Discovering New Intents Using Latent Variables

Yunhua Zhou, Peiju Liu, Yuxin Wang, Xipeng Qiu*

School of Computer Science, Fudan University
{zhouyh20,xpqiu}@fudan.edu.cn
{pjliu21, wangyuxin21}@m.fudan.edu.cn

Abstract

Discovering new intents is of great significance to establishing Bootstrapped Task-Oriented Dialogue System. Most existing methods either lack the ability to transfer prior knowledge in the known intent data or fall into the dilemma of forgetting prior knowledge in the follow-up. More importantly, these methods do not deeply explore the intrinsic structure of unlabeled data, so they can not seek out the characteristics that make an intent in general. In this paper, starting from the intuition that discovering intents could be beneficial to the identification of the known intents, we propose a probabilistic framework for discovering intents where intent assignments are treated as latent variables. We adopt Expectation Maximization framework for optimization. Specifically, In E-step, we conduct discovering intents and explore the intrinsic structure of unlabeled data by the posterior of intent assignments. In M-step, we alleviate the forgetting of prior knowledge transferred from known intents by optimizing the discrimination of labeled data. Extensive experiments conducted in three challenging real-world datasets demonstrate our method can achieve substantial improvements.

Introduction

Unknown intent detection (Zhou, Liu, and Qiu 2022) in Bootstrapped Task-Oriented Dialogue System (BTODS) has gradually attracted more and more attention from researchers. However, *detecting* unknown intent is only the first step. For BTODS, *discovering* new intents is not only the same basic but also more crucial and challenging. Because the preset intent set in BTODS is limited to cover all intents, BTODS should discover potential new intents actively during interacting with the users. Specifically, a large number of valuable unlabeled data will be generated within the interaction between users and the dialogue system. Considering the limited labeled corpus and time-consuming annotating, which also requires prior domain knowledge, the BTODS should adaptively identify known intents and discover unknown intents from those unlabeled data with the aid of limited labeled data.

Just as discovering new intents plays a crucial role in establishing BTODS, discovering new intents has raised a lot of research interest as unknown intent detection. Unsupervised cluster learning is one popular method to solve this problem. To discover new intents from a large number of unlabeled data, many works (Hakkani-Tür et al. 2013, 2015; Shi et al. 2018; Padmasundari 2018) formalize this problem as an unsupervised clustering process. However, these methods mainly focus on how to construct pseudo-supervised signals to assist in guiding the clustering process and do not fully utilize the prior knowledge contained in the existing labeled data.

In a more general real scenario, we often have a small (but containing prior knowledge that can be used to guide the discovery of new intents) amount of labeled data in advance and a large amount of unlabeled data (e.g., in the dialogue scene mentioned above, it is generated in the interaction with the dialogue system), which contains both known intents and unknown intents to be discovered.Our purpose is to identify the known intents and discover the potential intents contained in the unlabeled corpus using labeled data.

Recently, Lin, Xu, and Zhang (2020) propose that pairwise similarities can be used as pseudo supervision signals to guide the discovery of new intents. However, as in the analysis of Zhang et al. (2021), this method can not achieve effective performance when there are more new intents to be discovered. Inspired by Caron et al. (2018), Zhang et al. (2021) (DeepAligned) propose an effective method for discovering new intents. DeepAligned first finetunes the BERT (Devlin et al. 2018) using the labeled data to transfer the prior knowledge and generalize the knowledge into the semantic features of unlabeled data. Further, to learn friendly representations for clustering, DeepAligned assigns a pseudo label to each unlabeled utterance and re-trains the model under the supervision of the softmax which is calculated by those pseudo labels.

Nevertheless, DeepAligned may suffer from two critical problems. Firstly, when the model is re-trained with pseudo supervision signal, the model will forget the knowledge transferred in the transferring stage, the forgetting curves on different datasets as shown in Figure 1. During discovering intents in DeepAligned, we test the performance of the model on the validation set used in the transferring prior knowledge stage and show that with the advancement of clustering, the model constantly forgets the knowledge learned from labeled data. Furthermore, the model could be misled by inaccurate pseudo labels, particularly in large-sized intent space (Wang et al. 2021). More importantly,

* Corresponding author.

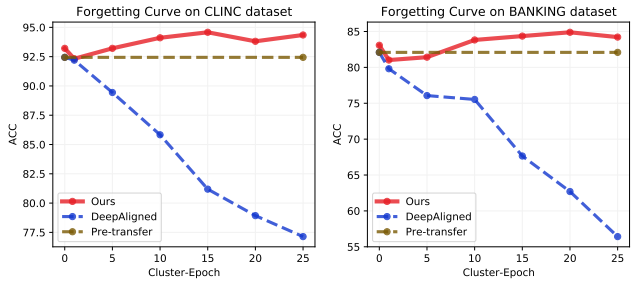

Figure 1: The forgetting curves of DeepAligned (Blue). During discovering intents in DeepAligned, the model constantly forgets the knowledge learned from labeled data. The brown line represents the baseline obtained by the model after transferring prior knowledge. In contrast, our method (Red) can alleviate forgetting well and See subsequent section for more discuss.

softmax loss formed by pseudo labels cannot explore the intrinsic structure of unlabeled data, so it can not provide accurate clustering supervised signals for discovering intents.

