Unknown Intent Detection using Multi-Objective Optimization on Deep Learning Classifiers

Prerna Prem[1], Zishan Ahmad[1], Asif Ekbal[1]
Shubhashis Sengupta[2], Sakshi Jain[2], Roshini Rammani[2]
[1]Indian Institute of Technology Patna [2]Accenture
prernaprem21@gmail.com zeeman.zishan@gmail.com asif@iitp.ac.in
shubhashis.sengupta@accenture.com sakshi.c.jain@accenture.com
roshni.r.ramnani@accenture.com

Abstract

Modelling and understanding dialogues in a conversation depends on identifying the user intent from the given text. Unknown or new intent detection has became an important task, as in a realistic scenario a user intent may frequently change over time and divert even to an intent previously not encountered. This task of separating the unknown intent samples from known intents, is challenging as the unknown user intent can range from intents similar to the known ones or to something completely different. Prior research for intent discovery often considers this problem as a classification task where an unknown intent can belong to a finite or predefined set of known intent classes, thus limiting the scope of such research. In this paper, we tackle the problem of detecting a completely unknown intent without any prior hints about the kind of classes belonging to unknown intents. We propose an effective post-processing method using multi-objective optimization to tune an existing neural network based intent classifier making it capable of detecting unknown intents. Thus our method can be plugged into existing deep-learning based classifiers and fine-tuned for the purpose of unknown intent detection. We perform experiments using existing state-of-the-art intent classifiers and use our method on top of them for unknown intent detection. Our experiments across different domains and real-world datasets show that our method yields significant improvement compared with the state-of-the-art methods for unknown intent detection.

1 Introduction

Detecting whether an intent is unknown or new in a dialogue system has become an important task for improving customer satisfaction. Since user intent may frequently change over time in many realistic scenarios, unknown (new) intent detection has become an essential problem so, solving which can enhance system interaction with the customer. This task is challenging since there is no prior knowledge of the type or the exact numbers of unknown intents that would be encountered in the future.

We model unknown intent detection as an (m+1)-class classification task as suggested by (Shu et al., 2017; Lin and Xu, 2019; Zhang et al., 2020) and group unknown classes into the (m+1)th class. We aim to identify the known intent samples accurately, while at the same time finding the unknown intent samples without having prior knowledge about the unknown intents. Recently, to solve this problem, researchers have used deep neural networks for open classification. OpenMax (Bendale and Boult, 2016) fits Weibull distribution to the outputs of the penultimate layer, but requires negative samples for selecting the best hyperparameters. The MSP (Hendrycks and Gimpel, 2016) calculates the softmax probability of known samples and discards the low confidence unknown samples with the threshold. In our approach, we attempt to solve the problem of unknown intent detection with added constraints such as not having prior knowledge of the finite set of intents. The main contributions of this paper are:

1. We develop a novel post processing method using multi-objective optimization (non-deterministic genetic algorithm-NSGA2) by optimising two objectives i.e. recall and precision in order to obtain the optimal thresholds for each intent class.

2. Our approach does not require any model architecture modification and can be applied on top of any deep neural network model.

The rest of the paper is organized as follows. In Section 2 we cover literature survey on previous work done for intent classification and open intent

detection. In Section 3 we elaborate on the proposed architecture. In Section 4 we discuss the experimental setup and the dataset used. In Section 5 we analyse the results of detecting the unknown intents. Finally, Section 6 concludes the paper with future work that can be done explored in this field.

