

# CLUSTERLLM: Large Language Models as a Guide for Text Clustering

Yuwei Zhang Zihan Wang Jingbo Shang\*

University of California, San Diego  
{yuz163, ziw224, jshang}@ucsd.edu

## Abstract

We introduce CLUSTERLLM, a novel text clustering framework that leverages feedback from an instruction-tuned large language model, such as ChatGPT. Compared with traditional unsupervised methods that builds upon “small” embedders, CLUSTERLLM exhibits two intriguing advantages: (1) it enjoys the emergent capability of LLM even if its embeddings are inaccessible; and (2) it understands the user’s preference on clustering through textual instruction and/or a few annotated data. First, we prompt ChatGPT for insights on clustering perspective by constructing hard triplet questions *<does A better correspond to B than C>*, where *A, B* and *C* are similar data points that belong to different clusters according to small embedder. We empirically show that this strategy is both effective for fine-tuning small embedder and cost-efficient to query ChatGPT. Second, we prompt ChatGPT for helps on clustering granularity by carefully designed pairwise questions *<do A and B belong to the same category>*, and tune the granularity from cluster hierarchies that is the most consistent with the ChatGPT answers. Extensive experiments on 14 datasets show that CLUSTERLLM consistently improves clustering quality, at an average cost of  $\sim \$0.6^1$  per dataset. The code will be available at <https://github.com/zhang-yu-wei/ClusterLLM>.

## 1 Introduction

Text clustering, as a fundamental task in natural language processing (NLP), has a wide spectrum of applications, such as identifying public perception from social media (Park et al., 2022), analysing cause of accidents (Xu et al., 2022), and detecting emerging research topics (Martínez et al., 2022). A common practice for text clustering is to apply clustering algorithms (MacQueen, 1967; Zhang et al.,

\* Corresponding author.

<sup>1</sup>The cost is calculated with gpt-3.5-turbo.

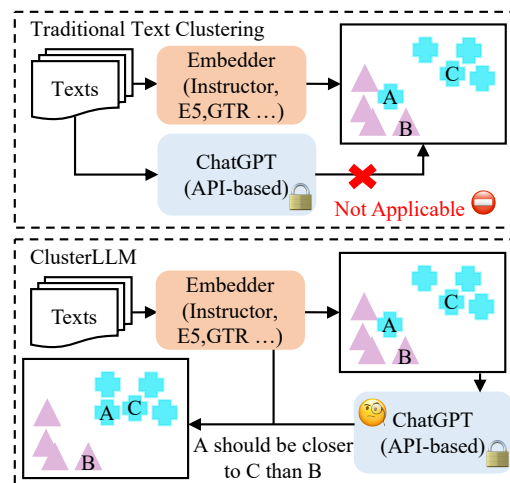


Figure 1: LLMs like ChatGPT are not applicable for text clustering directly because of the inaccessible embeddings. CLUSTERLLM resolves the dilemma by leveraging LLM as a guide on text clustering.

2021a) on top of pre-trained embedders (Muenighoff et al., 2022; Wang et al., 2022; Su et al., 2022) which could achieve higher performance with better pre-training quality. State-of-the-art large language models (LLMs) such as recent GPT series (Brown et al., 2020; Ouyang et al., 2022; OpenAI, 2023) have demonstrated extraordinary language capabilities for various NLP applications however, these GPT models can only be utilized through the APIs without accessible embedding vectors for clustering. Hence, LLMs cannot be directly applied on text clustering tasks.

In this paper, we provide insights on the question: *Can we leverage API-based LLMs to guide text clustering efficiently?* We attack this challenging question by drawing inspiration from an observation that *humans represent an instance through comparing with others* (Nosofsky, 2011). For instance, people often classify a new piece of music into a specific genre by relating to familiar ones. In fact, pairwise relationships have been utilized in spectral clustering (Donath and Hoffman, 1972; Cheeger, 1970) before. Nonetheless, naively

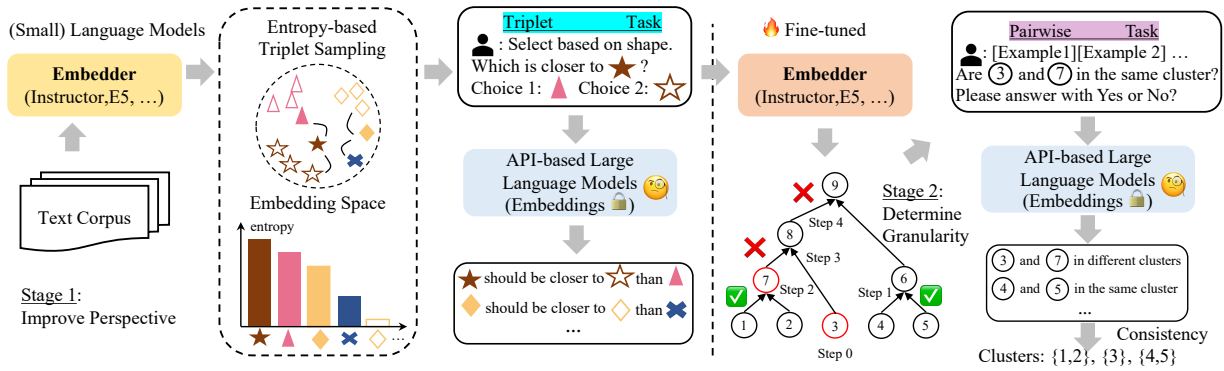


Figure 2: An overview of CLUSTERLLM. It utilizes LLM to guide an embedder for text clustering with a low cost.

traversing all the pairs within dataset is obviously intractable and too expensive for querying LLMs.

We propose CLUSTERLLM, a framework that utilizes LLM to guide a small embedder for finding text clusters with a low cost, as shown in Figure 1. It comprises two stages that are specially designed for two aspects of clustering: (1) perspective, i.e., the grouping criterion such as *topic*, *intent* and *emotion* and (2) granularity, i.e. the scope of clusters.

In Stage 1, we prompt LLMs with a triplet task that predicts which one of the two candidate choices is closer to anchor instance to understand the user-preferred perspectives. We choose this triplet task because (a) it is irrelevant with cluster granularity and (b) the produced triplets can fine-tune small embedder towards the right perspective. In order to improve sample efficiency, we further propose entropy-based triplet sampling to find the most informative triplets. Specifically, we first calculate entropy for each instance based on cluster assignment probabilities, and then identify those with highest entropy. Two candidate choices are then sampled from its nearest clusters to guarantee they are close enough to the anchor.

In Stage 2, we first obtain the cluster hierarchy that starts from instance-level clusters and iteratively merge two closest clusters until the entire dataset. And then we prompt LLMs to determine cluster granularity with a few annotated data pairs as demonstrations. We construct the data pairs to prompt by sampling from two clusters that are merged at each step of hierarchical clustering, so that they cover a wide range of granularities. And the final decision is made by measuring consistency between each level of clustering and predictions.

We extensively evaluate CLUSTERLLM on 14 datasets that include diverse tasks such as intent discovery, topic mining, type discovery, domain discovery, and emotion detection. Furthermore, these

datasets span a wide range of granularities that have 10 to 150 number of clusters. We show that CLUSTERLLM is effective overall on improving clustering quality, where the clustering performance is improved over both a deep clustering baseline and a self-supervise baseline. Moreover, the ablation study shows that our sampling strategy is effective compared to a random sampling baseline. Finally, CLUSTERLLM also outperforms clustering-error based methods on determining cluster granularity.

In summary, our contributions are three-fold: (i) We propose a framework CLUSTERLLM that utilizes sentence relations predicted from API-based LLMs to guide clustering. Furthermore, it allows users to provide textual instructions and/or few-shot annotations to specify preferences on clustering. (ii) In order to reduce API-queries, we propose a novel entropy-based sampling strategy to find the most informative triplets. Additionally, we utilize pairwise data sampled from hierarchical clustering to determine cluster granularity. (iii) Extensive experiments show that our proposed method can improve clustering performance at  $\sim \$0.2$  for perspective and  $\sim \$0.4$  for granularity with GPT-3.5.

## 2 Preliminary

Text clustering takes an unlabeled corpus  $\mathcal{D} = \{x_i\}_{i=1}^N$  as input, and outputs a clustering assignment  $\mathcal{Y} = \{y_i\}_{i=1}^N$  that maps the input text to cluster indices. To specify user’s needs, CLUSTERLLM integrates additional textual instruction (e.g. “Select the example that better corresponds with the Query in terms of entity type.”) to understand perspective and few-shot annotations (e.g. “Sentence1 and Sentence2 have the same entity type ...”) to determine cluster granularity.

## 3 Our CLUSTERLLM

CLUSTERLLM is based on a pre-trained small embedder (Wang et al., 2022; Su et al., 2022) (denoted

as  $f$ ) which usually represents sentences individually. In contrast, inspired by human cognitive ability (Nosofsky, 2011), CLUSTERLLM considers a pair or a triplet of sentences through prompting LLMs that are trained to follow human instructions (Ouyang et al., 2022; OpenAI, 2023). Specifically, CLUSTERLLM is a two-stage framework (See Figure 2). In Section 3.1 we introduce Stage 1 that utilizes triplet task to improve clustering quality with respect to user-specified perspectives, along with a sampling strategy that reduces number of API queries. In Section 3.2, we introduce Stage 2 that leverages pairwise task to determine cluster granularity based on predictions from LLMs.

### 3.1 Triplet Task for Perspective

In this section, we explore how to harness a triplet task to refine the cluster structures for a user-specified perspective. A triplet task takes as input a tuple of three sentences  $t = (a, c_1, c_2)$ , where  $a$  is the anchor and  $(c_1, c_2)$  are two choices. We then prompt LLMs to select one of  $(c_1, c_2)$  that better corresponds with  $a$  using a prompt  $\mathcal{P}_T$ . Moreover, in order to specify the user’s perspective,  $\mathcal{P}_T$  also requires a task instruction  $\mathcal{I}_T$  as input. The LLM should make a choice

$$c_j = \mathcal{P}_T(\mathcal{I}_T, t), \quad (1)$$

where  $c_j \in \{c_1, c_2\}$  indicates one of the choices that LLM selects as positive and we denote the other (or negative) one as  $c_{\setminus j}$ .

#### 3.1.1 Entropy-based Triplet Sampling

While one can randomly sample triplets to query the LLM, we demonstrate it non-efficient in experiments. In this section, we pose the question of mining *informative* triplets to both save the costs from querying LLMs and optimally improve the clustering. To achieve this, we resort to the current clustering results from the extracted embeddings  $\mathcal{Z} = \{z_i = f(x_i)\}_{i=1}^N$ . In summary, our algorithm contains two steps: **Step 1:** We find the most ambiguous instances as anchors based on entropy. **Step 2:** For each anchor instance, we sample two choices from two of its closest clusters. Refer to Algorithm 1 for entire process.

