Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/47/20241381

An open intent detection model optimized for datasets based on the Bert large model

Yichen Dong[1], Zhen Wang[2,4,5], Tianjun Wu[3]

[1]Department of Computer Science and Technology, Xidian University, Xian, China
[2]Department of Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing, China
[3]Department of Communication Engineering, Wuhan University of Technology, Wuhan, China

[4]b21041512@njupt.edu.cn
[5]corresponding author

Abstract. Within current task-oriented dialogue systems, the focus of intent detection predominantly centers on closed domains. Nevertheless, in real-world usage scenarios, a substantial proportion of interactions fall into the open-domain category. User intentions frequently transcend predefined boundaries, giving rise to a multitude of out-of-domain intents, which pose a formidable challenge to existing models, ultimately leading to diminished recognition rates and accuracy. The demand for open intent detection models is increasing in today's society to address this issue effectively. This paper proposes a method to optimize datasets, thereby enhancing the training accuracy of open intent detection models. Specifically, this paper employs the Adaptive Decision Boundary Learning algorithm, which is currently popular in open intent detection. Leveraging this algorithm, this paper suggests using the K-means clustering algorithm to refine the intent labels within the dataset. This process helps identify and remove outliers in the dataset, making the distinction between known domain and open-domain intent labels more precise. Experimental results on two datasets, banking77 and stackoverflow, demonstrate the effectiveness of our approach in significantly improving model accuracy.

Keywords: Open Intent Detection, K-means, Adaptive Decision Boundary Learning.

1. Introduction

In recent years, there has been significant progress in the field of open intent detection. Some researchers treat open intent detection as a classification problem, where known intents are divided into n classes, and the remaining open intents are categorized as one class, making it an (n+1) classification [1, 2]. The goal is to correctly categorize the known n classes of intent into (n+1) open intent patterns. To address this problem, one research team introduced the concept of open space risk as a criterion for classifying open intents [3]. Building on this, a method was proposed to reduce open space risk by learning the closed boundaries of each positive class in a similar space [4]. However, this approach does not capture high-level semantic concepts using Support Vector Machine (SVM). Using Deep Neural Networks (DNNs) to reduce open space risk also requires collecting open class labels to

DOI: 10.54254/2755-2721/47/20241381

adjust parameters [5]. Methods using outlier factors to distinguish known intents from open intents have also been proposed, but their drawbacks include relying on statistical data for thresholds and lacking specific decision boundaries to distinguish open intents [1, 2, 6].

Based on the current research on open intent detection, it is believed that directly using these methods for model training may not yield satisfactory results. This is because these model training methods have their own unique shortcomings as mentioned in the above paragraph, and they can not complement each other well, resulting in poor improvement in the results. This paper decide to start with the dataset and optimize the intent labels of the dataset to achieve better results.Analysis Therefore, a proposal is made to preprocess the training dataset by combining outlier factors and confidence thresholds. This preprocessing helps differentiate known classes from open classes in the dataset, making model training more efficient and accurate. This approach is suitable for datasets with fewer labels but more instances, as defining boundaries between known classes and open classes can become unclear when the dataset has fewer instances.

This paper selected an adaptive decision boundary open intent detection model that is currently relatively mature as the basis for our research [7]. This paper added a data preprocessing section to it, which includes clustering algorithms and confidence threshold preprocessing of the dataset. This paper compared the training results before and after adding this preprocessing step, using the banking dataset for comparison. According to the experimental data, this paper found that preprocessing the dataset indeed benefits training accuracy.

Our contributions are as follows: 1) This paper integrated the concepts of outlier factors and confidence thresholds into dataset preprocessing, reducing the pressure of distinguishing known classes from open classes during model training while improving training accuracy. 2) This paper introduced DataParallel into the existing model to accelerate training, optimizing training efficiency.

2. Method

2.1. Dataset introduction

This paper used two publicly available English datasets, namely banking77 and stackoverflow[8, 9]. The Banking77 dataset consists of online banking queries with corresponding intent annotations. It provides a fine-grained set of intents in the banking domain. The dataset contains 13,083 customer service queries, labeled with 77 intents. It focuses on fine-grained single-domain intent detection. The data structure consists of 'label' and 'text,' where 'text' is the text string and 'label' is the corresponding intent.The Stackoverflow dataset comprises comment data from the Stack Overflow website and consists of two parts: 'text' and 'label.' 'Text' represents the textual data, and 'label' corresponds to the specific question category, with 20 categories, such as whether it is related to Matlab or Apache.

