Enhancing Out-Of-Domain Utterance Detection with Data Augmentation Based on Word Embeddings

Yueqi Feng
Twilio AI
San Francisco, CA
rfeng@twilio.com

Jiali Lin
Twilio
San Francisco, CA
jlin@twilio.com

Abstract
For most intelligent assistant systems, it is essential to have a mechanism that detects out-of-domain (OOD) utterances automatically to handle noisy input properly. One typical approach would be introducing a separate class that contains OOD utterance examples combined with in-domain text samples into the classifier. However, since OOD utterances are usually unseen to the training datasets, the detection performance largely depends on the quality of the attached OOD text data with restricted sizes of samples due to computing limits. In this paper, we study how augmented OOD data based on sampling impact OOD utterance detection with small sample size. We hypothesize that OOD utterance samples chosen randomly can increase the coverage of unknown OOD utterance space and enhance detection accuracy if they are more dispersed. Experiments show that given the same dataset with the same OOD sample size, the OOD utterance detection performance improves when OOD samples are more spread-out.

1 Introduction
For most goal-oriented conversational systems, developers have to pre-define a list of intents containing training data that are close to what the end users are expected to say. These pre-defined training samples under different intents often lie within the same domain. For example, for a common “restaurant booking” virtual assistant, developers can define intent classes like “reserve a table” and “cancel reservations”. However, utterances from the end users are usually noisy and may not always fall into those pre-defined categories of intents, which the back end system is not able to handle. For example, if a user gives input as “what’s the population of San Francisco?” to the “restaurant booking” chatbot, the model does not contain related training data to classify it to the correct class, and the dialogue is going to end. These utterances are defined as out-of-domain (OOD) utterances in this work. It is important to recognize these OOD utterances and follow up with the corresponding prompt questions after getting detected by the text classifier.

Detecting OOD utterances for conversational assistant platforms is even more challenging than building chatbots for one single domain. Whereas building a domain-specific chatbot can rely on collecting OOD samples iteratively and improve the performance over time, conversational assistant platforms are unable to take advantage of tailored OOD corpora. This is because those assistants built on top of the platform may come from different domains and have different distributions. Especially when computation resources are limited, custom intent classification models will not be able to take a considerable amount of OOD samples. Thus, it is necessary to down-sample text data from the OOD utterance pool. However, since OOD utterances from production environments are most likely unseen to the models when they are being developed and trained, those classifiers can have difficulty differentiating OOD samples from in-domain (IND) samples and performances may vary significantly for each round of sampling.

We hold an assumption that OOD data samples, which are more spread out across the feature space, have better coverage for OOD data space. Intuitively speaking, when in-domain sample text is mapped to feature representations, such as word embeddings, these representations are distributed as a cluster with relatively small intra-cluster distance. When the classifier sees OOD samples, their representations are often not close to the IND sample cluster. Therefore, with a fixed number of OOD samples, the utterance detection could possibly be enhanced if those OOD samples can cover
more space with their feature representations.

In this paper, we mainly focus on studying how randomly selected OOD samples affect OOD utterance detection from feature space coverage perspective.

2 Related Work

In general, one can improve out-of-domain utterance detection with either data-driven or model-based techniques.

Rather than training the domain classifier and OOD detector separately, Kim and Kim (2018) propose a neural joint learning model that uses dynamic class weighting to optimize a given OOD false acceptance rate (FAR). Oh et al. (2018) use RNN(Recurrent Neural Network) encoders with the attention mechanism to train models where the OOD sentences are detected based on distances. Ryu et al. (2018) build a generative adversarial network (GAN) that is able to create OOD sentences of low scores with pre-trained sentence embeddings.

Random sampling is found as a more reliable approach than providing negative data for answer retrieval task (Saeidi et al., 2017). Chen et al. (2018) improves the negative sampler with an adaptive one that can take advantage of multi-dimensional word information instead of one-dimensional popularity. The sampling efficiency is boosted by dynamically oversampling high score negative words with embedding features.

For the field of conversational understanding systems precisely, using syntactic and semantic parse structure features can result in better performance (Tur et al., 2014). Heck and Hakkani-Tür (2012) propose an unsupervised training approach for spoken language understanding tasks where the structure of semantic knowledge graphs combines web search retrieval and syntax-based dependency parsing. Another approach would be calculating the generality of a set of text based on dispersion, which can be used to determine how much perceptual information should be included in a given model (Kiela et al., 2014). Lee and Shalyminov (2019) augment OOD data with counterfeit OOD samples in the context of a dialog. Lane et al. (2004) leverage classification confidence scores of topics and train a linear classifier based on deleted interpolation of the IND data.

