Dialog Intent Induction via Density-based Deep Clustering Ensemble

Jiashu Pu*, **Xiaoxi Mao, Guandan Chen, Yongzhu Chang**
Fuxi AI Lab, NetEase Inc., Hangzhou, China
{pujiashu,maoxiaoxi,chenguandan,changyongzhu}@corp.netease.com

Abstract

Existing task-oriented chatbots heavily rely on spoken language understanding (SLU) systems to determine a user's utterance's intent and other key information for fulfilling specific tasks. In real-life applications, it is crucial to occasionally induce novel dialog intents from the conversation logs to improve the user experience. In this paper, we propose the **D**ensity-based **D**eep **C**lustering **E**nsemble (DDCE) method for dialog intent induction. Compared to existing K-means based methods, our proposed method is more effective in dealing with real-life scenarios where a large number of outliers exist. To maximize data utilization, we jointly optimize texts' representations and the hyperparameters of the clustering algorithm. In addition, we design an outlier-aware clustering ensemble framework to handle the overfitting issue. Experimental results over seven datasets show that our proposed method significantly outperforms other state-of-the-art baselines.

Introduction

In recent years, applications built with task-oriented chatbots have become ubiquitous in many fields (Perkins and Yang 2019). Despite the considerable success that massive pre-training has achieved over open-domain response generation, task-oriented chatbots still heavily rely on SLU systems to convert a user's utterance to a specified dialog intent and corresponding slots information. In general, the SLU system is trained on a handful of examples corresponding to several pre-determined dialog intents before deployment. However, in real-life scenarios, it's common that dialog intents designed by developers may not cover actual users' utterances. User demands may shift with time or are simply not considered in advance. To perfect the user experience, developers need to induce novel dialog intents from the conversation logs generated by users.

In practice, due to the enormous size of conversation logs, developers typically perform a clustering analysis over the logs to find clusters with a large number of similar user utterances and then mark them as novel dialog intents. Because of the practicability of this task, there have been multiple works proposed by researchers. However, these works are mostly based on K-means algorithms (Hadifar et al. 2019; Perkins and Yang 2019; Lin, Xu, and Zhang 2020; Wang, Mi, and Ittycheriah 2016). In practice, these K-means based methods have two limitations. First, the hyperparameter K is challenging to determine before clustering. Choosing a proper K takes a lot of trial and error. Second, there are a large number of outliers in real-world conversation logs. These outliers are occasional, irrelevant user utterances that should not be mapped to any dialog intent. They cannot be effectively excluded by K-means based methods and take much human labor to clean. To demonstrate, we present an example in Figure 2.

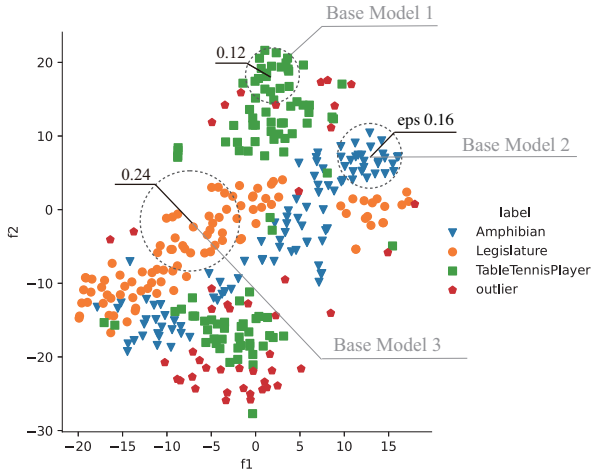

Figure 1: We apply t-SNE on embeddings of a small fraction of samples from DBpedia (Auer et al. 2007). The embeddings are computed with BERT-base (Devlin et al. 2019) without finetuning. It is difficult to distinguish labels from the density of aggregated samples, and the involvement of outliers makes the task even more difficult. Besides, the figure illustrates a clustering model with a specific set hyperparameters is effective on only a part of the data (eps refers to the radius of neighborhood defined in DBSCAN (Ester et al. 1996)).

