# Out-of-Domain Detection for Low-Resource Text Classification Tasks

**Ming Tan** [*†] **Yang Yu** [*†] **Haoyu Wang** [*†] **Dakuo Wang** [‡]
**Saloni Potdar** [†] **Shiyu Chang** [‡] **Mo Yu** [‡]
[†] IBM Watson [‡] IBM Research

## Abstract

Out-of-domain (OOD) detection for low-resource text classification is a realistic but understudied task. The goal is to detect the OOD cases with limited in-domain (ID) training data, since we observe that training data is often insufficient in machine learning applications. In this work, we propose an *OOD-resistant Prototypical Network* to tackle this **zero-shot OOD detection and few-shot ID classification** task. Evaluation on real-world datasets show that the proposed solution outperforms state-of-the-art methods in zero-shot OOD detection task, while maintaining a competitive performance on ID classification task.

## 1 Introduction

Text classification tasks in real-world applications often consists of 2 components- In-Doman (ID) classification and Out-of-Domain (OOD) detection components (Liao et al., 2018; Kim and Kim, 2018; Shu et al., 2017; Shamekhi et al., 2018). ID classification refers to classifying a user's input with a label that exists in the training data, and OOD detection refers to designate a special OOD tag to the input when it does not belong to any of the labels in the ID training dataset (Dai et al., 2007). Recent state-of-the-art deep learning (DL) approaches for OOD detection and ID classification task often require massive amounts of ID or OOD labeled data (Kim and Kim, 2018). In reality, many applications have very limited ID labeled data (i.e., few-shot learning) and no OOD labeled data (i.e., zero-shot learning). Thus, existing methods for OOD detection do not perform well in this setting.

One such application is the intent classification for conversational AI services, such as IBM Watson Assistant[1]. For example, Table 1 shows some of the utterances a chat-bot builder provided for training. Each class may only have less than 20 training utterances, due to the high cost of manual labelling by domain experts. Meanwhile, the user also expects the service to effectively reject irrelevant queries (as shown at the bottom of Table 1). The challenge of OOD detection is reflected by the undefined in-domain boundary. Although one can provide a certain amount of OOD samples to build a binary classifier for OOD detection, such samples may not efficiently reflect the infinite OOD space. Recent approaches, such as (Shu et al., 2017), make remarkable progress on OOD detection with only ID examples. However, such condition on ID data cannot be satisfied by the few-shot scenario presented in Table 1.

| Intent Label | Example |
|---|---|
| Help_List | List what you can help me with. |
| | Watson, I need your help |
| Schedule_Appointment | Can you book a cleaning with my dentist for me? |
| | Can you schedule my dentist's appointment? |
| End_Meeting | You can end the meeting now |
| | Meeting is over |
| ... | ... |
| OOD utterances | My birthday is coming! |
| | blah blah... |

Table 1: A few-shot ID training set for a conversation service for teleconference management, with OOD testing examples.

This work aims to build a model that can detect OOD inputs with limited ID data and zero OOD training data, while classifying ID inputs with a high accuracy. Learning similarities with

[*] Equal contributions from the corresponding authors: {mingtan,yu,wanghaoy}@us.ibm.com.

[1] https://www.ibm.com/cloud/watson-assistant/



*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3566–3572, Hong Kong, China, November 3–7, 2019. 

the meta-learning strategy (Vinyals et al., 2016) has been proposed to deal with the problem of limited training examples for each label (few-shot learning). In this line of work, *Prototypical Networks* (Snell et al., 2017), which was originally introduced for few-shot image classification, has proven to be promising for few-shot ID text classification (Yu et al., 2018). However the usage of prototypical network for OOD detection is unexplored in this regard.

To the best of our knowledge, this work is the first one to adopt a meta-learning strategy to train an OOD-Resistant Prototypical Network for simultaneously detecting OOD examples and classifying ID examples. The contributions of this work are two-fold: 1) Unified solution using a prototypical network model which can detect OOD instances and classify ID instances in a real-world low-resource scenario. 2) Experiments and analysis on two datasets prove that the proposed model outperforms previous work on the OOD detection task, while maintaining a state-of-the-art ID classification performance.