3 Methods

In order to describe the dispersion characteristic of OOD data samples, we define mean pairwise cosine similarity (MPCS) to measure how disperse the word representations are. For any given set of sentences, let \( \{ \vec{w}_i, \vec{w}_j \} \) be a pair of word vectors of two unique words appeared in the OOD training data. MPCS is defined by taking the average of cosine similarities between each pair of words based on word embeddings:

\[
\cos(\vec{w}_i, \vec{w}_j) = \frac{\vec{w}_i \cdot \vec{w}_j}{\|\vec{w}_i\| \|\vec{w}_j\|}
\]

\[
MPCS = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \cos(\vec{w}_i, \vec{w}_j)
\]

where \( n \) is the the size of unique words. High MPCS implies the words are similar to each other so that they are covering less space.

In order to study the relationship between the quality of OOD utterance detection and the dispersion of datasets, we select some datasets as IND samples and augment each dataset with random OOD samples. Then we train models on those augmented datasets and evaluate how the model performs on detecting OOD utterances. Therefore, we are able to analyze whether the dispersion of word representations affects model performances or not.

4 Experiments

We start by collecting multiple IND datasets as training data. Some of the datasets contain one class, while others could have multiple classes. We then sample a specific number of data points from the OOD utterance pool. The sample size will be chosen with respect to the dataset size. After sampling, we combine OOD samples to training datasets, and these OOD samples are treated as a new class called OOD. The output training datasets are stored separately as data snapshots, which are determined by the IND datasets with specific OOD samples.

A text classification model is trained on each snapshot of training data. Note that for the snapshots who share the same in-domain datasets, OOD samples selected randomly from the OOD utterance pool are different. After the model training is completed, the performance is evaluated by scoring OOD testing samples which are isolated from the OOD sample pool. At the same time,
MPCS scores are calculated for each set of OOD samples. Then we analyze how the MPCS measure of each OOD sample snapshot impacts the accuracy metrics.

4.1 Datasets

There are four datasets used in the experiments. First, we use Snips datasets as our in-domain samples. Apart from Snips, we also include a dataset containing 20000 StackOverflow questions from Xu et al. (2015). Table 1 shows some statistics along with some examples. Note that PlayMusic, SearchCreativeWork, SearchScreeningEvent come from Snips.

We also collect two sets of OOD utterances from two different sources generated randomly. The first dataset (10,000 sentences) will be used as the pool to generate candidates for training data. Another set (100,000 sentences) will be served as the testing dataset to evaluate the performance of classification models.

4.2 Models

We use a CNN-based model as the standard model across all comparisons for text classification. The model (Kim, 2014) is a multi-channel convolutional neural network model that contains two sets of word embedding layers. Both of them come from the pre-trained word vectors, but only one of them is updated during training. The embedding layers are followed by a convolutional layer combined with a ReLu activation function which generates a feature map. Then we apply a max-over-time pooling layer afterward to capture the important features. After concatenating the output from the pooling layer, we get the final feature representations and send them to the final fully-connected layer with dropout.

4.3 Evaluation

We use false acceptance rate (FAR) to measure the model accuracy for OOD detection (Lane et al., 2006).

\[
FAR = \frac{\text{Number of falsely accepted OOD samples}}{\text{Total number of OOD samples}}
\]

We also check false rejection rate (FRR) for each dataset.

\[
FRR = \frac{\text{Number of falsely accepted OOD samples}}{\text{Total number of OOD samples}}
\]

But we notice that for all datasets, FRR were close to perfect so we didn’t take FRR into comparison. We run each experiment for 30 times and compute the FAR for each snapshot of data.

4.4 Discussion

In this section, we examine the connection between model accuracy (via FAR) and data dispersion based on word embeddings.

As discussed in 3, the dispersion of datasets is measured based on MPCS. We first analyze how MPCS changes with respect to sample sizes. In order to achieve that, we obtain 200 sets of random samples and observed that as the sample size increases, the distribution would be shifted to the right. This is also understandable in the sense that as the sample size is large, the dataset is more likely to contain more unique words to have a
higher dispersion to cover more space (see Figure 1).

Text dispersion is an effective predictor of model accuracy. By using a CNN-based model for text classification, we find a positive correlation between dispersion and accuracy of text classification. As shown in Figure 2, for StackOverflow dataset, the FAR score has a positive correlation with its MPCS. Specifically, as illustrated in Table 2, we divide OOD samples into two groups, baseline and filtered, based on MPCS, and filtered sample group contains OOD utterances that has higher dispersion scores. We find on all four test datasets, low (baseline) MPCS, which means OOD text data is more disperse, leads to low (baseline) FAR, and vice versa.