In this paper, we propose a method called **D**ensity based **D**eep **C**lustering **E**nsemble(DDCE). In this method, we adopt a density-based clustering algorithm OP-

*corresponding author

TICS (Ankerst et al. 1999), to avoid the limitations mentioned above. In addition, it's widely known that general representations obtained with pre-trained language models are not sufficient to support effective clustering (Li et al. 2020; Reimers et al. 2019), as such representations sometimes may fail to distinguish between specific semantics (e.g. negation), which is illustrated in Figure 2 and Figure 1; besides, in practice, we also notice that a specific set of hyperparameters of a clustering algorithm may not be effective across the whole data.

Due to this, we propose to apply a clustering ensemble framework that combines multiple base clustering models with corresponding text encoders and hyperparameters. Cluster ensemble has been proven to enhance the robustness and thus improve clustering quality (Strehl and Ghosh 2002). Although the concept of ensemble learning has been well established for tasks such as classification and regression (Strehl and Ghosh 2002; Bauer and Kohavi 1999), there is very little work that applies it to the clustering problem of novel intent induction. Our framework that combines representation learning and clustering ensemble fills this void. We conduct detailed experiments on seven datasets to verify its effectiveness. The experimental results show that our method significantly outperforms other state-of-the-art baselines.

Related works

Dialog Intent Induction: Several recent works propose to exploit the information in labeled data by combining deep models with specific loss functions or training paradigms. Perkins et al. (Perkins and Yang 2019) propose a method exploiting multi-view data to learn representation and cluster jointly. The representations are updated iteratively using the K-means cluster assignments from the alternative view. Two other works use the self-training approach, and both incorporate K-means into the training process. The first (Wang, Mi, and Ittycheriah 2016) adopted the DEC method (Xie, Girshick, and Farhadi 2016), and the second (Hadifar et al. 2019) designed a loss function that combines clustering and classification. Different from previous works, Lin et al. (Lin, Xu, and Zhang 2020) suggested learning a model that predicts pair-wise similarities from labeled data. The model transforms label and unlabeled data into pair-wise constraints served as the surrogate for clustering. A finetuning stage follows, and clustering is optimized based on KLD loss (Xie, Girshick, and Farhadi 2016) and a process of eliminating low confidence similarity pairs is executed subsequently. However, none of the previous works considers the existence of outliers, which may make it difficult to estimate the hyperparameter K of K-means. Though there exists an ITER-DBSCAN algorithm (Chatterjee and Sengupta 2020) that considers the outliers and claims to handle the class imbalance issue well, it does not unify representation learning and clustering algorithms under one framework. Our work remedies the above-mentioned deficiencies.

Clustering Ensemble: Many successful applications of clustering ensemble exist in fields like image processing, cheminformatics, etc. (Boongoen and Iam-On 2018). However, applying the clustering ensemble in dialog intent induction is rare; the most relevant work is (Fraj, Hajkacem, and Essoussi 2019). It trains base clustering models with multi-views of text representations rather than different splits of data.

[1]https://qnm.163.com

[2]https://huggingface.co/bert-base-chinese

Density-based Deep Clustering Ensemble

Task Formulation

The task is to induce novel dialog intents from unlabeled user utterances with some labeled examples in the same domain. We define the unlabeled user utterances as $D_{ul} = \{u_i, i = 1, ..., M\}$ and the labeled examples as $D_l = \{(x_j, y_j), j = 1, ..., N\}$, where u_i denotes a user utterance, x_j denotes the example and $y_j \in Y$ denotes its label. Y is the collection of predefined dialog intents. M is the size of the unlabeled user utterances to be processed, and N is the size of the labeled examples. Given the unlabeled user utterances D_{ul} and labeled examples D_l, our goal is to find a set of clusters from $T_{ul} = \{t_i, i = 1, ..., M\}$, where t_i denotes the cluster label. Clusters with a size smaller than S are regarded as outliers and ignored. Other clusters will be further processed, merged with existing dialog intent examples, or summarized as a novel dialog intent.