2 Related work

Several works have been done for intent detection in dialogue systems in recent years (Min et al., 2020; Qin et al., 2020; Zhang et al., 2018; Niu et al., 2019; Qin et al., 2019). Most of the works are based on closed world classification without any open intent. (Srivastava et al., 2018) proposed zero-shot learning for intent detection. However, ZSL is different from our task as it only contains finite known set of classes during testing. (Kim and Kim, 2018) try to optimise the intent classifier together with an out-of-domain detector, which was trained using out-of-domain samples. The generative method (Yu et al., 2017) uses adversarial learning to generate positive and negative examples from known classes but the method does not work well in the discrete data space like text. (Ryu et al., 2018) proposed generative adversarial network (GAN) to train on the ID samples and use the discriminator to detect the OOD samples. (Nalisnick et al., 2018; Mundt et al., 2019) showed that deep generative models fail to capture high-level semantics on real world data. (Jain et al., 2014) fit the probability distributions to statistical Extreme Value Theory (EVT) using a Weibull-calibrated multi-class SVM to detect the unnormalized posterior probability of inclusion for open set problems. ODIN (Liang et al., 2017) enlarged the differences between known and unknown samples by using temperature scaling and input preprocessing but all the above method need negative samples for selecting the decision boundary or probability threshold. DOC (Shu et al., 2017) instead of using Softmax as the final output layer built a multi-class classifier with a 1-vs-rest final layer which contains a sigmoid function for each seen class to reduce the open space risk. Zero-shot intent classification aims to generalize knowledge and concepts learned from seen intents to recognize unseen intents. Early methods (Ferreira et al., 2015a; Ferreira et al., 2015b) explore the relationship between seen and unseen intents by introducing external resources such as manually defined attributes or label ontologies, but they are usually expensive to obtain. To deal with this, some methods (Chen et al., 2016; Kumar et al., 2017) map the utterances and intent labels to an embedding space and then model their relations in the same space. IntentCapsNet-ZS (Xia et al., 2018) extends capsule networks (Sabour et al., 2017) for zero-shot intent classification by transferring the prediction vectors from seen classes to unseen classes. ReCapsNet (Liu et al., 2019) shows that IntentCapsNet-ZS hardly recognizes utterances from unseen intents in the generalized zero-shot classification scenario, and proposes to solve this issue by transferring the transformation matrices from seen intents to unseen intents. These approaches also need unknown intents embedding for classifying the unknown intent sample.

3 Methodology

We train two different model architectures for intent classification and use our post-processing steps on top of these to obtain optimal results. The pipeline of the system processes is shown in Figure 1. We describe the models along with our novel post-processing steps in this section.

3.1 Models

3.1.1 Bi-LSTM

We train Bi-LSTM to get the prediction scores and use these prediction scores to get the optimal thresholds for each known intent class by optimizing the correct classification rate and the misclassification rate on the training data. Given an utterance with maximum word sequence length l, we transform a sequence of input words $w_{1:l}$ into m-dimensional word embedding $v_{1:l}$, which is used by forward and backward LSTM to produce feature representations x:

$$\vec{x_t} = LSTM(v_t, \vec{c}_{t-1})$$

$$\vec{x_t} = LSTM(v_t, \vec{c}_{t-1})$$

$$x = [\vec{x_l} : \vec{x_1}]$$

where v_t denotes the word embedding of input at time step t. $\vec{x_t}$ and $\vec{x_t}$ are the output vector of forward and backward LSTM respectively. $\vec{c_t}$ and $\vec{c_t}$

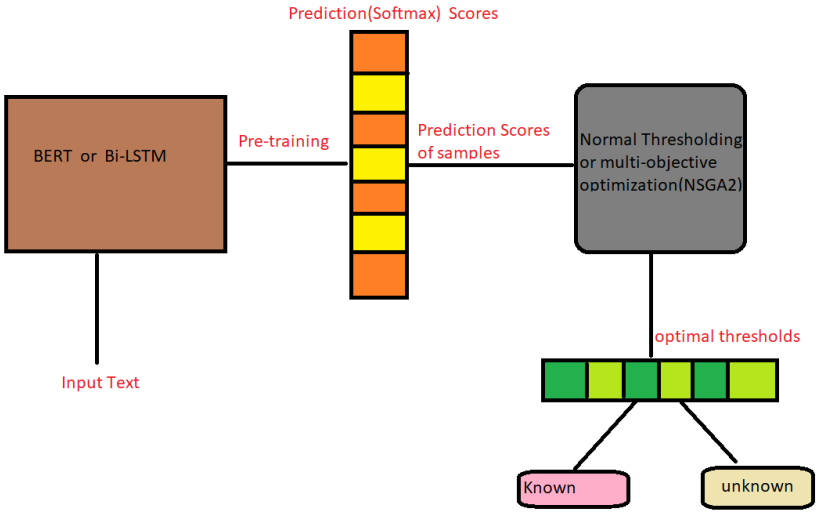

Figure 1: The system architecture consisting of two parts (i). BERT or Bi-LSTM model for softmax score prediction and (ii). Normal Thresholding or NSGA2 for tuning the thresholds of class scores

are the cell state vectors of forward and backward LSTM respectively. We concatenate the last output vector of forward LSTM $\vec{x_l}$ and the first output vector of backward LSTM $\vec{x_1}$ into x as the sentence representation. It captures high-level semantic concepts learned by the model. The representation x is then fed to an n neuron feed forward layer where n is the number of known intent classes in the dataset. The n dimensional representation obtained is converted to probability distribution by using a 'Softmax' function.

