A Probabilistic Framework for Discovering New Intents

Yunhua Zhou[*], **Guofeng Quan**[*], **Xipeng Qiu**[†]
School of Computer Science, Fudan University
{zhouyh20,xpqiu}@fudan.edu.cn
gfquan21@m.fudan.edu.cn

Abstract

Discovering new intents is of great significance for establishing the Task-Oriented Dialogue System. Most prevailing approaches either cannot transfer prior knowledge inherent in known intents or fall into the dilemma of forgetting prior knowledge in the follow-up. Furthermore, such approaches fail to thoroughly explore the inherent structure of unlabeled data, thereby failing to capture the fundamental characteristics that define an intent in general sense. In this paper, starting from the intuition that discovering intents should be beneficial for identifying known intents, we propose a probabilistic framework for discovering intents where intent assignments are treated as latent variables. We adopt the Expectation Maximization framework for optimization. Specifically, In the E-step, we conduct intent discovery and explore the intrinsic structure of unlabeled data by the posterior of intent assignments. In the M-step, we alleviate the forgetting of prior knowledge transferred from known intents by optimizing the discrimination of labeled data. Extensive experiments conducted on three challenging real-world datasets demonstrate the generality and effectiveness of the proposed framework and implementation. Codes is publicly available.[1]

1 Introduction

Unknown intent detection (Zhou et al., 2022) in the Task-Oriented Dialogue System (TODS) has gradually attracted more and more attention from researchers. However, detecting unknown intent is only the first step. For the TODS, intent discovery is crucial but also more challenging. Because the pre-defined intent set in the TODS is limited to cover all intents, the TODS should discover potential new intents automatically during interactions with the users. And as a practical matter, 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 expertise, the TODS should adaptively discover intents from those unlabeled data with the aid of limited labeled data.

Just as discovering new intents plays a crucial role in establishing the TODS, discovering new intents has raised a lot of research interest just like unknown intent detection. Unsupervised cluster learning is a popular paradigm to solve this problem. Specifically, previous works (Hakkani-Tür et al., 2013, 2015; Shi et al., 2018; Padmasundari, 2018) formulate intent discovery as an unsupervised clustering process. However, these methods mainly focus on constructing pseudo-supervised signals to guide the clustering process while neglecting the prior knowledge embedded in the available labeled data. In real user-facing scenarios, we often possess a small amount of labeled data in advance, which contains prior knowledge that can guide the intent discovery process, and a substantial volume of unlabeled data generated in the interaction with the dialogue system mentioned above, which contains both known intents and unknown intents to be discovered.

How do discover intents in the unlabeled corpus using the labeled data? Recently, the semi-supervised methods (Lin et al., 2020; Zhang et al., 2021) have become popular. DeepAligned (Zhang et al., 2021) is the most typical and has also inspired a series of effective works (Shen et al., 2021; Zhang et al., 2022) recently. DeepAligned first generalizes the prior knowledge into the semantic features of unlabeled data by pre-training. Then, to learn cluster-friendly representations, DeepAligned assigns a pseudo label to each unlabeled utterance and re-trains the model under the supervision of those pseudo labels.

[*] Equal contribution.

[†] Corresponding author.

[1] https://github.com/zyh190507/Probabilistic-discovery-new-intents

*Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers*, pages 3771–3784
July 9-14, 2023 ©2023 Association for Computational Linguistics

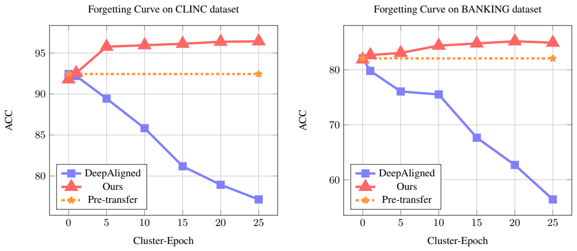

Figure 1: The catastrophic forgetting of DeepAligned (Blue). During clustering in DeepAligned, we test the performance of the model on the verification set used in the transferring prior knowledge stage and show that with the advancement of clustering, the model constantly forgets the knowledge learned from labeled data. The brown line represents the baseline obtained by the model after transferring prior knowledge. In contrast, our method (Red) can alleviate forgetting well. See Section 5.6 for more discussion.

