TeC: A Novel Method for Text Clustering with Large Language Models Guidance and Weakly-Supervised Contrastive Learning

Chen Yang, Bin Cao, Jing Fan *

College of Computer Science
Zhejiang University of Technology, Hangzhou, China
{yangchen, bincao, fanjing}@zjut.edu.cn

Abstract

Text clustering has become an important branch in unsupervised learning methods and has been widely used in social media. Recently, Large Language Models (LLMs) represent a significant advancement in the field of AI. Therefore, some works have been dedicated to improving the clustering performance of embedding models with feedback from LLMs. However, current approaches hardly take into consideration the cluster label information between text instances when fine-tuning embedding models, leading to the problem of cluster collision. To tackle this issue, this paper proposes TeC, a novel method operating through teaching and correcting phases. In these phases, LLMs take on the role of teachers, guiding embedding models as students to enhance their clustering performance. The teaching phase imparts guidance on cluster label information to embedding models by querying LLMs in a batch-wise manner and utilizes a proposed weakly-supervised contrastive learning loss to fine-tune embedding models based on the provided cluster label information. Subsequently, the correcting phase refines clustering outcomes obtained by the teaching phase by instructing LLMs to correct cluster assignments of low-confidence samples. The extensive experimental evaluation of six text datasets across three different clustering tasks shows the superior performance of our proposed method over existing state-of-the-art approaches.

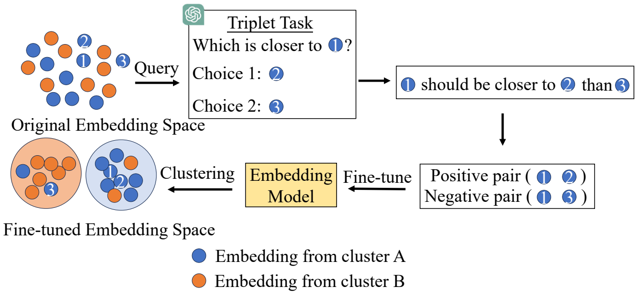


Figure 1: Illustration of cluster collision. Since the cluster label information is not taken into account when fine-tuning the model, some samples from the same semantic cluster are misassigned to other clusters.

Introduction

Text clustering finds diverse applications in social media networks, including content recommendation (Shepitsen et al. 2008), topic discovery (Yin et al. 2011), and user profiling (Tang et al. 2010). In the realm of text clustering, a prevalent practice is to deploy a classical clustering model, e.g. K-Means (MacQueen 1965; Steinbach, Karypis, and Kumar 2000), directly on the representations generated by embedding models (Muennighoff et al. 2023; Wang et al. 2022; Su et al. 2023). However, these methods are not explored in the clustering process and the complex relations among instances are often overlooked, leading to sub-optimal clustering results (Zhou et al. 2022).

Recent instruction-tuned large language models (LLMs) such as ChatGPT, have been shown to have the ability to reproduce or improve human-generated labels (Gilardi, Alizadeh, and Kubli 2023; He et al. 2023). Furthermore, several works (Zhang, Wang, and Shang 2023; Cheng et al. 2023) have been dedicated to improve clustering performance of embedding models with feedbacks from LLMs. A prevalent approach involves prompting LLMs with a triplet task that predicts which one of the two candidates is closer to the anchor instance. Then an embedding model is fine-tuned based on the triplet relationships with contrastive learning (Su et al. 2023). However, these methods disregard cluster label information between text instances, focusing exclusively on relationships between text pairs. As shown in Fig.1, this exclusive focus may result in cluster collision, where different instances from the same semantic cluster are treated as negative pairs and incorrectly pushed away, adversely affecting clustering outcomes.

To conquer the aforementioned limitation, this paper proposes a novel method, TeC. It operates through two distinct phases: the *teaching* phase and the *correcting* phase, where LLMs serve as the role of teachers and embedding models act as students throughout the entire process. During the teaching phase, LLMs provide guidance to embedding models to improve clustering performance with the prompts. Subsequently, in the correcting phase, LLMs refine the clustering outcomes obtained by embedding models in the teaching phase. This two-phase approach leverages the instructive capabilities of LLMs to enhance the clustering performance

*Corresponding author

of embedding models.

Specifically, in the teaching phase, our objective is for LLMs to impart guidance to embedding models about the cluster label information. However, due to the constraints of maximum input length of LLMs, processing the entire dataset in a single query is impractical. Even when dividing the dataset into batches and querying LLMs in a batch-wise manner, it does not yield the final clustering result for the complete dataset. Consequently, we propose a weakly-supervised contrastive learning loss to fine-tune embedding models with feedback from LLMs in batch-wise query. Firstly, a randomly sampled minibatch serves as input to prompt LLMs for clustering. Subsequently, the embedding models are fine-tuned based on the clustering results provided by LLMs. This iterative refinement process aims to enhance the clustering performance of embedding models with the instructive guidance received from LLMs in each batch-wise operation. Additionally, in the correcting phase, building upon the clustering results produced by the embedding models during the teaching phase, we instruct LLMs to correct cluster assignments of low-confidence samples, which can further boost the clustering performance.

In summary, the major contributions of this work are summarized as follows:

- Leveraging LLMs, we propose a weakly-supervised contrastive learning loss to inject cluster label information into embedding models to improve its clustering performance;
- To rectify cluster assignments, we adopt a confidence-based criterion to identify low-confidence samples. Experiments show that such a strategy could further boost the clustering performance;
- We conduct extensive experiments on 6 text datasets across 3 different clustering tasks to demonstrate the effectiveness of the proposed method.

Related Work

In this section, we briefly introduce some recent developments in three related topics, namely, text clustering, text embedding models, and LLMs as Annotators.

Text Clustering

Most existing text clustering methods involve text embedding followed by clustering algorithms, leveraging techniques such as bag-of-words (Blei, Ng, and Jordan 2003), tf-idf (Aggarwal and Zhai 2012), or pre-trained models like BERT (Kenton and Toutanova 2019), RoBERTa (Liu et al. 2019), Sentence-BERT (Reimers 2019), Whitening-BERT (Huang et al. 2021), and GPT-3 (Brown et al. 2020). However, these methods often neglect the intricate relations among instances, leading to sub-optimal clustering outcomes (Zhou et al. 2022). In response to these limitations, the emergence of deep text clustering (Zhang et al. 2021a; Xu et al. 2015; Hadifar et al. 2019) seeks to jointly optimize deep representation learning and clustering, garnering increased attention recently. Nevertheless, these methodologies heavily depend on self-supervised labels, potentially introducing noise and negatively impacting final clustering outcomes.