## 2 Related Work

**Out-of-Domain Detection** Existing methods often formulate the OOD task as a one-class classification problem, then use appropriate methods to solve it (e.g., one-class SVM (Schölkopf et al., 2001) and one-class DL-based classifiers (Ruff et al., 2018; Manevitz and Yousef, 2007). A group of researchers also proposed an auto-encoder-based approach and its variation to tackle OOD tasks (Ryu et al., 2017, 2018). Recently, a few papers have investigated ID classification and OOD detection simultaneously (Kim and Kim, 2018; Shu et al., 2017), but they fail in a low resource setting.

**Few-Shot Learning** While few-shot learning approaches may help with this low-resource setting, some recent work is promising in this regard. For example, (Vinyals et al., 2016; Bertinetto et al., 2016; Snell et al., 2017) use metric learning by learning a good similarity metric between input examples; some other methods adapt a meta-learning framework, and train the model to quickly adapt to new tasks with gradients on small samples, e.g., learning the optimization step sizes (Ravi and Larochelle) or model initialization (Finn et al., 2017). Though most of these approaches are explored for computer vision, recent studies suggests that few-shot learning is promising in the text domain, including text classification (Yu et al., 2018; Jiang et al., 2018), relation extraction (Han et al., 2018), link prediction in knowledge bases (Xiong et al., 2018) and fine-grained entity typing (Xiong et al., 2019), and we put it to test with the OOD detection task.

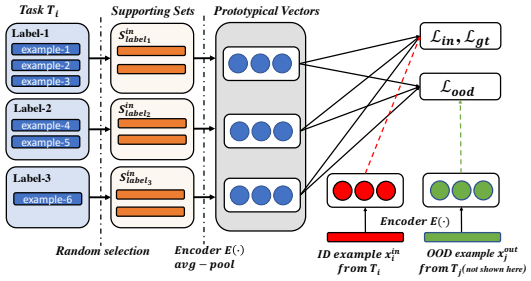


Figure 1: Model overview: the model maximizes likelihood of the ground-truth of ID example, minimizes distance between ID example and ground-truth, and maximizes the distance of OOD example and all ID labels.

## 3 Approach

In this paper, we target solving the zero-shot OOD detection problem for a few-shot **meta-test** dataset $D = (D^{train}, D^{test})$ by training a transferable prototypical network model from large-scale **independent source datasets** $\mathcal{T} = \{T_1, T_2, ..., T_N\}$ for dynamic construction of the **meta-train** set. Each task $T_i$ contains labeled training examples (note that a test set is not required in meta-train). $D$ is different from the traditional supervised close-domain classification dataset from two folds: 1) $D^{test}$ contains OOD testing examples, whereas $D^{train}$ only includes labeled examples for the target domain. 2) The training size for each label in $D^{train}$ is limited (e.g. less than 100 examples). Such limitations prevent existing methods from efficiently training a model for either ID classification or OOD detection using $D^{train}$ only.

We propose an OOD-resistant prototypical network for both OOD detection and few-shot ID classification. We follow (Snell et al., 2017) in few-shot image classification by training a prototypical network on $\mathcal{T}$ and directly perform prediction on $D$ without additional training. But our method is different from the prior work in that during the meta-training, while we maximize the likelihood of the true label for an example in $T_i$, we also sample an example from another meta-train task $T_j$ for the purpose of OOD training by maximizing the distance between the OOD instance and the prototypical vector of each ID label.

### 3.1 General Framework

As in Fig. 1, on a large-scale source dataset $\mathcal{T}$ with the following steps:

1. Sample a training task $T_i$ from $\mathcal{T}$ (e.g., the Book category of Amazon Review in Section 4), and another task $T_j$ from $\mathcal{T} - T_i$ (e.g. the Apps-for-Android category).
2. Sample an ID training example $x_i^{in}$ from $T_i$, and a simulated OOD example $x_j^{out}$ from $T_j$.
3. Sample $N$ labels ($N$=4) from $T_i$ in addition to the label of $x_i^{in}$. For the ground-truth label and $N$ negative labels, we select $K$ training examples for each label ($K$-shot learning, we set $K$=20). If a label has less than $K$ examples, we replicate the selected example to satisfy $K$. Therefore, $(N+1) \times K$ examples serve as a supporting set $\mathcal{S}^{in} = \{S_l^{in}\}_{l=1}^{N}$.
4. Given a batch of dynamically-constructed meta-train set $(x_i^{in}, x_j^{out}, \mathcal{S}^{in})$, $E(\cdot)$ encodes $x_i^{in}$, $x_j^{out}$ and the examples in $S_l^{in}$ using a deep network (Any DL structure can be used for the encoder, such as LSTM and CNN. Here we use a one-layer CNN with a mean pooling. The detailed CNN hyper-parameters are introduced in Section 5).
5. Following (Snell et al., 2017), a Prototypical Vector representation for each label is generated, by averaging all the examples' representations of that label.
6. The model is optimized by an objective function, defined by $x_i^{in}$ , $x_j^{out}$ and $\mathcal{S}^{in}$. Details in Section 3.2.
7. Repeat these steps for multiple epochs (5k in this paper) to train the model, and select the best model based on an independent meta-valid set $\mathcal{T}^{valid}$. $\mathcal{T}^{valid}$ contains tasks that are homogeneous to the meta-test task $D$.

The only trainable parameters of this model are in the encoder $E(\cdot)$. Therefore the trained model can be easily transferred to the few-shot target domain.

### 3.2 Training Objective and Runtime

Prototypical networks (Snell et al., 2017) minimize a cross-entropy loss defined on the distance metrics between $x_i^{in}$ and the supporting sets,

$$\mathcal{L}_{in} = -\log \frac{\exp \alpha F(x_i^{in}, S_{l_i}^{in})}{\sum_{l'} \exp \alpha F(x_i^{in}, S_{l'}^{in})} \quad (1)$$

where $l_i$ is the ground-truth label of $x_i$, $\alpha$ is a re-scaling factor. Here we define $F$ as a cosine similarity score (mapped to the range between 0 and 1)[2] between the $E(\cdot)$-encoded representations of $x$ and the prototypical vector of a label. Our experiments show this meta-learning approach is efficient for ID classification, but is not good enough for detecting the OOD examples.

We propose two more training losses in addition to the $L_{in}$ for OOD detection. The rationale behind this addition is to adopt the examples from other tasks as simulated OOD examples for the current meta-train tasks. Specifically, we first define a hinge loss on $x_j^{out}$ and the closest ID supporting set in $\mathcal{S}^{in}$, then we push the examples from another task away from the prototypical vectors of ID supporting sets.

$$\mathcal{L}_{ood} = max[0, \max_{l}(F(x_j^{out}, S_l^{in}) - \mathcal{M}_1)] \quad (2)$$

We expect optimizing only on $\mathcal{L}_{in}$ and $L_{ood}$ will lead to lower confidences on ID classification, because the system tends to mistakenly reduce the scale of $F$ in order to minimize the loss for OOD examples. Therefore we add another loss to improve the confidence of classifed ID labels.

$$\mathcal{L}_{gt} = max[0, \mathcal{M}_2 - F(x_i^{in}, S_{l_i}^{in}))] \quad (3)$$

The model is optimized on the three losses.

$$\mathcal{L} = \mathcal{L}_{in} + \beta \mathcal{L}_{ood} + \gamma \mathcal{L}_{gt} \quad (4)$$

where $\alpha$, $\beta$, $\gamma$, $\mathcal{M}_1$ and $\mathcal{M}_2$ are hyper-parameters, whose detailed values are shown in Section 5 .

During inference, the supporting set per label is generated by averaging the encoded representations of all instances of that label in $D^{train}$ and the prediction is based on $F(x, S_l^{in})$. OOD detection is decided with a *confidence* threshold.

## 4 Datasets

Our methods are evaluated on two datasets and each has many tasks and is divided into meta-train, meta-valid and meta-test sets, which are respectively used for background model training, evaluation and hyper-parameter selection.

**Amazon Review** [3]**:** We follow (Yu et al., 2018) to construct multiple tasks using the Amazon Review dataset (He and McAuley, 2016) . We convert it into a binary classification task of labeling the review sentiment (positive/negative). It has 21

[2]Following (Snell et al., 2017), we also tried squared Euclidean distance, but did not achieve better results.