Nevertheless, DeepAligned suffers from many problems. Firstly, when the model is re-trained with the pseudo supervision signal, the model will forget the knowledge transferred in the transferring stage, which is demonstrated in Figure 1. Then, the model could be misled by inaccurate pseudo labels, particularly in large-sized intent space (Wang et al., 2021). More importantly, softmax loss formed by pseudo labels cannot explore the intrinsic structure of unlabeled data, so it can not provide accurate clustering supervised signals.

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

Therefore, with the help of optimizing the identification of labeled data given the whole data corpus, we propose a principled probabilistic framework for intent discovery, where intent assignments as a latent variable. We adopt Expectation Maximization as a principal template for optimizing 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 to explore the intrinsic structure of data. In the M-step, the probability of identification of labeled data including those newly discovered from unlabeled data, and the posterior probability of intent assignments, which is to help learn friendly-discovery features, are maximized simultaneously to optimize and update model parameters. Extensive experiments conducted in three benchmark datasets demonstrate our method can achieve substantial improvements over strong baselines. Our contributions are as follows:

(Theory) We introduce a principled probabilistic framework for discovering new 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 yet effective method to alleviate forgetting. Furthermore, we propose a new contrastive paradigm to explore the intrinsic structure of unlabeled data, which avoids the model shift towards inaccurate pseudo labels but helps to better learn the friendly-discovery features.

(Experiments and Analysis) We conduct extensive experiments and detailed analyses on a suite of real-world datasets to demonstrate the generality and effectiveness of our proposed framework and implementation.

2 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 et al. (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 By virtue of a few labeled data, semi-supervised clustering usually produces better results compared with unsupervised counterparts. **PCK-Means** (Basu et al., 2004) proposes that the clustering can be supervised by pairwise constraints between samples in the dataset. **KCL** (Hsu et al., 2017) transfers knowledge in the form of pairwise similarity predictions first 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 paradigm to various settings. **DTC** (Han et al., 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 et al., 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. Recently, **SCL** (Shen et al., 2021) incorporates a strong backbone MPNet in the Siamese Network structure with pairwise contrastive loss to learn the sentence representations. Similarly, **MTP** (Zhang et al., 2022) enhances sentence representation through multi-task pre-training strategy and extra data. 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.

3 Approach

3.1 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 partitioning) 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 Section 1, discovering intents should not damage but be beneficial for the identification of known intents, which can be formulated to optimize $p(Y^l|D^l, D; \theta)$. Since $D^l \subset D$, the optimization objective can be written as: $p(Y^l|D; \theta)$.

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, $p(Y^l|D; \theta)$ can be calculated as:

$$p(Y^l|D) = \sum_{\mathcal{Z}_\mathcal{D} \in Z} p(Y^l|\mathcal{Z}_\mathcal{D}, D; \theta) p(\mathcal{Z}_\mathcal{D}|D; \theta). \tag{1}$$

Exactly optimizing Eq.(1) is intractable due to its combinatorial nature. Consider a specific value $\mathcal{Z}_D$, the log-likelihood can be simplified as:

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

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

3.2 Intent Representation and Transferring Knowledge

Before optimizing $\mathcal{L}_{obj}$, we want to transfer knowledge from the 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 as suggested in (Zhang et al., 2021). Given the i-th labeled utterance x_i, we first get its contextual embeddings by utilizing BERT and then perform mean-pooling to get sentence semantic representation z_i. The objective of fine-tune $\mathcal{L}_{ce}$ as:

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

where $\phi(\cdot)$ represents a linear classifier and $\phi(z_i)^j$ denotes the logits of the j-th class, K^l denotes the total number of known intents.

3.3 EM Framework for Optimization

Intent Assignments $\mathcal{Z}$ (In the following, we omit the subscript D of $\mathcal{Z}_D$ for clarity.) 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 et al., 2019; Shen et al., 2021) have been proposed to estimate K. Considering the tradeoff between efficiency and effect, we follow Zhang et al. (2021) (see Appendix D for discussions on more accurate estimating K under our framework) and first set a rough value $\mathcal{K}$ (e.g., the multiple of the ground truth number) for K and then refine it by dropping clusters (formed by grouping the dataset D into $\mathcal{K}$ semantic clusters using k-means) whose size is less than a certain threshold. 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 $p(\mathcal{Z}|D;\theta)$:

$$\begin{aligned} p(\mathcal{Z}|D;\theta) &= \prod_{C_k \in \mathcal{Z}} p(C_k|D;\theta) && (4)\\ &= \prod_{C_k \in \mathcal{Z}} \prod_{x \in C_k} p(x \in C_k|D;\theta) && (5)\\ &\propto \prod_{C_k \in \mathcal{Z}} \prod_{x \in C_k} \frac{\sum_{x^+ \in C_k} exp(x \cdot x^+)}{\sum_{x^p \in D \setminus \{x\}} exp(x \cdot x^p)}, && (6) \end{aligned}$$