Method Description

The DDCE method consists of two steps. First, we train several base clustering models over labeled examples D_l. The training consists of two aspects: finetuning the text encoder and searching for the best hyperparameters. Specifically, we split, the D_l into D_l^{rl} and D_l^{hs}. D_l^{rl} is the training set for finetuning the text encoder and D_l^{hs} is the validation set for searching the best hyperparameters, with outliers injected. To better generalize to unseen dialog intents, we intentionally make the intents corresponding to instances in D_l^{rl} and D_l^{hs} not overlap each other at splitting, i.e., for $\forall(x_i, y_i) \in D_l^{rl}$ and $\forall(x_j, y_j) \in D_l^{hs}$, $y_i \neq y_j$. To achieve this effect, we split D_l by the dialog intents. Given split ratio α, the collection of predefined intents Y, assume the size of Y is O, we let the examples corresponding to αO intents enter the validation set D_l^{hs}, the rest of examples corresponding to other $(1 - \alpha)O$ intents enter the training set D_l^{rl}. There are many ways to split D_l, from which we randomly choose K. For each k-th split, we finetune the text encoder on D_l^{rl} with a classification task and the cross-entropy objective function. We split another validation set from D_l^{rl} to select the best model according to the classification performance. Given the best text encoder, we conduct a random search of hyperparameters on D_l^{hs} and choose the best one with the highest $score$, a metric later described in the evaluation setting. At last, we obtain K base clustering models, respectively corresponding to K text encoders and K groups of hyperparameters. We shall calculate the performance score $score_c$ of the K base clustering models on their respective corresponding validation sets D_l^{hs} for the following ensemble. The details of calculating $score_c$ are covered in Evaluation Metric Section.

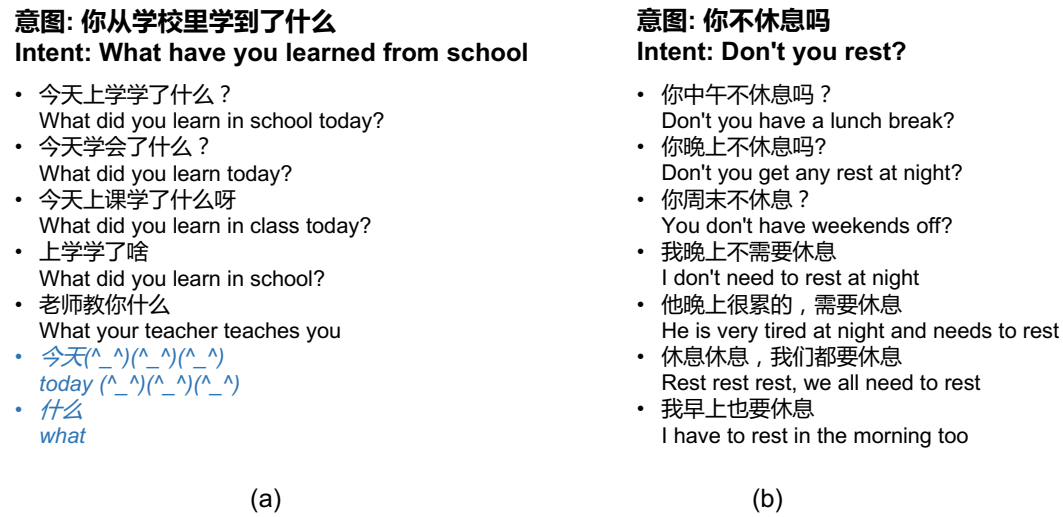

Figure 2: Mining intents from dialogue logs is an efficient way to update a task-oriented robot's intent database. We show two typical examples of real intent clusters induced from conversation logs of an MMORPG game–*Ghost Story*[1]. These two examples are clustered by the K-means algorithm, and sentence embeddings are extracted by a pre-trained bert-base-chinese[2] model. Subfigure (a) presents an intent cluster contaminated by two outliers, which are in blue and italics. Subfigure (b) shows non-finetuned Bert embeddings can ensure the semantics of the sentences within a cluster are roughly similar but may fail to distinguish some subtle differences, such as negation, personal pronouns, and descriptions of time.