[3]We will release Amazon data and our code at https://github.com/SLAD-ml/few-shot-ood

categories of products, each of which is treated as a task. We randomly picked 13 categories as meta-train, 4 as meta-test and 4 as meta-valid. (another 3 original categories are discarded due to not enough examples to make a dataset). We construct a 2-way 100-shot problem per meta-test task by sampling 100 reviews per label in a category. For the test examples in meta-test and meta-valid, we sample other categories' examples as OOD, merged with a equal number of ID instances. We used all available data for meta-train.

**Conversation Dataset:** An intent classification dataset for a AI conversational system. It has 539 categories/tasks. We allocate 497 tasks as meta-train, and 42 tasks as meta-test. This dataset is different and more difficult than the typical ID few-shot learning data: 1) Both the meta-test and meta-train tasks are not restricted to $N$-way $K$-shot classification, and the source dataset is highly imbalanced across labels; 2) Each task has a variety of labels (utterance intents), whereas Amazon data always has two labels. There are 29% OOD testing instances in meta-test, which are human-labeled and not generated from other tasks.

## 5 Experimental Results

**Baselines:** We compare our model **O-Proto** with 4 baselines: 1) **OSVM (Schölkopf et al., 2001)**: OSVM is trained on meta-test set, and learn a domain boundary by only examining ID examples. 2) **LSTM-AutoEncoder (Ryu et al., 2017)**: Recent work on OOD detection that uses only ID examples to train an autoencoder for OOD detection. 3) **Vanilla CNN**: A classifier with a typical CNN structure that uses a confidence threshold for OOD detection. 4) **Proto. Network (Snell et al., 2017)**: A native prototypical network trained on $\mathcal{T}$ with only the loss $\mathcal{L}_{in}$, which uses a confidence threshold for OOD detection. We test the Proto. Network with both CNN and bidirectional LSTM as the encoder $E(\cdot)$.

**Hyper Parameters:** We introduce the hyper-parameters of our model and all baselines below.

We use Python scikit-learn One-Class SVM as the basis of our OSVM implementation. We use Radial Basis Function (RBF) as the kernel and the gamma parameter is set to auto. We use squared hinge loss and L2 regularization.

We follow the same architecture as proposed in (Ryu et al., 2017) for the LSTM-Autoencoder. In LSTM, we set the input embedding dimension as 100 and hidden as 200. We use RMSprop as the optimizer with a learning rate of 0.001. We train the LSTM with a batch size of 32 and 100 epochs. For Autoencoder, we set the hidden size as 20. We use Adam as the optimizer with a learning rate of 0.001. We train the model with a batch size of 32 for 10 epochs.

For vanilla CNN, we use the most common CNN architecture used in NLP tasks, where the convolutional layer on top of word embedding has 128 filters followed by a ReLU and max pooling layer before the final softmax. We use Adam as the optimizer with a learning rate of 0.001. We train the model with a batch size 64 for 100 epochs.

Our proposed model O-Proto uses the similar CNN architecture, the optimizer and the learning rate in the previous Vanilla CNN. The input word embeddings are pre-trained by 1-billion-token Wikipedia corpus. We set the batch size as 10. In Eq. 1, 2, 3 and 4, $\alpha$, $\beta$, $\gamma$, $\mathcal{M}_1$ and $\mathcal{M}_2$ are hyper-parameters, which we fix $\beta$, $\gamma$ as 1.0 by default, and set $\alpha$, $\mathcal{M}_1$ and $\mathcal{M}_2$ as 10.0, 0.4 and 0.8 according to the meta-valid performance of Amazon dataset. The sentence encoder, CNN, has 200 filters, followed by a tanh and mean pooling layer before the final regression layer. The maximum length of tokens per example is 40, and any words out of this range will be discarded. During training, we set the size of sampled negative labels Step 3 (section 3.1) to at most four, so there will be maximum five labels involved in a training step (1 positive, 4 negative). The supporting set size for each label are 20.

To make a fair comparison, we follow the same hyper-parameters as O-Proto in Proto. Network, except that the weight of $\mathcal{L}_{ood}$ and $\mathcal{L}_{gt}$, $\beta$ and $\gamma$, are set to zero.