Recent advancements in Large Language Model-based methods have sought to enhance clustering efficacy. ClusterLLM (Zhang, Wang, and Shang 2023) employs LLMs to infer sentence relationships, providing guidance for clustering results. Notably, this approach overlooks cluster information, focusing solely on relationships between text instances. Additionally, Wang, Shang, and Zhong employs LLMs for clustering by assigning instances to different explanations. (Viswanathan et al. 2023) generate keyphrases with LLMs to facilitate semi-supervised clustering.

Text Embedding Models

Text embedding models (Kenton and Toutanova 2019; Liu et al. 2019; Brown et al. 2020; Huang et al. 2021; Reimers 2019) measure the relatedness of text instances. these models find applications in retrieval (Xiao et al. 2022), text similarity (Gao, Yao, and Chen 2021), and classification (Gunel et al. 2021), among other tasks. Recently, two text embedding models, E5 (Wang et al. 2022) and Instructor (Su et al. 2023), have demonstrated superior performance compared to earlier models. Specifically, E5 trains high-quality embeddings through self-supervised pre-training exclusively on web-scraped data pairs. Instructor(Su et al. 2023) annotates instructions for a diverse set of 330 tasks, training on this multitask mixture with a contrastive loss to generate embeddings based on both text input and task input. Our method enhances clustering performance on these models with the assistance of LLMs.

LLMs as Annotators

Creating human-annotated data is a labor-intensive and expensive process, especially for complex tasks or specialized domains where sufficient data may be lacking. Recent research has explored the potential of LLMs to serve as an annotator for textual data, offering insights into various NLP tasks. Examples include the use of ChatGPT for annotating misinformation (Bang et al. 2023) and hate speech (Huang, Kwak, and An 2023). LLMs have been demonstrated the ability to reproduce or enhance human-generated labels (Gilardi, Alizadeh, and Kubli 2023; He et al. 2023). Additionally, there have been efforts to fine-tune models with feedback from LLMs (Cheng et al. 2023; Zhang, Wang, and Shang 2023; Bai et al. 2022). In our work, we specifically focus on clustering tasks and explore how predictions from LLMs regarding cluster information can be leveraged to enhance the clustering quality of embedding models and rectify clustering assignments for low-confidence samples.

Methodology

In this section, we present our method for Large Language Model (LLM)-based clustering. As illustrated in Fig.2, our method is composed of two stages: the $teaching$ phase and the $correcting$ phase. Within the teaching phase, we propose weakly supervised contrastive learning loss to fine-tune the Language Model (LM) with the collaborative assistance of LLM. Subsequently, In the phase of correcting, based on the clustering results obtained by $teaching$, we identify the low-confidence texts first, and then LLM is utilized to rectify these texts into correct clusters, thereby refining the overall clustering results. Next, we will provide detailed explanations for each of the two components.

Teaching

In this section, we explore how to use LLMs to cluster. Acknowledging the limitations imposed by the maximum input length of LLMs, we recognize the incapacity of LLMs to handle the entire dataset $X = \{x_i\}_{i=1}^{N}$ within a single query, particularly when the cumulative length exceeds the specified maximum input limit. Even though dividing the dataset into batches and querying the LLMs in a batch-wise manner does not yield the final clustering result for the entire dataset. Consequently, we advocate a batch-wise fine-tuning embedding models approach, where a randomly sampled minibatch $\mathcal{B} = \{x_i\}_{i=1}^{M}$ is presented as input to LLMs for clustering. LLMs are prompted to cluster the texts within the batch using a designated prompt $\mathcal{P}$. Moreover, clustering results usually are highly related to the user's goal, which can be clustered based on topic, sentiment, genre, or other properties (Aharoni and Goldberg 2020). Hence, rather than forcing the clustering algorithm to mine these key factors from scratch, it is better to highlight these aspects globally beforehand and thus focus on task priorities. To do this, LLMs are employed to generate clustering results aligned with user-defined goals $\mathcal{G}$ and descriptions of dataset $\mathcal{D}$. Thus the LLM prediction process is,

$$\Psi = \mathcal{P}(\mathcal{B}, \mathcal{G}, \mathcal{D}) \tag{1}$$

where $\Psi = \{y_i\}_{i=1}^{M}$ indicates cluster label corresponding to the texts in batch returned by LLMs. The prompt $\mathcal{P}$ is as follows:

You are now an excellent algorithm expert for clustering. {the description of dataset $\mathcal{D}$ }, in which each row represents an text instance, including id and text. Please cluster the following text instance based on {user's goals $\mathcal{G}$} in text instance. Please don't leave out any instances. Output the result in JSON format. Do not provide any additional information except the JSON, like "ids":"type", where "ids" indicates the id corresponding to all texts contained in this cluster, i.e. 0,1,2, and "type" represents the type of this cluster. Let's think step by step.

{*Example:*}

{*Input: $\mathcal{B}$*}

Now that we have accurate cluster labels of the texts in batch, the subsequent challenge lies in effectively incorporating them into the clustering process. In this paper, we focus on fine-tuning the base embedding model f with the aid of Ψ in order to produce an embedding space that aligns with the user's perspective. To mitigate the embedding model being biased towards hard examples after fine-tuning, we need to contrast the set of all samples from the same cluster as positives against the negatives from the remainder of the batch. Specifically, we propose a weakly-supervised contrastive learning loss. This loss incorporates pseudo-labels Ψ derived from LLMs. Hence, we optimize the following training objective,

$$\mathcal{L} = \frac{1}{M}\sum_{i\in\mathcal{B}} -\log\left\{\frac{1}{|P(i)|}\sum_{p\in P(i)}\frac{\exp\left(\boldsymbol{z}_i\cdot\boldsymbol{z}_p/\tau\right)}{\sum_{a\in A(i)}\exp\left(\boldsymbol{z}_i\cdot\boldsymbol{z}_a/\tau\right)}\right\} \tag{2}$$

Here, $A(i) = \{1,\ldots,M\}$ is the set of all instances in a batch, $P(i) = \{p \in A(i) : y_i = y_p\}$ is the set that belongs to the same cluster as sample i in $\mathcal{B}$, τ is the temperature hyper-parameter that allows the model to learn better difficult samples, and $\boldsymbol{z}$ is the vector representation of text. Finally, fine-tuned embedding models can be applied to generate the final clustering assignment $\mathcal{Y} = \{y_i\}_{i=1}^{N}$ of the dataset with a clustering algorithm.