Different from the previous methods, we start from the essential intuition that the discovery of intents should not damage the identification of the known intents and the two processes should achieve a win-win situation. The knowledge contained in labeled data corpus can be used to guide the discovery of the new intents, and the information learned from the unlabeled corpus (in the process of discovering) could improve the identification of the known intents.

Based on this intuition, with the help of optimizing identification of labeled data given the whole data corpus, we propose a principled probabilistic framework for intents discovery, where intent assignments as a latent variable. Expectation maximization provides a principal template for learning this typical latent variable model. Specifically, in the E-step, we use the current model to discover intents and calculate a specified posterior probability of intent assignments, which is to explore the intrinsic structure of data. In the M-step, maximize the probability of identification of labeled data (which is to mitigate catastrophic forgetting) and the posterior probability of intent assignments (which is to help learn friendly features for discovering new intents) simultaneously to optimize and update model parameters. Extensive experiments conducted in three benchmark datasets demonstrate our method can achieve substantial improvements over strong baselines. We summarize our contributions as follows:

(Theory) We introduce a principled probabilistic framework for discovering intents and provide a learning algorithm based on Expectation Maximization. To the best of our knowledge, this is the first complete theoretical framework in this field and we hope it can inspire follow-up research.

(Methodology) We provide an efficient implementation based on the proposed probabilistic framework. After transferring prior knowledge, we use a simple and effective method to alleviate the forgetting. Furthermore, we use the contrastive learning paradigm to explore the intrinsic structure of unlabeled data, which not only avoids misleading the model caused by relying on pseudo labels but also helps to better learn the features that are friendly to intent discovery.

(Experiments and Analysis) We conduct extensive experiments on a suite of real-world datasets and establish substantial improvements.

Related Work

Our work is mainly related to two lines of research: Unsupervised and Semi-supervised clustering.

Unsupervised Clustering Extracting meaningful information from unlabeled data has been studied for a long time. Traditional approaches like **K-means** (MacQueen et al. 1967) and Agglomerative Clustering **(AC)** (Gowda and Krishna 1978) are seminal but hardly perform well in high-dimensional space. Recent efforts are devoted to using the deep neural network to obtain good clustering representations. Xie, Girshick, and Farhadi (2016) propose Deep Embedded Cluster **(DEC)** to learn and refine the features iteratively by optimizing a clustering objective based on an auxiliary distribution. Unlike DEC, Yang et al. (2017) propose Deep Clustering Network **(DCN)** that performs nonlinear dimensionality reduction and k-means clustering jointly to learn friendly representation. Chang et al. (2017) **(DAC)** apply unsupervised clustering to image clustering and proposes a binary-classification framework that uses adaptive learning for optimization. Then, **DeepCluster** (Caron et al. 2018) proposes an end-to-end training method that performs cluster assignments and representation learning alternately. However, the key drawback of unsupervised methods is their incapability of taking advantage of prior knowledge to guide the clustering.

Semi-supervised Clustering With the aid of a few labeled data, semi-supervised clustering usually produces better results compared with unsupervised counterparts. **PCK-Means** (Basu, Banerjee, and Mooney 2004) proposes that the clustering can be supervised by pairwise constraints between samples in the dataset. **KCL** (Hsu, Lv, and Kira 2017) transfers knowledge in the form of pairwise similarity predictions firstly and learns a clustering network to transfer learning. Along this line, **MCL** (Hsu et al. 2019) further formulates multi-classification as meta classification that predicts pairwise similarity and generalizes the framework to various settings. **DTC** (Han, Vedaldi, and Zisserman 2019) extends the DEC algorithm and proposes a mechanism to estimate the number of new images categories using labeled data. When it comes to the field of text clustering, **CDAC+** (Lin, Xu, and Zhang 2020) combines the pairwise constraints and target distribution to discover new intents while **DeepAligned** (Zhang et al. 2021) introduces an alignment strategy to improve the clustering consistency. Very recently, **SCL** (Shen et al. 2021) incorporates a strong backbone MPNet in the Siamese Network structure with contrastive loss (or rely on a large amount of additional external data (Zhang et al. 2022)) to learn the better sentence representations. Although these methods take known intents into account, they may suffer from knowledge forgetting during the training process. More importantly, these methods are

insufficient in the probe into the intrinsic structure of unlabeled data, making it hard to distinguish the characteristics that form an intent.

Approach

Problem Definition

Given as input an labeled dataset $D^l = \{x_i^l, i = 1, \ldots, N\}$ where intents $Y^l = \{y_i^l, i = 1, \ldots, N\}$ are known and an unlabeled dataset $D^u = \{x_i^u, i = 1, \ldots, M\}$. Our goal is to produce intent assignments as output by clustering (or partition) the whole dataset D, which denotes $D = D^l \cup D^u$. Directly optimizing the goal is intractable as the lack of knowledge about new intents and the intrinsic structure of unlabeled data. As analyzed in introduction, discovering intents should not damage but be beneficial for the identification of known intents, which can be formulated to optimize $p(Y^l|D^l, D; \theta)$.