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 $p(Y^l|\mathcal{Z}, D;\theta)$. Exactly computing is difficult as the label space in $\mathcal{Z}$ does not match that of Y^l. Consider the catastrophic forgetting as in Deepaligned mentioned above, we approximate $p(Y^l|\mathcal{Z}, D;\theta)$:

$$\begin{aligned} p(Y^l|\mathcal{Z}, D;\theta) =& p(Y^l|\mathcal{Z}, D^l, D^u;\theta) && (7)\\ \propto &\ p(Y^l|D^l, \hat{D}^l(D^u, \mathcal{Z});\theta) && (8)\\ \propto &\prod_{x \in D^l \cup \hat{D}^l} \frac{exp(\phi(x)^y)}{\sum_{j=1}^{K^l} exp(\phi(x)^j)}, && (9) \end{aligned}$$

where $\phi(\cdot)$ denotes same linear classifier as Eq.(3), y denotes the label of x, K^l denotes the total number of known intents and D^l denotes labeled data in D. $\hat{D}^l(D^u, \mathcal{Z})$ refers to the set of samples in D^u that can be considered as known intents after the operation of $\mathcal{Z}$.

$$\hat{D}^l = \{(x, y^l)|x \in \mathcal{N}_{\mathcal{Z}}(x^l), (x^l, y^l) \in D^l\}, \quad (10)$$

where x^l is the sample from D^l, y^l is the label of x^l. $\mathcal{N}_{\mathcal{Z}}(x^l)$ is the unlabeled nearest neighbor samples set that belongs to the same cluster (divided by $\mathcal{Z}$) as x^l. See Appendix E for specific benefits from $\hat{D}^l$. The labeled data is tailored to 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 bringing Eq. (4) and Eq. (7) into Eq. (2), we introduce two hyper-parameters to help optimize objectives. The overall loss $\mathcal{L}$ can be formulated as follows:

$$\begin{aligned} \mathcal{L} = \lambda \cdot \sum_{C_k \in \mathcal{Z}} \sum_{x \in C_k} \log \frac{\sum_{x^+ \in C_k} exp(\frac{x \cdot x^+}{\tau})}{\sum_{x^p \in D \setminus \{x\}} exp(\frac{x \cdot x^p}{\tau})} && (11)\\ + (1-\lambda) \cdot \sum_{x \in D^l \cup \hat{D}^l} \log \frac{exp(\phi(x)^y)}{\sum_{j=1}^{K^l} exp(\phi(x)^j)}, && (12) \end{aligned}$$

where λ is to balance the proportion of two log-likelihoods (discussed in Section 5.3) during training, τ is a hyper-parameter for temperature scaling which often appears in contrastive learning.

We summarize the whole training process of the 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.