Correcting

The correcting stage aims to discern instances within the sample dataset characterized by low-confidence attributes, subsequently rectifying these instances to their appropriate clustering assignments. Given clustering assignment $\mathcal{Y}$, suppose our data consists of K semantic categories, and each category is characterized by its centroid in the representation space. The cluster center $\boldsymbol{\mu}_k$ for cluster C_k is computed by averaging the embeddings assigned to it:

$$\boldsymbol{\mu}_k = \frac{1}{|C_k|}\sum_{i\in C_k}\boldsymbol{z}_i \tag{3}$$

Then, the cluster assignment posterior probability p_{ik}, indicating the likelihood of sample i belonging to cluster C_k, is computed as follows:

$$p_{ik} = \frac{\left(\|\boldsymbol{z}_i - \boldsymbol{\mu}_k\|^2\right)^{-1}}{\sum_{k'}\left(\|\boldsymbol{z}_i - \boldsymbol{\mu}_{k'}\|^2\right)^{-1}} \tag{4}$$

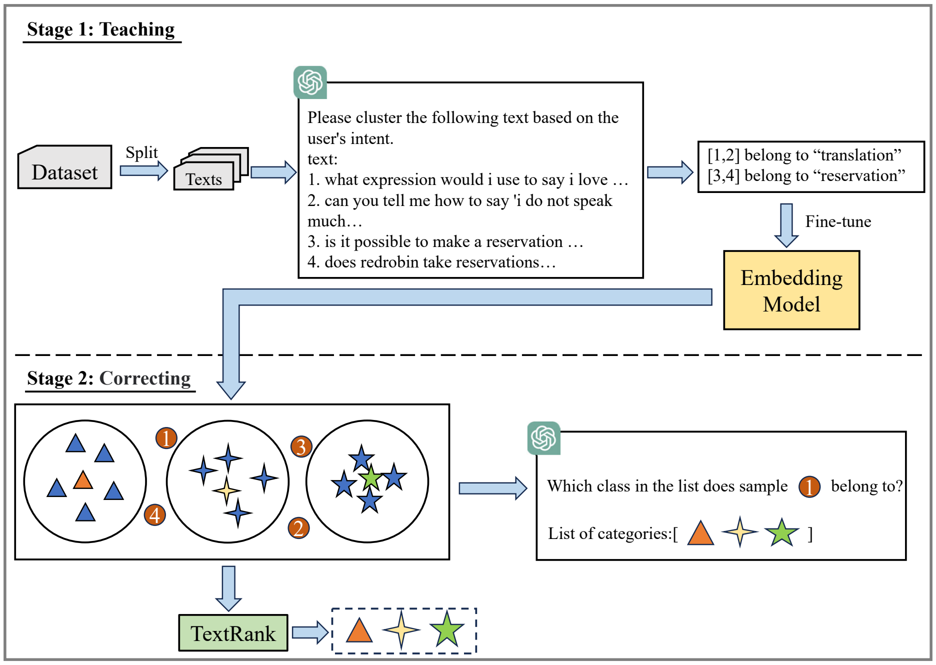
Hence, a probability matrix $\mathcal{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N\}^T \in \mathbb{R}^{N*K}$ is derived.

A low-confidence sample denotes a data point where the model exhibits diminished certainty or confidence in its predictions (DeVries and Taylor 2018). Formally, the low-confidence samples are selected in two ways:

$$\begin{aligned}\aleph_{unselected} = \{x_i \mid i \notin \bigcup\{&\text{argtopk}(\boldsymbol{P}_{:,1}, \frac{N}{K}), \ldots, \\ &\text{argtopk}(\boldsymbol{P}_{:,K}, \frac{N}{K})\}, \forall i = 1, 2, \ldots, N\}\end{aligned} \tag{5}$$

$$\begin{aligned}\aleph_{overlapped} = \{x_i \mid i \in (&\text{argtopk}(\boldsymbol{P}_{:,k}, \frac{N}{K}) \bigcap \\ &\text{argtopk}(\boldsymbol{P}_{:,j}, \frac{N}{K}), \forall i = 1, 2, \ldots, N\}\end{aligned} \tag{6}$$

where, $\boldsymbol{P}_{:,k}$ denotes the k-th column of matrix $\boldsymbol{P}$ and $\text{argtopk}(\boldsymbol{P}_{:,k}, \frac{N}{K})$ yields the top N/K confident sample indices from $\boldsymbol{P}_{:,k}$. Consequently, low-confidence samples encompass those unselected in all clusters. Table. 1 visually illustrates the process of selecting low-confidence samples, where 7 samples are distributed across 3 clusters (A, B, C). The top 2 confident samples for each cluster are selected and



Please cluster the following text based on the user's intent.
text:
1. what expression would i use to say i love …
2. can you tell me how to say 'i do not speak much…
3. is it possible to make a reservation …
4. does redrobin take reservations…

[1,2] belong to "translation"
[3,4] belong to "reservation"

Which class in the list does sample 1 belong to?

List of categories:[ ]

Figure 2: The method framework of the proposed TeC. In the correcting stage, the brown sample points are regarded as low-confidence samples. Moreover, orange, yellow, and green samples are selected by TextRank to represent every cluster, respectively.

	A	B	C	Label
1	0.89	0.10	0.01	A
2	0.70	0.20	0.10	A
3	0.30	0.22	0.48	-1
4	0.02	0.96	0.02	B
5	0.00	0.40	0.60	B C
6	0.05	0.05	0.90	C
7	0.30	0.30	0.40	-1

Table 1: Illustration for low-confidence samples, where 1,2,3, etc, indicate the index of the samples and A, B, and C indicate the index of the clusters.

labeled according to the predicted probabilities. According to the Eq. 5, the samples that are not selected are those with low confidence, i.e. labeled as -1. Moreover, there may exist overlapped samples between different clusters. As shown in Table. 1, sample 5 belongs to both cluster B and cluster C. Here, we also recognize these overlapped samples as low-confidence samples according to the Eq.6.

$$\aleph = \aleph_{unselected} \bigcup \aleph_{overlapped} \tag{7}$$

In summary, low-confidence samples constitute the union of sets $\aleph_{unselected}$ and $\aleph_{overlapped}$ as defined in Eq 7.