Denote our latent variable representing intent assignments obtained by clustering on D by Z and let $\mathcal{Z}_D$ be a possible value of Z. Using Bayes rule, $p(Y^l|D^l, D; \theta)$ can be calculated as:

$$p(Y^l|D^l, D) = \sum_{\mathcal{Z}_D \in Z} p(Y^l|\mathcal{Z}_D, D^l) p(\mathcal{Z}_D|D^l). \quad (1)$$

Exactly optimizing Eq. (1) is intractable as it is combinatorial in nature. Consider a specific value $\mathcal{Z}$ (omitting subscript D for clarity), the log-likelihood can be simplified as:

$$\mathcal{L}_{obj} = \log p(Y^l|\mathcal{Z}, D^l; \theta) + \log p(\mathcal{Z}|D^l; \theta). \quad (2)$$

Our goal is get better $\mathcal{Z}$ (i.e.intent discovery) by optimizing $\mathcal{L}_{obj}$, and a better $\mathcal{Z}$ can also help optimize $\mathcal{L}_{obj}$.

Intent Representation and Transfer Knowledge

Before optimizing $\mathcal{L}_{obj}$, we want to transfer knowledge from labeled corpus to initialize the model. Transferring knowledge has been widely studied and types of transferred knowledge have been proposed for a variety of circumstances. Considering the excellent generalization of the pre-trained model, we fine-tune BERT (Devlin et al. 2018) with labeled corpus under the supervision of cross entropy. Given the i-th labeled utterance, we first get its contextual embeddings $[[CLS], T_1, T_2, ..., T_N]$ by utilizing BERT and then perform mean-pooling to get sentence semantic representation Z_i = Mean-Pooling($[[\text{CLS}], T_1, ...T_N]$), where $Z_i \in \mathcal{R}^H$, N is the sequence length and H is the hidden dimension. The objective of fine-tune $\mathcal{L}_{ce}$ as:

$$\mathcal{L}_{\text{ce}} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(\phi(z_i)^{y_i})}{\sum_{j=1}^{K^l} \exp(\phi(z_i)^j)}, \quad (3)$$

where $\phi(\cdot)$ denotes linear classifier and $\phi(z_i)^j$ denotes the logits of the j-th class.

EM Framework for Optimization

Intent Assignments $\mathcal{Z}$ Specific intent assignments $\mathcal{Z}$ involves two components: how to determine K representing how many intents in dataset D and how to assign the utterance in the dataset to corresponding intent. Many methods (Han, Vedaldi, and Zisserman 2019; Shen et al. 2021) have been proposed to estimate K. Considering the trade-off between the efficiency and effect, we follow Zhang et al. (2021) (see subsequent analysis for improvements by us). We first set a rough value $\mathcal{K}$ (e.g., the multiple of the ground truth number) for K and extract the semantic feature of the utterance using above fine-tuned BERT. Furtherly, we group the dataset D as K semantic clusters using k-means and drop clusters whose size is less than a certain threshold. The K is calculated as:

$$K = \sum_{i=1}^{\mathcal{K}} \mathbb{I}(|C_i| >= \theta), \quad (4)$$

where $|C_i|$ is the size of i-th produced cluster, $\mathbb{I}$ is an indicator function and θ is the threshold which is set to the same value as suggested in Zhang et al. (2021).

After estimating how many intents are contained in the dataset, we perform k-means to assign cluster assignments as (pseudo) intent to each utterance. Next, We discuss in detail how to further optimize Eq. (2) with Expectation-Maximization (EM) algorithm framework.

E-Step We have assigned a specific intent assignment $\mathcal{Z}$ to latent variable Z based on prior knowledge. We expect that the intent assignments $\mathcal{Z}$ should reflect what characteristics make a good intent in general rather than specific intents. Therefore, the standard cross entropy loss formed by specific pseudo labels adopted by Caron et al. (2018); Zhang et al. (2021) can not achieve this purpose, and even the model may be confused by the false pseudo labels according to Wang et al. (2021). To better reflect the intrinsic structure of dataset D and learn friendly features for intent assignments, we hope that intent assignments $\mathcal{Z}$ can make utterances with the same intent close enough and pull utterances with different intents far away in the semantic feature space. Inspired by contrastive learning paradigm, we estimate the posterior $p(\mathcal{Z}|D^l; \theta)$:

$$p(\mathcal{Z}|D^l; \theta) = \prod_{C_k \in \mathcal{Z}} p(C_k|D^l; \theta) \quad (5)$$

$$= \prod_{C_k \in \mathcal{Z}} \prod_{x \in C_k} p(x \in C_k|D^l; \theta) \quad (6)$$

$$\propto \prod_{C_k \in \mathcal{Z}} \prod_{x \in C_k} \frac{\sum_{x^+ \in C_k} exp(x \cdot x^+)}{\sum_{x^p \in D \setminus \{x\}} exp(x \cdot x^p)}, \quad (7)$$

where C_k is a cluster produced by $\mathcal{Z}$, and $x \cdot x^+$ is calculated by consine between features. To optimize Eq. (2), we also need to compute $p(Y^l|\mathcal{Z}, D^l; \theta)$. Exactly computing is difficult as the label space in $\mathcal{Z}$ does not match that of Y^l. Consider the disaster forgetting as in Deepaligned mentioned above, we approximate $p(Y^l|\mathcal{Z}, D^l; \theta)$:

$$p(Y^l|\mathcal{Z}, D^l; \theta) \propto \quad p(Y^l|D^l; \theta) \quad (8)$$

$$\propto \prod_{x \in D^l} \frac{exp(\phi(x)^y)}{\sum_{j=1}^{K^l} exp(\phi(x)^j)}, \quad (9)$$

