data, while using only a subset to train a model more efficiently. Lastly, we performed an analysis of performance improvements on different slices, and found RETRONLU to be more effective on rarer complex frames, compared to a traditional seq2seq model.

RETrONLU extensions, we focus on joint training of retrieval and parsing components. Having task specific utterances representation can benefit i.e. finding utterances with similar semantic parse. Exploring few/zero-shot performance could be interesting direction. Having an easily-updateable index enables you to amend annotations, add new ones, or remove existing ones, without affecting the model. Finally, using cross-lingual representations such as mBART (Liu et al., 2020), could help multilingual semantic parsing.
References


Below, we provide some qualitative example from the TOPv2 datasets, in the first box, we shows the input utterance, nearest neighbour utterance and it corresponding semantic parse i.e. semparse-nn (from the training set). In the second box we showed expected gold semantic parse (expected-sparse) and the semantic parse prediction with baseline i.e. without nearest neighbor (without-nn), and our models utterance only nearest neighbor, where the input is appending with nearest-nn utterance (utterance-nn) and semantic augmented model i.e. (semparse-nn), where the input is augmented with semantic parse of first nearest neighbor. We can clearly see from the examples that augmented nearest neighbor enhance models in identifying correct intent and slots.

A Qualitative Examples

| example 1 input | nearest-nn utterance : message kira and lena saying want to get drinks this week? | nearest-nn semparse : [sl:recipient kira ] and [sl:recipient lena ] saying [sl:content_exact want to get drinks this week]?
---|---|---
| nearest-nn utterance : message jizzie and trent from my group if they have any updates yet? | nearest-nn semparse :
example 1 output

expected-parse: [in:send_message [sl:recipient lizzie] [sl:recipient trent] [sl:content_exact they have any updates yet]]

without-nn: [in:get_message [sl:content_exact they have any updates yet] [sl:group lizzie] [sl:group trent]]

semparse-nn: [in:send_message [sl:recipient lizzie] [sl:recipient trent] [sl:content_exact they have any updates yet]]

utterance-nn: [in:send_message [sl:recipient lizzie] [sl:recipient trent] [sl:content_exact they have any updates yet]]

example 2 input

utterance: no more country

nearest-nn utterance: no more music

nearest-nn semparse: [in:stop_music [sl:music_type music]]

example 2 output

expected-parse: [in:remove_from_playlist_music [sl:music_artist_name mariah carey] [sl:music_genre country]]

without-nn: [in:unsupported_music [sl:music_type songs]]

semparse-nn: [in:remove_from_playlist_music [sl:music_type songs] [sl:music_artist_name mariah carey]]

utterance-nn: [in:remove_from_playlist_music [sl:music_type songs] [sl:music_artist_name mariah carey]]

example 3 input

utterance: block all songs of mariah carey

nearest-nn utterance: delete mariah carey songs

nearest-nn semparse: [in:remove_from_playlist_music [sl:music_artist_name mariah carey] [sl:music_type songs]]

example 3 output

expected-parse: [in:remove_from_playlist_music [sl:music_artist_name mariah carey]]

without-nn: [in:unsupported_music [sl:music_type songs]]

semparse-nn: [in:remove_from_playlist_music [sl:music_type songs] [sl:music_artist_name mariah carey]]

utterance-nn: [in:remove_from_playlist_music [sl:music_type songs] [sl:music_artist_name mariah carey]]

In example 1, the model misses the correct intent and corresponding slots completely, the correct intent is sending a message rather than receiving a message is correctly identified by both semparse-nn and utterance-nn.

In example 2, the baseline model without nearest neighbour augmentation struggled to identify the intent from utterance text token “block” therefore prediction unsupported music as the intent and the music type as songs, however the model with augmented nearest neighbour example with “delete” intended slot correctly identified both the intent and slots. Furthermore, the model also resolve the active passive voice confusion with nearest neighbor augmentation.

B Domain based Limited Training Setting

In Table 4 shows the performance of model for each domain on original baseline (without-nn), and RetroNLU model utterance-nn and semparse-nn with varying amount of supervised training data. Overall, semparse-nn outperform utterance-nn over most of the domains. Surprisingly, we also found that for few domain (with large number of samples) utterance-nn perform marginally better than semparse-nn, need to investigate exact reason for that. As expected both model utterance-nn and semparse-nn perform much better than original baseline which is without any nearest neighbour augmentation.
<table>
<thead>
<tr>
<th>Percentage</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain w/o nn</td>
<td>uttr-nn</td>
<td>sem-nn</td>
<td>w/o nn</td>
</tr>
<tr>
<td>Alarm</td>
<td>80.50</td>
<td>84.05</td>
<td>83.60</td>
</tr>
<tr>
<td>Event</td>
<td>68.56</td>
<td>78.33</td>
<td>79.38</td>
</tr>
<tr>
<td>Music</td>
<td>69.12</td>
<td>75.74</td>
<td>73.23</td>
</tr>
<tr>
<td>Timer</td>
<td>71.63</td>
<td>76.76</td>
<td>76.27</td>
</tr>
<tr>
<td>Navigation</td>
<td>74.30</td>
<td>73.86</td>
<td>76.44</td>
</tr>
<tr>
<td>Messaging</td>
<td>84.38</td>
<td>87.30</td>
<td>89.44</td>
</tr>
</tbody>
</table>

Table 4: Limited training setting results on various domain with baseline (without-nn), RETRONLU model utterance-nn and semparse-nn, shown here as w/o nn, utter-nn and sem-nn respectively.

<table>
<thead>
<tr>
<th>#neighbour's one</th>
<th>two</th>
<th>three</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain w/o nn</td>
<td>uttr-nn</td>
<td>sem-nn</td>
</tr>
<tr>
<td>Alarm</td>
<td>86.67</td>
<td>87.17</td>
</tr>
<tr>
<td>Event</td>
<td>83.83</td>
<td>85.03</td>
</tr>
<tr>
<td>Music</td>
<td>79.80</td>
<td>80.73</td>
</tr>
<tr>
<td>Timer</td>
<td>81.21</td>
<td>81.75</td>
</tr>
<tr>
<td>Messaging</td>
<td>93.50</td>
<td>94.52</td>
</tr>
<tr>
<td>Navigation</td>
<td>82.96</td>
<td>84.16</td>
</tr>
</tbody>
</table>

Table 5: Effect of number of nearest neighbours of RETRONLU performance with varying domains

C Domain Specific Effect of Nearest Neighbours

In Table 5 we shows the performance of model for each domain on original baseline (without-nn), and RetroNLU model utterance-nn and semparse-nn with varying number of nearest neighbour augmented. We found the utterance-nn performance increases with increasing number of neighbours where semparse performance remain mostly constant after the first neighbour augmentation for many domains. We suspect this is due to the fact that the data contains a large number of utterances with identical semparse output. There is also frame redundancy, since many unique utterance inquiries have comparable semantic parse frames structure with differences only on slot values.

D Hyperparameters Details

For training we use 100 epochs, Adam optimizer (Kingma and Ba, 2014) with learning rate $\alpha$ of $1e^{-4}$ and decay rate $\beta_1$ and $\beta_2$ of 0.9 and 0.98 respectively in all our experiments. Also, we didn’t added any left or right padding and rely on variable length encoding in our experiments. We use warm-up steps of 4000, dropout ratio of 0.4, and weight decay 0.0001, but no clip normalization as regularization during the training. We use batch size of 128 and maximum token size of 2048. Furthermore, to ensure both encoder and decoder BART, can utilise the extra nearest neighbour information, we increase the embedding dimension to 1024.