Towards Zero and Few-shot Knowledge-seeking Turn Detection in Task-orientated Dialogue Systems

Di Jin, Shuyang Gao, Seokhwan Kim, Yang Liu, Dilek Hakkani-Tur
Amazon Alexa AI, Sunnyvale, CA, USA
{djinamzn,shuyag,seokhwk,yangliud,hakkanit}@amazon.com

Abstract

Most prior work on task-oriented dialogue systems is restricted to supporting domain APIs. However, users may have requests that are out of the scope of these APIs. This work focuses on identifying such user requests. Existing methods for this task mainly rely on fine-tuning pre-trained models on large annotated data. We propose a novel method, REDE, based on adaptive representation learning and density estimation. REDE can be applied to zero-shot cases, and quickly learns a high-performing detector with only a few shots by updating less than 3K parameters. We demonstrate REDE’s competitive performance on DSTC9 data and our newly collected test set.

1 Introduction

Current task-oriented dialog systems often rely on pre-defined APIs to complete target tasks (Williams et al., 2017; Eric et al., 2017) and filter out any other requests beyond the APIs as out-of-domain cases. However, some in-domain user requests can be addressed by incorporating external domain knowledge from the web or any other sources (Kim et al., 2020). To address this problem, Kim et al. (2021) recently organized a benchmark challenge on task-oriented conversational modeling with unstructured knowledge access in DSTC9 (Gunasekara et al., 2020). This challenge includes the knowledge-seeking turn detection task to determine whether to invoke a knowledge-driven responder or just rely on available API functions. One data sample is provided in Table 1. The state-of-the-art systems (He et al., 2021; Tang et al., 2021; Mi et al., 2021; Jin et al., 2021) implemented this detector by fine-tuning a large pre-trained model on the training dataset (about 72K samples) as a binary classifier, and achieved an F1 score of over 95% on the benchmark test set. However, after close investigation, we find those user queries in the test set are very limited in topic coverage and language variation.

To evaluate the detector performance on real-world user queries, we specially curate a new contrast set following Gardner et al. (2020) by manually collecting questions posted by real users on Tripadvisor forums. We found that the detector trained on DSTC9 Track 1 training samples had a large performance degradation on this contrast set (F1 score dropped by over 15%), suggesting the need for methods with better generalization.

In this work, we propose a method that can quickly learn a knowledge-seeking turn detector with much fewer out-of-domain samples, such as only a few shots or even zero shot. Our method is composed of two stages: REpresentation learning and DEnsity estimation (REDE). First, we learn a representation model via fine-tuning a pre-trained sentence encoder on all non-knowledge-seeking turns (utterances that can be supported by APIs) via masked language modeling (MLM). Then we learn a density estimator using these representation vectors. During inference, the density estimator produces a density score for a given user utterance. If it is above a threshold, this utterance is counted as an in-domain API turn, otherwise as a knowledge-seeking turn.

To incorporate out-of-domain examples, we propose to use principle component analysis to quickly learn a projection matrix with few knowledge-seeking turn samples and then use this matrix to linearly transform the representation vectors. We conduct experiments on the DSTC9 Track 1 data as well as our new contrast test set. We demonstrate that REDE can achieve competitive performance as other supervised methods in the full-shot setting and outperform them by a large margin in the low-resource setting. More importantly, our approach generalizes much better in the new contrast test set that we created.

Overall, our contributions are summarized as follows:

- We propose a new approach, REDE, for
1 User I’m looking for a place to stay in the south of town. It doesn’t need to have free parking.
2 Agent There are 4 hotels that are in the area you are looking for. Would you prefer a 3 or 4 star rated hotel?
3 User I don’t care about the star rating as long as it’s expensive.
4 Agent The **Lensfield Hotel** is the only expensive hotel in the south area. Would you like any more information on this location?
5 User I’m interested in knowing, do they have a workout facility on the premises?
6 Agent There are both a fitness center and gym available on the premises. Does this sound ok?
7 User That is perfect can you book that for me please.
8 Agent The **Lensfield Hotel** is located in the South. It has a 3 star rating and is expensive. There is free parking and internet. I have booked it for you.
9 User Great, thank you!