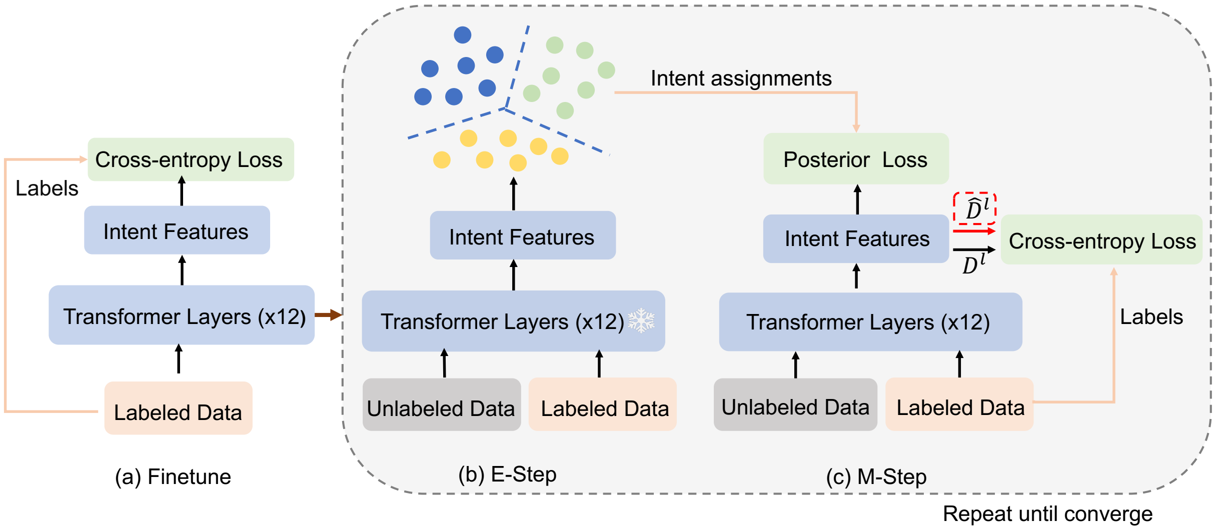

Figure 2: The model architecture of our implementation based on the proposed probabilistic framework. (a) Firstly, we transfer knowledge by fine-tuning BERT with labeled data. (b) Then, we perform intent assignments on full data (labeled and unlabeled data) and reflect the intrinsic structure of data 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.

Algorithm 1 EM algorithm for optimization

```
**Input**: $D^l = \{x_i^l, i = 1, \dots, N\}$, $Y^l = \{y_i^l, i = 1, \dots, N\}$, $D^u = \{x_i^u, i = 1, \dots, M\}$.
**Parameter**: Model parameters $\theta$.
Intialize $\theta$ by transferring knowledge.
while not converged do
    Perform intent assignment ($\mathcal{Z}$) using K-means; \\ E-Step
    Compute $P(Y^l|\mathcal{Z}, D; \theta)$ and $P(\mathcal{Z}|D; \theta)$ using current parameters $\theta$; \\ E-Step
    Update model parameters $\theta$ to maximize the log-likelihood $\mathcal{L}$ in Eq. (11). \\ M-Step
end while
return $\theta$
```

4 Experiments

4.1 Datasets

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

CLINC (Larson et al., 2019) is a popular intent 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 banking dataset covering 77 intents and containing 13083 utterances.

StackOverflow represents a dataset dispersed through Kaggle.com, encompassing 20 intents and 20000 utterances. We adopt the dataset processed by Xu et al. (2015).

4.2 Baseline and Evaluation Metrics

We follow Lin et al. (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 (Section 2). For fairness, we uniformly use BERT as the backbone network when compared with the above methods. We also note that SCL (Shen et al., 2021) uses a stronger backbone network to obtain semantically meaningful sentence representations, and we also use the same backbone network in comparison with these methods. Similarly, when comparing with MTP-CLNN (Zhang et al., 2022), we use the same additional data and multi-task pre-training to enhance sentence representation.

To evaluate clustering results, we follow existing methods (Lin et al., 2020; Zhang et al., 2021) and adopt three widely recognized 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).

	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	34.26
	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	93.95	80.33	87.29	79.91	54.34	66.59	76.47	62.52	80.26
	Ours	**95.01**$_{0.49}$	**83.00**$_{1.54}$	**88.99**$_{1.05}$	**84.02**$_{0.82}$	**62.92**$_{2.00}$	**74.03**$_{1.37}$	**77.32**$_{1.02}$	**65.70**$_{2.07}$	**80.50**$_{1.14}$

Table 1: The main results on three datasets. We re-run the result of DeepAligned by its release code. The other baselines on CLINC and BANKING are retrieved from Zhang et al. (2021). The baselines on StackOverflow are retrieved from Lin et al. (2020). All reported results are averaged over different seeds and the subscripts represent the corresponding standard deviations. See text for details.

4.3 Experimental Settings

For each dataset, 75% of all intents are randomly selected as known intent, with the remaining designated as unknown. Furthermore, 10% of the known intents data are chosen randomly as labeled data. We set the number of intents as ground truth in line with previous methods Lin et al. (2020); Zhang et al. (2021, 2022). Our other experimental settings are mostly the same as Lin et al. (2020); Zhang et al. (2021, 2022) for a fair comparison. We take different random seeds to run at least three rounds on the test set and report the final averaged results.