2.2. Data Preprocessing

In addressing issues related to open intent models, this paper opted for an adaptive decision boundary open intent detection model as the foundation [7]. This paper employed the k-means clustering algorithm to identify outliers in the dataset. Our approach involved four iterations: first, training and testing on the raw dataset; then, testing on the dataset with outliers removed; next, training on the dataset with both outlier removal and clustering; finally, comparing and analyzing the four training results in terms of accuracy, F1-Score, and training time.

2.2.1. Introduction of k-means

The K-means algorithm operates with the 'k' representing the number of clusters, and 'means' indicating that the mean of each cluster's data points is taken as the centroid [10, 11].

The general algorithm proceeds as follows:

Step (1): Selection of the number of clusters 'k'

Randomly select 'k' samples from the dataset as cluster centers.

Step (2): Calculation of distances between data points and cluster centers

DOI: 10.54254/2755-2721/47/20241381

Assigning data points to the cluster that has the nearest cluster center involves using different methods depending on the data type. In Euclidean space, the Euclidean distance is used, while for text data, cosine similarity functions are employed. Occasionally, Manhattan distance may be used as a metric. The choice of the distance metric varies depending on the specific context. In this paper, author used the Euclidean distance for measurement.

$$d(x,y):=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+(x_3-y_3)^2+\ldots+(x_n-y_n)^2} \quad (4)$$

Step (3): Updating Cluster Centers

Based on the new cluster assignments, recalculate the distances to different clusters, reassign data points to their nearest clusters, update the cluster centers, and then continue iterating through the above operations until reaching the specified loop limit or until the centroids no longer change [12].

2.2.2. Outlier detection based on the K-means

For any intent dataset, each label for a sentence is determined by humans and carries subjective intent. Consequently, during the process of converting data into multi-dimensional vectors, it's quite common to encounter outliers, leading to redundancy. This paper employs K-means to handle the original dataset, removing outliers, thereby achieving the goal of improving the dataset and avoiding redundancy.

To be specific, create a boolean mask to select samples with target labels, extract embedding representations from the selected samples with target labels, and apply the K-means clustering algorithm. Subsequently, based on the cluster labels of each sample, identify the indices of the most outlier points. After the training is complete, remove these outlier points from the original data to obtain a higher-quality dataset.

2.3. Model

In this section, this paper employed the work of Zhang et al. from 2021 [7], which is an adaptive open intent detection boundary model. Initially, this paper pre-trained the model using labeled samples with known intent. Subsequently, this paper utilized well-trained features to learn an adaptive spherical decision boundary for each known class. This paper employed a specific loss function to balance empirical risk and open-space risk.

2.3.1. Bert

The Bert model is a type of pre-trained model. Bert stands for Bidirectional Encoder Representations from Transformers [13]. It directly incorporates the Encoder module from the Transformer architecture while omitting the Decoder module. This design gives it bidirectional encoding capabilities and strong feature extraction capabilities. Therefore, this paper use Bert to extract deep intent features.In the absence of open intent samples, known intents are employed as prior knowledge to pre-train the model. This paper uses softmax loss to learn intent features. Subsequently, this paper utilize the pre-trained model to extract intent features for learning decision boundaries.

2.3.2. Adaptive Decision Boundary Learning

Utilizing the known decision boundary formula and an optimized boundary learning strategy. Then, use the learned decision boundary for open classification. The decision boundary formula selects a spherical decision boundary [4], and the boundary learning strategy is based on the work of Hanlei Zhang from 2021[7]. With this strategy, under the boundary loss, the boundary can adapt to the intent feature space and learn an appropriate decision boundary.

The learned decision boundary is not only effective in enclosing the majority of known intent samples but also positioned far from the centroids of each known class. This enables effective identification of open intent samples.

After training, this paper uses the centroids of each known class and the learned decision boundary for inference. This paper assumes that known intent samples are constrained within their respective

DOI: 10.54254/2755-2721/47/20241381

centroids and the enclosed spherical regions created by the decision boundary. In contrast, open intent samples lie outside any bounded spherical region [7].

2.4. Implementation details

2.4.1. DataParallel accelerating

To speed up the training speed, this paper adopted DataParallel[14] for accelerating training. In the model training, this paper allels the training task to two 3090 GPUs for training. Each process maintains the same model parameters and the same computation task, but has different data to improve the training throughput.In order to evaluate the overall performance, this paper used accuracy and F1-score as evaluation indicators.