In Step 1, since the granularity is unknown at current stage, we perform clustering on top of  $\mathcal{Z}$ , where the clustering hyperparameters<sup>2</sup> are consis-

<sup>2</sup>It can be number of clusters in K-means or maximum distance to merge two clusters in hierarchical clustering

---

#### Algorithm 1: Entropy-based Triplet Sampling

---

**Input:** embeddings  $\mathcal{Z} = \{z_i = f(x_i)\}_{i=1}^N$ , interval boundaries  $\gamma_{\text{high}}$  and  $\gamma_{\text{low}}$ , closest clusters fraction  $\epsilon = 2\%$ , maximum number of queries  $Q$ .

```

1 Step 1:  $K$  clusters  $\leftarrow$  Clustering( $\mathcal{Z}$ );
2 Compute  $\mu_k$  for each cluster  $k$  by averaging;
3 Compute  $K_{\text{closest}}$  with Eq. 3;
4 Compute entropy  $H$  with Eq. 4;
5  $\text{Ind} \leftarrow \text{argsort}(H)[::-1]$ ;
6  $\text{Ind} \leftarrow \text{Ind}[\gamma_{\text{high}}N:\gamma_{\text{low}}N]$ ;
7 Step 2: Initialize triplets  $\{t_q\} \leftarrow \{\}$ ;
8 while  $\text{len}(\{t_q\}) < Q$  do
9   for  $a$  in  $\text{Ind}$  do
10     Obtain  $K_{\text{closest}}$  closest clusters;
11     Sample  $C_1, C_2$  from closest clusters;
12      $c_1 \sim C_1, c_2 \sim C_2, t = (a, c_1, c_2)$ ;
13     if  $t$  not in  $\{t_q\}, c_1 \neq a, c_2 \neq a$  then
14       Append  $t$  to  $\{t_q\}$ ;
```

**Output:** A set of triplets  $\{t_q\}_{q=1}^Q$

---

tent across datasets and only specific to the embedder model  $f$ . Cluster center  $\mu_k$  will thereafter be calculated for cluster  $k$  by averaging embeddings assigned to it. Following (Xie et al., 2016; Van der Maaten and Hinton, 2008), we calculate instance-wise soft assignments with Student’s  $t$ -distribution,

$$p_{ik} = \frac{(1 + \|z_i - \mu_k\|^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{k'} (1 + \|z_i - \mu_{k'}\|^2/\alpha)^{-\frac{\alpha+1}{2}}} \quad (2)$$

where  $\alpha = 1$  is the degree of freedom. We then define closest clusters for instance  $i$  as  $K_{\text{closest}}$  clusters with largest soft assignment  $p_{ik}$ . Here,  $K_{\text{closest}}$  is proportional to the total number of clusters  $K$ .

$$K_{\text{closest}} = \max(\epsilon K, 2) \quad (3)$$

where we fix  $\epsilon$  to be a small value, such as 2%. We then compute entropy based on these closest clusters with renormalized probabilities  $p'_{ik}$ ,

$$h_i = - \sum_{k=1}^{K_{\text{closest}}} p'_{ik} \log(p'_{ik}) \quad (4)$$

where  $p'_{ik} = \frac{p_{ik}}{\sum_{k'=1}^{K_{\text{closest}}} p_{ik'}}$ . We sort the entire dataset in descending order according to the entropies  $H = \{h_i\}_{i=1}^N$ . We introduce two hyperparameters  $\gamma_{\text{high}}$  and  $\gamma_{\text{low}}$  that control the proportion interval to filter out from ordered dataset. Our hypothesis is that higher entropy (smaller  $\gamma_{\text{high}}$  and  $\gamma_{\text{low}}$ ) anchors form more informative triplets that we verify in Section 4.6. In Step 2, we randomly sample two clusters  $C_1, C_2$  from  $K_{\text{closest}}$  closest clusters,

and then sample two sentences  $c_1, c_2$  from each of them as choices (see line 11 and line 12). In another word, these choices should be either a positive or a hard negative to the anchor. Finally, we also remove triplets that are either repeated or have identical choice and anchor. We continue to sample triplets until reaching budget  $Q$ .

**Remarks.** (1) Since  $Q$  is defined by the user and is independent with the dataset size, our sampling is cost-efficient. For example, in our experiments, using 1,024 queries can improve performance on both dataset scales of  $\sim 3,000$  and  $\sim 50,000$ . (2) From the view of ground truth, the sampled triplets might contain “both are correct” or “none of the above”. However, we argue that even these triplets might provide soft aligning information, i.e. the ranking of closeness between choices. (3) Our sampling method may also be utilized in active learning to acquire human annotations when no prior knowledge is available on the categories.

### 3.1.2 Finetuning Embedder

Now that we have the triplet predictions, it is still not clear how to utilize them in clustering. Previous research rely on deep constrained clustering (Zhang et al., 2020; Manduchi et al., 2021) which are often sensitive to noisy labels (Basu et al., 2008). In this paper, we instead focus on finetuning the base embedder  $f$  towards producing an embedding space that better explains the user’s perspective. We exploit both hard and in-batch negatives. Following (Su et al., 2022; Ni et al., 2022b), for a triplet  $t = (a, c_j, c_{\setminus j})$  with positive  $c_j$  and hard negative  $c_{\setminus j}$ , we optimize the following objective,

$$l_j = \frac{\exp(s(a, c_j)/\tau)}{\sum_{c_l \in \mathcal{B}} \exp(s(a, c_l)/\tau)} \quad (5)$$

where  $\mathcal{B}$  combines  $c_j, c_{\setminus j}$  and other in-batch negatives.  $\tau$  is a temperature parameter. Following the original implementation, we also compute the loss with  $a$  and  $c_j$  swapped. Finally fine-tuned embedders can be applied to find even more informative triplets with our sampling method which will further improve performance in an iterative manner. We acquire clustering assignments by running clustering algorithms on top of extracted embeddings.

## 3.2 Pairwise Task for Granularity

In this section, we build upon the refined embedding space in Section 3.1 to determine cluster granularity. In this paper, we convert the problem of

determining granularity into finding the best step in a cluster hierarchy (see Figure 2 right), where each step denotes a unique granularity (or equally number of clusters). It is non-trivial, since different granularities can be applied to the same dataset (such as domains or topics). To tackle this challenge, we query LLM with pairwise task that predicts whether a pair of data  $p$  belong to the same cluster with a prompt  $\mathcal{P}_P$ ,

$$w = \mathcal{P}_P(\mathcal{I}_P, \{\tilde{p}_d\}_{d=1}^D, p) \quad (6)$$

where  $w \in \{\text{same, different}\}$  is the binary decision,  $\mathcal{I}_P$  is the task instruction and  $\{\tilde{p}_d\}_{d=1}^D$  are few-shot demonstration pairs used for in-context learning (typically  $D = 4$ ). We assume these demonstration pairs are annotated by users who have a desired cluster granularity in mind. We also combine a brief justification for each demonstration pair (see Table 12 bottom for example).

### 3.2.1 Determine Granularity with Pairwise Hierarchical Sampling

We then introduce how to sample pairs from cluster hierarchy to query LLMs and determine granularity. We assume a maximum and a minimum number of clusters (denoted as  $k_{\max}$  and  $k_{\min}$ ) following Pelleg et al. (2000) which depend on the user’s expectation on the granularity. We then randomly sample  $\lambda$  (1 or 3 in our experiments) pairs of data from the two clusters to be merged at each step to form candidate pairs  $\{p_i\}_{i=1}^{N_p}$ , where  $N_p = \lambda(k_{\max} - k_{\min})$ . These pairs cover the entire range of granularity between  $k_{\max}$  and  $k_{\min}$ , and will be used to query LLMs. After that, each level of granularity can be examined against LLM predictions to choose the one with the highest consistency measure  $\mathcal{M}$ ,

$$k^* = \underset{k}{\operatorname{argmax}} \mathcal{M}(W^p, W^k) \quad (7)$$

where  $W^p = \{w_i^p\}_{i=1}^{N_p}$  denotes the predictions obtained from Eq. 6 and  $W^k$  represents a set of binary values indicating whether each pair of data is in the same cluster at granularity  $k$ . Empirically, we found the following performs better in our framework: use F-beta score, a weighted harmonic mean of precision and recall, as measurement  $\mathcal{M}$  and set  $W^p/W^k$  as labels/predictions. Finally, for large-scale datasets, we address the high time complexity of hierarchical clustering by applying it on top of mini-batch K-means. See details in Appendix A.

**Remarks.** Similar to Section 3.1.1, pairwise hierarchical sampling can also be used to acquire human

annotations. Nonetheless, the reliability of the algorithm still depends on the quality of clusters. In an extreme case where the clusters are completely random, it is unable to find granularity even though all the pairwise predictions are correct.

## 4 Experiments

We first evaluate CLUSTERLLM on clustering quality with ground truth number of clusters in Section 4.4. Then we conduct ablation studies in Section 4.6 to further analyze the effectiveness of CLUSTERLLM. Finally, we show results of determining cluster granularity in Section 4.7.

### 4.1 Datasets

We provide a high-level summary of evaluated datasets in this section, and see Appendix E for more descriptions. In this paper, we evaluate on a broad range of clustering datasets with various perspectives and granularities. Furthermore, to better analyze the effect of scale, each dataset has both a small-scale and a large-scale version. The two versions are different in number of data while keeping the same number of clusters. A summary of dataset statistics is shown in Table 1. Note that there is no data splits in clustering.

**Intent (Domain) Discovery.** Intent discovery (Zhang et al., 2021b, 2022) discovers unknown intents in unlabeled customer utterances. For CLINC, Massive and MTOP, we also use domains as labels to convert them into domain discovery.

**Type Discovery.** Type Discovery (Li et al., 2022) resolves the closed-world set-up of traditional Information Extraction. In this work, we focus on three tasks: entity, relation and event type discovery. To indicate specific mentions (entities or event triggers), we directly append them behind sentences with natural language formats, such as “The relation between [ENTITY1] and [ENTITY2]”.

**Topic Mining.** We adapt three topic mining datasets from MTEB (Muennighoff et al., 2022).

**Emotion.** We adapt GoEmo (Demszky et al., 2020), a fine-grained emotion detection dataset by removing multi-label or neutral instances.

### 4.2 Experiment Details

**Query LLMs.** The prompt only contains a task-specific instruction (see Table 11). We set generation temperature to 0.5. Explanations are suppressed by adding a postfix: “Please respond with ‘Choice 1’ or ‘Choice 2’ without explanation” and

Task	Name	#clusters	#data(small)	#data(large)
Intent	Bank77	77	3,080	10,003
	CLINC(I)	150	4,500	15,000
	MTOP(I)	102	4,386	15,638
	Massive(I)	59	2,974	11,510
Type	FewRel	64	4,480	40,320
	FewNerd	58	3,789	50,000
	FewEvent	34	4,742	18,969
Topic	StackEx	121	4,156	50,000
	ArxivS2S	93	3,674	50,000
	Reddit	50	3,217	50,000
Emotion	GoEmo	27	5,940	23,485
Domain	CLINC(D)	10	4,500	15,000
	MTOP(D)	11	4,386	15,667
	Massive(D)	18	2,974	11,514

Table 1: Dataset statistics.

set up a max token of 10. We then assign them to binary choices by directly checking whether one of the texts “Choice 1” or “Choice 2” is in the response. We also find that a very small amount of responses do not contain any choices and we discard them during fine-tuning. We use the Python API tool provided by OpenAI.

**Triplet Sampling.** For both small- or large-scale experiments, we set a budget of  $Q = 1,024$  triplets. We set  $\gamma_{\text{low}} = 20\%$  and  $\gamma_{\text{high}} = 0$ . For clustering methods, we fix hyperparameters of these algorithms across datasets in Stage 1. We choose agglomerative clustering with fixed distance threshold 67 for small-scale experiments on Instructor, and 77 on E5 (the embeddings are preprocessed by standard scaler). For large-scale datasets, we choose mini-batch K-means with fixed number of clusters 100 due to its lower latency. Clustering algorithms are implemented by scikit-learn (Pedregosa et al., 2011).