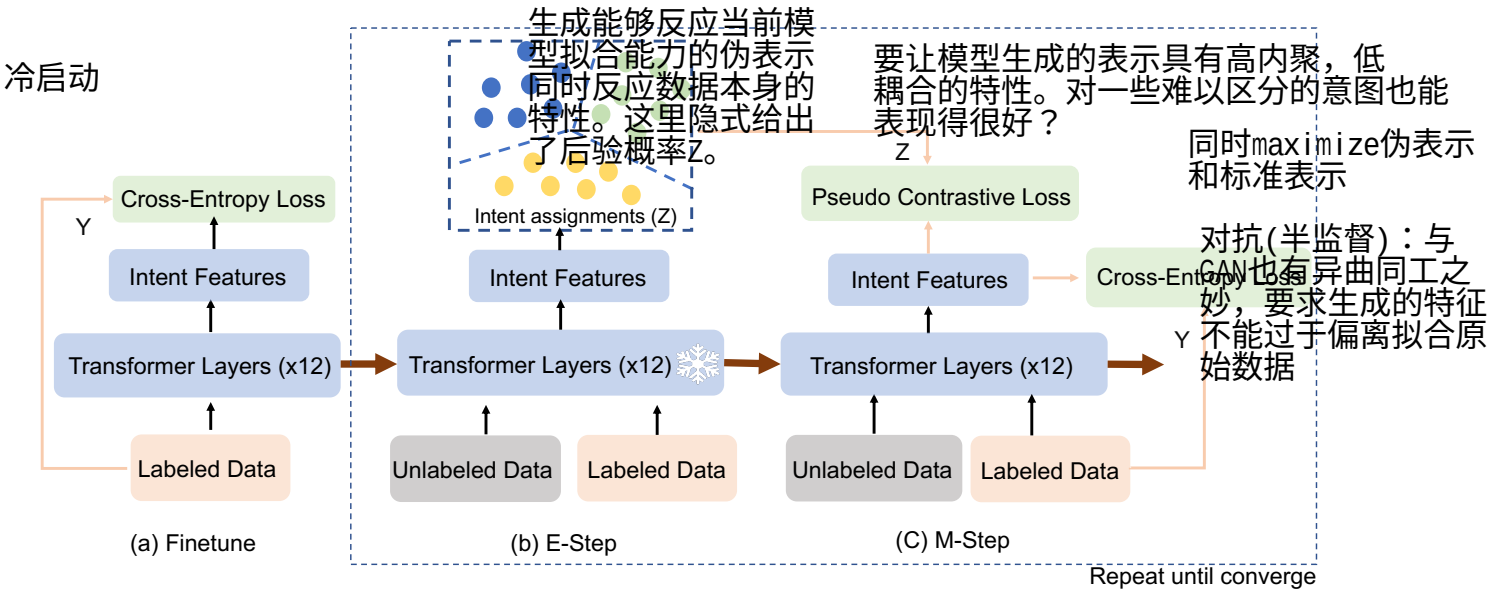

Figure 2: The model architecture of our implementation based on proprosed probabilistic framework. (a) Firstly, we transfer knowledge by fine-tuning BERT with labeled data. (b) Then, we perform intent assignments on full data (labeled and unlabeled data) and reflect the intrinsic structure of data in E-step. (c)And alleviate the forgetting of prior knowledge and update model parameters in M-step. The snow mark represents this step only needs forward without calculating the gradient.

where $\phi(\cdot)$ denotes same linear classifier as Eq. (3), y denotes the intent of x, K^l denotes the total number of known intents and D^l denotes labeled data in D.

Our goal is to tailor the labeled data into model training. On the one hand, the model will not lose the knowledge transferred from labeled data, on the other hand, the model can constantly explore the intrinsic structure of the dataset by utilizing it.

M-Step In the M-step, we update the θ in Eq. (2). In addition to bring Eq. (5) and Eq. (8) into Eq. (2), we introduce two hyper-parameters to help optimize objectives. The overall loss $\mathcal{L}$ can be formulated as follows:

$$\mathcal{L} = \lambda \cdot \sum_{C_k \in \mathcal{Z}} \sum_{x \in C_k} \log \frac{\sum_{x^+ \in C_k} exp(\frac{x \cdot x^+}{\tau})}{\sum_{x^p \in D \setminus \{x\}} exp(\frac{x \cdot x^p}{\tau})} \quad (10)$$

$$+ (1-\lambda) \cdot \sum_{x \in D^l} \log \frac{exp(\phi(x)^y)}{\sum_{j=1}^{K^l} exp(\phi(x)^j)}, \quad (11)$$

where λ is to balance the proportion of two log-likelihoods during training, τ is a hyper-parameter for temperature scaling which often appears in contrastive learning.

We summarize the whole training process of EM framework in Algorithm 1 and the model architecture of our approach as shown in Figure 2. It is worth noting that our method actually proposes a framework where probability estimation can flexibly adopt different ways for a variety of circumstances.