**Evaluation Metrics:** Following (Ryu et al., 2017; Lane et al., 2007), we use a commonly used OOD detection metric, equal error rate (**EER**), which is the error rate when the confidence threshold is located where false acceptance rate (**FAR**) is equivalent to false rejection rate (**FRR**).

$$FAR = \frac{Number\ of\ accepted\ OOD\ sentences}{Number\ of\ OOD\ sentences}$$
$$FRR = \frac{Number\ of\ rejected\ ID\ sentences}{Number\ of\ ID\ sentences}$$

We use class error rate **CER** to reflect ID performance. Lastly, we applied the threshold used in the EER to ID test examples to test how many ID

| (%) | Conversation EER | CER | Comb. | Amazon Review EER | CER | Comb. |
|---|---|---|---|---|---|---|
| OSVM | 63.6 | - | - | 47.6 | - | - |
| LSTM AutoEnc. | 48.0 | 78.4 | 79.5 | 45.4 | 29.3 | 38.6 |
| Vanilla CNN | 26.4 | 76.8 | 77.6 | 47.7 | 34.4 | 42.8 |
| Proto. Network | 26.9 | 32.5 | 44.5 | 46.5 | **7.3** | 47.6 |
| O-Proto ($\mathcal{L}_{in} + \mathcal{L}_{gt}$) | 27.6 | 33.3 | 46.2 | 47.8 | 7.4 | 48.9 |
| O-Proto ($\mathcal{L}_{in} + \mathcal{L}_{ood}$) | 24.5 | 30.1 | 41.2 | 24.7 | 9.7 | 30.1 |
| O-Proto (all) | **24.1** | **29.6** | **40.8** | **24.0** | 9.1 | **29.1** |
| Proto. with bilstm | 25.0 | 32.5 | 42.6 | 45.1 | **6.8** | 46.0 |
| O-Proto with bilstm | **22.0** | **30.5** | **39.8** | **21.9** | 9.0 | **27.1** |

Table 2: O-Proto is compared with other baselines for Conversation and Amazon data

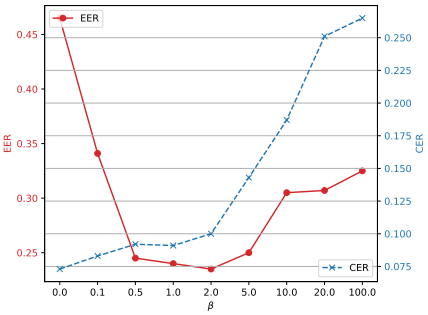


Figure 2: Various $\beta$ for Amazon data

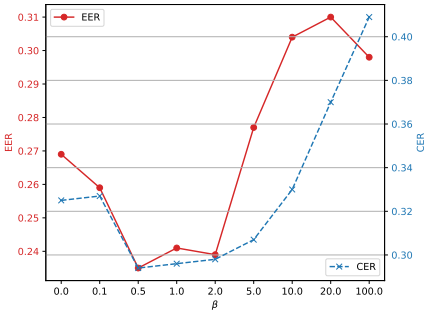


Figure 3: Various $\beta$ for Conversation data

| EER (%) | K=1 | =5 | =10 | =20 | =100 |
|---|---|---|---|---|---|
| Proto. | 53.9 | 46.4 | 44.5 | 41.0 | 46.5 |
| O-Proto. | 31.1 | 25.1 | 24.0 | 23.4 | 24.1 |

Table 3: Compare our model with Prototype Network on EER when choosing various $K$-shot values

examples are rejected by the OOD detection, as **Combined-CER** (Comb. in Table 2).

**Results:** Table 2 compares the proposed model and baselines on EER and CER on the two datasets. [4] For prototypical-network related models, we randomly initialize the model parameters by 10 times and report the averaged metrics. *OSVM, LSTM-AutoEncoder and Vanilla CNN* perform poorly on OOD detection as expected, as they require larger number of ID examples which is not a few-shot scenario. *Proto.Network* has a better performance on ID classification. But it is not designed for OOD, thus it does not perform well on OOD detection. *O-Proto* achieves significantly better EERs (2.8% improvement on Conversation, and 22.5% on Amazon), yielding competitive results on CER compared to *Proto.Network*. These lead to a remarkable improvement on Combined-CER. *O-Proto* improves less on EER in Conversation than Amazon, because some Conversation tasks actually come from the conversational service providers belonging to similar business domains. Our model is better even when meta-train datasets are from slightly different domains. Moreover, in Table 2 we show the ablation study by removing $\mathcal{L}_{ood}$ and $\mathcal{L}_{gt}$ from *O-Proto*, respectively. *O-Proto* without $\mathcal{L}_{ood}$ completely loses the ability of OOD detection. Compared to the one without $\mathcal{L}_{gt}$, *O-Proto* with all losses gives a mild improvement. We also observe a more stable testing performance among epochs during training. Finally, we replace the CNN encoders with bidirectional LSTMs (the bottom of Table 1), which yields the same dimension of sentence representations as CNN. For Conversation data, we achieve the best performance on validation set when $\alpha$ and $\gamma$ are 0.5. We observe comparable performances with respect to *Proto.Network* and *O-Proto*, showing that our proposed OOD approach is not limited to a specific sentence encoding architecture.