Our main experiments use pre-trained BERT, which is implemented in the Huggingface Transformers[2], as the network backbone. We also replace the backbones of the compared baselines with the same BERT as ours. 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. Similarly, we will use the same additional data and the same pre-training strategy for fair comparison only when we compare with MTP (Zhang et al., 2022).

Moreover, considering the efficiency of the training process and the capacity of GPU, we only finetune the last transformer layer parameters during transferring knowledge and freeze all but the latter 6 transformer layers parameters during performing the EM algorithm. See Appendix B for training details and parameters.

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	**95.94**$_{0.24}$	**85.69**$_{0.90}$	**90.44**$_{0.77}$	**86.85**$_{0.40}$	**69.28**$_{0.32}$	**79.32**$_{0.91}$

Table 2: 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).

Methods	BANKING			Stackoverflow		
	NMI	ARI	ACC	NMI	ARI	ACC
MTP	85.17	64.37	74.20	80.70	71.68	83.74
MTP(DAC)	85.78	65.28	75.43	80.89	71.17	84.20
MTP(CLNN)	87.68	70.43	79.61	81.30	73.29	86.56
Ours	**88.61**$_{0.96}$	**73.61**$_{2.61}$	**83.15**$_{2.93}$	**81.93**$_{0.24}$	**74.76**$_{0.55}$	**87.03**$_{0.21}$

Table 3: The results compared with MTP and variants. DAC and CLNN are different strategies for intent discovery (see (Zhang et al., 2022) for details). We re-run the result of MTP(CLNN) by its released code. The other baselines are retrieved from Zhang et al. (2022).

[2] https://github.com/huggingface/transformers

5 Results and Discussion

5.1 Main results

We present the main results in table 1, 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 that 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 Section 1, 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 2, where the best results are also highlighted in bold. Table 3 is the result of the comparison between our method and MTP, where *DAC* and *CLNN* are different strategies for intent discovery after pre-training. To make a fair comparison, we only adopt the same additional data and pre-training strategies (based on its released code) as MTP in the first step (Finetune stage in Figure 2), and the rest of the methods remain unchanged.

Methods	Known			Unknown		
	NMI	ARI	ACC	NMI	ARI	ACC
DeepAligned	95.45	85.69	91.05	91.69	78.91	86.31
Ours(Clinc)	**97.16**	**91.61**	**95.20**	**92.50**	**81.10**	**87.37**
DeepAligned	82.13	60.62	72.00	78.11	61.23	74.74
Ours(Banking)	**88.06**	**74.53**	**85.23**	**78.46**	**61.89**	**74.78**
DeepAligned	78.77	61.83	81.86	59.36	52.83	75.20
Ours(Stackover.)	**80.34**	**74.85**	**87.55**	**60.72**	**57.96**	**81.07**

Table 4: Comparison results on Known and Unknown intents respectively. From top to bottom, there are CLINC, BANKING and Stackoverflow datasets (the name of the dataset is filled in parentheses).

5.2 A Closer Look at Effectiveness

To better verify the effectiveness of our proposed method, we analyze the comparison results between our method and DeepAligned in a more fine-grained way. We separate the known intents and the unknown intents from the test set and compare our method with DeepAligned on these two sub-datasets respectively (the experimental settings remain unchanged). The results are shown in Table 4, which demonstrates that our method can not only effectively apply to known intents, but also can more effectively discover new intents, and the effect of improvement is substantial. This also fully conforms to our expectations that the two processes of intent discovery and recognition of known intents can be "win-win".

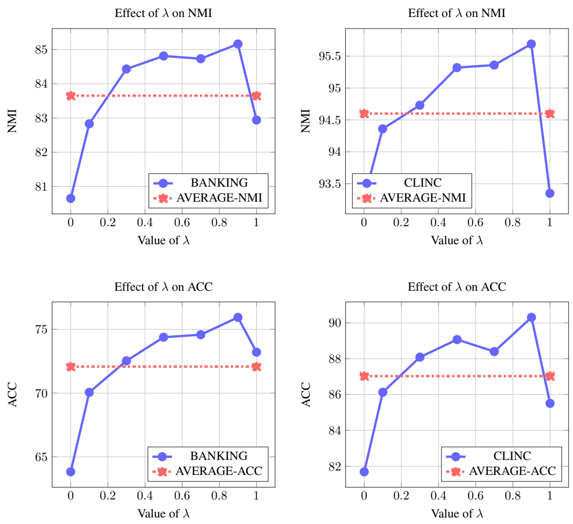

Figure 3: The effects of λ on CLINC and BANKING. Detailed performance are available in Appendix C.