**Fine-tune Embedders.** In this work, we focus on two state-of-the-art pre-trained embedders: Instructor (Su et al., 2022) and E5 (Wang et al., 2022). We only use the large versions. Refer to Appendix D for details.

**Evaluation.** To reduce cost, we run CLUSTERLLM once for each dataset. We then run (mini-batch) K-means on (large) small-scale datasets for 5 seeds with ground truth K. We show two metrics. The first one is clustering accuracy calculated after Hungarian alignment (Kuhn, 1955) that permute prediction classes back to label classes. Another popular metric for clustering is normalized mutual information (NMI) that calculates mutual information between two assignments, and normalized by their individual entropies.

Method	Intent Discovery								Emotion	
	Bank77		CLINC(I)		MTOP(I)		Massive(I)		GoEmo	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
E5 (Wang et al., 2022)	59.90 <sub>(0.91)</sub>	77.71 <sub>(0.42)</sub>	75.83 <sub>(0.79)</sub>	91.16 <sub>(0.42)</sub>	33.54 <sub>(0.92)</sub>	70.79 <sub>(0.26)</sub>	52.52 <sub>(0.62)</sub>	70.67 <sub>(0.50)</sub>	22.13 <sub>(1.04)</sub>	20.98 <sub>(0.53)</sub>
SCCL-E (Zhang et al., 2021a)	63.60 <sub>(1.37)</sub>	77.34 <sub>(0.62)</sub>	77.96 <sub>(1.78)</sub>	91.89 <sub>(0.49)</sub>	33.82 <sub>(1.07)</sub>	70.42 <sub>(0.34)</sub>	54.48 <sub>(1.80)</sub>	71.57 <sub>(0.89)</sub>	22.03 <sub>(0.69)</sub>	20.05 <sub>(0.63)</sub>
self-supervise-E	64.76 <sub>(2.09)</sub>	80.75 <sub>(0.67)</sub>	78.91 <sub>(0.69)</sub>	92.06 <sub>(0.23)</sub>	33.81 <sub>(0.81)</sub>	71.71 <sub>(0.42)</sub>	54.23 <sub>(1.57)</sub>	71.78 <sub>(0.25)</sub>	21.35 <sub>(0.42)</sub>	20.92 <sub>(0.55)</sub>
CLUSTERLLM-E	69.09 <sub>(1.99)</sub>	83.17 <sub>(0.48)</sub>	79.51 <sub>(1.10)</sub>	92.73 <sub>(0.36)</sub>	34.66 <sub>(1.31)</sub>	73.19 <sub>(0.41)</sub>	54.80 <sub>(0.72)</sub>	73.97 <sub>(0.38)</sub>	22.69 <sub>(0.41)</sub>	22.07 <sub>(0.23)</sub>
CLUSTERLLM-E-iter	70.13 <sub>(1.34)</sub>	84.16 <sub>(0.36)</sub>	80.48 <sub>(0.93)</sub>	92.92 <sub>(0.29)</sub>	<b>37.22</b> <sub>(1.18)</sub>	<b>74.46</b> <sub>(0.11)</sub>	56.08 <sub>(1.01)</sub>	74.39 <sub>(0.21)</sub>	22.22 <sub>(1.15)</sub>	22.23 <sub>(0.17)</sub>
Instructor (Su et al., 2022)	64.49 <sub>(1.52)</sub>	81.43 <sub>(0.61)</sub>	79.29 <sub>(1.03)</sub>	92.60 <sub>(0.19)</sub>	33.35 <sub>(1.32)</sub>	70.63 <sub>(0.27)</sub>	54.08 <sub>(1.53)</sub>	73.42 <sub>(0.62)</sub>	25.19 <sub>(0.98)</sub>	21.54 <sub>(0.46)</sub>
SCCL-I (Zhang et al., 2021a)	65.48 <sub>(1.36)</sub>	81.77 <sub>(1.36)</sub>	80.85 <sub>(0.74)</sub>	92.94 <sub>(0.44)</sub>	34.28 <sub>(0.58)</sub>	73.52 <sub>(0.38)</sub>	54.10 <sub>(1.05)</sub>	73.90 <sub>(0.36)</sub>	<b>34.33</b> <sub>(0.86)</sub>	<b>30.54</b> <sub>(0.68)</sub>
self-supervise-I	68.18 <sub>(0.73)</sub>	83.31 <sub>(0.59)</sub>	80.82 <sub>(0.75)</sub>	93.88 <sub>(0.17)</sub>	34.06 <sub>(0.64)</sub>	72.50 <sub>(0.47)</sub>	55.07 <sub>(1.25)</sub>	72.88 <sub>(0.77)</sub>	24.11 <sub>(2.02)</sub>	22.05 <sub>(0.53)</sub>
CLUSTERLLM-I	70.77 <sub>(0.49)</sub>	85.07 <sub>(0.33)</sub>	82.77 <sub>(1.20)</sub>	93.88 <sub>(0.17)</sub>	35.84 <sub>(2.07)</sub>	73.52 <sub>(0.38)</sub>	59.89 <sub>(2.05)</sub>	76.96 <sub>(0.54)</sub>	27.49 <sub>(1.25)</sub>	24.78 <sub>(0.56)</sub>
CLUSTERLLM-I-iter	<b>71.20</b> <sub>(1.59)</sub>	<b>85.15</b> <sub>(0.41)</sub>	<b>83.80</b> <sub>(0.41)</sub>	<b>94.00</b> <sub>(0.21)</sub>	35.04 <sub>(0.97)</sub>	73.83 <sub>(0.79)</sub>	<b>60.69</b> <sub>(0.96)</sub>	<b>77.64</b> <sub>(0.21)</sub>	26.75 <sub>(1.76)</sub>	23.89 <sub>(0.68)</sub>

Method	Type Discovery						Topic Mining			
	FewRel		FewNerd		FewEvent		StackEx		Reddit	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
E5 (Wang et al., 2022)	39.62 <sub>(1.22)</sub>	55.78 <sub>(0.25)</sub>	25.49 <sub>(0.44)</sub>	40.62 <sub>(0.16)</sub>	37.30 <sub>(1.97)</sub>	59.28 <sub>(0.52)</sub>	37.31 <sub>(0.95)</sub>	58.59 <sub>(0.62)</sub>	39.03 <sub>(0.62)</sub>	48.63 <sub>(0.75)</sub>
SCCL-E (Zhang et al., 2021a)	39.93 <sub>(0.83)</sub>	56.49 <sub>(0.10)</sub>	27.86 <sub>(1.04)</sub>	43.12 <sub>(0.47)</sub>	35.39 <sub>(0.53)</sub>	57.23 <sub>(0.35)</sub>	37.85 <sub>(0.71)</sub>	59.17 <sub>(0.33)</sub>	44.45 <sub>(0.79)</sub>	53.29 <sub>(0.71)</sub>
self-supervise-E	43.01 <sub>(1.46)</sub>	58.78 <sub>(0.45)</sub>	29.25 <sub>(0.74)</sub>	44.77 <sub>(0.21)</sub>	37.07 <sub>(0.70)</sub>	62.14 <sub>(0.70)</sub>	42.19 <sub>(0.79)</sub>	63.06 <sub>(0.39)</sub>	48.69 <sub>(2.40)</sub>	57.02 <sub>(0.64)</sub>
CLUSTERLLM-E	47.53 <sub>(1.00)</sub>	62.89 <sub>(0.30)</sub>	28.52 <sub>(0.63)</sub>	44.45 <sub>(0.20)</sub>	42.17 <sub>(1.24)</sub>	67.55 <sub>(0.31)</sub>	43.01 <sub>(1.58)</sub>	63.81 <sub>(0.50)</sub>	47.72 <sub>(1.53)</sub>	56.41 <sub>(0.43)</sub>
CLUSTERLLM-E-iter	<b>52.95</b> <sub>(1.15)</sub>	<b>67.64</b> <sub>(0.29)</sub>	32.16 <sub>(0.83)</sub>	48.16 <sub>(0.25)</sub>	44.64 <sub>(0.90)</sub>	70.74 <sub>(0.35)</sub>	43.93 <sub>(0.93)</sub>	64.66 <sub>(0.36)</sub>	49.05 <sub>(1.15)</sub>	57.35 <sub>(0.31)</sub>
Instructor (Su et al., 2022)	41.23 <sub>(0.60)</sub>	57.55 <sub>(0.41)</sub>	30.02 <sub>(1.24)</sub>	46.12 <sub>(0.54)</sub>	41.99 <sub>(2.04)</sub>	62.88 <sub>(0.67)</sub>	44.81 <sub>(0.94)</sub>	65.76 <sub>(0.32)</sub>	54.98 <sub>(1.51)</sub>	62.51 <sub>(0.62)</sub>
SCCL-I (Zhang et al., 2021a)	41.15 <sub>(1.51)</sub>	57.04 <sub>(0.34)</sub>	31.09 <sub>(0.87)</sub>	46.47 <sub>(0.47)</sub>	39.97 <sub>(0.52)</sub>	57.36 <sub>(0.43)</sub>	45.11 <sub>(0.93)</sub>	65.36 <sub>(0.16)</sub>	53.66 <sub>(0.94)</sub>	61.34 <sub>(0.57)</sub>
self-supervise-I	41.72 <sub>(0.47)</sub>	57.83 <sub>(0.34)</sub>	31.39 <sub>(0.74)</sub>	47.25 <sub>(0.27)</sub>	43.94 <sub>(1.15)</sub>	64.71 <sub>(0.29)</sub>	46.15 <sub>(1.17)</sub>	66.49 <sub>(0.32)</sub>	55.41 <sub>(0.93)</sub>	63.45 <sub>(0.86)</sub>
CLUSTERLLM-I	47.94 <sub>(1.37)</sub>	62.43 <sub>(0.43)</sub>	34.75 <sub>(1.58)</sub>	51.03 <sub>(0.57)</sub>	46.17 <sub>(2.18)</sub>	70.73 <sub>(0.34)</sub>	47.21 <sub>(1.07)</sub>	66.78 <sub>(0.29)</sub>	56.79 <sub>(1.90)</sub>	63.87 <sub>(0.56)</sub>
CLUSTERLLM-I-iter	51.22 <sub>(1.43)</sub>	65.53 <sub>(0.40)</sub>	<b>40.60</b> <sub>(0.77)</sub>	<b>57.35</b> <sub>(0.23)</sub>	<b>50.60</b> <sub>(0.79)</sub>	<b>73.89</b> <sub>(0.47)</sub>	<b>47.75</b> <sub>(1.24)</sub>	<b>67.08</b> <sub>(0.30)</sub>	<b>57.02</b> <sub>(0.87)</sub>	<b>63.92</b> <sub>(0.38)</sub>