After selecting the low-confidence samples, each cluster is textually represented using TextRank (Mihalcea and Tarau 2004) to identify the most representative sentence r in the cluster. For each low-confidence point q, the LLMs are tasked with selecting the semantically closest cluster among the m closest clusters, including the original cluster:

$$\mathcal{P}(\mathcal{G}, q, \mathcal{R}) \tag{8}$$

where, $\mathcal{P}$ is prompt of correcting phase and $\mathcal{R} = \{r_i\}_{i=1}^{m}$. The prompt is as follows:

Select the user utterance that better corresponds with the Query in terms of $\{\mathcal{G}\}$*. Each row represents an instance, including id and text. Please respond with 'id' that better corresponds with the Query in terms of* $\{\mathcal{G}\}$ *without explanation.*

Query: $\{q\}$

$\{\mathcal{R}\}$

Experiment

In this section, we conduct experiments to verify the effectiveness of the proposed TeC.

Datasets

In the experiments, we assess our method on a diverse set of clustering tasks and datasets, including various perspectives and granularities. Table 2 provides a comprehensive overview of the main statistics. To mitigate the costs associated with LLMs guidance, we only conduct main experiments on small-scale datasets provided by (Zhang, Wang, and Shang 2023).

Intent Discovery. Intent discovery is the task of inferring latent intents from a set of unlabeled utterances (Zhang, Xu, and Lin 2021). We conduct experiments on three challenging real-world datasets to evaluate our approach, including CLINC (Larson et al. 2019), MTOP (Li et al. 2021) and Massive (FitzGerald et al. 2023).

Information Extraction (IE). Information extraction is the task of automatically extracting structured information from unstructured and/or semi-structured documents (Mausam et al. 2012). When dealing with fine-grained labels (Choi et al. 2018), the exploration of new labels and the expansion of supported information types become especially crucial. In our paper, we focus on two tasks of IE, including entity type recognition and event type detection. We adapt FewNerd (Ding et al. 2021) and FewEvent (Deng et al. 2020) to evaluate our approach.

Sentiment Analysis. Sentiment analysis is the process of analyzing text to determine the emotional tone of the message (Kenyon-Dean et al. 2018). We adapt Goemo (Demszky et al. 2020), a fine-grained emotion detection dataset for evaluation.

Task	Dataset	#Text	#Clusters	#L/S
Intent	CLINC	4500	150	1
	MTOP	4386	102	473
	Massive	2974	59	209
IE	FewNerd	3789	58	113.8
	FewEvent	4742	34	32.3
Emotion	GoEmo	5940	27	84.25

Table 2: A summary of datasets used for evaluations. #Text: the number of texts; #Cluster: the number of clusters; L/S: the ratio of the size of the largest cluster to that of the smallest cluster.

Baselines

K-Means on Embeddings. We directly apply (mini-batch) K-means (MacQueen 1965) on top of extracted embeddings from E5 (Wang et al. 2022) and Instructor (Su et al. 2023) in a zero-shot manner. For the Instructor, we use the same prompts provided by the original paper.

Contrast. Contrast (Vaze et al. 2022) is a method used in computer vision to detect unknown classes of images and we adapt this algorithm to natural language processing for our problem settings.

CLNN. CLNN (Zhang et al. 2022) proposes a pre-training strategy for multi-task learning, which leverages unlabeled and labeled data for better representation learning. Then, a contrast loss is designed to take advantage of the self-supervised signals in the unlabeled data for better clustering.

DAC. DAC (Zhang et al. 2021b) proposes an iterative clustering method to obtain pseudo-cluster labels by K-means. It performs representation learning and cluster assignment in a pipeline way.

DPN. DPN (An et al. 2023) proposes decoupling known and new intents from unlabeled data to acquire different knowledge for capturing high-level semantics.

SCCL-I. We also adopt deep text clustering algorithm SCCL (Zhang et al. 2021a) equipped with Instructor for comparison. It jointly optimizes a top-down clustering loss with a bottom-up instance-wise contrastive loss, where cluster loss follows the approach proposed by (Hadifar et al. 2019) and contrastive loss follows the approach proposed by (Gao, Yao, and Chen 2021). We use the same prompts provided by the Instructor.

CLUSTER-LLM. CLUSTER-LLM (Zhang, Wang, and Shang 2023) proposes to prompt LLMs such as ChatGPT, with a triplet task that predicts which one of the two candidates is closer to the anchor instance. The predicted triplets thereafter are used to fine-tune a small embedding model with bi-directional in-batch sampled softmax loss. CLUSTERLLM-I adopts Instructor as its embedding model. CLUSTERLLM-I-iter applies the entire framework in an iterative manner twice.

Experimental Details

In the teaching phase, the prompts for querying LLMs only contain a task-specific instruction. For all experiments, we use a temperature of 0.2 and top_p of 0.9 with gpt-3.5-$turbo$. We use the Python API tool provided by OpenAI. In our work, we focus on a state-of-the-art embedding model: Instructor (Su et al. 2023) provided by Hugging Face. We adopt the same hyper-parameters as in Instructor. We use the Adam optimizer (Kingma and Ba 2015) with a learning rate of 8e-5. The training batch size is 64 and the temperature parameter τ is set to 0.1 in our proposed weakly-supervised contrastive loss. Moreover, we use K-Means (MacQueen 1965) to obtain clustering results on fine-tuned Instructor. The descriptions of $\mathcal{D}$ and user-defined goals $\mathcal{G}$ can be seen in Table 3. In the correcting phase, we select the top 3 sentences with the highest TextRank scores in each cluster to represent this cluster and the number m of nearest clusters is set to 5. Moreover, 200 samples are selected to rectify its clustering assignments in the correcting phase. The experiments are carried out on two NVIDIA RTX 4090 GPUs.

Evaluation Metric

We follow the previous work (Zhang, Wang, and Shang 2023), using a common clustering performance metric to evaluate our method, i.e., Accuracy (ACC) (Wu 2006). The ACC ranges between 0 and 1. A larger ACC indicates a better clustering result. ACC is computed as follows:

$$\mathrm{ACC} = \frac{\sum_{i=1}^{N} \delta\left(y_i, map\left(c_i\right)\right)}{N} \tag{9}$$

where y_i is the true cluster label, c_i is the cluster label obtained by clustering, and $\delta(x, y)$ is an indicator function returning 0 ($x \neq y$) or 1 ($x = y$). $map(\cdot)$ transforms the cluster label c_i to its true cluster label by the Hungarian algorithm (Papadimitriou and Steiglitz 1998).