Second, we use K base clustering models to do clustering over D_{ul}, obtaining K groups of cluster labels, of which the kth labels can be noted as T_{ul}^k. Finally, we apply a consensus function over the results $(T_{ul}^1, ..., T_{ul}^K; score_c^1, ..., score_c^K)$ to obtain final clustering labels T_{ul}. The complete process of the algorithm is summarized in Algorithm 1.

Consensus Function Combining the clustering results of base models is non-trivial because of the label correspondence problem (Strehl and Ghosh 2002). We introduce three consensus functions here and test BOKV and CHM in experiments. We denote $NMI(\cdot,\cdot)$ as the normalized mutual information, and $\mathbb{T}_K = \{T_{ul}^1, ..., T_{ul}^K\}$ as the set consisting all base models' partitions.

CSPA/HGPA/MCLA (CHM) (Strehl and Ghosh 2002): CHM uses three partition methods to generate different cluster labels. The optimal labels T_{ul}^* for CHM is defined as

$$T_{ul}^* = \arg\max_{T \in \mathbb{T}_{chm}} \sum_{j=1} \mathrm{NMI}\left(T, T_j\right) \tag{1}$$

, where $\mathbb{T}_{chm} = \{T_{CSPA}, T_{HGPA}, T_{MCLA}\}$. Abbreviations **CSPA**, **HGPA** and **MCLA** denote Cluster-based Similarity Partitioning, HyperGraphs Partitioning, and Meta-CLustering Algorithm respectively (Strehl and Ghosh 2002).

Best Of K (BOK) (Vega-Pons and Ruiz-Shulcloper 2011): In BOK, the optimal labels T_{ul}^* is defined as

$$T_{ul}^* = \arg\max_{T \in \mathbb{T}_K} \sum_{j=1}^{K} \mathrm{NMI}\left(T, T_j\right) \tag{2}$$

Best Of K with outlier Voting (BOKV, ours): As all base models share the same label for outliers, to boost performance, we choose to aggregate outliers' predictions via simple voting (Bauer and Kohavi 1999). Because the low performances of base models may have a negative impact on the ensemble result (Wang 2008), only when more than half of base models' non-outliers recall scores on validation set are higher than 0.5 do we vote to predict outliers, otherwise, BOKV is degraded to BOK. When BOKV is adopted, for each sample t_i, we first obtain the prediction u_i by majority voting, to decide whether it is outlier or non-outlier,

$$u_i = \begin{cases} 1 & \text{if } \mathrm{argmax}_{t_j} \left\{ \sum_{j=1}^{K} t_k \right\} = l^{out} \\ 0 & \text{if } \mathrm{argmax}_{t_j} \left\{ \sum_{j=1}^{K} t_k \right\} \neq l^{out} \end{cases} \tag{3}$$

, where l^{out} is the label of outlier. We denote $I_{out} = \{i \,|\, u_i = 1\}$ and $I_{nout} = \{i \,|\, u_i = 0\}$ as index sets of outlier and non-outlier respectively. While the ensemble labels of outliers $T_{ul}^{out} = \{t_i, i \in I_{out}\}$ are determined by voting, the ensemble labels of non-outliers T_{ul}^{nout} still need to be determined by BOK,

$$T_{ul}^{nout} = \arg\max_{T^{nout} \in \mathbb{T}_K^{nout}} \sum_{j=1}^{K} \mathrm{NMI}\left(T^{nout}, T_j^{nout}\right) \tag{4}$$

where $T^{nout} = \{t_i, i \in I_{nout}\}$ denotes the partition of a base learner with non-outliers. At last, we obtain the optimal label T_{ul}^* by combining T_{ul}^{out} and T_{ul}^{nout}.