5.3 Effect of Exploration and Utilization

In objective function Eq. (11), 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 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 ($\lambda = 0.0$) or only exploring($\lambda = 1.0$) 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.

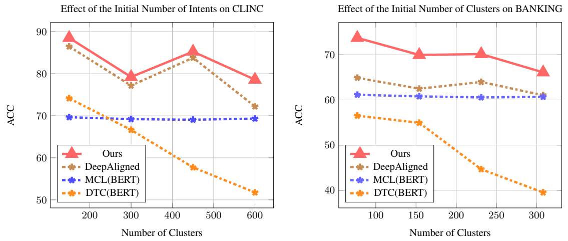

Figure 4: The effect of the initial number of intents on datasets(Left: CLINC, Right: BANKING). The compared results are retrieved from Zhang et al. (2021).

5.4 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 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.

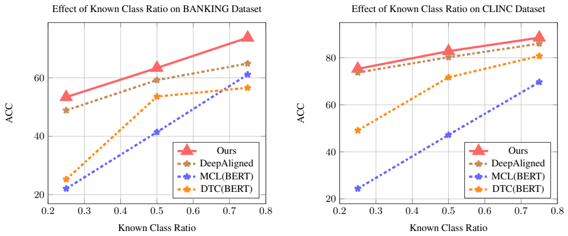

Figure 5: The effect of known class ratio on datasets (Left: BANKING, Right: CLINC). The compared results are retrieved from Zhang et al. (2021).

5.5 Effect of the Known Intent Ratio

We also investigate the effect of known intent ratios on performance by adopting different known class ratios (25%, 50% and 75%). As shown in Figure 5, our method also shows better performance compared with other baselines. Interestingly, The advantage of our method in dataset BANKING is significant. We speculate that this may be related to the unbalanced number of samples in BANKING. Although there are more known intents, it does not mean that enough labeled and balanced samples are provided. 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. This also provides room for future research.

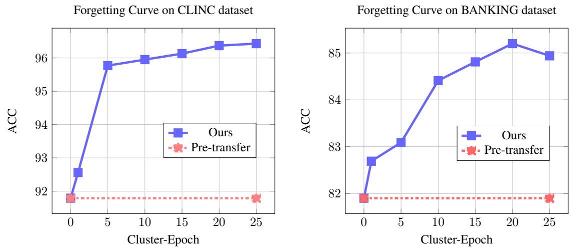

Figure 6: The knowledge curves of our method (Blue). During intent assignments, our performance is not only not forgotten, but also constantly strengthened compared with the pre-transfer stage (Red, approximated by the initial performance in clustering stage).

5.6 More Than Remembering Knowledge

We show knowledge forgetting in DeepAligned in Section 1. 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 in previous works).

However, 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 improvement is brought by the sample set $\hat{D}^l$ discovered in the unlabeled data (also can improve the intent discovery in Appendix E) corpus helps the identification of the known intents.

6 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 shows 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 $p(Y^l|\mathcal{Z}, D; \theta)$ and $p(\mathcal{Z}|D; \theta)$.

Limitations

To better inspire the follow-up work, we summarize the limitations of our method as follows: 1) From our experimental results the Appendix D, we can see that the estimation of the number of intents in our proposed can be further improved. 2) We do not try more means to prevent knowledge from forgetting. We can probe into the intrinsic structure of unlabeled data in a more fine-grained way by improving the posterior estimation. 3) According to Section 5.3, we have verified that both exploration and utilization are indispensable, but at the same time, we only empirically choose the specific proportion of both, without theoretical analysis of the most appropriate proportion for each dataset. We look forward to making progress in the follow-up research on the above limitations.

Acknowledgements

We thank Dr.Wang yuxin and Dr.Liu Peiju for their patience and meaningful suggestions in this work early. This work was supported by the National Key Research and Development Program of China (No.2020AAA0108700), National Natural Science Foundation of China (No.62022027) and CAAI-Huawei MindSpore Open Fund.