2.4.2. Evaluation index

In order to evaluate the overall performance, this paper used accuracy and F1-score as evaluation indicators. They are calculated on all classes (known and public). This paper also further established a macro F1-score evaluation standard for known and open classes, which can analyze whether the intention of a sentence belongs to the domain or outside the domain, and better evaluate the fine-grained performance of the model.

2.4.3. Training methods and parameters

Author selected Bert-base-uncased model for training. In order to speed up the training process and achieve better performance, this paper reference some work done by others and freeze all bert parameters except the last layer parameters.The training batch size is 128, the learning rate is 2e-5, and the formal training epochs are 100. The pre-trained Loss function is Cross Entropy Loss, and the formally trained loss function is BoundaryLoss. This paper employs Adam to optimize the boundary parameters at a learning rate of 0.05. In order to avoid randomness, this paper trained 10 times under the same conditions, and then took the average value of each index after 10 training as the evaluation standard.

3. Results and discussion

3.1. Processing of the Banking77 Dataset and Results

Table 1. Training results of ADB model without optimizing the training set

seed	Known	Open	F1-score	Accuracy
0	87.9397	90.0419	88.1308	89.10
1	84.8779	87.3362	85.1013	86.07
2	83.2983	86.8623	83.6223	85.03
3	84.9466	87.2926	85.1599	86.07
4	84.3120	87.5869	84.6097	85.98
5	85.3385	87.3016	85.5169	86.27
6	87.7046	89.4984	87.8677	88.65
7	85.7695	88.1049	85.9818	87.05
8	84.6745	86.8062	84.8683	85.75
9	85.4021	87.0411	85.5511	86.08
Avg	85.4264	87.7872	85.6409	86.6050

DOI: 10.54254/2755-2721/47/20241381

Table 2. Training results of ADB model with optimizing the training set

seed	Known	Open	F1-score	Accuracy
0	87.7361	89.8995	87.9328	88.95
1	84.7132	87.2456	84.9434	85.95
2	83.2335	86.5171	83.5322	84.78
3	84.9201	87.3362	85.1398	86.07
4	83.5394	86.7996	83.8358	85.17
5	85.4509	87.9762	85.6805	86.78
6	87.9536	89.8991	88.1305	89.02
7	85.7896	88.0874	85.9985	87.05
8	84.9706	87.2767	85.1802	86.17
9	85.6923	87.7034	85.8751	86.63
Avg	85.4000	87.8741	85.6249	86.6570

In the experiment, this paper only carried out optimized comparison tests for the training set. The test sets used were all the same data set, and the ratio of known classes to unknown classes was 0.5. As can be seen from the above Table 1 and Table 2, when the paper optimizes the training set using k-means algorithm, all indicators have a certain improvement. In the training without optimizing the training set, the average values of F1-score and accuracy are 80.7522 and 78.79, respectively. After the k-means algorithm is used to optimize the training set, the average values of F1-score and accuracy of training results are 81.3786 and 79.55, respectively. The macro F1-score of open class and known class also increased by about 1% respectively. Therefore, this method has certain positive effects, and the overall experimental data are relatively stable, general and reliable. At the same time, it can be seen that the macro F1-score of open class and known class both have an accuracy rate of more than 80%, which indicates that our model can perform intention detection well, and can analyze whether it belongs to the domain or outside the domain, and has certain openness, thus meeting the basic needs of open intention detection.

In the multidimensional vector, the vector-value corresponding to these sentences has a great deviation from the real clustering points. So training with a dataset that contains these data can lead to overfitting. After the k-means algorithm is optimized and outliers are deleted, the data intent label of the training set is more objective, reducing the possibility of overfitting, and helping to improve the accuracy of model training, which is consistent with the results obtained by our experiment.

3.2. Processing of the Stackoverflow Dataset and Results

Table 3. Training results of ADB model without optimizing the training set

seed	Known	Open	F1-score	Accuracy
0	87.9397	90.0419	88.1308	89.10
1	84.8779	87.3362	85.1013	86.07
2	83.2983	86.8623	83.6223	85.03
3	84.9466	87.2926	85.1599	86.07
4	84.3120	87.5869	84.6097	85.98
5	85.3385	87.3016	85.5169	86.27
6	87.7046	89.4984	87.8677	88.65

DOI: 10.54254/2755-2721/47/20241381

Table 3. (continued)

7	85.7695	88.1049	85.9818	87.05
8	84.6745	86.8062	84.8683	85.75
9	85.4021	87.0411	85.5511	86.08
Avg	85.4264	87.7872	85.6409	86.6050