Dataset	$\mathcal{D}$	$\mathcal{G}$
CLINC	Here is a fine-grained dataset in the customer service query domain	the customer service query intent
MTOP	Here is a fine-grained intent dataset in the customer service query domain	the customer service query intent
Massive	Here is a intent dataset in the user utterance query domain	the user utterance query intent
FewNerd	Here is a named entity recognition dataset	the entity type expressed in text instance
FewEvent	Here is a event detection dataset	the event type expressed in event trigger of text instance
GoEmo	Here is a fine-grained emotion detection dataset	the emotion expressed in text instance

Table 3: The prompt template of description of datasets and user's goal in teaching phase.

	Method	**Intent Discovery**			**Information Extraction**		**Emotion**	**Avg**
		MTOP	CLINC	Massive	FewNerd	FewEvent	GoEmo	
Few-shot	Contrast	29.25	34.68	33.07	30.42	47.61	16.34	31.90
	DAC	31.43	61.84	34.45	40.84	33.22	18.88	36.78
	DPN	33.64	45.56	33.86	38.13	43.43	14.86	34.91
	CLNN	29.77	75.64	46.22	40.59	28.05	20.01	40.05
	TeC (Ours)	**39.25**	**84.15**	**62.96**	**43.65**	**55.51**	**35.62**	**53.52**
Zero-shot	E5	33.54	75.83	52.52	25.49	37.30	22.13	41.14
	Instructor	33.35	79.29	54.08	30.02	41.99	25.19	43.99
	SCCL-I	34.28	80.85	54.05	31.09	39.97	34.33	45.76
	ClusterLLM-I	35.84	82.77	59.89	34.75	46.17	27.49	47.82
	ClusterLLM-I-iter	35.04	83.80	60.69	40.60	50.60	26.75	49.58
	TeC (Ours)	**37.02**	**83.92**	**60.86**	**43.36**	**53.40**	**34.63**	**52.20**

Table 4: The clustering performance on 6 benchmarks accross 3 clustering tasks. The ”Few-shot” indicates experiments are conducted with 16-way 8-shot labels.

Comparisons with State of the Arts

The clustering results across six benchmarks and three distinct clustering tasks are summarized in Table 4. In this study, we contrast our proposed method, TeC, with nine baseline approaches. The term ”Few-shot” denotes experiments conducted with 16-way 8-shot labels, where we randomly select such labels in original datasets to fill the *Example:* section of our prompt. Conversely, ”Zero-shot” signifies an empty *Example:* section. Our observations are as follows: Firstly, our method outperforms the baselines in the Zero-shot setting, suggesting that large language models (LLMs) enhance our approach, and our training strategy is crucial for improving clustering performance. Secondly, the Few-shot setting outperforms the Zero-shot, indicating that Few-shot is advantageous in bolstering the instructive capacity of LLMs. Finally, our method, TeC, surpasses all state-of-the-art baselines across all datasets, demonstrating its robust and powerful clustering capabilities.

Ablation Study on Teaching Stage

In this section, we present ablation studies on the teaching stage of TeC based on the Instructor model to showcase the effectiveness of the weakly-supervised contrastive learning loss we proposed and prompt.

Effect of the proposed loss To verify the effectiveness the weakly-supervised contrastive learning loss, we conduct a set of ablation experiments on six experimental datasets. Specifically, as shown in Table 5, we initially employ the same training objective as (Su et al. 2023) to fine-tune Instructor. This process utilizes positive pairs generated from independently sampled dropout masks (Gao, Yao, and Chen 2021), denoted as ”Self-supervised.” Furthermore, ClusterLLM-I-iter indicates the same training objective as (Su et al. 2023) is used to fine-tune Instructor, using positive pairs from a triplet task that predicts which one of the two candidates is closer to the anchor instance. Additionally, we use supervised contrastive learn-

Method	Intent Discovery			Information Extraction		Emotion
	MTOP	CLINC	Massive	FewNerd	FewEvent	GoEmo
Instructor	33.35	79.29	54.08	30.02	41.99	25.19
Self-supervised	34.10	80.13	55.21	31.25	43.85	24.08
ClusterLLM-I-iter	35.04	83.80	60.69	40.60	50.60	26.75
Zero-shot-TeC w/o correcting	**36.01**	**83.72**	**60.66**	**43.41**	**53.50**	**34.51**
Few-shot-TeC w/o correcting	**38.15**	**84.01**	**61.81**	**43.75**	**55.65**	**34.62**
Supervised	51.14	85.27	68.29	66.93	84.52	66.31

Table 5: The effect of weakly supervised contrastive learning loss on 6 benchmarks accross 3 clustering tasks.

Method	Intent Discovery			Information Extraction		Emotion
	MTOP	CLINC	Massive	FewNerd	FewEvent	GoEmo
Zero-shot-TeC w/o $\mathcal{G}$	34.01	80.10	57.21	40.25	51.85	32.17
Zero-shot-TeC w/o $\mathcal{D}$	35.84	82.60	60.09	42.60	52.68	33.76
Zero-shot-TeC w/o correcting	**36.01**	**83.72**	**60.66**	**43.41**	**53.50**	**34.51**
Few-shot-TeC w/o $\mathcal{G}$	37.10	83.13	60.21	42.32	52.08	32.68
Few-shot-TeC w/o $\mathcal{D}$	38.04	83.80	60.69	42.65	53.64	34.05
Few-shot-TeC w/o correcting	**38.15**	**84.01**	**61.81**	**43.75**	**55.65**	**34.62**

Table 6: The effect of different components in prompt on 6 benchmarks accross 3 clustering tasks in the teaching stage.

ing (Gunel et al. 2021) to fine-tune Instructor, denoted as ”Supervised”, where positive pairs are from the same class, to provide a performance upper bound. We can observe that Self-supervised increases the performance of Instructor, which demonstrates the significance of further fine-tuning on experimental datasets. Moreover, the performance ClusterLLM-I-iter surpasses that of Self-supervised, highlighting the importance of positive and hard negative samples in the training process. However, the performance of Self-supervised and ClusterLLM-I-iter are lower than our method, which indicates the critical role of category information in advancing clustering performance. Finally, when the model is supplied with human labels, i.e. gold category information, it attains the highest clustering performance that demonstrates further the importance of category information.