Method	Topic Mining		Domain Discovery						Avg	
	ArxivS2S		MTOP(D)		Massive(D)		CLINC(D)		ACC	NMI
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI		
E5 (Wang et al., 2022)	30.85 <sub>(0.37)</sub>	54.49 <sub>(0.15)</sub>	91.23 <sub>(4.23)</sub>	86.75 <sub>(1.94)</sub>	<b>63.70</b> <sub>(0.86)</sub>	66.20 <sub>(0.72)</sub>	59.64 <sub>(2.73)</sub>	57.23 <sub>(0.66)</sub>	47.72	61.35
SCCL-E (Zhang et al., 2021a)	32.78 <sub>(0.69)</sub>	55.77 <sub>(0.30)</sub>	92.28 <sub>(0.18)</sub>	86.01 <sub>(0.22)</sub>	61.22 <sub>(1.15)</sub>	65.27 <sub>(0.99)</sub>	59.88 <sub>(2.74)</sub>	56.21 <sub>(0.94)</sub>	48.82	61.70
self-supervise-E	34.41 <sub>(0.52)</sub>	57.09 <sub>(0.26)</sub>	91.40 <sub>(4.61)</sub>	88.20 <sub>(1.71)</sub>	61.79 <sub>(2.50)</sub>	65.02 <sub>(1.32)</sub>	57.29 <sub>(2.51)</sub>	57.07 <sub>(0.88)</sub>	49.87	63.60
CLUSTERLLM-E	34.93 <sub>(0.36)</sub>	57.90 <sub>(0.09)</sub>	89.18 <sub>(5.25)</sub>	86.19 <sub>(2.05)</sub>	60.73 <sub>(2.81)</sub>	66.15 <sub>(0.69)</sub>	56.25 <sub>(2.94)</sub>	56.32 <sub>(0.81)</sub>	50.77	64.77
CLUSTERLLM-E-iter	<b>35.73</b> <sub>(0.87)</sub>	<b>58.42</b> <sub>(0.38)</sub>	89.58 <sub>(5.08)</sub>	87.25 <sub>(1.96)</sub>	58.63 <sub>(1.32)</sub>	65.59 <sub>(0.81)</sub>	<b>60.84</b> <sub>(2.88)</sub>	58.55 <sub>(2.09)</sub>	52.40	66.18
Instructor (Su et al., 2022)	24.31 <sub>(0.77)</sub>	48.04 <sub>(0.39)</sub>	90.56 <sub>(3.34)</sub>	87.30 <sub>(1.53)</sub>	61.81 <sub>(2.56)</sub>	67.31 <sub>(1.79)</sub>	52.50 <sub>(2.44)</sub>	56.87 <sub>(2.03)</sub>	49.90	63.85
SCCL-I (Zhang et al., 2021a)	25.63 <sub>(0.53)</sub>	49.07 <sub>(0.22)</sub>	89.08 <sub>(3.77)</sub>	84.77 <sub>(2.18)</sub>	61.34 <sub>(2.77)</sub>	68.69 <sub>(1.42)</sub>	54.22 <sub>(3.15)</sub>	51.08 <sub>(1.05)</sub>	50.74	63.85
self-supervise-I	25.65 <sub>(0.37)</sub>	49.41 <sub>(0.17)</sub>	92.12 <sub>(2.66)</sub>	88.49 <sub>(1.25)</sub>	53.97 <sub>(2.14)</sub>	<b>71.53</b> <sub>(0.77)</sub>	58.58 <sub>(2.56)</sub>	<b>60.84</b> <sub>(1.04)</sub>	51.39	65.33
CLUSTERLLM-I	26.61 <sub>(0.48)</sub>	50.06 <sub>(0.26)</sub>	<b>93.53</b> <sub>(0.10)</sub>	<b>89.36</b> <sub>(0.11)</sub>	61.06 <sub>(1.91)</sub>	68.62 <sub>(0.90)</sub>	52.39 <sub>(1.84)</sub>	54.98 <sub>(2.08)</sub>	53.09	66.58
CLUSTERLLM-I-iter	26.34 <sub>(0.38)</sub>	50.45 <sub>(0.19)</sub>	92.13 <sub>(3.81)</sub>	89.23 <sub>(1.21)</sub>	60.85 <sub>(4.33)</sub>	68.67 <sub>(1.59)</sub>	51.82 <sub>(1.91)</sub>	54.81 <sub>(1.15)</sub>	<b>53.99</b>	<b>67.53</b>

Table 2: Comparison of clustering accuracy and NMI with known granularity for evaluation. Average over all 14 datasets are shown in the last two columns. Best results are bolded.

### 4.3 Compared Methods

**E5 and Instructor.** We directly apply (mini-batch) K-means on extracted embeddings from instructor-large and e5-large.

**self-supervise-I(E).** To verify that the performance improvement of CLUSTERLLM does not only come from domain-specific fine-tuning, instead of the more accurate triplet prediction. We propose a self-supervise fine-tuning that uses exactly the same triplets as CLUSTERLLM but only switch to self-supervised triplet predictions that select closest choices in embedding space.

**SCCL-I(E).** We also combine Instructor and E5 with SCCL (Zhang et al., 2021a), an unsupervised deep clustering algorithm that utilizes entire dataset

for training. Notice that our method uses fewer data for training. See Appendix D for details.

### 4.4 Main Results

We show main results with small-scale datasets in Table 2. We show several variants of our method: CLUSTERLLM-I(E) adopt Instructor or E5 as embedders. CLUSTERLLM-I(E)-iter applies the entire framework twice in an iterative manner by using previous fine-tuned model as initialization and the 1,024 triplets inferred from new embeddings for fine-tuning. All of these use GPT-3.5 for prediction. We make the following observations: (1) CLUSTERLLM consistently improves upon both embedders. For example, CLUSTER-

Type	Model	Bank77	CLINC(I)	MTOP(I)	Massive(I)	GoEmo	FewRel	FewNerd
Random	#GT Triplets	23	6	102	61	117	41	156
	Instructor	100	100	98.04	88.52	68.38	80.49	71.15
	GPT3.5	100	100	85.29	85.25	68.38	85.37	82.05
	$\Delta$	(+0)	(+0)	(-12.75)	(-3.27)	(+0)	(+4.88)	(+10.90)
Entropy-based	#GT Triplets	510	462	140	98	206	266	347
	Instructor	64.12	76.19	65.74	63.56	64.08	62.41	59.65
	GPT3.5 †	76.67	79.44	67.41	68.76	64.56	76.69	68.88
	$\Delta$	(+12.55)	(+3.25)	(+1.67)	(+5.20)	(+0.48)	(+14.28)	(+9.23)
	GPT4	79.41	80.74	76.04	74.84	61.65	87.22	82.13
$\Delta$	(+15.29)	(+4.55)	(+10.30)	(+11.28)	(-2.43)	(+24.81)	(+22.48)	

Type	Model	FewEvent	StackEx	Reddit	ArxivS2S	MTOP(D)	Massive(D)	CLINC(D)
Random	#GT Triplets	105	14	40	22	184	148	189
	Instructor	98.10	85.71	80	95.45	96.74	80.41	76.72
	GPT3.5	94.29	71.43	70	81.82	85.87	82.43	68.25
	$\Delta$	(-3.81)	(-14.28)	(-10)	(-13.63)	(-10.87)	(+2.02)	(-18.47)
Entropy-based	#GT Triplets	259	271	92	145	144	108	208
	Instructor	70.66	68.27	61.98	59.31	70.31	63.82	75.06
	GPT3.5 †	83.78	71.22	63.28	73.79	69.79	72.09	75.78
	$\Delta$	(+13.12)	(+2.95)	(+1.30)	(+14.48)	(-0.52)	(+8.27)	(+0.72)
	GPT4	85.71	79.70	67.71	77.93	72.40	70.28	74.58
$\Delta$	(+15.05)	(+11.43)	(+5.73)	(+18.62)	(+2.09)	(+6.46)	(-0.48)	

Table 3: Analysis on the triplet prediction accuracy († is used to produce results of CLUSTERLLM-I in Table 2). Red and green mean decreased or increased performances respectively. “#GT Triplets” means triplets that have ground truth (see Section 4.5 for details).

LLM-I increases the performance by 6.71% on FewRel. CLUSTERLLM-E increases the performance by 9.19 on Bank77. However, we do observe that on Massive(D) and CLINC(D), there are no improvements. (2) CLUSTERLLM outperforms deep clustering and self-supervise baselines. For instance, CLUSTERLLM-I surpasses self-supervise-I on most datasets except for two and it is also better than SCCL-I on 11 over 14 datasets. Furthermore, these improvements are consistent across both reported metrics. (3) Combined with the results in Appendix F, applying CLUSTERLLM iteratively is beneficial, emphasizing the potential of further improvements.

#### 4.5 Analysis on Triplet Prediction Accuracy

We attribute the improvements on clustering quality to more accurate triplet predictions. In Table 3, we show the accuracy on predicted triplets that have ground truth (exactly one positive and one negative choices based on ground truth) with two different sampling methods. Random triplet sampling uniformly samples three random instances as query and two choices, and we guarantee the two choices are different from the anchor by filtering. Furthermore, we also show a selection accuracy with Euclidean distances between embeddings as a comparison. We observe that, GPT-3.5/4 consistently improves upon Instructor on high entropy ex-

amples, demonstrating our hypothesis. In contrast, with random sampling, the ground truth triplets is significantly fewer and the accuracy gap is much smaller or even decreases performance.

#### 4.6 Ablation Study

**Clustering Quality.** We show ablation studies on CLUSTERLLM based on Instructor in Table 4. Specifically, we present results with 3 kinds of predictions on the same set of triplets for fine-tuning: GPT-3.5/4, replace triplet predictions of GPT-3.5 to ground truth on those triplets that have ground truth. We observe that GPT-4 marginally improves upon GPT-3.5 given the much higher cost. When provided with human labels, CLUSTERLLM-GT&GPT3.5 achieves the highest performance, which indicates the possibility for further improvement with more accurate predictions. We make similar observations for large-scale datasets in Table 6.