3.1.2 BERT

We fine tune the pre-trained BERT model to get the 'softmax' classification scores of the input samples. Given ith input sentence s_i we append a $[CLS]$ token at the beginning of the sentence. We obtain the token embeddings of the sequence $[CLS, T1, , TN] \quad \epsilon \quad R^{(N+1)*H}$ from the last hidden layer of BERT. Here the $[CLS]$ vector representation is used for text classification, N is the sequence length and H is the hidden layer size. We calculate the prediction scores by applying 'Softmax' function to the last layer output(logits(x_i)) of the trained BERT model.

3.2 Pre-training

For getting the optimal thresholds we require prediction scores of the samples for which we had to first train our base models by following the procedure mentioned in section 3.1. As we don't have unknown intent samples we use known intents as prior knowledge to train the model. In order to reflect the effectiveness of the learned optimal thresholds we use cross-Entropy loss L_s to train our both the base models.

$$L_s = \frac{-1}{N} \sum_{i=1}^{N} y_i log(\hat{y_i})$$

where N is total number of training samples y_i is true label and $\hat{y_i}$ is predicted label Then, we use the pre-trained model to get the prediction scores of the input samples that is used further for threshold tuning of each known intent class.

3.3 Finding optimal threshold for each known intent class

We use the pre-trained models to get the prediction scores corresponding to each sample by giving the training data samples to the base models. After getting the prediction scores we apply different techniques for getting the optimal thresholds for each

Datasets	Classes (intents)	#Training	#Validation	#Test
Bank_catridge	14	1020	120	240
Banking	77	9003	1000	3080
MultiWOZ	8	37542	4643	4720

Table 1: Statistics of the dataset being used in our experiment.

Datasets	*Few examples of intents*
Bank_catridge	['Bal_Inquiry', 'Card_Activation', 'card_declined', 'cheque_book_Req', 'credit_query', 'direct_deposit', 'freeze_account', 'inter_transfer', 'mortgage_processing', 'replacement_card_duration', 'report_fraud', 'report_lost_card', 'update_so_dd']
Banking	['transfer_timing', 'order_physical_card', 'card_acceptance', 'balance_not_updated_after_bank_transfer','card_swallowed', 'top_up_by_bank_transfer_charge', 'card_delivery_estimate', 'transfer_not_received_by_recipient']
MultiWOZ	['Find_restaurant','book_train', 'Find_attraction', 'book_taxi', 'book_restaurant']

Table 2: Few intents present in each of the dataset.

known intent class. The different techniques are as follows:

3.3.1 Normal thresholding

First, the input text containing the training data samples is fed to the deep learning classifier to get the prediction scores corresponding to each samples. These prediction scores (PS) and the list of thresholds (T) ranging from 0.1 to 0.99 increasing by 0.01 in each step is used to calculate the correct classification matrix (CCM) and the mis-classification matrix (MCM). The set of prediction scores is a matrix of $N \times M$ where N is the total number of training samples and M is the number of known intent classes. This set of prediction scores and the list of thresholds containing K threshold values is used to calculate correct classification matrix(CCM) and the miss-classification matrix(MCM).

Let $C(X)$ be the output class, Y the ground truth class, and $(.)$ the enumeration function, the standard definition for correctly classified sample (or true positives) rate of an intent class i is given by Equation 1:

$$CC_i = \frac{(C(X) = i \; AND \; Y = i)}{Y = i} \tag{1}$$

We can also write the standard definition of misclassified sample rate (or false negatives) of an intent class i as given by Equation 2:

$$MC_i = \frac{(C(X) \neq i \; AND \; Y = i)}{Y = i} \tag{2}$$

The correct classification rate (CC) and misclassification (MC) rate of an intent i can be extended by introducing the thresholds τ_i and by adding the unsure classification (UC) rate, for each intent as shown in Equation 3, 4 and 5.

$$CC_i(\tau_i) = \frac{(C(X) = i \; AND \; S(X) > \tau_i) \; AND \; (Y = i)}{Y = i} \tag{3}$$

$$MC_i(\tau_i) = \frac{(X) \neq i \; AND \; S(X) > \tau_i) \; AND \; (Y = i)}{Y = i} \tag{4}$$

$$UC_i(\tau_i) = \frac{\begin{array}{c}((C(X) = i) or (C(X) \neq i) \; AND \\ (S(X) < \tau_i) \; AND \; (Y = i)\end{array}}{Y = i} \tag{5}$$