References

Sugato Basu, Arindam Banerjee, and Raymond J Mooney. 2004. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the 2004 SIAM international conference on data mining*, pages 333–344. SIAM.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149.

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

Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. 2017. Deep adaptive image clustering. In *Proceedings of the IEEE international conference on computer vision*, pages 5879–5887.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Dilek Hakkani-Tür, Asli Celikyilmaz, Larry Heck, and Gokhan Tur. 2013. A weakly-supervised approach for discovering new user intents from search query logs.

Dilek Hakkani-Tür, Yun-Cheng Ju, Geoffrey Zweig, and Gokhan Tur. 2015. Clustering novel intents in a conversational interaction system with semantic parsing. In *Sixteenth Annual Conference of the International Speech Communication Association*.

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

Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. 2017. Learning to cluster in order to transfer across domains and tasks. *arXiv preprint arXiv:1711.10125*.

Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. 2019. Multi-class classification without multi-class labels. *arXiv preprint arXiv:1901.00544*.

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*.

Ting-En Lin, Hua Xu, and Hanlei Zhang. 2020. Discovering new intents via constrained deep adaptive clustering with cluster refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8360–8367.

James MacQueen 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, pages 281–297. Oakland, CA, USA.

Srinivas Bangalore Padmasundari. 2018. Intent discovery through unsupervised semantic text clustering. *Proc. Interspeech 2018*, pages 606–610.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Xiang Shen, Yinge Sun, Yao Zhang, and Mani Najmabadi. 2021. Semi-supervised intent discovery with contrastive learning. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 120–129.

Chen Shi, Qi Chen, Lei Sha, Sujian Li, Xu Sun, Houfeng Wang, and Lintao Zhang. 2018. Auto-dialabel: Labeling dialogue data with unsupervised learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 684–689, Brussels, Belgium. Association for Computational Linguistics.

Ximei Wang, Jinghan Gao, Mingsheng Long, and Jianmin Wang. 2021. Self-tuning for data-efficient deep learning. In *International Conference on Machine Learning*, pages 10738–10748. PMLR.

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR.

Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. 2015. Short text clustering via convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 62–69.

Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, pages 3861–3870. PMLR.

Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lyu. 2021. Discovering new intents with deep aligned clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14365–14373.

Yuwei Zhang, Haode Zhang, Li-Ming Zhan, Xiao-Ming Wu, and Albert Lam. 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)*, pages 256–269, Dublin, Ireland. Association for Computational Linguistics.

Yunhua Zhou, Peiju Liu, and Xipeng Qiu. 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)*, pages 5129–5141, Dublin, Ireland. Association for Computational Linguistics.

A Statistics of Datasets

We present detailed statistics of datasets in our experiments in Table 6.

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 (Xu et al., 2015) represents a dataset dispersed through Kaggle.com, encompassing 20 intents and 20000 utterances. We adopt the dataset processed by (Xu et al., 2015).

λ	BANKING			CLINC		
	NMI	ARI	ACC	NMI	ARI	ACC
0.0	80.65	54.11	63.83	93.27	76.12	81.69
0.1	82.83	58.99	70.06	94.36	80.23	86.13
0.3	84.43	62.62	72.53	94.73	81.79	88.09
0.5	84.81	63.91	74.38	95.32	83.41	89.07
0.7	84.73	63.95	74.58	95.36	83.26	88.40
0.9	85.16	65.34	75.94	95.69	84.97	90.31
1.0	82.94	61.98	73.21	93.35	78.46	85.51

Table 5: Detailed Results about the Effect of Exploration and Utilization.

B Experiment Details

Our main experiments use pre-trained BERT (bert-uncased, with 12-layer transformer), which is implemented in the Huggingface Transformers[3]. We try learning rate in $\{1e-5, 5e-5\}$ and λ in $\{0.5, 0.6\}$. The training batch size is 512, 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.

C More Results on Effect of Exploration and Utilization

In this section, we detail the results of varying λ in the Table 5. This result can be used as a supplement to Section 5.3, which further proves that if the model wants to achieve better results, both exploration and utilization are indispensable.