Algorithm 1: EM algorithm for optimization

Input: $D^l = \{x_i^l, i = 1, \ldots, N\}$, $Y^l = \{y_i^l, i = 1, \ldots, N\}$, $D^u = \{x_i^u, i = 1, \ldots, M\}$.
Parameter: Model parameters θ.

1: Intialize θ by transferring knowledge.
2: **while** not converged **do**
3: Perform intent assignment $\mathcal{Z}$ using K-means; \\ E-Step
4: Compute $P(Y^l|\mathcal{Z}, D^l; \theta)$ and $P(\mathcal{Z}|D^l; \theta)$ using current parameters θ; \\ E-Step
5: Update model parameters θ to maximize the log-likelihood in Eq. (10). \\ M-Step
6: **end while**
7: **return** θ

Experiments

Datasets

We conduct experiments on three challenging datasets to verify the effectiveness of our proposed method. The detailed statistics are shown in Table 1.

CLINC (Larson et al. 2019) is a dataset designed for Out-of-domain intent detection, which contains 150 intents from 10 domains and 22500 utterances.

BANKING (Casanueva et al. 2020) is a dataset covering 77 intents and containing 13083 utterances.

StackOverflow is a dataset published in Kaggle.com, which has 20 intents and 20000 utterances. We adopt the dataset processed by Xu et al. (2015).

Baseline and Evaluation Metrics

We follow Lin, Xu, and Zhang (2020); Zhang et al. (2021) and divide the baselines to be compared into two categories: Unsupervised (Unsup.) and Semi-supervised (Semi-sup.). All methods are introduced in Related Work. For a fairness, we uniformly use BERT as the backbone network

Dataset	Classes	#Training	#Validation	#Test	Vocabulary Size	Length (Avg)
CLINC	150	18000	2250	2250	7283	8.32
BANKING	77	9003	1000	3080	5028	11.91
StackOverflow	20	12000	2000	6000	17182	9.18

Table 1: Statistics of datasets. # denotes the total number of utterances.

when compared with the above methods. We also note that SCL (Shen et al. 2021) uses stronger backbone network to obtain semantically meaningful sentence representations, and we also use the same backbone network in comparison with these methods.

To evaluate clustering results, we follow existing methods (Lin, Xu, and Zhang 2020; Zhang et al. 2021) and adopt three widely used metrics: Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and clustering accuracy (ACC). It should be noted that when calculating ACC, the Hungarian algorithm is adopted to find the optimal alignment between the pseudo labels and the ground-truth labels as following Zhang et al. (2021).

Experimental Settings

For each dataset, we randomly select 75% of all intents as known and regard the remaining as unknown. Furthermore, we randomly choose 10% of the known intents data as labeled data. We set the number of intents as ground-truth. Our experimental settings is the same as Lin, Xu, and Zhang (2020); Zhang et al. (2021) for fair comparison. We take different random seeds to run three rounds on the test set and report the averaged results.

Our main experiments use pre-trained BERT (bert-uncased, with 12-layer transformer), which is implemented in PyTorch, as the network backbone. We also replace the backbones of the compared baselines with the same BERT as ours. We adopt most of the suggested hyper-parameters for learning. We try learning rate in $\{1e-5, 5e-5\}$ and λ in $\{0.5, 0.6\}$. The training batch size is 256, and the temperature scale τ is 0.1. All experiments were conducted in the Nvidia Ge-Force RTX-3090 Graphical Card with 24G graphical memory. Only when comparing with SCL (Shen et al. 2021), which definitely point out that they use pre-trained MPNet (Reimers and Gurevych 2019) as the backbone network, will we adopt the same backbone network for a fair comparison.

Moreover, considering the efficiency of the training process and the capacity of GPU, we only fine-tune the last transformer layer parameters during transferring knowledge and freeze all but the latter 6 transformer layers parameters during performing EM algorithm.

Results and Discussion

Main results

We present the main results in table 2, where the best results are highlighted in bold. It is clear from the results that our method achieves substantial improvements in all metrics and all datasets, especially in the BANKING dataset, where the number of samples in each class is imbalanced. These results illustrate the effectiveness and generalization of our method. At the same time, we note most semi-supervised methods are better than unsupervised as a whole, which further verifies the importance of labeled data. From this perspective, we can explain why our method can be better than DeepAligned as it will constantly forget the knowledge existing in labeled data as shown in Introduction, and our method tailors the labeled data into model training to guide clustering so that our method can achieve better results.

To make a fair comparison with SCL (Shen et al. 2021), we also replace the backbone network in our method with the same MPNet as SCL, keeping other parts of our method unchanged. We present the results of our comparison with SCL and various variants (See Shen et al. (2021) for the calculation of specific strategies) on CLINC and BANKING in Table 3, where the best results are also highlighted in bold.