Effect of different components in prompt To assess the impact of different prompt components in the teaching stage, we conduct ablation studies on six benchmarks, as shown in the Table 6. These experiments aim to scrutinize the contribution of users' goals ($\mathcal{G}$) and the dataset description ($\mathcal{D}$) in both Zero-shot and Few-shot settings. Specifically, we systematically remove the sentence *’{user’s goals* $\mathcal{G}$*}’* and *’{the description of dataset* $\mathcal{D}$ *}’* from the original prompt in separate experiments. Obviously, regardless of the zero-shot or few-shot settings, a significant decline in clustering performance is observed when either users' goals or the dataset description is omitted from the prompt. Furthermore, the experiments indicate that users' goals ($\mathcal{G}$) contribute more significantly to the enhancement of clustering performance compared to the dataset description ($\mathcal{D}$). Additionally, the Few-shot setting demonstrates greater efficacy when querying LLMs.

Ablation Study on Correcting Stage

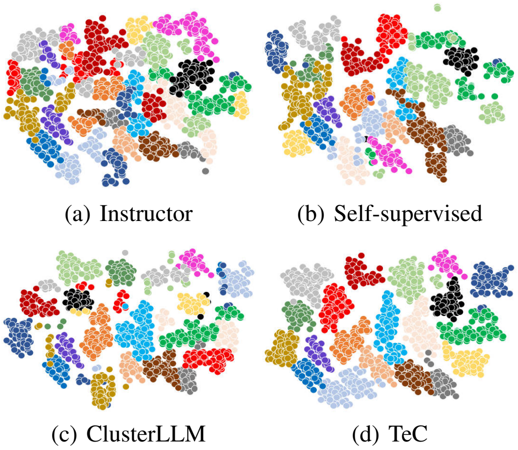
In this section, we conduct ablation studies on the correcting stage of TeC, utilizing the Instructor model to evaluate the effectiveness of our proposed method for selecting low-confidence samples. The results are presented in Table 7. We initially select some samples randomly (referred to as TeC $random$) to query LLMs. It is observed that the clustering performance of TeC $random$ is inferior to our method, indicating the validity of the proposed approach for selecting low-confidence samples. Furthermore, a decline in performance is noted on FewNerd and FewEvent after correction. Upon investigation, it is found that the top 3 sentences selected by TextRank in some clusters erroneously include sentences that should belong to other clusters. This may contribute to LLMs failing to correct samples with low confidence. Nevertheless, the overall performance of our method surpasses that of other approaches. Therefore, we assert that our method for selecting low-confidence samples remains effective.

Qualitative Study

In this section, we deeply analyze the impact of different training objectives on the CLINC dataset embedding space based on the instructor model. By improving the supervision

Method		Intent Discovery			Information Extraction		Emotion
		MTOP	CLINC	Massive	FewNerd	FewEvent	GoEmo
Few-shot	TeC w/o correcting	38.15	84.01	61.81	**43.75**	**55.65**	34.62
	TeC *random*	37.23	83.74	60.81	43.41	53.62	34.62
	TeC w/o $\aleph_{unselected}$	37.43	83.84	60.89	43.45	53.63	34.68
	TeC w/o $\aleph_{overlapped}$	37.64	83.90	60.86	43.48	53.53	34.68
	TeC	**39.25**	**84.15**	**62.96**	43.65	55.51	**35.62**
Zero-shot	TeC w/o correcting	36.01	83.72	60.66	**43.41**	**53.50**	34.51
	TeC *random*	37.01	83.74	60.75	43.41	53.48	34.58
	TeC w/o $\aleph_{unselected}$	37.55	83.82	60.79	43.42	53.50	34.59
	TeC w/o $\aleph_{overlapped}$	37.88	83.85	60.80	43.39	53.47	34.60
	TeC	**37.02**	**83.92**	**60.86**	43.36	53.40	**34.63**

Table 7: The effect of correcting stage on 6 benchmarks accross 3 clustering tasks.



(a) Instructor

(b) Self-supervised

(c) ClusterLLM

(d) TeC

Figure 3: The visualization of embedding space of CLINC dataset on different training objectives.

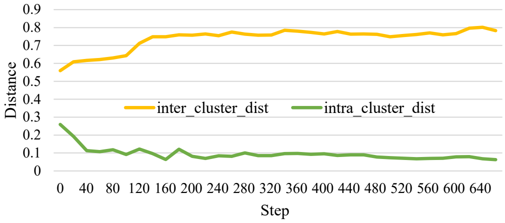


Figure 4: Intra- and inter-cluster distance across the training process on CLINC dataset.

signal of traning objectives, the model should learn discriminative representations and yield improved clustering outcomes. To see how our model converges to the goal, we compare the outcomes of using the same training objective as (Su et al. 2023) (referred to as Self-supervised), triplet relationships (Zhang, Wang, and Shang 2023) (referred to as ClusterLLM) to fine-tune Instructor. We randomly select 20 classes from the CLINC dataset. Obviously, as the enhancement of the supervision signal, clustering results become more reasonable, and the feature representations become more dispersed, forming more and well-defined clusters. This phenomenon indicates class label information is more important when fune-tuning model.

Moreover, we provide a visualization that illustrates the evolution of both intra-cluster and inter-cluster distances with respect to the training step on the CLINC dataset. Ideally, a successful clustering outcome exhibits a low intra-cluster distance, indicating tight cohesion within clusters, and a high inter-cluster distance, reflecting clear separation between clusters. Specifically, for a given cluster, the intra-cluster distance represents the average distance between the cluster's centroid and all the samples assigned to it. Conversely, the inter-cluster distance measures the separation between a cluster and its nearest neighboring cluster. In Figure 4, we present the mean values of both types of distances, computed by averaging across all clusters. Notably, the results demonstrate that as the training progresses, the inter-cluster distance gradually increases, while the intra-cluster distance decreases. This trend underscores the effectiveness of our method in compactly grouping similar samples within each cluster while effectively differentiating distinct clusters.

The Cost and Time of TeC

Our findings have demonstrated that incorporating LLMs to guide the clustering process yields improvements in performance. However, it is essential to acknowledge that utilizing LLMs can incur significant expenses. Employing a commercial LLM API during clustering introduces additional costs to the overall process. Table 8 provides a comprehensive summary of the cost associated with using the OpenAI API on six benchmarks. The average cost for Zero-shot TeC is \$0.64 per dataset, while that of Few-shot TeC is \$2.03. Consequently, we assert that, despite the expenses, our method remains cost-effective, considering the observed enhancements in clustering performance. Moreover, we assess the efficiency of our proposed method by measuring the training times across six benchmark datasets in Table 9. Notably, the zero-shot learning approach offers quicker training times and broader applicability without the need for task-specific data, while the few-shot learning approach, despite requiring longer training, potentially enhances the model's clustering accuracy and adaptability by utilizing a small set of relevant examples.