**Improvement in different K-shot settings:** On the Amazon data, we construct different $K$-shot tasks as meta-test (results shown in Table 3), and observe consistent improvements on EER.

**Effect of $\beta$:** Fig. 2 and 3 show EER and CER with different $\beta$ values on the two datasets. We observe within a proper range of $\beta$ (between 0.5 and 2.0), the model can provide stable improvement on EER, guaranteeing competitive CER results.

## 6 Conclusion

Inspired by the Prototypical Network, we propose a new method to tackle the OOD detection task in low-resource settings. Evaluation on real-world datasets demonstrates that our method performs favorably against state-of-the-art algorithms on the OOD detection, without adversely affecting performance on the few-shot ID classification.

[4] No CER reported for *OSVM* as it treats ID as one class and does not support ID classification.

## References


Luca Bertinetto, João F. Henriques, Jack Valmadre, Philip H. S. Torr, and Andrea Vedaldi. 2016. Learning feed-forward one-shot learners. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 523–531, USA. Curran Associates Inc.

Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. 2007. Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 210–219.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135, International Convention Centre, Sydney, Australia. PMLR.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Xiang Jiang, Mohammad Havaei, Gabriel Chartrand, Hassan Chouaib, Thomas Vincent, Andrew Jesson, Nicolas Chapados, and Stan Matwin. 2018. Attentive task-agnostic meta-learning for few-shot text classification. *The Second Workshop on Meta-Learning at NeurIPS*.

Joo-Kyung Kim and Young-Bum Kim. 2018. Joint learning of domain classification and out-of-domain detection with dynamic class weighting for satisficing false acceptance rates. In *Proc. Interspeech 2018*, pages 556–560.

I. Lane, T. Kawahara, T. Matsui, and S. Nakamura. 2007. Out-of-domain utterance detection using classification confidences of multiple topics. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):150–161.

Q. Vera Liao, Mas-ud Hussain, Praveen Chandar, Matthew Davis, Yasaman Khazaeni, Marco Patricio Crasso, Dakuo Wang, Michael Muller, N. Sadat Shami, and Werner Geyer. 2018. All work and no play? In *In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*.

Larry Manevitz and Malik Yousef. 2007. One-class document classification via neural networks. *Neurocomput.*, 70(7-9):1466–1481.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. *International Conference on Learning Representations, 2017*.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4393–4402, Stockholmsmssan, Stockholm Sweden. PMLR.

Seonghan Ryu, Seokhwan Kim, Junhwi Choi, Hwanjo Yu, and Gary Lee. 2017. Neural sentence embedding using only in-domain sentences for out-of-domain sentence detection in dialog systems. *Pattern Recognition Letters*, 88:26–32.

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. Association for Computational Linguistics.

Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471.

Ameneh Shamekhi, Q Vera Liao, Dakuo Wang, Rachel KE Bellamy, and Thomas Erickson. 2018. Face value? exploring the effects of embodiment for a group facilitation agent. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 391. ACM.

Lei Shu, Hu Xu, and Bing Liu. 2017. Doc: Deep open classification of text documents. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2911–2916, Copenhagen, Denmark. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4077–4087. Curran Associates, Inc.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 3637–3645, USA. Curran Associates Inc.

Wenhan Xiong, Jiawei Wu, Deren Lei, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Imposing label-relational inductive bias for extremely fine-grained entity typing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 773–784.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2018. One-shot relational learning for knowledge graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1980–1990.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1206–1215. Association for Computational Linguistics.