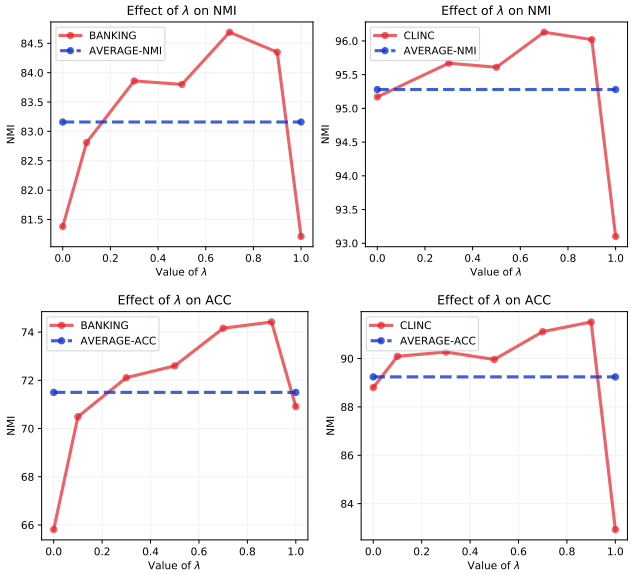

Figure 3: The effects of λ on datasets (Right: CLINC, Left: BANKING). Only utilizing labeled data or only exploring the intrinsic structure will not achieve good results.

Effect of Exploration and Utilization

In objective function Eq. (10), we use λ to reconcile the effects of the two log-likelihoods. Intuitively, the first term is used to explore the intrinsic structure of unlabeled data, and the second term is used to strengthen the knowledge transferred from labeled data to utilize. We vary the value of λ

	Methods	CLINC			BANKING			StackOverflow		
		NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC
Unsup.	K-means	70.89	26.86	45.06	54.57	12.18	29.55	8.24	1.46	13.55
	AC	73.07	27.70	44.03	57.07	13.31	31.58	10.62	2.12	14.66
	SAE-KM	73.13	29.95	46.75	63.79	22.85	38.92	32.62	17.07	34.44
	DEC	74.83	27.46	46.89	67.78	27.21	41.29	10.88	3.76	13.09
	DCN	75.66	31.15	49.29	67.54	26.81	41.99	31.09	15.45	34.56
	DAC	78.40	40.49	55.94	47.35	14.24	27.41	14.71	2.76	16.30
	DeepCluster	65.58	19.11	35.70	41.77	8.95	20.69	-	-	-
Semi-sup.	PCKMeans	68.70	35.40	54.61	48.22	16.24	32.66	17.26	5.35	24.16
	KCL(BERT)	86.82	58.79	68.86	75.21	46.72	60.15	8.84	7.81	13.94
	MCL(BERT)	87.72	59.92	69.66	75.68	47.43	61.14	-	-	-
	CDAC+	86.65	54.33	69.89	72.25	40.97	53.83	69.84	52.59	73.48
	DTC(BERT)	90.54	65.02	74.15	76.55	44.70	56.51	-	-	-
	DeepAligned	93.89	79.75	86.49	79.56	53.64	64.90	76.47	62.52	80.26
	Ours	**95.13**$_{0.46}$	**82.65**$_{1.77}$	**88.35**$_{1.22}$	**83.40**$_{1.44}$	**61.19**$_{3.15}$	**72.59**$_{1.77}$	**77.29**$_{0.80}$	**63.93**$_{2.71}$	**80.90**$_{1.28}$

Table 2: The main results on three datasets. The baselines on CLINC and BANKING are retrieved from Zhang et al. (2021). The baselines on StackOverflow are retrieved from Lin, Xu, and Zhang (2020). We get the baseline of DeepAligned on StackOverflow by running its release code. All reported results are percentages and mean by conducting with different seeds (The subscripts are the corresponding standard deviations).

Methods	CLINC			BANKING		
	NMI	ARI	ACC	NMI	ARI	ACC
SMPNET	93.39	74.28	83.24	82.22	58.82	71.82
SCL	94.75	81.64	86.91	85.04	65.43	76.55
SCL(EP)	95.25	83.44	88.68	84.77	64.44	75.18
SCL(IP)	94.95	82.32	88.28	84.82	64.51	74.81
SCL(AA)	95.11	83.09	88.49	85.02	64.91	75.66
SCL(AC)	94.04	78.99	84.58	83.52	62.18	73.09
Ours	**95.69**	**84.81**	**89.57**	**85.97**	**67.54**	**76.82**

Table 3: The results compared with SCL and variants. IP, EP, AA, and AC represent four pseudo label training strategies:inclusive pairing, exclusive pairing, Alignment-A, and Alignment-C respectively. The baselines are retrieved from Shen et al. (2021).

and conduct experiments on CLINC and BANKING to explore the effect of λ, which also reflects the inference of exploration and utilization. As shown in Figure 3, only utilizing labeled data (λ = 0.0) or only exploring(λ = 1.0) the intrinsic structure will not achieve good results (below average). Interestingly, on all metrics and datasets, the effect of λ shows a similar trend (increase first and then decrease), which indicates that we can adjust the value of λ to give full play to the role of both so that the model can make better use of known knowledge to discover intents accurately. This result shows that if the model wants to achieve good results, exploration and utilization are indispensable.