Table 1: One example of task-oriented conversations with unstructured knowledge access. The user utterance of turn 5 is beyond the coverage of API and needs external knowledge support, therefore it is a knowledge-seeking turn while other user turns are non-knowledge-seeking turns. To address turn 5, three sampled FAQ pairs for the entity “Lensfield Hotel” extracted from the external knowledge are listed in the rightmost column. The most appropriate FAQ pair is highlighted in bold font.

knowledge-seeking turn detection that be applied to zero or few shot cases. It can be quickly adapted to new knowledge-seeking turns data with much less training samples, which can achieve over 90% F1 score with only five shots;

- Once the continuous pretraining stage on non-knowledge-seeking turns data is finished, our model can be quickly adapted to any kinds of knowledge-seeking turns data within seconds with only a few parameters to be learned.

- We curate and release a contrast set to examine the generalization capability of the knowledge seeking-turn detectors.\(^1\) We demonstrate that our model is better at generalizing to this contrast set than the previous best models.

2 Related Work

Our work is closely related to those participating systems in DSTC9 Track 1 (Kim et al., 2020, 2021). All the systems proposed to treat the problem of knowledge-seeking turn detection as a binary classification task and fine-tuned pre-trained models such as RoBERTa, UniLM, PLATO, GPT2, on the whole training set (He et al., 2021; Tang et al., 2021; Mi et al., 2021), which yielded around 99% and 96% F1 scores on the development and test sets, respectively. Our method differs in two aspects: 1) We do not need to fine-tune the pre-trained model on the training set; 2) Our model is at least 5 times smaller and we need less than 5% of training data to achieve similar performance.

Our method is inspired by previous work for out-of-domain (OOD) detection (Ren et al., 2019; Gangal et al., 2020; Hendrycks et al., 2019) and one-class classification (Sohn et al., 2021). Kim et al. (2020) also tried tackling this problem by applying an unsupervised anomaly detection algorithm, Local Outlier Factor (LOF) (Breunig et al., 2000), which compares the local densities between a given input instance and its nearest neighbors, but did not obtain good results (F1 score is less than 50%). Sohn et al. (2021) proposed to first learn a representation model via contrastive learning, then learn a density estimator on the obtained representations. They showed decent performance for one-class classification. All these previous work assumed no access to OOD samples, however, we would like to make use of those OOD samples efficiently when they are available. Therefore we extend the general representation learning framework by proposing a novel representation transformation method to learn OOD samples, which leads to significantly boosted detection performance.

Our work is also related to few/zero-shot learn-
ing, which has been widely studied previously (Gao et al., 2020; Jin et al., 2020c,b). Transfer learning (Zhou et al., 2019c; Jin et al., 2020a; Zhou et al., 2019a,b; Yan et al., 2020) and data augmentation (Jindal et al., 2020b,a) have been two major methods for this direction, while our work focuses on manipulation of learned representations, which provides a new perspective.

3 Methods

Our method includes three steps: encoder adaptation, representation transformation, and density estimation. The representation transformation step is only applicable when there are OOD examples (i.e., knowledge-seeking turns).

3.1 Encoder Adaptation

In this step, we adapt a pre-trained sentence encoder $E$ to the in-domain data, i.e., non-knowledge-seeking turns, $X^{NK} = \{x^K_1, \ldots, x^K_N\}$, via a masked language modeling (Devlin et al., 2019). Specifically, 15% of tokens of $x^K_i$ are masked and $E$ is trained to predict these masked tokens.

3.2 Representation Transformation

To incorporate the knowledge-seeking turns $X^K = \{x^K_1, \ldots, x^K_M\}$, a standard solution is to fine-tune $E$ on the combined data of knowledge-seeking and non-knowledge-seeking turns, $X = X^K \cup X^{NK}$, as a supervised binary classifier. However, in few-shot settings where $M << N$, there is an extreme class imbalance problem. In addition, fine-tuning large models may take a long time and much computation power with large data size. Instead, we propose a simple linear transformation to the sentence representation $e = E(x)$ without updating the model parameters, following (Su et al., 2021):

$$\tilde{e} = T(e) = (e - \mu)W \quad (1)$$

where $\mu = \frac{1}{M} \sum_{i=1}^{M} E(x^K_i)$. To calculate $W$, we first calculate the covariance matrix, $\Sigma = \frac{1}{M} \sum_{i=1}^{M} (E(x^K_i) - \mu)(E(x^K_i) - \mu)^T$, then perform Singular Value Decomposition (SVD) over $\Sigma$ such that: $\Sigma = U\Lambda U^T$, and finally we obtain $W = U\sqrt{\Lambda^{-1}}$.