This finding validates the assumption we introduce previously that more spread-out samples can improve the detection of OOD utterances.

5 Conclusion and future work

In this paper, we introduce the notion of augmenting out-of-domain training data with OOD utterances sampled based on word embeddings. Our experiments show that mean pairwise cosine similarity (MPCS) is a useful measure to describe the dispersion of OOD text data in the word embedding space, and this metric can be exploited to filter good OOD samples. With the same small size of OOD samples, one could significantly improve the model’s ability to detect OOD utterance by appending text data that are more dispersed based on representations and increasing coverage in feature space. This method can also be easily extended to other models and datasets.

We can possibly expand our research in multiple directions in the future. First of all, this work studies the internal structure of out-of-domain samples, but it has not linked the sample selection to the in-domain data samples. It would be interesting to discover methods to choose high-quality out-of-domain text data adapted to in-domain data. Another area that we could explore is the metric to measure dispersion. In this paper, we use MPCS to describe this characteristic of text data, but we would like to extend it to other metrics.

| Dataset            | MPCS - Baseline | FAR - Baseline | MPCS - Filtered | FAR - Filtered |
|--------------------|-----------------|----------------|-----------------|----------------|
| PlayMusic          | 0.135           | 0.096          | 0.128           | 0.090          |
| StackOverflow      | 0.126           | 0.170          | 0.119           | 0.144          |
| SearchCreativeWork | 0.109           | 0.354          | 0.105           | 0.344          |
| SearchScreeningEvent | 0.110       | 0.233          | 0.105           | 0.213          |

Table 2: FAR for low/high dispersion scores by datasets

Figure 1: Dispersion Score Distribution

Figure 2: MPCS v.s. FAR joint distribution for StackOverflow dataset
References

Long Chen, Fajie Yuan, Joemon M Jose, and Weinan Zhang. 2018. Improving negative sampling for word representation using self-embedded features. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pages 99–107. ACM.

Larry Heck and Dilek Hakkani-Tür. 2012. Exploiting the semantic web for unsupervised spoken language understanding. In 2012 IEEE Spoken Language Technology Workshop (SLT), pages 228–233. IEEE.

Douwe Kiela, Felix Hill, Anna Korhonen, and Stephen Clark. 2014. Improving multi-modal representations using image dispersion: Why less is sometimes more. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 835–841.

Joo-Kyung Kim and Young-Bum Kim. 2018. Joint learning of domain classification and out-of-domain detection with dynamic class weighting for satisfying false acceptance rates. arXiv preprint arXiv:1807.00072.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Ian Lane, Tatsuya Kawahara, Tomoko Matsui, and Satoshi Nakamura. 2006. Out-of-domain utterance detection using classification confidences of multiple topics. IEEE Transactions on Audio, Speech, and Language Processing, 15(1):150–161.

LR Lane, Tatsuya Kawahara, Tomoko Matsui, and Satoshi Nakamura. 2004. Out-of-domain detection based on confidence measures from multiple topic classification. In 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 1, pages I–757. IEEE.

Sungjin Lee and Igor Shalyninov. 2019. Contextual out-of-domain utterance handling with counterfeit data augmentation. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7205–7209. IEEE.

Kyo-Joong Oh, DongKun Lee, Chanyong Park, Young-Seob Jeong, Sawook Hong, Sungtae Kwon, and Ho-Jin Choi. 2018. Out-of-domain detection method based on sentence distance for dialogue systems. In 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), pages 673–676. IEEE.

Seonghan Ryu, Sangjun Koo, Hwanjo Yu, and Gary Geunbae Lee. 2018. Out-of-domain detection based on generative adversarial network. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 714–718.

Marzieh Saeidi, Ritwik Kulkarni, Theodosia Togia, and Michele Sama. 2017. The effect of negative sampling strategy on capturing semantic similarity in document embeddings. In Proceedings of the 2nd Workshop on Semantic Deep Learning (SemDeep-2), pages 1–8.

Gokhan Tur, Anoop Deoras, and Dilek Hakkani-Tür. 2014. Detecting out-of-domain utterances addressed to a virtual personal assistant. In Fifteenth Annual Conference of the International Speech Communication Association.

Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. 2015. Short text clustering via convolutional neural networks. In Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, pages 62–69, Denver, Colorado. Association for Computational Linguistics.