Algorithm 1: DDCE

Input: $D_l = \{(x_j, y_j), j = 1, ..., N\}$, $D_{ul} = \{u_i, i = 1, ..., M\}$;
Ouput: $T_{ul} = \{t_i, i = 1, ..., M\}$;
Require: The number of base cluster models K, the collection of predefined intents Y, the size of Y is O, the split ratio α, hyperparameter search space hp;
Training:
for $k = 1, ..., K$ **do**
  split D_l into D_l^{rl} which contains examples corresponding to $(1 - \alpha)O$ intents, and D_l^{hs} which contains examples corresponding to αO intents
  Initialize text encoder f_θ^k with pre-trained weights
  Update θ^k after trained on D_l^{rl}
  Compute embeddings E_l^{hs} of D_l^{hs} using f_θ^k
  Search the best hyperparameters hp^k on E_l^{hs}
  Calculate $score_c^k$ on D_l^{hs} with hp^k
end
Inference:
for $k = 1, ..., K$ **do**
  Compute embeddings E_{ul} of D_{ul} using f_θ^k
  Do clustering over E_{ul} with hyperparameters hp^k
end
Apply a consensus function (e.g. BOKV) on $(T_{ul}^1, ..., T_{ul}^K; score_c^1, ..., score_c^K)$ to obtain T_{ul}

Experiments

Datasets and Preprocessing

We present the statistics for datasets used in the experiments in Table 1, including three English datasets and three Chinese datasets. CLINC150 (Larson et al. 2019) is an intent classification dataset with 150 in-domain intent classes. DBpedia (Auer et al. 2007) is an active project dealing with structured data and Wikipedia. IWSDS (Xingkun Liu and Rieser 2019) is a multi-domain SLU benchmarking dataset built from 25K user utterances. THUCNews is a news classification dataset[3]. SMP2019 is the dataset of SMP2019 ECDT Task1[4]. AgentDialog and HumanDialog are user utterances respectively extracted from agent-human and human-human conversation logs of an MMORPG game– *Ghost Story*[5], where agents are intelligent kids. To make the experiments closer to the real-life scenario, we inject outliers into the test sets. The outliers are samples from other datasets. For example, we may randomly pick one sample per intent from DBpedia and add them to the test set of CLINC150 as outliers. Ratios of injected outliers in experiments are presented in Table 2.

[3]http://thuctc.thunlp.org

[4]https://conference.cipsc.org.cn/smp2019/evaluation.html

[5]https://qnm.163.com

Dataset	Class	Text	Len	U.Token
CLINC150 (EN)	150	150	40	6391
DBpedia (EN)	219	1566	121	418737
IWSDS (EN)	68	377	35	10585
THUCNews (CN)	14	59720	20	266060
SMP2019 (CN)	23	156	9	3482
AgentDialog (CN)	354	21	7	2883
HumanDialog (CN)	1226	15	6	2790

Table 1: The columns from left to right show the number of classes, the average number of text in each class, the average length of the text, and the total number of unique tokens.

Evaluation Metrics

To measure the performance of novel intents detection, we considered two metrics: $score_c$ to measure the recall of non-outlier samples, following Lin etc. (Lin, Xu, and Zhang 2020), we adopt the Adjusted Rand Index score (ARI) (Yeung and Ruzzo 2001) to measure the clustering quality of T_{ul}, denoted as $score_{ari}$. The final $score$ is defined as the harmonic mean of $score_c$ and $score_{ari}$. We use the harmonic mean because we believe that the ability to detect outliers and the clustering quality are equally important. We hope as many reasonable clusters to be found as possible, at the same time, the quality of which is good enough to be readily merged with old intents or form as new ones with less or no post-processing. In practice, we found numerous outliers during the process of inducing new intents from human-human conversation logs in online games, thus we believe the new metric is more in line with the real world.

Baselines

We include unsupervised clustering algorithms — K-means (Steinley 2006), Hierarchical Clustering using Ward Linkage (Murtagh and Legendre 2011) and OPTICS (Ankerst et al. 1999). We also compare with the highly influential DEC (Xie, Girshick, and Farhadi 2016), a method that proposes an iterative refinement via soft assignment. For semi-supervised clustering, we compare with SOTA works including BERT-MCL (Hsu et al. 2018) and CDAC+ (Lin, Xu, and Zhang 2020). We use BERT-base as the text encoder in every method to ensure a fair comparison. All text encoders are in-domain fine-tuned except for the OPTICS baseline.