**Sampling Strategy.** In this section, we show ablation study on entropy-based sampling. In Figure 3, we observe that clustering accuracy increases when increasing entropies (or equally decreasing mean of interval) except for GoEmo. We make two hypothesis: (1) LLMs are much better than small embedders on harder instances. (2) high-entropy instances are generally more informative. In Table 4, we observe that training with randomly se-

Method	Intent Discovery								Emotion	
	Bank77		CLINC(I)		MTOPI(I)		Massive(I)		GoEmo	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
CLUSTERLLM-GPT3.5(random)	59.88 <sub>(2.56)</sub>	79.69 <sub>(0.63)</sub>	74.40 <sub>(0.91)</sub>	90.38 <sub>(0.20)</sub>	28.05 <sub>(1.69)</sub>	61.76 <sub>(0.62)</sub>	51.66 <sub>(2.41)</sub>	68.87 <sub>(0.73)</sub>	28.62 <sub>(1.95)</sub>	25.88 <sub>(1.02)</sub>
CLUSTERLLM-GPT3.5	70.77 <sub>(0.49)</sub>	85.07 <sub>(0.33)</sub>	82.77 <sub>(1.20)</sub>	93.88 <sub>(0.17)</sub>	35.84 <sub>(2.07)</sub>	73.52 <sub>(0.38)</sub>	<b>59.89</b> <sub>(2.05)</sub>	76.96 <sub>(0.54)</sub>	27.49 <sub>(1.25)</sub>	24.78 <sub>(0.56)</sub>
CLUSTERLLM-GPT4	69.71 <sub>(1.13)</sub>	84.68 <sub>(0.40)</sub>	81.91 <sub>(1.20)</sub>	93.76 <sub>(0.24)</sub>	34.48 <sub>(0.38)</sub>	73.57 <sub>(0.40)</sub>	59.10 <sub>(1.12)</sub>	76.59 <sub>(0.41)</sub>	27.41 <sub>(1.13)</sub>	23.77 <sub>(0.42)</sub>
CLUSTERLLM-GT&GPT3.5	<b>71.35</b> <sub>(1.97)</sub>	<b>85.12</b> <sub>(0.45)</sub>	<b>84.00</b> <sub>(1.04)</sub>	<b>94.34</b> <sub>(0.30)</sub>	<b>36.86</b> <sub>(0.42)</sub>	<b>75.36</b> <sub>(0.08)</sub>	59.27 <sub>(1.43)</sub>	<b>77.37</b> <sub>(0.54)</sub>	<b>30.91</b> <sub>(1.16)</sub>	<b>27.71</b> <sub>(0.46)</sub>

Method	Type Discovery						Topic Mining			
	FewRel		FewNerd		FewEvent		StackEx		Reddit	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
CLUSTERLLM-GPT3.5(random)	40.65 <sub>(0.89)</sub>	56.54 <sub>(0.30)</sub>	27.15 <sub>(0.53)</sub>	43.56 <sub>(0.49)</sub>	44.23 <sub>(1.72)</sub>	66.75 <sub>(0.87)</sub>	40.81 <sub>(0.94)</sub>	62.10 <sub>(0.34)</sub>	54.60 <sub>(2.23)</sub>	61.82 <sub>(1.64)</sub>
CLUSTERLLM-GPT3.5	47.94 <sub>(1.37)</sub>	62.43 <sub>(0.43)</sub>	34.75 <sub>(1.58)</sub>	51.03 <sub>(0.57)</sub>	46.17 <sub>(2.18)</sub>	70.73 <sub>(0.34)</sub>	47.21 <sub>(1.07)</sub>	66.78 <sub>(0.29)</sub>	56.79 <sub>(1.90)</sub>	63.87 <sub>(0.56)</sub>
CLUSTERLLM-GPT4	<b>48.96</b> <sub>(1.14)</sub>	<b>63.58</b> <sub>(0.39)</sub>	<b>37.54</b> <sub>(0.54)</sub>	<b>53.94</b> <sub>(0.27)</sub>	47.98 <sub>(1.45)</sub>	71.32 <sub>(0.70)</sub>	46.82 <sub>(0.78)</sub>	66.72 <sub>(0.11)</sub>	55.38 <sub>(0.37)</sub>	63.45 <sub>(0.49)</sub>
CLUSTERLLM-GT&GPT3.5	48.91 <sub>(1.20)</sub>	63.34 <sub>(0.47)</sub>	37.27 <sub>(0.61)</sub>	53.57 <sub>(0.32)</sub>	<b>48.12</b> <sub>(1.52)</sub>	<b>72.31</b> <sub>(0.84)</sub>	<b>47.55</b> <sub>(1.17)</sub>	<b>67.04</b> <sub>(0.31)</sub>	<b>58.33</b> <sub>(1.26)</sub>	<b>65.34</b> <sub>(0.51)</sub>

Method	Topic Mining				Domain Discovery				Avg	
	ArxivS2S		MTOPI(D)		Massive(D)		CLINC(D)		ACC	NMI
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
CLUSTERLLM-GPT3.5(random)	22.03 <sub>(0.28)</sub>	45.50 <sub>(0.16)</sub>	87.00 <sub>(2.27)</sub>	82.09 <sub>(1.54)</sub>	56.40 <sub>(2.35)</sub>	64.39 <sub>(1.12)</sub>	<b>60.27</b> <sub>(4.20)</sub>	<b>58.11</b> <sub>(2.93)</sub>	48.27	61.96
CLUSTERLLM-GPT3.5	<b>26.61</b> <sub>(0.48)</sub>	50.06 <sub>(0.26)</sub>	<b>93.53</b> <sub>(0.10)</sub>	<b>89.36</b> <sub>(0.11)</sub>	61.06 <sub>(1.91)</sub>	68.62 <sub>(0.90)</sub>	52.39 <sub>(1.84)</sub>	54.98 <sub>(2.08)</sub>	53.09	66.58
CLUSTERLLM-GPT4	26.16 <sub>(0.22)</sub>	50.06 <sub>(0.20)</sub>	92.04 <sub>(2.67)</sub>	88.39 <sub>(1.33)</sub>	60.16 <sub>(2.97)</sub>	67.98 <sub>(1.04)</sub>	57.45 <sub>(2.48)</sub>	59.98 <sub>(1.14)</sub>	53.22	66.98
CLUSTERLLM-GT&GPT3.5	26.14 <sub>(0.57)</sub>	<b>50.19</b> <sub>(0.33)</sub>	92.26 <sub>(3.62)</sub>	89.36 <sub>(1.42)</sub>	<b>61.65</b> <sub>(3.50)</sub>	<b>69.51</b> <sub>(1.50)</sub>	52.87 <sub>(2.63)</sub>	56.43 <sub>(1.21)</sub>	<b>53.96</b>	<b>67.64</b>

Table 4: Ablation study on clustering quality with Instructor as backbone and known granularity for evaluation. See more results with large-scale datasets in Table 6.

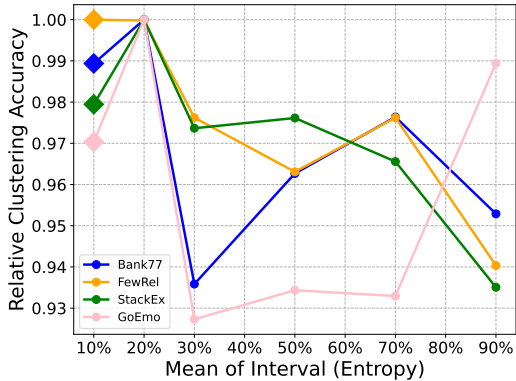


Figure 3: Relative clustering accuracy (divided by maximum for better aligning across datasets) of CLUSTERLLM-GPT3.5 with different range of entropy selected. x-axis shows the mean of interval where interval length is set to 20%. For example, “mean of interval= 50%” means  $\gamma_{\text{high}} = 40\%$  and  $\gamma_{\text{low}} = 60\%$  (see Section 3.1.1).  $\blacklozenge$  marks the setting for main experiments.

lected triplets even decreases performance, which demonstrates the cruciality of triplet sampling.

#### 4.7 Determining Cluster Granularity

In this section, we show the results for determining cluster granularity. We evaluate on a subset of 8 datasets including various cluster granularities with  $k_{\text{max}} = 200$  and  $k_{\text{min}} = 2$ . We compare with different methods that rely on clustering errors. For our methods, we show results with  $\lambda = \{1, 3\}$  (except for GPT-4 to reduce costs), which involve

198 & 594 pairs in total respectively. To simulate experts for providing demonstrations, we directly sample 16 pairs from small-scale datasets when  $\lambda = 3$  and then choose 2 positive and 2 negative as demonstrations. Notice that we use the same demonstrations for large-scale experiments. See more details in Appendix B.

We make several observations from Table 5 and Table 7: (1) Our methods have higher ranks. Most baseline methods predict similar number of clusters for domain and intent, while our methods can effectively distinguish between the two. For instance, on MTOPI(I)/(D) in Table 5, BIC predicts number of clusters 69/64 while our method (GPT-3.5,  $\lambda = 3$ ) predicts 92/18. (2) Increasing  $\lambda$  generally helps (MTOPI(D) in Table 5) but might not always make a large difference. (3) GPT-4 significantly improves upon GPT-3.5, probably due to its better understanding of demonstrations.

## 5 Related Works

**Clustering.** As a fundamental task in machine learning, clustering has been applied on diverse data types, including texts (Xu et al., 2015; Hadifar et al., 2019; Zhang et al., 2021a), images (Yaling Tao, 2021; Yang et al., 2016; Caron et al., 2018; Niu et al., 2020; Xie et al., 2016) and graphs (Huang et al., 2014; Chiang et al., 2019). Recent research has been shifted to deep cluster-



Method	Bank77	FewRel	Massive(I)	Massive(D)	MTOP(I)	MTOP(D)	CLINC(I)	CLINC(D)	Rank
GT #clusters	77	64	59	18	102	11	150	10	-
Silhouette (Rousseeuw, 1987)	118 (53.25)	10 (84.38)	38 (35.59)	41 (127.8)	11 (89.22)	11 (0)	172 (14.67)	163 (1530)	10
Elbow (Thorndike, 1953)	53 (31.17)	43 (32.81)	45 (23.73)	46 (155.6)	33 (67.65)	34 (209.1)	66 (56.00)	68 (580.0)	9
X-means (Pelleg et al., 2000)	69 (10.39)	30 (53.13)	32 (45.76)	34 (88.89)	28 (72.55)	27 (145.5)	130 (13.33)	135 (1250)	8
BIC (Goutte et al., 2001)	123 (59.74)	58 (9.38)	56 (5.08)	60 (233.3)	69 (32.35)	64 (481.8)	167 (11.33)	176 (1660)	7
ClusterSize (Zhang et al., 2021b)	86 (11.69)	71 (10.94)	72 (22.03)	90 (400.0)	82 (19.61)	85 (672.7)	105 (30.00)	106 (960.0)	6
Ours (GPT3.5, $\lambda = 1$ )	64 (16.88)	46 (28.13)	43 (27.12)	90 (400.0)	43 (57.84)	40 (263.6)	151 (0.67)	96 (860.0)	5
Ours (GPT3.5, $\lambda = 3$ )	64 (16.88)	46 (28.13)	52 (11.86)	37 (105.6)	92 (9.80)	18 (63.63)	142 (5.33)	107 (970.0)	2
Ours (GPT4, $\lambda = 1$ )	56 (27.27)	46 (28.13)	41 (30.51)	20 (11.11)	53 (48.04)	8 (27.27)	146 (2.67)	39 (290.0)	3
Ours (GT, $\lambda = 1$ )	100 (29.87)	91 (42.19)	42 (28.81)	18 (0)	41 (59.80)	11 (0)	141 (6.00)	39 (290.0)	4
Ours (GT, $\lambda = 3$ )	99 (28.57)	94 (46.88)	62 (5.08)	20 (11.11)	37 (63.73)	11 (0)	142 (5.33)	31 (210.0)	1

Table 5: Cluster granularity on small-scale datasets. Maximum & minimum number of clusters are set to 200 & 2. The results are shown in format of “[#clusters] (errors)”. “Rank” column is computed with 1-level ranking (Colombo et al., 2022) with inverse errors. “GT” is ground truth. See results for large-scale datasets in Table 7.

ing (Zhou et al., 2022) which focuses on how to leverage deep neural network in clustering. Zhou et al. (2022) has categorized deep clustering research into four types including multi-stage (Yaling Tao, 2021; Huang et al., 2014), iterative (Yang et al., 2016; Caron et al., 2018; Van Gansbeke et al., 2020; Niu et al., 2022; Chang et al., 2017; Niu et al., 2020), generative (Dilokthanakul et al., 2016) and simultaneous (Xie et al., 2016; Zhang et al., 2021a; Hadifar et al., 2019) depended on how representation learning and clustering modules interact with each other. Most recently, a concurrent work (Wang et al., 2023) studies a similar problem by assigning instances to different explanations proposed by LLMs. Another recent work IDAS (Raedt et al., 2023) directly encode the concatenation of sentence and abstractive summarizations from LLMs for clustering.