Estimate the Number of Intents (K)

A key point of intent discovery is whether the model can accurately predict the number of intents. DeepAligned proposes a simple yet effective estimation method. However, due to the alignment operation in the iterative process of clustering (see Zhang et al. (2021) for details), DeepAligned needs to determine K in advance and only limited labeled data is used, while a large number of unlabeled data are ignored. Unlikely, our method does not directly rely on pseudo labels so that we can continue to refine K during subsequent clustering. We use the same settings as Zhang et al. (2021) and firstly assign the number of intents (i.e., $\mathcal{K}$ in intent assignments) as two times the ground truth number to investigate the ability to estimate K. In the process of executing EM algorithm, we refine K per 10 epochs using the method as suggested in above Section. We get the final performance of the model and the results are shown in Table 4 show that our method can predict the number of intents more accurately and achieve better results at the same time.

Methods	CLINC ($\hat{K}$ = 150)			BANKING ($\hat{K}$ = 77)		
	K	Error ↓	ACC ↑	K	Error ↓	ACC ↑
MCL(BERT)	112	25.33	69.2	58	24.68	60.8
DTC(BERT)	195	30.00	66.65	110	42.86	54.94
DeepAligned	129	14.00	77.18	67	12.99	62.49
Ours	**130**	**13.3**	**80.8**	**73**	**5.48**	**69.68**

Table 4: The results of predicting K. The $\hat{K}$ denotes the ground truth number of K. The closer $\hat{K}$ and K is, the more accurate the prediction is. The compared results are retrieved from Zhang et al. (2021).

Effect of the Initial Number of Intents

Because we do not know the actual number of intents, we usually need to assign an initial number of intents (i.e., $\mathcal{K}$) in advance as we do earlier. This also requires us to investigate

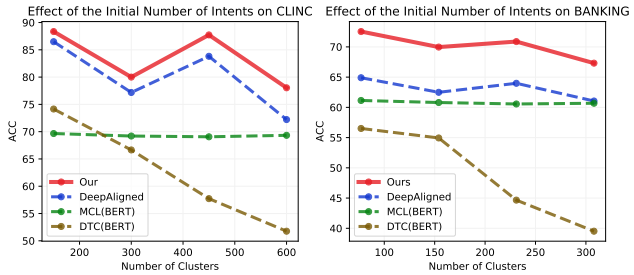

Figure 4: The effect of the Initial Number of Intents on datasets(Left: CLINC, Right: BANKING). The compared results are retrieved from Zhang et al. (2021).

the sensitivity of the model to the initial K. We investigate the performance of our method in the datasets by varying initial values (leaving others unchanged). As shown Figure 4, compared with others, our method can better adapt to different initial values. Combined with the experiments in previous section,we suppose the main reason why our method can achieve better performance is that our method can refine K more accurately than other methods by making the most of the knowledge of the whole data.

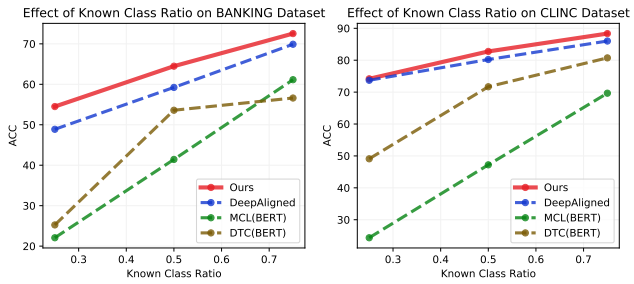

Figure 5: The effect of Known Class Ratio on datasets (Left: BANKING, Right: CLINC). The compared results are retrieved from Zhang et al. (2021).

Effect of the Known Intent Ratio

We also investigate the effect of known intent ratios on the model. We adopt different known class ratios (25%, 50% and 75%) and experiment on datasets. As shown in Figure 5, our method also shows better performance compared with other methods. Interestingly, The advantage of our method in dataset BANKING is particularly obvious. We speculate that this may be related to the imbalance of samples in each intent in the BANKING dataset. Although there are more known intents, it is unable to provide enough labeled and balanced samples. As a result, the previous methods (e.g. DeepAligned) not only failed to transfer more prior knowledge but also exacerbated the speed of forgetting in the follow-up process.

More Than Remembering Prior Knowledge

We showed knowledge forgetting in DeepAligned in introduction. Interestingly, there is a phenomenon called *model shift* in the field of computer vision, which means fine-tuned model would shift towards labeled data, see Wang et al. (2021) for details. However, in our scenario and our experiments, the model is moving in the opposite direction and is shifting away from labeled data, which reasonable explanation should be the forgetting of knowledge. After fine-tuning with labeled data, the prior knowledge is stored in the model in the form of model parameters. With the subsequent clustering steps, the parameters change gradually (the forgetting process is step by step from the forgetting curve).

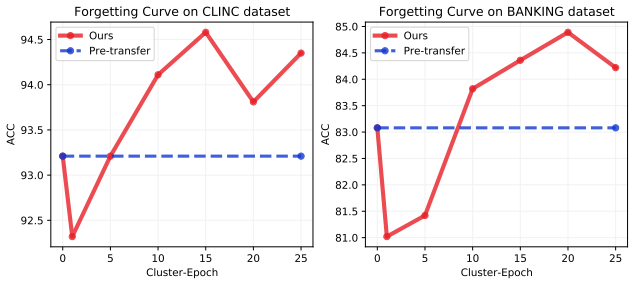

Figure 6: The knowledge curves of our method (Red). During intent assignments in our method, we also test the performance of the model on the validation set used in the pre-transfer stage and show that with the advancement of clustering, our performance can be better than in the pre-transfer stage. The blue line represents the baseline obtained by the model in the pre-transfer stage.