Dataset	Zeroshot-TeC	Fewshot-TeC
MTOP	\$0.33	\$0.97
CLINC	\$0.35	\$1.04
Massive	\$0.21	\$0.63
FewNerd	\$0.98	\$2.95
FewEvent	\$1.35	\$4.50
GoEmo	\$0.66	\$2.10

Table 8: The API query cost of our method on six benchmarks.

Dataset	Zeroshot-TeC	Fewshot-TeC
MTOP	0.70 h	1.17 h
CLINC	0.92 h	1.57 h
Massive	1.47 h	1.88 h
FewNerd	1.33 h	2.0 h
FewEvent	1.0 h	1.55 h
GoEmo	1.0 h	1.43 h

Table 9: The total training time (hours) of our method on six benchmarks.

Conclusion

In this paper, we propose TeC, a new method for text clustering. TeC is structured as a two-stage process guided by LLMs, comprising the teaching and correcting phases. In the teaching phase, we propose a weakly-supervised contrastive learning loss to fine-tune embedding models based on feedback from LLMs through batch-wise queries. Subsequently, in the correcting phase, we leverage LLMs to instruct and refine cluster assignments of low-confidence samples, thereby enhancing overall clustering performance. Our method demonstrates superior performance compared to state-of-the-art approaches across six benchmark datasets spanning three distinct clustering tasks, all achieved at a reasonable cost. Additionally, we conduct ablation studies to substantiate the efficacy of our proposed method.

Acknowledgements

This research was partially supported by STI 2030-Major Projects 2021ZD0200400, National Natural Science Foundation of China (62276233 and 62072405) and Key Research Project of Zhejiang Province (2023C01048).

References

Aggarwal, C. C.; and Zhai, C. 2012. A survey of text clustering algorithms. *Mining text data*, 77–128.

Aharoni, R.; and Goldberg, Y. 2020. Unsupervised Domain Clusters in Pretrained Language Models. In *ACL*.

An, W.; Tian, F.; Zheng, Q.; Ding, W.; Wang, Q.; and Chen, P. 2023. Generalized Category Discovery with Decoupled Prototypical Network. In *AAAI 2023*.

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; and Chen, C. 2022. Constitutional AI: Harmlessness from AI Feedback. *CoRR*, abs/2212.08073.

Bang, Y.; Cahyawijaya, S.; Lee, N.; Dai, W.; Su, D.; Wilie, B.; Lovenia, H.; Ji, Z.; Yu, T.; Chung, W.; Do, Q. V.; Xu, Y.; and Fung, P. 2023. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. *CoRR*, abs/2302.04023.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan): 993–1022.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *NIPS*.

Cheng, Q.; Yang, X.; Sun, T.; Li, L.; and Qiu, X. 2023. Improving Contrastive Learning of Sentence Embeddings from AI Feedback. *arXiv preprint arXiv:2305.01918*.

Choi, E.; Levy, O.; Choi, Y.; and Zettlemoyer, L. 2018. Ultra-Fine Entity Typing. In *ACL)*. Melbourne, Australia: Association for Computational Linguistics.

Demszky, D.; Movshovitz-Attias, D.; Ko, J.; Cowen, A.; Nemade, G.; and Ravi, S. 2020. GoEmotions: A Dataset of Fine-Grained Emotions. In *ACL*.

Deng, S.; Zhang, N.; Kang, J.; Zhang, Y.; Zhang, W.; and Chen, H. 2020. Meta-Learning with Dynamic-Memory-Based Prototypical Network for Few-Shot Event Detection. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. New York, NY, USA: Association for Computing Machinery. ISBN 9781450368223.

DeVries, T.; and Taylor, G. W. 2018. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*.

Ding, N.; Xu, G.; Chen, Y.; Wang, X.; Han, X.; Xie, P.; Zheng, H.; and Liu, Z. 2021. Few-NERD: A Few-shot Named Entity Recognition Dataset. In *ACL*.

FitzGerald, J.; Hench, C.; Peris, C.; Mackie, S.; Rottmann, K.; and Sanchez, A. 2023. MASSIVE: A 1M-Example Multilingual Natural Language Understanding Dataset with 51 Typologically-Diverse Languages. In *ACL 2023*.

FORCE11. 2020. The FAIR Data principles. https://force11.org/info/the-fair-data-principles/.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP*.

Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J. W.; Wallach, H.; Iii, H. D.; and Crawford, K. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92.

Gilardi, F.; Alizadeh, M.; and Kubli, M. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.

Gunel, B.; Du, J.; Conneau, A.; and Stoyanov, V. 2021. Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning. In *ICLR*.

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)*.

He, X.; Lin, Z.; Gong, Y.; Jin, A.; Zhang, H.; Lin, C.; Jiao, J.; Yiu, S. M.; Duan, N.; Chen, W.; et al. 2023. Annollm: Making large language models to be better crowdsourced annotators. *arXiv preprint arXiv:2303.16854*.

Huang, F.; Kwak, H.; and An, J. 2023. Is ChatGPT better than Human Annotators? Potential and Limitations of ChatGPT in Explaining Implicit Hate Speech. In *WWW 2023*, 294–297. ACM.

Huang, J.; Tang, D.; Zhong, W.; Lu, S.; Shou, L.; Gong, M.; Jiang, D.; and Duan, N. 2021. WhiteningBERT: An Easy Unsupervised Sentence Embedding Approach. In *EMNLP*.

Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Kenyon-Dean, K.; Ahmed, E.; Fujimoto, S.; Georges-Filteau, J.; Glasz, C.; Kaur, B.; Lalande, A.; Bhanderi, S.; Belfer, R.; Kanagasabai, N.; Sarrazingendron, R.; Verma, R.; and Ruths, D. 2018. Sentiment Analysis: It's Complicated! In *NAACL*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR, 2015, Conference Track Proceedings*.

Larson, S.; Mahendran, A.; Peper, J. J.; Clarke, C.; Lee, A.; Hill, P.; and Kummerfeld, J. K. 2019. An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. In *EMNLP-IJCNLP*. Hong Kong, China.

Li, H.; Arora, A.; Chen, S.; Gupta, A.; Gupta, S.; and Mehdad, Y. 2021. MTOP: A Comprehensive Multilingual Task-Oriented Semantic Parsing Benchmark. In *ACL*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

MacQueen, J. 1965. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symposium on Math., Stat., and Prob*, 281.

Mausam; Schmitz, M.; Soderland, S.; Bart, R.; and Etzioni, O. 2012. Open Language Learning for Information Extraction. In *EMNLP*.