Experimental Settings

The experiment is repeated three times for each dataset, with different splits of D_l and D_{ul}. D_{ul} includes examples corresponding to around 15% of total dialog intents before outliers injection.

To approximate the situation in real applications where there are only a small number of labeled examples per intent, the maximum size of samples per intent is set to 50.

We set the number of base clustering models K to 5 and split ratio α to 0.5 throughout all experiments. For both languages, we chose BERT-base as our text encoder because

	CLINC150	DBpedia	IWSDS	THUCNews	SMP2019	AgentDialog	HumanDialog
Outlier ratio	0.547	0.279	1.143	2.0	2.0	1.144	0.156
K-means	0.208±0.031	0.144±0.009	0.128±0.004	0.458±0.083	0.365±0.064	0.305±0.014	0.605±0.012
Hierarchical	0.248±0.043	0.169±0.016	0.146±0.021	0.504±0.027	0.521±0.126	0.356±0.018	0.622±0.003
DEC	0.102±0.034	0.089±0.011	0.125±0.045	**0.533±0.135**	0.427±0.042	0.288±0.054	0.683±0.016
BERT-MCL	0.307±0.237	0.025±0.030	0.036±0.033	0.046±0.079	0.231±0.204	0.200±0.104	0.015±0.017
CDAC+	0.203±0.002	0.287±0.044	0.119±0.044	0.007±0.013	0.118±0.046	0.360±0.042	0.585±0.011
OPTICS	0.247±0.079	0.185±0.063	0.199±0.111	0.375±0.176	0.487±0.195	0.348±0.122	0.759±0.058
DDEC-CHM (ours)	**0.563±0.046**	0.315±0.035	**0.517±0.115**	0.263±0.241	0.408±0.274	0.515±0.059	0.744±0.044
DDEC-BOKV-BM (ours)	0.525±0.008	0.300±0.023	0.411±0.017	0.469±0.077	0.700±0.067	0.605±0.015	0.853±0.011
DDEC-BOKV (ours)	0.557±0.057	**0.363±0.021**	0.506±0.058	0.504±0.140	**0.760±0.080**	**0.641±0.010**	**0.855±0.015**

Table 2: The values in the table correspond to the $score$ described in the Evaluation Metrics Section. The *outlier ratio* is the ratio of the size of injected outliers to the original size of D_{ul}. **BOKV-BM** refers to the average performance of K BOKV's base models.

BERT is the basis for most SOTA text encoders[6]. We set the batch size to 32 and the learning rate to $5e^{-5}$. The embedding extracted for clustering is the average pooling of the second last layer of BERT-base.

Regarding the hyperparameter searching phase, we chose OPTICS (Ankerst et al. 1999) as the density-based clustering algorithm; compared to the popular algorithm DBSCAN, it has the advantage of finding clusters with varying densities. We conducted random searches (Bergstra and Bengio 2012) on three hyperparameters of OPTICS: $max\ eps$, xi, and $min\ sample$. The interval of $min\ sample$ is set to $(2, 20)$, and the intervals of $max\ eps$, xi are set to $(0.0, 0.5)$ respectively. We repeat the search process a hundred times per trial.

For K-means based methods, which require setting the number of clusters K_c, we first estimate the average size of examples per intent in the labeled data D_l and infer the K_c for test set D_{ul} accordingly. To boost baseline methods' performance, we increase the K_c by a factor of 4 for a rough estimate of outliers. We leave any example in clusters with a size smaller than 2 as outliers.