The elements in diagonal matrix $\Lambda$ derived from SVD are sorted in descending order. Therefore, we can retain the first $L$ columns of $W$ to reduce the dimension of transformed vectors $\tilde{e}$, which is theoretically equivalent to Principal Component Analysis (PCA). However, to be noted, both $\mu$ and $W$ parameters are obtained using those knowledge-seeking turns instead of non-knowledge-seeking turns and the number of knowledge-seeking turns is much smaller, which can be as small as just a few shots. In another word, we only need a very small size of out-of-domain samples to learn the parameters needed for our representation transformation as defined in Eq. 1 to transform the representations of in-domain data. This is in contrast to the conventional PCA based density estimation method that assumes only having access to in-domain data, i.e. non-knowledge-seeking turns, and needs to learn and perform PCA transformation both on a good amount of those in-domain data.

This step of transformation can be viewed as another round of unsupervised representation learning with knowledge-seeking turns, which helps us obtain a better representation and is extremely critical to our claimed great performance for few-shot learning, as analyzed in Section 5.2.2.

3.3 Density estimation

In this step, we encode all the non-knowledge-seeking turns in the training set and transform them to obtain $\{\tilde{e}_{1}^{NK}, \ldots, \tilde{e}_{N}^{NK}\}$, normalize them into unit vectors, and then learn a shallow density estimator $D$ over them, such as Gaussian Mixture Model (GMM). Note that in the zero-shot setting when no knowledge-seeking turns are available, the representation transformation step (in Section 3.2) is skipped.

During inference, given a test sample $x$, we encode it with the encoder $E$, transform it with $T$ defined in Eq. 1, and then use the learned density estimator $D$ to produce a density score $D(T(E(x)))$. If it is above a pre-set threshold $\eta$, $x$ is considered as a non-knowledge-seeking turn, otherwise as a knowledge-seeking turn. This whole pipeline is motivated by the assumption that the well learned representations of in-domain (non-knowledge-seeking turns) and OOD samples (knowledge-seeking turns) should be distributed separately in the latent space, and thus the estimated density of in-domain data by the density estimator should be higher than that of OOD data.

4 Experiments

4.1 Dataset

We use the DSTC9 Track 1 competition data (Kim et al., 2020, 2021), and focus on the sub-task of binary knowledge-seeking turn detection. The data
statistics are summarized in Table 2. We further curate a new contrast test set by first collecting questions posted by real users in the Tripadvisor forums, then obtain the questions as knowledge-seeking turns that cannot be addressed by MultiWOZ API schema (Eric et al., 2019) (this schema was also used for constructing the DSTC9 Track 1 dataset), and finally manually paraphrasing them to make them more like dialogue utterances. We obtained 617 knowledge-seeking turns and mixed them with those non-knowledge-seeking turns in the original test set to form the contrast set.

Table 3 shows several data samples for the newly curated contrast set. These user queries collected from real users are much more diverse than those in the benchmark test set of DSTC9 Track 1 dataset. Among these examples, the user query of “How much do you charge for parking?” is actually quite challenging for knowledge-seeking turn detection since this query is very close to one of the available API functions that is responsible for checking whether there is free parking. However, in order to answer this query, we still need to invoke the knowledge module to retrieve external unstructured knowledge.

<table>
<thead>
<tr>
<th>Domains</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attraction</td>
<td>Is it necessary to buy tickets in advance?</td>
</tr>
<tr>
<td>Attraction</td>
<td>How long could one take to see it all? 4 hours it would be enough?</td>
</tr>
<tr>
<td>Hotel</td>
<td>Is there a minimum check in age?</td>
</tr>
<tr>
<td>Hotel</td>
<td>How much do you charge for parking?</td>
</tr>
<tr>
<td>Restaurant</td>
<td>Would there be room for a stroller with a sleeping baby during dinner?</td>
</tr>
<tr>
<td>Restaurant</td>
<td>Can I order crab cakes take out for eight servings?</td>
</tr>
</tbody>
</table>

Table 3: Examples of newly collected user questions in the contrast set. These user queries collected from real users are much more diverse than those in the benchmark test set of DSTC9 Track 1 dataset.

The threshold $\eta$ is chosen based on the highest F1 score on the development set. For the density estimator, we have tried OC-SVM, KDE with various kinds of kernels, and GMM, and we find GMM performs the best and its inference time is the lowest. We set the number of components to 1 for GMM. Dimensionality $L$ is set as 650 for PCA transformation by tuning on the development set. Details of comparison and tuning results are in the appendix. For evaluation metrics, we report precision (P), recall (R), and F1 scores.