Mihalcea, R.; and Tarau, P. 2004. Textrank: Bringing order into text. In *EMNLP*.

Muennighoff, N.; Tazi, N.; Magne, L.; and Reimers, N. 2023. MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*.

Papadimitriou, C. H.; and Steiglitz, K. 1998. *Combinatorial optimization: algorithms and complexity*. Courier Corporation.

Reimers, N. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP*.

Shepitsen, A.; Gemmell, J.; Mobasher, B.; and Burke, R. 2008. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 ACM conference on Recommender systems*, 259–266.

Steinbach, M.; Karypis, G.; and Kumar, V. 2000. A comparison of document clustering techniques.

Su, H.; Shi, W.; Kasai, J.; Wang, Y.; Hu, Y.; Ostendorf, M.; Yih, W.-t.; Smith, N. A.; Zettlemoyer, L.; and Yu, T. 2023. One Embedder, Any Task: Instruction-Finetuned Text Embeddings. In *ACL*.

Tang, J.; Yao, L.; Zhang, D.; and Zhang, J. 2010. A combination approach to web user profiling. *ACM Transactions on Knowledge Discovery from Data (TKDD)*.

Vaze, S.; Han, K.; Vedaldi, A.; and Zisserman, A. 2022. Generalized Category Discovery. In *CVPR*.

Viswanathan, V.; Gashteovski, K.; Lawrence, C.; Wu, T.; and Neubig, G. 2023. Large Language Models Enable Few-Shot Clustering. *CoRR*.

Wang, L.; Yang, N.; Huang, X.; Jiao, B.; Yang, L.; Jiang, D.; Majumder, R.; and Wei, F. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *arXiv preprint arXiv:2212.03533*.

Wang, Z.; Shang, J.; and Zhong, R. 2023. Goal-Driven Explainable Clustering via Language Descriptions. In *EMNLP*.

Wu. 2006. A local learning approach for clustering. In *NIPS*.

Xiao, S.; Liu, Z.; Shao, Y.; and Cao, Z. 2022. Retro-MAE: Pre-Training Retrieval-oriented Language Models Via Masked Auto-Encoder. In *EMNLP*.

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*.

Yin, Z.; Cao, L.; Han, J.; Zhai, C.; and Huang, T. 2011. Geographical topic discovery and comparison. In *WWW*.

Zhang, D.; Nan, F.; Wei, X.; Li, S.; Zhu, H.; McKeown, K. R.; Nallapati, R.; Arnold, A. O.; and Xiang, B. 2021a. Supporting Clustering with Contrastive Learning. In *NAACL*.

Zhang, H.; Xu, H.; Lin, T.; and Lyu, R. 2021b. Discovering New Intents with Deep Aligned Clustering. In *AAAI*.

Zhang, H.; Xu, H.; and Lin, T.-E. 2021. Deep open intent classification with adaptive decision boundary. In *AAAI*, 14374–14382.

Zhang, Y.; Wang, Z.; and Shang, J. 2023. ClusterLLM: Large Language Models as a Guide for Text Clustering. In *EMNLP*.

Zhang, Y.; Zhang, H.; Zhan, L.-M.; Wu, X.-M.; and Lam, A. 2022. New Intent Discovery with Pre-training and Contrastive Learning. In *ACL*.

Zhou, S.; Xu, H.; Zheng, Z.; Chen, J.; Bu, J.; Wu, J.; Wang, X.; Zhu, W.; Ester, M.; et al. 2022. A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. *arXiv preprint arXiv:2206.07579*.

Paper Checklist

1. For most authors...
   (a) Would answering this research question advance science without violating social contracts, such as violating privacy norms, perpetuating unfair profiling, exacerbating the socio-economic divide, or implying disrespect to societies or cultures? Our research is conducted on public datasets. Hence, social contracts are not violated in our work.
   (b) Do your main claims in the abstract and introduction accurately reflect the paper's contributions and scope? Yes
   (c) Do you clarify how the proposed methodological approach is appropriate for the claims made? Yes
   (d) Do you clarify what are possible artifacts in the data used, given population-specific distributions? Yes
   (e) Did you describe the limitations of your work? No
   (f) Did you discuss any potential negative societal impacts of your work? No. Our work don't have any negative impact on society.
   (g) Did you discuss any potential misuse of your work? No
   (h) Did you describe steps taken to prevent or mitigate potential negative outcomes of the research, such as data and model documentation, data anonymization, responsible release, access control, and the reproducibility of findings? No
   (i) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes
2. Additionally, if your study involves hypotheses testing...
   (a) Did you clearly state the assumptions underlying all theoretical results? N/A
   (b) Have you provided justifications for all theoretical results? N/A
   (c) Did you discuss competing hypotheses or theories that might challenge or complement your theoretical results? N/A
   (d) Have you considered alternative mechanisms or explanations that might account for the same outcomes observed in your study? N/A
   (e) Did you address potential biases or limitations in your theoretical framework? N/A
   (f) Have you related your theoretical results to the existing literature in social science? N/A
   (g) Did you discuss the implications of your theoretical results for policy, practice, or further research in the social science domain? N/A
3. Additionally, if you are including theoretical proofs...
   (a) Did you state the full set of assumptions of all theoretical results? N/A
   (b) Did you include complete proofs of all theoretical results? N/A
4. Additionally, if you ran machine learning experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? No. We will release our code in the final version due to the concern of copyright.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? Yes
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? Yes
   (e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? Yes
   (f) Do you discuss what is "the cost" of misclassification and fault (in)tolerance? No
5. Additionally, if you are using existing assets (e.g., code, data, models) or curating/releasing new assets, **without compromising anonymity**...
   (a) If your work uses existing assets, did you cite the creators? Yes
   (b) Did you mention the license of the assets? No
   (c) Did you include any new assets in the supplemental material or as a URL? No
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? No. Experimental datasets in our work are public datasets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? No
   (f) If you are curating or releasing new datasets, did you discuss how you intend to make your datasets FAIR (see FORCE11 (2020))? N/A
   (g) If you are curating or releasing new datasets, did you create a Datasheet for the Dataset (see Gebru et al. (2021))? N/A
6. Additionally, if you used crowdsourcing or conducted research with human subjects, **without compromising anonymity**...
   (a) Did you include the full text of instructions given to participants and screenshots? N/A
   (b) Did you describe any potential participant risks, with mentions of Institutional Review Board (IRB) approvals? N/A
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? N/A
   (d) Did you discuss how data is stored, shared, and deidentified? N/A