In our proposed framework, we just use a simple yet effective method (which can be dynamically adjusted according to different needs) to re-memorize this prior knowledge. Therefore, as shown in Figure 6, we observe that our method does not have the catastrophic forgetting that occurs in DeepAligned. On the contrary, with the iteration (EM algorithm), our performance is better than that in the pre-transfer stage. We surmise that this is presumably because the knowledge contained in the unlabeled data corpus helps the identification of the known intents.

Conclusion

In this paper, we provide a probabilistic framework for intent discovery. This is the first complete theoretical framework for intent discovery. We also provide an efficient implementation based on this proposed framework. Compared with the existing methods, our method effectively alleviates the forgetting of prior knowledge transferred from known intents and provides intensive clustering supervised signals for discovering intents. Extensive experiments conducted in three challenging datasets demonstrate our method can achieve substantial improvements. The subsequent analysis also show that our method can better estimate the number of intents and adapt to various conditions. In the future, we will try different methods to perform intent assignments and explore more methods to approximate $p(Y^l|\mathcal{Z}, D^l; \theta)$ and $p(\mathcal{Z}|D^l\theta)$. 寻找更有效的建模方法

References

Basu, S.; Banerjee, A.; and Mooney, R. J. 2004. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the 2004 SIAM international conference on data mining*, 333–344. SIAM.

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 132–149.

Casanueva, I.; Temčinas, T.; Gerz, D.; Henderson, M.; and Vulić, I. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

Chang, J.; Wang, L.; Meng, G.; Xiang, S.; and Pan, C. 2017. Deep adaptive image clustering. In *Proceedings of the IEEE international conference on computer vision*, 5879–5887.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Gowda, K. C.; and Krishna, G. 1978. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern recognition*, 10(2): 105–112.

Hakkani-Tür, D.; Celikyilmaz, A.; Heck, L.; and Tur, G. 2013. A weakly-supervised approach for discovering new user intents from search query logs. In *Annual Conference of the International Speech Communication Association (Interspeech)*.

Hakkani-Tür, D.; Ju, Y.-C.; Zweig, G.; and Tur, G. 2015. Clustering novel intents in a conversational interaction system with semantic parsing. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Han, K.; Vedaldi, A.; and Zisserman, A. 2019. Learning to discover novel visual categories via deep transfer clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8401–8409.

Hsu, Y.-C.; Lv, Z.; and Kira, Z. 2017. Learning to cluster in order to transfer across domains and tasks. *arXiv preprint arXiv:1711.10125*.

Hsu, Y.-C.; Lv, Z.; Schlosser, J.; Odom, P.; and Kira, Z. 2019. Multi-class classification without multi-class labels. *arXiv preprint arXiv:1901.00544*.

Larson, S.; Mahendran, A.; Peper, J. J.; Clarke, C.; Lee, A.; Hill, P.; Kummerfeld, J. K.; Leach, K.; Laurenzano, M. A.; Tang, L.; et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*.

Lin, T.-E.; Xu, H.; and Zhang, H. 2020. Discovering new intents via constrained deep adaptive clustering with cluster refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 8360–8367.

MacQueen, J.; et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, 281–297. Oakland, CA, USA.

Padmasundari, S. B. 2018. INTENT DISCOVERY THROUGH UNSUPERVISED SEMANTIC TEXT CLUSTERING. *Proc. Interspeech 2018*, 606–610.

Reimers, N.; and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Shen, X.; Sun, Y.; Zhang, Y.; and Najmabadi, M. 2021. Semi-supervised Intent Discovery with Contrastive Learning. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, 120–129.

Shi, C.; Chen, Q.; Sha, L.; Li, S.; Sun, X.; Wang, H.; and Zhang, L. 2018. Auto-Dialabel: Labeling Dialogue Data with Unsupervised Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 684–689. Brussels, Belgium: Association for Computational Linguistics.

Wang, X.; Gao, J.; Long, M.; and Wang, J. 2021. Self-tuning for data-efficient deep learning. In *International Conference on Machine Learning*, 10738–10748. PMLR.

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

Xu, J.; Wang, P.; Tian, G.; Xu, B.; Zhao, J.; Wang, F.; and Hao, H. 2015. Short text clustering via convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 62–69.

Yang, B.; Fu, X.; Sidiropoulos, N. D.; and Hong, M. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, 3861–3870. PMLR.

Zhang, H.; Xu, H.; Lin, T.-E.; and Lyu, R. 2021. Discovering new intents with deep aligned clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 14365–14373.

Zhang, Y.; Zhang, H.; Zhan, L.-M.; Wu, X.-M.; and Lam, A. 2022. New Intent Discovery with Pre-training and Contrastive Learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 256–269. Dublin, Ireland: Association for Computational Linguistics.

Zhou, Y.; Liu, P.; and Qiu, X. 2022. KNN-Contrastive Learning for Out-of-Domain Intent Classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5129–5141. Dublin, Ireland: Association for Computational Linguistics.