## 5 Results & Discussion

### 5.1 Main Results

**Full supervised setting** Table 4 summarizes the comparison of our method REDE with baselines where all knowledge-seeking turn samples in the DSTC9 Track 1 training set are used for training. REDE has two advantages: (1) Once the first step of adaptive pre-training on non-knowledge-seeking turns is done, it only needs to update less than 3K parameters of the density estimator for learning the knowledge-seeking turns, but it can still achieve superior performance on the test set; (2) It can be better generalized to the new contrast set that has distribution shift with respect to the training data.

**Low-resource setting** We are more interested in exploring how our method performs under the low-resource setting compared with baselines. Therefore, we sub-sampled different numbers of knowledge-seeking turn samples and kept using all non-knowledge-seeking turn samples. We then trained the model and obtained F1 scores on the test set. We performed five times of random sub-sampling and report the average and standard deviation in Figure 1. Since the error bar of REDE is
5.2 Analysis

5.2.1 Effect of MLM Adaptation

As shown in Table 6, after removing the MLM adaptation step, our method has significant performance degradation, especially for the contrast set, indicating the importance of adapting the general pre-trained model to the target dataset via unsupervised learning. We have also tried adopting contrastive learning for such unsupervised adaptation (i.e., SimCSE), which has shown state-of-the-art performance for unsupervised representation learning (Gao et al., 2021). Results in Table 6 show that it is worse than MLM. The reason could be that contrastive learning used in SimCSE would lead to more uniform and dispersed distribution over the latent space, however, the density estimation based OOD detection favors more dense and collapsed distribution for in-domain data.

Table 5: Averaged F1 score and standard deviation under the low-resource setting by randomly sub-sampling different number of knowledge-seeking turns for five times. DistillBERT is DistilBERT-Base-NLI-STSB while RoBERTa is RoBERTa-Large.

<table>
<thead>
<tr>
<th>Samples</th>
<th>DistillBERT</th>
<th>RoBERTa</th>
<th>REDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5.70 ± 0.09</td>
<td>0.00 ± 0.00</td>
<td>90.56 ± 0.69</td>
</tr>
<tr>
<td>10</td>
<td>13.88 ± 7.37</td>
<td>0.00 ± 0.00</td>
<td>91.34 ± 0.92</td>
</tr>
<tr>
<td>20</td>
<td>32.51 ± 3.84</td>
<td>0.00 ± 0.00</td>
<td>92.85 ± 0.26</td>
</tr>
<tr>
<td>50</td>
<td>53.22 ± 4.70</td>
<td>0.00 ± 0.00</td>
<td>94.29 ± 0.35</td>
</tr>
<tr>
<td>200</td>
<td>78.35 ± 2.57</td>
<td>47.87 ± 39.28</td>
<td>95.05 ± 0.20</td>
</tr>
<tr>
<td>1,000</td>
<td>89.14 ± 0.49</td>
<td>90.76 ± 0.94</td>
<td>95.74 ± 0.13</td>
</tr>
<tr>
<td>2,000</td>
<td>91.49 ± 0.97</td>
<td>91.92 ± 0.79</td>
<td>95.97 ± 0.12</td>
</tr>
<tr>
<td>4,000</td>
<td>93.49 ± 0.53</td>
<td>93.55 ± 0.49</td>
<td>95.95 ± 0.18</td>
</tr>
<tr>
<td>10,000</td>
<td>94.56 ± 0.32</td>
<td>95.06 ± 0.46</td>
<td>96.07 ± 0.17</td>
</tr>
</tbody>
</table>

Table 6: Ablation study for MLM adaptation by removing it or replacing it with SimCSE (a contrastive learning method). All training samples are used here.

<table>
<thead>
<tr>
<th>Settings</th>
<th>Test Set F1 (%)</th>
<th>Contrast Set F1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>REDE</td>
<td>96.18</td>
<td>90.43</td>
</tr>
<tr>
<td>no MLM</td>
<td>93.49</td>
<td>75.65</td>
</tr>
<tr>
<td>MLM → SimCSE</td>
<td>92.00</td>
<td>74.39</td>
</tr>
</tbody>
</table>

Table 7: F1 score on test set for top 5, 50, 500, and all principle components under three different settings: zero-shot (PCA over non-knowledge-seeking turns), ten-shot (PCA over ten knowledge-seeking turns), and full-shot (PCA over all knowledge-seeking turns).
Table 8: Comparison of different density estimators. Inference time is measure on the whole test set using the same machine. We have also tried other kernels for the KDE estimator, such as ‘tophat’, ‘epanechnikov’, ‘linear’, and ‘cosine’, but they all perform poorly.