Results

The results on seven datasets are shown in Table 2. We tested our method **DDEC** with two consensus functions: CHM and BOKV. Concerning performance, DDEC-CHM and DDEC-BOKV outperform all other methods on all datasets except for the THUCNews dataset, where DEC ranks first. However, we find DEC quite sensitive to the choice of degree of freedom; in Table 2, the reported $scores$ of DEC are the best ones chosen from trails of different hyperparameters. From the above observations, we conclude that K-means related methods are more susceptible to inappropriate hyperparameters while the **DDEC** is more robust and ranks at the top for all datasets.

The overall best consensus function is BOKV, though it slightly underperforms CHM on CLINC150 and DBpedia. The ensemble method DDEC-BOKV exhibits a consistent advantage over its base models in all datasets, demonstrating better generalization in discovering new intents.

[6]https://gluebenchmark.com/leaderboard

The best split ratio α: We test how the choice of α impacts the performance of a base clustering model. For each α value, we repeat the same experimental procedure described in the Experimental Section. For both English and Chinese datasets, scores are averaged over datasets. In Figure 3, we show how the mean and variance of the scores are influenced by α. We can conclude that setting α at 0.5 achieve an optimal balance regardless of the language of the dataset.

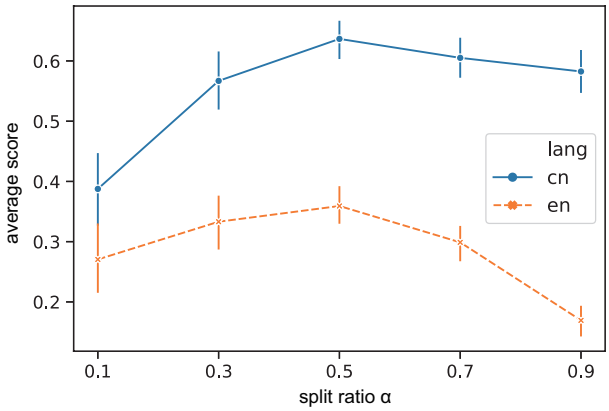

Figure 3: The impact of split ratio α for a base clustering model.

Effects of different ratios of outliers: Trends in Figure 4 show that the score of DDEC-BOKV decreases much more slowly than the average score of base clustering models as the outlier ratio increases, indicating that BOKV introduces more robustness against the number of outliers.

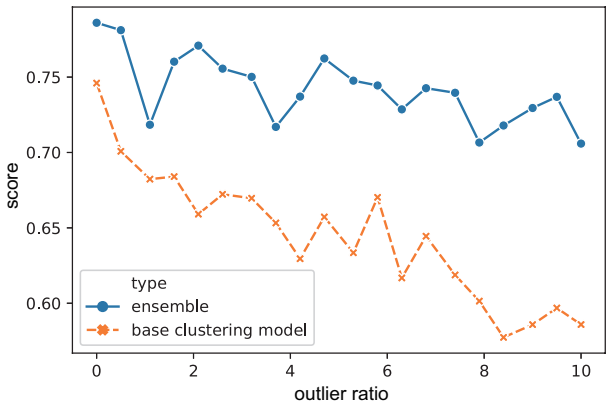

Figure 4: We present the effect of outliers ratios on the results. We choose SMP2019 as the experimental dataset to test a larger order of magnitude of outlier ratios.

Effects of different sizes of training data: We conduct experiments to analyze if DDEC-BOKV is sensitive to the labeled data's size D_l. Table 3 proves convincingly that the clustering ensemble consistently outperforms base clustering models, except the difference is only significant at $p = 0.07$ when O is minimal. This experiment demonstrates that BOKV significantly outperforms the base model even when the data volume is very small.

O	4	8	16	32	64	128
	10.4%	15.2%†	9.9%†	10.0%†	4.2%†	6.3%†

Table 3: We report the relative $score$ improvement of DDEC-BOKV over DDEC-BOKV-BM when training size varies. O denotes the size of the collection of predefined intents Y in D_l. Significant tests are performed (Woolson 2007) and † indicates $p < 0.05$.