D 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

[3] https://github.com/huggingface/transformers

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	18000	1000	1000	17182	9.18

Table 6: Statistics of datasets. || denotes the total number of utterances. The StackOverflow is drawn from Lin et al. (2020)

Methods	CLINC ($\hat{k}$ = 150)					BANKING ($\hat{k}$ = 77)				
	K(Pred)	Error(↓)	NMI	ARI	ACC	K(Pred)	Error(↓)	NMI	ARI	ACC
MCL(BERT)	112	25.33	87.15	59.22	69.20	58	24.68	75.33	47.35	60.80
DTC(BERT)	195	30.00	89.15	63.18	66.65	110	42.86	77.61	47.50	54.94
DeepAligned	129	14.00	92.50	72.26	77.18	67	12.99	78.88	51.71	62.49
Ours	**130**	**13.30**	**93.58**	**75.30**	**80.80**	**73**	**5.48**	**83.56**	**60.92**	**69.68**

Table 7: The results of predicting K. The $\hat{k}$ denotes the ground truth number of K.The compared results are retrieved from (Zhang et al., 2021).

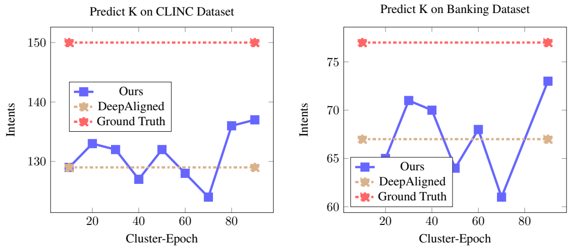

Figure 7: The results of predicting K on CLINC and BANKING. The Red line denotes ground truth, the Brown line denotes the result of DeepAligned and the Bule line denotes the refinement of K by our method.

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 the EM algorithm, we refine K per 10 epochs using the method as suggested in Section 3.3. To effectively demonstrate the impact and efficiency of our proposed framework on the estimation of K, we did not consider dataset $\hat{D}^l$ in the experiment. We get the final performance of the model and the results are shown in Table 7 (Figure 7 shows the intermediate values of K per epoch.) shows that our method can predict the number of intents more accurately and achieve better results at the same time. During the experiment, we observed that the performance of model exhibited fluctuations attributed to the setting of hyperparameters. A more comprehensive and in-depth investigation of the estimation of K will be left for future research endeavors.

E Effect of $\hat{D}^l$ discovered in unlabeled data

In addition to the labeled data in hand, in Section 3.3, we also use the sample set $\hat{D}^l$ predicted known intents in unlabeled data during discovery (See Section 3.3 for the specific construction of $\hat{D}^l$. The nearest neighbor measure is based on the cosine similarity of the sample representation in the semantic space). In this section, we will further analyze the benefits brought by these discovered sample set. We have compared the effects of adding $\hat{D}^l$ and not adding $\hat{D}^l$, and the comparison results are shown in Table 8. From Table 8, we can easily conclude that the added sample set $\hat{D}^l$ can improve the effectiveness. This also proves the importance of exploring the intrinsic structure of unlabeled data, which can not only improve the effect of preventing knowledge forgetting (Section 5.6) to improve the identification of IND, but also improve the effect of intent discovery, which is completely in line with our expectations.

Methods	CLINC			BANKING			Stackoverflow		
	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC
Our (D^l)	94.78	82.32	88.29	83.40	61.19	72.59	77.29	63.93	**80.90**
+$\hat{D}^l$	**95.01**	**83.00**	**88.99**	**84.02**	**62.92**	**74.03**	**77.32**	**65.70**	80.50

Table 8: Ablation study on effect of $\hat{D}^l$. $\hat{D}^l$ is the set of samples in D^u that can be considered as known intents after the operation of $\mathcal{Z}$.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section "Limitations" (7th Section)

☑ A2. Did you discuss any potential risks of your work?
Section "Limitations" (7th Section)

☑ A3. Do the abstract and introduction summarize the paper's main claims?
"Abstract" and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4.1 and Appendix A

☑ B1. Did you cite the creators of artifacts you used?
Section 4.1

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
These datasets are available for all researchers in the NLP community.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
These datasets are only for scientific research and are available for all members of the NLP research community. We have adhered to the typical method of utilizing these resources.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
These datasets are only for scientific research and are available for all members of the NLP research community. We have adhered to the typical method of utilizing these resources.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4.1

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix A

C ☑ Did you run computational experiments?

Section 4 and Section 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix B

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4.3 and Appendix B

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4.3 and Section 5.1

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix B

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.