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Test Set (%)</th>
<th>Contrast Set (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>OC-SVM</td>
<td>92.30</td>
<td>88.34</td>
</tr>
<tr>
<td>KDE-Gaussian</td>
<td>92.81</td>
<td>91.17</td>
</tr>
<tr>
<td>KDE-Exponential</td>
<td>92.29</td>
<td>91.87</td>
</tr>
<tr>
<td>GMM</td>
<td>97.76</td>
<td>94.65</td>
</tr>
</tbody>
</table>

Table 9: Comparison of performance (in percentage) by using different number of components for the GMM estimator.

<table>
<thead>
<tr>
<th>Components #</th>
<th>Dev Set F1</th>
<th>Test Set F1</th>
<th>Contrast Set F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>98.71</td>
<td>96.18</td>
<td>90.43</td>
</tr>
<tr>
<td>2</td>
<td>98.88</td>
<td>95.82</td>
<td>90.61</td>
</tr>
<tr>
<td>3</td>
<td>98.97</td>
<td>96.12</td>
<td>90.35</td>
</tr>
<tr>
<td>4</td>
<td>99.03</td>
<td>96.04</td>
<td>89.72</td>
</tr>
</tbody>
</table>

5.2.2 Understanding PCA Transformation

In Section 3.2, the sentence representation is transformed with PCA learned from knowledge-seeking turns. Table 7 shows the F1 score on the test set using top \( L \) principle components with PCA learned using different data. Overall, we can see that PCA with knowledge-seeking turns achieves better performance, and using more principle components is always beneficial. PCA is well-known to help construct new subspaces by maximizing the global variance. Intuitively, by learning PCA over knowledge-seeking turns, we expect the manifolds on knowledge-seeking turns to spread out and non-knowledge seeking turns condense. Figure 2 shows the scatter plot of the top two principle components of transformed features. In Figure 2a, we learn PCA from non-knowledge-seeking turns, which results in the manifold of knowledge-seeking turns (red dots) to be within that of non-knowledge-seeking turns (blue dots). It hurts the performance since the density estimation is performed over non-knowledge-seeking turns, as confirmed by the zero shot result in Table 7 in comparison to that in Fig 1. In contrast, in Figure 2b, we learn PCA with knowledge-seeking turns, which makes knowledge-seeking turns (red dots) spread out and non-knowledge-seeking turns (blue dots) condense. By estimating the density of this condensed blue area, we obtain higher F1 score because all the red dots falling outside of the region of blue dots will be classified as out-of-distribution correctly.

Figure 3: Development set F1 scores by retraining different values of first \( L \) columns of \( W \). Full dimension is 768.

5.2.3 Comparison of Density Estimators

For the density estimator, we have tried OC-SVM, KDE with various kinds of kernels, and GMM, which are summarized in Table 8. All these estimators are implemented using Scikit-Learn library. From Table 8, we see that GMM performs the best while being the fastest for inference, therefore we chose it as the density estimator in our work.

---

5 https://scikit-learn.org/stable/
Table 9 shows the performance under different number of components for the GMM density estimator. From it, we see that the number of components has minor influence on the performance so we decide to use 1 as the number of components in this work.

5.2.4 Effects of L
We can retrain only the first $L$ columns of $W$ for the PCA transformation, which can help us reduce the dimension of transformed representation vector $\tilde{e}$. Figure 3 shows the development set performance under different values of $L$ when all knowledge-seeking turns are used for training. We see that the first 50 dimensions can achieve over 95% F1 score and 300 dimensions are already enough to realize the peak performance, whereas the full dimension is 768.

6 Conclusion
In this work, we propose a novel method REDE based on domain-adapted representation learning and density estimation for knowledge-seeking turn detection in task-oriented dialogue systems. Compared with previous SOTA models, REDE can achieve comparable performance in the full supervised setting and significantly superior performance for the low-resource setting. Besides, REDE has much better generalization capability onto a new contrast set we curated.

References