Conclusion

In practice, we find that clusters mined in conversation logs by K-means based clustering algorithms often contain many outliers, partly because of the characteristics of the data itself, and partly because K-mean based clustering algorithms alone cannot handle outliers properly. To compensate for the shortcomings of the K-means based methods, we propose a deep clustering ensemble method as well as a new outlier-aware metric for the dialog intent induction task. Our approach encourages base models to learn from different parts of the labeled data. We maximize the use of data through finetuning a text encoder and searching a proper set of hyperparameters for OPTICS simultaneously. To avoid overfitting, separate clustering results are integrated via a novel consensus function BOKV. Our method is proved effective in extensive experiments, even if the size of labeled data is extremely small or the unlabeled data contains a large number of outliers.

References

Ankerst, M.; Breunig, M. M.; Kriegel, H.-P.; and Sander, J. 1999. OPTICS: ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2): 49–60.

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, 722–735. Springer.

Bauer, E.; and Kohavi, R. 1999. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1): 105–139.

Bergstra, J.; and Bengio, Y. 2012. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2).

Boongoen, T.; and Iam-On, N. 2018. Cluster ensembles: A survey of approaches with recent extensions and applications. *Computer Science Review*, 28: 1–25.

Chatterjee, A.; and Sengupta, S. 2020. Intent Mining from past conversations for Conversational Agent. In *Proceedings of the 28th International Conference on Computational Linguistics*, 4140–4152.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 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)*, 4171–4186.

Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X.; et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, 226–231.

Fraj, M.; Hajkacem, M. A. B.; and Essoussi, N. 2019. Ensemble method for multi-view text clustering. In *International Conference on Computational Collective Intelligence*, 219–231. Springer.

Hadifar, A.; Sterckx, L.; Demeester, T.; and Develder, C. 2019. A self-training approach for short text clustering. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, 194–199.

Hsu, Y.-C.; Lv, Z.; Schlosser, J.; Odom, P.; and Kira, Z. 2018. Multi-class classification without multi-class labels. In *International Conference on Learning Representations*.

Larson, S.; Mahendran, A.; Peper, J. J.; Clarke, C.; Lee, A.; Hill, P.; Kummerfeld, J. K.; Leach, K.; Laurenzano, M. A.; Tang, L.; and Mars, J. 2019. An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. 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)*, 1311–1316. Hong Kong, China: Association for Computational Linguistics.

Li, B.; Zhou, H.; He, J.; Wang, M.; Yang, Y.; and Li, L. 2020. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*.

Lin, T.-E.; Xu, H.; and Zhang, H. 2020. Discovering New Intents via Constrained Deep Adaptive Clustering with Cluster Refinement. In *AAAI*, 8360–8367.

Murtagh, F.; and Legendre, P. 2011. Ward's hierarchical clustering method: Clustering criterion and agglomerative algorithm. *arXiv preprint arXiv:1111.6285*.

Perkins, H.; and Yang, Y. 2019. Dialog Intent Induction with Deep Multi-View Clustering. 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)*, 4007–4016.

Reimers, N.; Gurevych, I.; Reimers, N.; Gurevych, I.; Thakur, N.; Reimers, N.; Daxenberger, J.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Steinley, D. 2006. K-means clustering: a half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, 59(1): 1–34.

Strehl, A.; and Ghosh, J. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec): 583–617.

Vega-Pons, S.; and Ruiz-Shulcloper, J. 2011. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(03): 337–372.

Wang, W. 2008. Some fundamental issues in ensemble methods. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2243–2250. IEEE.

Wang, Z.; Mi, H.; and Ittycheriah, A. 2016. Semi-supervised Clustering for Short Text via Deep Representation Learning. *CoNLL 2016*, 31.

Woolson, R. 2007. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials*, 1–3.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, 478–487.

Xingkun Liu, P. S., Arash Eshghi; and Rieser, V. 2019. Benchmarking Natural Language Understanding Services for building Conversational Agents. In *Proceedings of the Tenth International Workshop on Spoken Dialogue Systems Technology (IWSDS)*, xxx–xxx. Ortigia, Siracusa (SR), Italy: Springer.

Yeung, K. Y.; and Ruzzo, W. L. 2001. Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9): 763–774.