For each intent we have:

$$CC_i(\tau) + MC_i(\tau) + UC_i(\tau) = 1$$

CCM is a matrix of $K \times M$ dimension containing the correct classification rate of each intent class

corresponding to each threshold in the threshold list i.e each entry CC_{ij} is calculated using equation 6.

$$CC_{ij} = \sum_{i=1}^{N} \frac{(C(X) = i \; AND \; S(X) > \tau_j) \; AND \; (Y = i)}{Y = i} \tag{6}$$

MCM is a matrix of $K \times M$ dimension containing the mis-classification rate of each intent class corresponding to each threshold in the threshold list i.e each entry MC_{ij} is calculated using equation 7.

$$MC_{ij} = \sum_{i=1}^{N} \frac{(C(X) \neq i \; AND \; S(X) > \tau_j) \; AND \; (Y = i)}{Y = i} \tag{7}$$

After obtaining these two matrices we obtain optimal τ_j for each known intent class by the following technique. We keep the best correct classification rate while reducing the mis-classification rate. For this, we used two steps. First, we identified the threshold(s) τ which maximizes $CC_i(\tau)$. Since several thresholds could reach this maximum, we get a set of threshold(s) Seg_1. Then, we selected the threshold with the lower$MC_i(\tau)$. This can be mathematically written as:

$$s = argmax_{\tau}(CC_i(\tau))$$

$$\tau_i = argmin_{\tau' \epsilon s}(MC_i(\tau'))$$

3.3.2 Multi-objective optimization(NSGA2)

To get the optimal threshold we used Non-dominated Sorting Genetic Algorithm II (NSGA-II) which is a multi-objective genetic algorithm, proposed by (Deb et al., 2002). In the structure of NSGA-II, in addition to genetic operators, crossover and mutation, two specialized multi-objective operators and mechanisms are defined and utilized. These are as follows:

- **Non-dominated Sorting:** The population is sorted and partitioned into fronts (F1, F2, etc.), where F1 (first front) indicates the approximated Pareto front.
- **Crowding Distance**: It is a mechanism of ranking among members of a front, which are dominating or dominated by each other.

We optimize for two objective (1). Correct classification rate (CC) and (2). Precision of the known intents. The NSGA2 takes threshold values of an intent as the input variable (values ranging from 0.1 to 0.99). It then uses prediction scores of samples from the pre-trained base model to perform optimization on the two objective function explained in detail in section 3.3.1 to get an optimal threshold for each known intent class. We initialize the population by randomly selecting the values from the range of the threshold variable and then we calculate the two objective values for each entry in the initial population. Next we perform a non-dominated sorting in the combination of parent and offspring populations and classify them by fronts , i.e. they are sorted according to an ascending level of non-domination. After that we fill new population according to front raking. If one front is taking partially, we perform Crowding-sort that uses crowding distance that is related with the density of solutions around each solution. The less dense are preferred. Then we create offspring population(children) from this new population using crowded tournament selection (It compares by front ranking, if equal then by crowding distance), crossover and mutation operators. We keep the best entries of the population in fronts.

We run the same procedure 1000 times to get a set of optimal thresholds for each known intent class. From this set of thresholds we choose the maximum threshold. This optimal threshold is used to decide upon known and unknown intent samples.

3.4 Testing

During testing, when a new sample (unseen class) is encountered it is first fed to the base model (BiLSTM or BERT) to get the corresponding prediction scores. After getting the prediction scores we compare each entries in the prediction scores with the corresponding optimal thresholds and if we find all the entries to be less than the corresponding optimal thresholds we classify that sample as unknown else we classify the sample to the one known intent class for which the prediction score is higher than the corresponding optimal threshold.

Text	True Label	Predicted Label (ADB)
Is Visa or Mastercard available?	visa or mastercard	supported cards and currencies
The app is showing an ATM withdrawal that I didn't make.	cash withdrawal not recognized	declined cash withdrawal
I did what you told me earlier and contacted the seller for a refund directly, but nothing is happening! It's been a week and I still haven't got anything. Please just give me back my money	refund not showing uo	balance not updated after cheque or cash deposit

Table 3: Samples texts whose intents are mis-classified by the ADB model but are correctly identified by out BERT+NSGA2 model