**Clustering with Relations.** Clustering with relations has been explored in different situations. To start with, spectral clustering (Donath and Hoffman, 1972; Cheeger, 1970) makes use of similarity matrix where each entry measures the similarity between a pair of data. More recently, several works in deep clustering utilize relational supervision (Yang et al., 2016; Niu et al., 2020; Van Gansbeke et al., 2020; Chang et al., 2017) via pseudo-labelling which could be noisy. Another line of works that is closely related to ours is constrained clustering. It usually incorporates pairwise must-link or cannot-link constraints (Basu et al., 2004; Wagstaff et al., 2001; Basu et al., 2008; Zhang et al., 2020; Manduchi et al., 2021). Nonetheless, these constraints are often sampled from labels as a prior which significantly limits its application in our scenario. In this work, we study how to utilize contemporary API-based LLMs to infer relations.

**Pre-trained Embedding Model.** Generic pre-

trained text embedding models (Reimers and Gurevych, 2019; Gao et al., 2021; Ni et al., 2022a,b) are widely applied in text similarity, classification, clustering and information retrieval. Recently, two embedding models E5 (Wang et al., 2022) and Instructor (Su et al., 2022) have shown superior performance on a popular benchmark (Muennighoff et al., 2022). Specifically E5 is pre-trained on web-scraped data pairs with contrastive objective. Instructor is pre-trained on 330 tasks with instructions. CLUSTERLLM aims at improving these models with LLMs.

## 6 Conclusion

In this paper, we study how to leverage API-based LLMs to guide small embedders for text clustering in order to benefit from high-level language capability of LLMs and user’s instructions on clustering. We propose to prompt LLMs with two kinds of sentence relationship tasks: triplet task and pairwise task. Triplet task chooses the sentence that is most similar with anchor combining with a perspective instruction from users. The predicted triplets are used for fine-tuning small embedders. Pairwise task judges whether a pair of sentences belong to the same category hinted by few-shot demonstrations, and then the predictions are used to determine cluster granularity with a consistency measure. Extensive experiments show that our proposed framework CLUSTERLLM can improve clustering quality and propose reasonable cluster granularity at a negligible cost. However, CLUSTERLLM still relies on the embedding model itself, which is inefficient and inapplicable on black-box embedding models. We encourage future works to explore the potential of model-free training such as constrained clustering.

## Limitations

We list several limitations of our work that we hope to be improved in the future:

**Reliance on pre-trained embedder.** To find the most informative data, we have to rely on a pre-trained embedder that can indicate the largest clustering assignment entropy. We hope that self-supervise triplets and LLM-predicted triplets can be combined to solve such an issue.

**Computational cost for fine-tuning.** Our initial idea is to utilize constrained clustering which is a light-weight algorithm that do not need to update small embedders. However, the inevitable unstable training will be heavily affected by the errors in LLM predictions. We make a compromise by introducing embedder into fine-tuning to temporarily solve the issue, but we hope to reduce the computational cost in a future work.

**Sub-optimal performance on domain discovery.** We notice that on domain discovery datasets such as Massive(D) and CLINC(D), the performance is usually sub-optimal compared with original Instructor embedding. We provide discussions on this issue in Appendix H.

## Ethics Statement

Our work employs LLMs which are accessed through OpenAI APIs. For some applications, uploading privacy-sensitive data is risky and might require efforts to remove sensitive information.

## Acknowledgements

Our work is sponsored in part by NSF CAREER Award 2239440, NSF Proto-OKN Award 2333790, NIH Bridge2AI Center Program under award 1U54HG012510-01, Cisco-UCSD Sponsored Research Project, as well as generous gifts from Google, Adobe, and Teradata. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes not withstanding any copyright annotation hereon.

## References

Wenbin An, Feng Tian, Qinghua Zheng, Wei Ding, QianYing Wang, and Ping Chen. 2022. General-

ized category discovery with decoupled prototypical network. *arXiv preprint arXiv:2211.15115*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

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.

Sugato Basu, Ian Davidson, and Kiri Wagstaff. 2008. *Constrained clustering: Advances in algorithms, theory, and applications*. CRC Press.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Kaidi Cao, Maria Brbic, and Jure Leskovec. 2022. [Open-world semi-supervised learning](#). In *International Conference on Learning Representations*.

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.

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. [Efficient intent detection with dual sentence encoders](#). In *Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020*. Data available at <https://github.com/PolyAI-LDN/task-specific-datasets>.

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.

Jeff Cheeger. 1970. A lower bound for the smallest eigenvalue of the laplacian, problems in analysis (papers dedicated to salomon bochner, 1969).

Qinyuan Cheng, Xiaogui Yang, Tianxiang Sun, Linyang Li, and Xipeng Qiu. 2023. [Improving contrastive learning of sentence embeddings from ai feedback](#).

Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. 2019. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 257–266.

- Pierre Colombo, Nathan Noiry, Ekhine Irurozki, and Stephan Clemencon. 2022. What are the best systems? new perspectives on nlp benchmarking. *arXiv preprint arXiv:2202.03799*.
- Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. **GoEmotions: A dataset of fine-grained emotions**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.
- Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 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, WSDM '20*, page 151–159, New York, NY, USA. Association for Computing Machinery.
- Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumar, and Murray Shanahan. 2016. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*.
- Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. **Few-NERD: A few-shot named entity recognition dataset**. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3198–3213, Online. Association for Computational Linguistics.
- William E Donath and Alan J Hoffman. 1972. Algorithms for partitioning of graphs and computer logic based on eigenvectors of connection matrices. *IBM Technical Disclosure Bulletin*, 15(3):938–944.
- Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Nataraajan. 2022. **Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages**.
- Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. **FewRel 2.0: Towards more challenging few-shot relation classification**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6251–6256, Hong Kong, China. Association for Computational Linguistics.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. **SimCSE: Simple contrastive learning of sentence embeddings**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Gregor Geigle, Nils Reimers, Andreas Rücklé, and Iryna Gurevych. 2021. **TWEAC: transformer with extendable QA agent classifiers**. *CoRR*, abs/2104.07081.
- Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.
- Cyril Goutte, Lars Kai Hansen, Matthew G Liptrot, and Egill Rostrup. 2001. Feature-space clustering for fmri meta-analysis. *Human brain mapping*, 13(3):165–183.
- Amir Hadifar, Lucas Sterckx, Thomas Demeester, and Chris Develder. 2019. **A self-training approach for short text clustering**. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 194–199, Florence, Italy. Association for Computational Linguistics.
- Xingwei He, Zhenghao Lin, Yeyun Gong, A Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, Weizhu Chen, et al. 2023. Annollm: Making large language models to be better crowdsourced annotators. *arXiv preprint arXiv:2303.16854*.
- William Hogan, Jiacheng Li, and Jingbo Shang. 2023. **Open-world semi-supervised generalized relation discovery aligned in a real-world setting**.
- Peihao Huang, Yan Huang, Wei Wang, and Liang Wang. 2014. Deep embedding network for clustering. In *2014 22nd International conference on pattern recognition*, pages 1532–1537. IEEE.
- Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.
- Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. **An evaluation dataset for intent classification and out-of-scope prediction**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.
- Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. **MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark**. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962, Online. Association for Computational Linguistics.
- Sha Li, Heng Ji, and Jiawei Han. 2022. **Open relation and event type discovery with type abstraction**.

- In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6864–6877, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Ting-En Lin, Hua Xu, and Hanlei Zhang. 2020. Discovering new intents via constrained deep adaptive clustering with cluster refinement. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.
- J MacQueen. 1967. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297. University of California Los Angeles LA USA.
- Laura Manduchi, Kieran Chin-Cheong, Holger Michel, Sven Wellmann, and Julia E Vogt. 2021. [Deep conditional gaussian mixture model for constrained clustering](#). In *Advances in Neural Information Processing Systems*.
- José Manuel Guaita Martínez, Patricia Carracedo, Dolores Gorgues Comas, and Carlos H Siemens. 2022. An analysis of the blockchain and covid-19 research landscape using a bibliometric study. *Sustainable Technology and Entrepreneurship*, 1(1):100006.
- Yutao Mou, Keqing He, Yanan Wu, Zhiyuan Zeng, Hong Xu, Huixing Jiang, Wei Wu, and Weiran Xu. 2022. [Disentangled knowledge transfer for OOD intent discovery with unified contrastive learning](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 46–53, Dublin, Ireland. Association for Computational Linguistics.
- Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.
- Fionn Murtagh and Pedro Contreras. 2011. Methods of hierarchical clustering. *arXiv preprint arXiv:1105.0121*.
- Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022a. [Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.
- Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022b. [Large dual encoders are generalizable retrievers](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Chuang Niu, Hongming Shan, and Ge Wang. 2022. Spice: Semantic pseudo-labeling for image clustering. *IEEE Transactions on Image Processing*, 31:7264–7278.
- Chuang Niu, Jun Zhang, Ge Wang, and Jimin Liang. 2020. Gatcluster: Self-supervised gaussian-attention network for image clustering. In *European Conference on Computer Vision (ECCV)*.
- Robert M Nosofsky. 2011. The generalized context model: An exemplar model of classification. *Formal approaches in categorization*, pages 18–39.
- OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- June Young Park, Evan Mistur, Donghwan Kim, Yunjeong Mo, and Richard Hoefler. 2022. [Toward human-centric urban infrastructure: Text mining for social media data to identify the public perception of covid-19 policy in transportation hubs](#). *Sustainable Cities and Society*, 76:103524.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Dan Pelleg, Andrew W Moore, et al. 2000. X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml*, volume 1, pages 727–734.
- Maarten De Raedt, Frédéric Godin, Thomas Demeester, and Chris Develder. 2023. [Idas: Intent discovery with abstractive summarization](#).
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Mamshad Nayeem Rizve, Navid Kardan, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. 2022a. Openldn: Learning to discover novel classes for open-world semi-supervised learning. In *European Conference on Computer Vision*, pages 382–401. Springer.
- Mamshad Nayeem Rizve, Navid Kardan, and Mubarak Shah. 2022b. Towards realistic semi-supervised learning. In *European Conference on Computer Vision*, pages 437–455. Springer.
- Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

- Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. [One embedder, any task: Instruction-finetuned text embeddings](#).
- Yiyou Sun and Yixuan Li. 2022. [Opencon: Open-world contrastive learning](#). In *Transactions on Machine Learning Research*.
- Robert Thorndike. 1953. Who belongs in the family? *Psychometrika*, 18(4):267–276.
- Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).
- Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. 2020. Scan: Learning to classify images without labels. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X*, pages 268–285. Springer.
- Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. 2022. Generalized category discovery. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. 2001. Constrained k-means clustering with background knowledge. In *Icml*, volume 1, pages 577–584.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.
- Zihan Wang, Jingbo Shang, and Ruiqi Zhong. 2023. [Goal-driven explainable clustering via language descriptions](#).
- Joe H Ward Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.
- Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR.
- Hui Xu, Yi Liu, Chi-Min Shu, Mingqi Bai, Mailidan Motalifu, Zhongxu He, Shuncheng Wu, Penggang Zhou, and Bing Li. 2022. Cause analysis of hot work accidents based on text mining and deep learning. *Journal of Loss Prevention in the Process Industries*, 76:104747.
- 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, Denver, Colorado. Association for Computational Linguistics.
- Kouta Nakata, Yaling Tao, Kentaro Takagi. 2021. [Clustering-friendly representation learning via instance discrimination and feature decorrelation](#). *Proceedings of ICLR 2021*.
- Jianwei Yang, Devi Parikh, and Dhruv Batra. 2016. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5147–5156.
- Muli Yang, Yuehua Zhu, Jiaping Yu, Aming Wu, and Cheng Deng. 2022. Divide and conquer: Compositional experts for generalized novel class discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14268–14277.
- Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen McKeown, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2021a. [Supporting clustering with contrastive learning](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5419–5430, Online. Association for Computational Linguistics.
- Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lyu. 2021b. Discovering new intents with deep aligned clustering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14365–14373.
- Hanlei Zhang, Hua Xu, Xin Wang, Fei Long, and Kai Gao. 2023. Usnid: A framework for unsupervised and semi-supervised new intent discovery. *arXiv preprint arXiv:2304.07699*.
- Hongjing Zhang, Sugato Basu, and Ian Davidson. 2020. A framework for deep constrained clustering-algorithms and advances. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part I*, pages 57–72. Springer.
- 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.
- Sheng Zhou, Hongjia Xu, Zhuonan Zheng, Jiawei Chen, Jiajun Bu, Jia Wu, Xin Wang, Wenwu Zhu, Martin Ester, et al. 2022. A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. *arXiv preprint arXiv:2206.07579*.
- Yiming Zhu, Peixian Zhang, Ehsan-Ul Haq, Pan Hui, and Gareth Tyson. 2023. Can chatgpt reproduce human-generated labels? a study of social computing tasks. *arXiv preprint arXiv:2304.10145*.

## A Details of Scaling up Hierarchical Clustering

A major drawback of hierarchical clustering is its  $\mathcal{O}(N^3)$  time complexity which makes the algorithm hard to be deployed on larger datasets. However, since we are only interested in a specific range of granularity in our scenario, the hierarchical clustering can start from an intermediate step. We address this issue by first running mini-batch K-means with  $k_{\max}$  and then run hierarchical clustering with current assignments as inputs. Specifically, we use agglomerative clustering with ward’s method (Ward Jr, 1963). We first calculate distances between each pair of clusters according to Murtagh and Contreras, 2011 and then provide them as inputs to *nearest neighbor chain* algorithm. Finally the returned hierarchy is combined with the K-means assignments to infer clusters.

## B More Details about Determining Cluster Granularity

Previous methods often employ clustering errors as a metric and they ignore user’s need on the granularity. **Silhouette coefficient** (Rousseeuw, 1987) indicates the clustering quality without ground truths, which exploits the inter-cluster distance with nearest clusters and the intra-cluster distance. We find the granularity by choosing the one with the best silhouette coefficient. **Elbow method** (Thorndike, 1953) is a heuristic method that plots the clustering error with respect to different levels of granularity in the hierarchy. And then the best granularity is determined with the largest elbow length. **X-means** (Pelleg et al., 2000) is a variation of K-means that starts with the lowest number of clusters, and then repeatedly attempt to split the clusters by running 2-means on them and evaluate with Bayesian Information Criterion (BIC) (Goutte et al., 2001). **BIC** (Goutte et al., 2001) calculates BIC for each of the granularity. **Cluster-Size** (Zhang et al., 2021b) uses a confidence threshold to filter small clusters starting from the maximum number of cluster. For all methods, we use the same fine-tuned embeddings (CLUSTERLLM-I in Table 2). The same cluster hierarchy is used (except for X-means that relies on K-means), which is either acquired from hierarchical clustering for small-scale or our proposed two-step method in Section 3.2 for large-scale. For our methods, the weight in F-beta score is set to 0.92 through empirical selection on Bank77. Because of the high

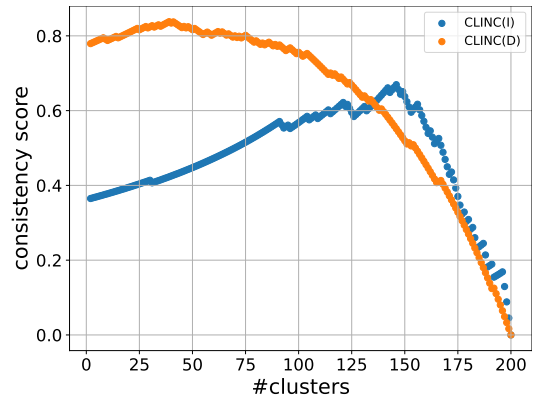


Figure 4: Consistency score v.s. various number of clusters with GPT-4 and  $\lambda = 1$ .

latency, results for Silhouette and X-means are not shown on large-scale datasets. After sampling 16 data pairs, we tend to choose positives with finer granularity or negatives with coarser granularity. We also consider the sentence length to minimize the cost. We use label names as justifications and we always put 2 positive before 2 negative (See Table 12 bottom).

## C Analysis for Determining Granularity

**Prompt Design.** We show the analysis results of prompt design for determining granularity in Table 8. We experiment with two settings: (1) remove justification for all demonstrations and only keep the “Yes” or “No” answer. (2) remove all demonstrations and any granularity-related words (such as domain)<sup>3</sup>. We observe that demonstrations are necessary and adding justifications have a positive impact.

**Visualization of Consistency Score.** We visualize consistency score with respect to the number of clusters. The consistency scores exhibit continuous variations and peak at the best number of clusters.

## D Details of Embedders and Fine-tuning

For all the experiments (including those with or without fine-tuning), we use large version of both Instructor and E5 (i.e. hkunlp/instructor-large & intfloat/e5-large). For Instructor, we use the same or similar prompt as original paper. See Table 10.

For fine-tuning, we adopt the same hyperparameters as in (Su et al., 2022), but modify the

<sup>3</sup>After removing, the prompt will be in the format “Determine whether the two sentences below belong to the same category.”

learning rate to  $2e-6$ , the maximum gradient steps to 3,840 for Instructor ( $\sim 15$  epochs) and 1,280 for E5, and batch size to 4. We choose this gradient in the beginning of our experiments by observing no performance increase after that on several datasets. Training is conducted with a single NVIDIA Quadro RTX 8000 GPU.

For SCCL-I(E), we change the maximum token length to 128 due to the limited compute resource<sup>4</sup>. We use the same learning rate  $2e-6$  as before for Instructor and  $2e-7$  for E5 since we found that the performance is unstable with large learning rate. Batch size is set to 16 and we evaluate representations with K-means after 200 iterations. Also notice that we do not interrupt prompts in Instructor during data augmentation.

## E Description of Datasets

**Bank77** (Casanueva et al., 2020) is a popular dataset in intent discovery that focuses on creating fine-grained intent categories for a single-domain, “banking”. **CLINC(I)** (Larson et al., 2019) is originally created for detecting utterances that falls outside of supported intents. The dataset also contains multiple domains, such as “travel”, “utility” and “work”. In this experiment, we discard all the out-of-scope utterances and only focus on in-domain ones. Moreover, we create a domain discovery dataset **CLINC(D)** that uses domains as labels. **Massive(I)/(D)** (FitzGerald et al., 2022) and **MTOP(I)/(D)** (Li et al., 2021) are both from MTEB (Muennighoff et al., 2022). Here “I” denotes intent and “D” for domain (or scenario). These datasets are originally used for classification but are adapted here for clustering. We also remove those intents with only a few instances and keep English-only data. For all datasets, we use the train & test sets as large- & small- scale datasets respectively. For **FewRel** (Gao et al., 2019) and **FewEvent** (Deng et al., 2020), we first randomly split datasets into train & test sets, and then sample from train set as large-scale and test set as small-scale. For **FewNerd** (Ding et al., 2021), we use the original train & test splits. For **StackEx**, **Reddit** (Geigle et al., 2021) and **ArxivS2S**, we combine all the splits into a single dataset and remove topics that only have few instances. Finally, the datasets are randomly splitted into large- & small- scale versions. To show the dataset balancy, we show the

<sup>4</sup>We also tried maximum token length 512 with smaller batch size, but the performance is not better.

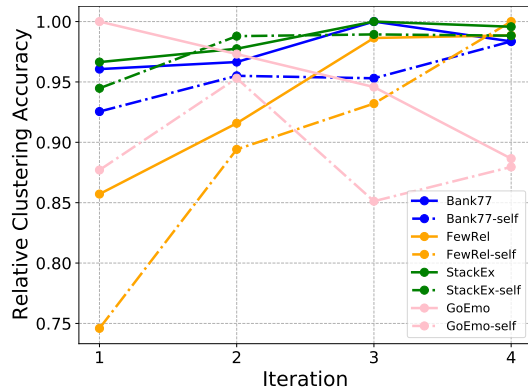


Figure 5: Relative clustering accuracy (divided by maximum for better aligning across datasets) of CLUSTERLLM-GPT3.5 over 4 iterations.

entropy of class distribution in Table 9.

## F Results of More Iterations

We show the results over 4 iterations of CLUSTERLLM in Figure 5. During iteration, we sample triplets from previously fine-tuned embedding space and continue to fine-tune the model with previous checkpoint as initialization. We also show the self-supervise results using the same checkpoint fine-tuned with GPT-3.5 predictions as initialization at each iteration. We observe that using GPT-3.5 predictions is almost always beneficial. The performance generally increases and saturate at the fourth iteration with the exception of GoEmo.

## G More Related Works

**Generalized Category Discovery (GCD).** GCD (Vaze et al., 2022; Lin et al., 2020; Zhang et al., 2021b, 2022; Mou et al., 2022; An et al., 2022) assume partial known classes with annotations which can also be used to infer user’s requirement on clustering. As an infant research area, most previous works employ pseudo-labelling, via optimal transport (Rizve et al., 2022b; Yang et al., 2022), similarity learning (Rizve et al., 2022a; Cao et al., 2022) or prototype-based learning (Sun and Li, 2022). Furthermore, new intent discovery (Zhang et al., 2022, 2021b, 2023; An et al., 2022; Lin et al., 2020) is proposed to study a similar research problem in the domain of intent detection. Most recently, Hogan et al., 2023 adapts the setting into relation type discovery. However, GCD relies on sufficient annotated and unlabeled data for training. In contrast, CLUSTERLLM seeks for minimal supervision and studies a setting with controlled