Table 4. Training results of ADB model with optimizing the training set

seed	Known	Open	F1-score	Accuracy
0	87.7361	89.8995	87.9328	88.95
1	84.7132	87.2456	84.9434	85.95
2	83.2335	86.5171	83.5322	84.78
3	84.9201	87.3362	85.1398	86.07
4	83.5394	86.7996	83.8358	85.17
5	85.4509	87.9762	85.6805	86.78
6	87.9536	89.8991	88.1305	89.02
7	85.7896	88.0874	85.9985	87.05
8	84.9706	87.2767	85.1802	86.17
9	85.6923	87.7034	85.8751	86.63
Avg	85.4000	87.8741	85.6249	86.6570

In order to further find out whether deleting outliers is fully beneficial to the training of the model, this paper selected another dataset which is stackoverflow, and then used the k-means algorithm again to delete a larger number of outliers. Finally, the dataset after processing and the data set before processing were respectively trained. As shown in Table 3 and Table 4, it can be observed that when this paper deletes more outliers, the training effect does not increase significantly. This paper guesses the reason is that when deleting a part of outliers in the data set, there are fewer very outliers and more gathering points. Then, if this paper further find the most outlier points among these aggregation points for deletion, the training effect will deteriorate due to the reduction in the number of data sets. Therefore, this paper believe that the deletion of outlier points has certain limitations, and it is not possible to unconditionally delete a large number of outlier points, not the more the better, but at a certain equilibrium point is the best.

4. Conclusion

Following our extensive experimental analysis and comparative assessments, this paper has ascertained that optimizing the training dataset with the K-means algorithm results in notable enhancements in training outcomes. This optimization proves particularly efficacious when training large models for open intent detection, substantially improving training accuracy. Nevertheless, our experiments on Stack Overflow data revealed that increasing the removal of outliers did not yield a significant enhancement in training performance. This may be attributed to the adverse effects of excessive data removal, leading to a reduction in the dataset size and subsequently deteriorating training results.Nevertheless, from another perspective, employing the K-means algorithm to remove a substantial portion of the dataset can achieve similar model performance with a smaller dataset. This approach serves as an effective means to accelerate model training and reduce computational load.Moreover, when the amount of data removal is controlled to maintain a balance, it results in a training dataset with more objective intent labels, reducing the risk of overfitting. This can, to some extent, enhance the model's accuracy.

DOI: 10.54254/2755-2721/47/20241381

Authors Contribution

All the authors contributed equally, and their names were listed in alphabetical order.

References

[1] Shu L et al 2017 DOC: Deep Open Classifification of Text Documents. In Proceedings of EMNLP, 2911–2916.
[2] Lin T E 2019 Deep Unknown Intent Detection with Margin Loss. In Proceedings of ACL, 5491–5496
[3] Scheirer W J et al 2013 Toward Open Set Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 7(35): 1757–1772.
[4] Fei G and Liu B 2016 Breaking the Closed World Assumption in Text Classifification. In Proceedings of NAACLHLT, 506–514.
[5] Bendale A and Boult T E 2016 Towards open set deep networks In Proceedings of CVPR, 1563–1572.
[6] Breunig M M et al 2000 LOF: identifying density-based local outliers. In ACM sigmod record, volume 29, 93–104.
[7] Zhang H Xu H and Lin T 2020 Deep Open Intent Classification with Adaptive Decision Boundary. AAAI Conference on Artificial Intelli3gence.
[8] Casanueva I et al 2020 Efficient Intent Detection with Dual Sentence Encoders. In Proceedings of ACL WorkShop.
[9] Xu J et al 2015 Short Text Clustering via Convolutional Neural Networks. In Proceedings of NAACL-HLT, 62–69
[10] Ahmed M et al 2020 The k-means algorithm: A comprehensive survey and performance evaluation. Electronics 9(8) 1295
[11] Yu Q et al 2020 Clustering Analysis for Silent Telecom Customers Based on K-means++. In 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (Vol. 1, pp. 1023-1027). IEEE.
[12] MacQueen J 1967 Some Methods for classification and Analysis of Multivariate Observations
[13] Kenton J D M W C Toutanova L K 2019 Bert: Pre-training of deep bidirectional transformers forlanguage understanding Proceedings of naacL-HLT 1: 2
[14] Li S Zhao Y Varma R et al. 2006 Pytorch distributed: Experiences on accelerating data parallel training. arXiv preprint arXiv:2006.15704, 2020.