Model/Dataset	***Bank catridge***			***Banking***			***MultiWOZ***		
	75%	50%	25%	75%	50%	25%	75%	50%	25%
Bi-LSTM +NT	0.28	0.42	0.48	0.22	0.34	0.46	0.03	0.04	0.05
Bi-LSTM+ NSGA2	0.45	0.54	0.64	0.35	0.4	0.52	0.12	0.56	0.76
BERT+NT	0.33	0.49	0.53	0.66	0.70	0.73	0.06	0.07	0.11
BERT+NSGA2	**0.82**	**0.9**	**0.85**	**0.68**	**0.80**	**0.9**	**0.42**	**0.84**	**0.93**
ADB	0.74	0.77	0.8	0.67	0.78	0.85	0.25	0.72	0.86

Table 4: F1 score of detecting unknown intent class samples with 75% ,50% and 25% of total intent class as known class on BANKING, Bank_Catridge and MultiWOZ dataset.

4 Datasets and Experiments

4.1 Dataset

We use three datasets on which we conduct our experiment. The detailed statistics of the datasets are shown in Table 1. Few intents of each dataset is shown in Table2

4.1.1 Banking

A fine-grained dataset in the banking domain (Casanueva et al., 2020). It contains 77 intents and 13,083 customer service queries.

4.1.2 Bank-catridge

A dataset which contain manually updated samples using paraphrasing tools along with manual modification. It contains 36 intents but we clubbed some of the intents into one to make overall 14 intents, So that the samples per intent is constant. It has almost 100 samples per intents.

4.1.3 MultiWOZ

MultiWOZ is a dialogue dataset which contains multiple domains such as "restaurant booking", "train booking", "attraction booing" and "taxi booking". It contains 2 main intents per domain namely "find" and "book". The total number of intents are 8 and number of samples per intents are not uniform.

4.2 Experimental Setups

We have kept 25% of the overall intent classes in training and validation set as masked but we keep those intents unmasked in the test set. To have a fair evaluation on the imbalanced dataset, we randomly select known classes by weighted random sampling without replacement in the training and validation set. If a class has more examples, it is more likely to be chosen as the known class. However, a class with fewer examples still has a chance to be selected.

For BERT we use 'bert-base-uncased' with 12-layer transformer model and fine-tune it using training set. We keep the learning-rate as 2e-5, the train-

Text	True Label	Predicted Label (BERT+NT)
What is the number of days I have to wait for my Europe transfer?	balance not updated after bank transfer	transfer timing
I need to find out why my transfer didn't get there.	declined transfer	transfer not received by recipient
I have a pending cash withdrawal	balance not updated after cheque or cash deposit	pending cash withdrawal
I don't find your services useful anymore, how do I delete my account?	edit personal details	terminate account
Will it cost more money if my currency needs to be exchanged?	exchange via app	exchange charge

Table 5: Samples texts whose intents are mis-classified by the BERT + NT model but are correctly identified by out BERT + NSGA2 model

ing batch size 128 and train for 50 epochs. For Bi-LSTM we set keep output dimension as 128 and train for 50 epochs with early stopping.

In NSGA2 we keep the chromosome size as 1 as we require only 1 optimal threshold per intent class. We have experiment with different threshold values as input and found that a range between (0.1-0.9) gives better result. The number of generations is kept 1000 and the population size is 100, num_of_tour_participants is 2, tournament_probability is 0.9, crossover_parameters is 2, mutation_parameters is 5.

For evaluating the models we use macro F1 score. We compare our method with following state of the art model: ADB (Shu et al., 2017) and with different variants of the proposed model as follows: (i). Bi-Lstm + normal-thresholding, (ii). Bi-Lstm + NSGA2 and (iii). BERT + normal-thresholding and (iv). BERT+NSGA2.

5 Result and Analysis

Table 4 shows the F1 score of detecting unknown intent class samples with 75%, 50% and 25% of total intent classes as known class on Banking, Bank_Catridge and SNIPS dataset. The best results are highlighted in bold. Comparing with the best scores of previous state-of-the-art and different variants of our approach we can see that our final model BERT+NSGA2 gives better results than the state-of-the-art and the different variants of our proposed model. Using BERT as the baseline, our model improves significantly in terms of F1 score on the Banking dataset as this dataset has more training samples as compared to other two dataset. Comparing with ADB our approach gives 8%, 13% and 5% improvement on Bank_catridge dataset, 1%, 2% and 5% improvement on Banking dataset and 17% ,12% and 7% improvement on MultiWOZ dataset. It can be explained from the results that our BERT+NSGA2 approach is able to learn tighter thresholds to decide upon known and unknown intent samples. Using Normal thresholding technique where the objective functions are optimised sequentially doesn't work well as optimizing one objective function can counter the optimization of another objective. This problem is addressed by multi-objective optimization technique which to satisfy all objective functions, finds a set of optimal solutions instead of one optimal solution. Some examples that are correctly classified by the BERT+NSGA2 and not by BERT+NT are shown in Table5. We can see that multi-objective optimization plays a vital role in predicting the unknown samples correctly as compared to normal optimization. Some examples that are correctly classified by the BERT+NSGA2 and not by ADB are shown in Table 3: From the examples in the table we can say that our BERT+NSGA2 is giving importace to words that are there in the unknown intent like "refund", "visa", "master_card" and "didn't make" to make the decision between

known and unknown intent class. Our model is learning tighter thresholds because of parallel optimization of objective functions, giving better results in many cases.

6 Conclusion and Future Work

We propose a novel post-processing method for unknown intent classification. After pre-training the model with labeled samples, our model can automatically learn precise thresholds to separate the known intent from unknown intent sample. Our method has no requirement for unknown intent or model architecture modification. Extensive experiments on three benchmark datasets show that our method yields significant improvements over the compared baseline models. After getting the samples classified as unknown from the model we are trying to cluster those samples. We are working on improving the cluster quality by varying the input features. This kind of work using clustering is not done so far.

References

Abhijit Bendale and Terrance E Boult. 2016. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572.

Inigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

Yun-Nung Chen, Dilek Hakkani-Tür, and Xiaodong He. 2016. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6045–6049. IEEE.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.

Emmanuel Ferreira, Bassam Jabaian, and Fabrice Lefevre. 2015a. Online adaptative zero-shot learning spoken language understanding using word-embedding. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5321–5325. IEEE.

Emmanuel Ferreira, Bassam Jabaian, and Fabrice Lefevre. 2015b. Zero-shot semantic parser for spoken language understanding. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Lalit P Jain, Walter J Scheirer, and Terrance E Boult. 2014. Multi-class open set recognition using probability of inclusion. In *European Conference on Computer Vision*, pages 393–409. Springer.

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. *arXiv preprint arXiv:1807.00072*.

Anjishnu Kumar, Pavankumar Reddy Muddireddy, Markus Dreyer, and Björn Hoffmeister. 2017. Zero-shot learning across heterogeneous overlapping domains. In *INTERSPEECH*, pages 2914–2918.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. 2017. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*.

Ting-En Lin and Hua Xu. 2019. Deep unknown intent detection with margin loss. *arXiv preprint arXiv:1906.00434*.

Han Liu, Xiaotong Zhang, Lu Fan, Xuandi Fu, Qimai Li, Xiao-Ming Wu, and Albert YS Lam. 2019. Reconstructing capsule networks for zero-shot intent classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4801–4811.

Qingkai Min, Libo Qin, Zhiyang Teng, Xiao Liu, and Yue Zhang. 2020. Dialogue state induction using neural latent variable models. *arXiv preprint arXiv:2008.05666*.

Martin Mundt, Iuliia Pliushch, Sagnik Majumder, and Visvanathan Ramesh. 2019. Open set recognition through deep neural network uncertainty: Does out-of-distribution detection require generative classifiers? In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. 2018. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*.

Peiqing Niu, Zhongfu Chen, Meina Song, et al. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. *arXiv preprint arXiv:1907.00390*.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. *arXiv preprint arXiv:1909.02188*.

Libo Qin, Wanxiang Che, Yangming Li, Mingheng Ni, and Ting Liu. 2020. Dcr-net: A deep co-interactive relation network for joint dialog act recognition and sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8665–8672.

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.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829*.

Lei Shu, Hu Xu, and Bing Liu. 2017. Doc: Deep open classification of text documents. *arXiv preprint arXiv:1709.08716*.

Shashank Srivastava, Igor Labutov, and Tom Mitchell. 2018. Zero-shot learning of classifiers from natural language quantification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 306–316.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S Yu. 2018. Zero-shot user intent detection via capsule neural networks. *arXiv preprint arXiv:1809.00385*.

Yang Yu, Wei-Yang Qu, Nan Li, and Zimin Guo. 2017. Open-category classification by adversarial sample generation. *arXiv preprint arXiv:1705.08722*.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S Yu. 2018. Joint slot filling and intent detection via capsule neural networks. *arXiv preprint arXiv:1812.09471*.

Hanlei Zhang, Hua Xu, and Ting-En Lin. 2020. Deep open intent classification with adaptive decision boundary. *arXiv preprint arXiv:2012.10209*.