Method	Intent Discovery								Emotion	
	Bank77		CLINC(I)		MTOPI(I)		Massive(I)		GoEmo	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
Instructor	60.30 <sub>(2.39)</sub>	78.37 <sub>(0.78)</sub>	79.52 <sub>(1.96)</sub>	92.65 <sub>(0.42)</sub>	35.53 <sub>(1.05)</sub>	70.78 <sub>(0.74)</sub>	54.72 <sub>(2.00)</sub>	72.29 <sub>(0.61)</sub>	24.02 <sub>(1.11)</sub>	20.15 <sub>(0.19)</sub>
+self-supervise	61.48 <sub>(2.84)</sub>	79.28 <sub>(1.02)</sub>	81.87 <sub>(0.97)</sub>	93.55 <sub>(0.41)</sub>	35.27 <sub>(1.53)</sub>	71.88 <sub>(0.66)</sub>	58.30 <sub>(1.42)</sub>	73.73 <sub>(0.56)</sub>	24.34 <sub>(1.25)</sub>	21.17 <sub>(0.56)</sub>
+CLUSTERLLM-GPT3.5	65.47 <sub>(2.28)</sub>	81.60 <sub>(0.92)</sub>	82.29 <sub>(1.09)</sub>	93.91 <sub>(0.11)</sub>	36.80 <sub>(0.83)</sub>	73.16 <sub>(0.37)</sub>	57.70 <sub>(2.92)</sub>	74.24 <sub>(0.68)</sub>	25.23 <sub>(1.21)</sub>	22.19 <sub>(0.42)</sub>
+CLUSTERLLM-GT&GPT3.5	67.30 <sub>(1.35)</sub>	82.46 <sub>(0.17)</sub>	80.90 <sub>(1.58)</sub>	93.80 <sub>(0.25)</sub>	37.75 <sub>(1.64)</sub>	74.51 <sub>(0.51)</sub>	58.92 <sub>(1.80)</sub>	75.07 <sub>(0.43)</sub>	27.96 <sub>(2.59)</sub>	25.90 <sub>(0.69)</sub>

Method	Type Discovery						Topic Mining			
	FewRel		FewNerd		FewEvent		StackEx		Reddit	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
Instructor	41.38 <sub>(0.75)</sub>	53.97 <sub>(0.23)</sub>	29.62 <sub>(1.02)</sub>	42.83 <sub>(0.31)</sub>	41.42 <sub>(2.09)</sub>	61.13 <sub>(1.11)</sub>	46.76 <sub>(0.80)</sub>	61.20 <sub>(0.37)</sub>	55.04 <sub>(2.69)</sub>	59.55 <sub>(0.98)</sub>
+self-supervise	41.09 <sub>(0.99)</sub>	54.02 <sub>(0.29)</sub>	30.57 <sub>(0.18)</sub>	43.64 <sub>(0.26)</sub>	45.54 <sub>(1.70)</sub>	64.70 <sub>(0.50)</sub>	46.24 <sub>(0.46)</sub>	60.78 <sub>(0.26)</sub>	56.45 <sub>(1.59)</sub>	60.11 <sub>(0.39)</sub>
+CLUSTERLLM-GPT3.5	47.22 <sub>(0.89)</sub>	59.87 <sub>(0.21)</sub>	33.86 <sub>(1.19)</sub>	46.91 <sub>(0.52)</sub>	47.55 <sub>(1.51)</sub>	70.21 <sub>(0.59)</sub>	47.42 <sub>(1.35)</sub>	61.34 <sub>(0.30)</sub>	55.47 <sub>(2.44)</sub>	58.73 <sub>(1.09)</sub>
+CLUSTERLLM-GT&GPT3.5	49.20 <sub>(0.47)</sub>	61.20 <sub>(0.30)</sub>	34.85 <sub>(1.42)</sub>	48.59 <sub>(0.33)</sub>	46.90 <sub>(1.77)</sub>	71.51 <sub>(0.83)</sub>	48.12 <sub>(0.93)</sub>	62.04 <sub>(0.36)</sub>	57.95 <sub>(1.96)</sub>	61.39 <sub>(0.92)</sub>

Method	Topic Mining		Domain Discovery				Avg			
	ArxivS2S		MTOPI(D)		Massive(D)		CLINC(D)			
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI		
Instructor	24.55 <sub>(0.42)</sub>	39.86 <sub>(0.06)</sub>	85.01 <sub>(2.18)</sub>	83.96 <sub>(0.93)</sub>	56.11 <sub>(5.07)</sub>	61.86 <sub>(3.27)</sub>	51.74 <sub>(2.97)</sub>	55.28 <sub>(2.56)</sub>	48.98	60.99
+self-supervise	24.49 <sub>(0.75)</sub>	40.70 <sub>(0.16)</sub>	89.54 <sub>(4.56)</sub>	86.69 <sub>(2.09)</sub>	58.14 <sub>(3.88)</sub>	64.49 <sub>(2.28)</sub>	50.93 <sub>(4.11)</sub>	53.01 <sub>(4.48)</sub>	50.30	61.98
+CLUSTERLLM-GPT3.5	25.60 <sub>(0.51)</sub>	41.50 <sub>(0.14)</sub>	84.08 <sub>(3.34)</sub>	84.57 <sub>(1.97)</sub>	58.14 <sub>(3.97)</sub>	65.50 <sub>(1.35)</sub>	50.12 <sub>(4.13)</sub>	53.46 <sub>(2.13)</sub>	51.21	63.37
+CLUSTERLLM-GT&GPT3.5	25.29 <sub>(0.41)</sub>	41.81 <sub>(0.13)</sub>	84.99 <sub>(4.24)</sub>	86.59 <sub>(1.53)</sub>	58.55 <sub>(1.40)</sub>	65.66 <sub>(1.00)</sub>	57.08 <sub>(1.75)</sub>	58.35 <sub>(1.78)</sub>	52.55	64.92

Table 6: Ablation study on clustering quality for large-scale datasets.

Method	Bank77	FewRel	Massive(I)	Massive(D)	MTOPI(I)	MTOPI(D)	CLINC(I)	CLINC(D)	Rank
GT #clusters	77	64	59	18	102	11	150	10	-
Elbow (Thornðike, 1953)	50 (35.06)	38 (40.63)	47 (20.34)	45 (150.0)	35 (65.69)	35 (218.2)	64 (57.33)	71 (610.0)	7
BIC (Goutte et al., 2001)	183 (137.7)	183 (185.9)	148 (150.8)	141 (683.3)	169 (65.69)	170 (1445)	179 (19.33)	178 (1680)	8
ClusterSize (Zhang et al., 2021b)	90 (16.88)	95 (23.38)	79 (33.90)	81 (350.0)	84 (17.65)	83 (654.5)	105 (30.00)	109 (990.0)	5
Ours (GPT3.5, $\lambda = 1$ )	79 (2.60)	32 (50.00)	122 (106.8)	55 (205.6)	56 (50.00)	13 (18.18)	143 (4.67)	118 (1080)	6
Ours (GPT3.5, $\lambda = 3$ )	79 (2.60)	52 (18.75)	108 (83.05)	54 (200.0)	61 (40.20)	20 (81.81)	145 (3.33)	118 (1080)	4
Ours (GPT4, $\lambda = 1$ )	64 (16.88)	40 (37.50)	58 (1.69)	16 (11.11)	24 (76.47)	8 (27.27)	143 (4.67)	32 (220.0)	3
Ours (GT, $\lambda = 1$ )	59 (23.38)	79 (23.44)	61 (3.39)	17 (5.56)	27 (73.53)	11 (0)	148 (1.33)	32 (220.0)	2
Ours (GT, $\lambda = 3$ )	77 (0)	78 (21.88)	54 (8.47)	37 (105.6)	19 (81.37)	11 (0)	148 (1.33)	32 (220.0)	1

Table 7: Inferred granularity on large-scale datasets. The setting is the same as in Table 5.

Method	small-scale				large-scale			
	Massive(I)	Massive(D)	CLINC(I)	CLINC(D)	Massive(I)	Massive(D)	CLINC(I)	CLINC(D)
GT	59	18	150	10	59	18	150	10
Ours	41	20	146	39	58	16	143	32
w/o Justification	41	50	137	41	80	36	129	32
w/o Demonstration	41	64	141	108	77	74	120	105

Table 8: Prompt designs in determining granularity. We use the Instructor embedding with prompts, and we report results of GPT-4 with  $\lambda = 1$ .

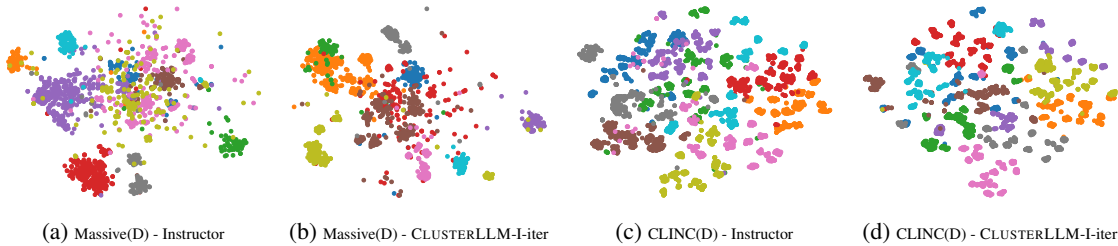


Figure 6: Scatter plots for t-SNE of embeddings. We select 10 classes from each datasets, denoted by colors.



Task	Name	entropy(small)	entropy(large)
Intent	Bank77	1.00	1.00
	CLINC(I)	1.00	1.00
	MTOP(I)	0.74	0.75
	Massive(I)	0.91	0.92
Type	FewRel	1.00	1.00
	FewNerd	0.82	0.82
	FewEvent	0.85	0.85
Topic	StackEx	0.98	0.98
	ArxivS2S	1.00	1.00
	Reddit	1.00	1.00
Emotion	GoEmo	0.91	0.91
Domain	CLINC(D)	1.00	1.00
	MTOP(D)	0.98	0.98
	Massive(D)	0.94	0.94

Table 9: Entropy of class distribution.

Dataset	Prompt
Bank77	Represent the bank purpose for retrieval:
CLINC(I)	Represent the sentence for retrieving the purpose:
FewRel	Represent the relation between two entities for retrieval:
FewNerd	Represent the entity type for retrieval:
FewEvent	Represent the event type for retrieval:
StackEx	Represent the question for retrieval:
ArxivS2S	Represent the science statement for retrieval:
GoEmo	Represent an emotion sentence for retrieval:
Massive(I)	Represent the sentence for retrieving the purpose:
MTOP(I)	Represent the sentence for retrieval:
Reddit	represent a reddit community title:
Massive(D)	Represent the scene for retrieval:
MTOP(D)	Represent a sentence:
CLINC(D)	Represent a sentence:

Table 10: Prompts for Instructor.

computation- & data- cost.

**LLMs as Annotators.** Recent instruction-tuned LLMs, such as ChatGPT, have been shown to have the ability to reproduce or improve human-generated labels (Gilardi et al., 2023; He et al., 2023; Zhu et al., 2023). Furthermore, several works dedicate to fine-tune models with feedbacks from LLMs (Cheng et al., 2023; Bai et al., 2022). This paper instead focuses on clustering tasks.

## H Sub-optimal Performance on Domain Discovery

We noticed that the performance of domain discovery (MTOP(D), Massive(D) and CLINC(D)) is barely improved or even decreased with CLUSTERLLM from original embedders (see Table 2). Furthermore, the ablation studies reveal that even with CLUSTERLLM-GT&GPT3.5, clustering performance is not as good as self-supervise or CLUSTERLLM-random (see CLINC(D) in Table 4 and MTOP(D) in Table 6). We also observe that, CLUSTERLLM-I-iter will further decrease the performance (see Massive(D) in Table 2). While we do not have rigorous explanations, one hypothesize is that the embedding space after fine-tuning

tends to be more compact than before and forming small cliques, making it better for clustering fine-grained clusters but not for coarse-grained clusters. We showcase scatterplots on two datasets with both Instructor and CLUSTERLLM-I-iter. It can be observed that the clusters in embedding space are tighter and more separated especially on CLINC(D).

## I Dataset Leakage

Since LLMs like ChatGPT are trained on web-scraped texts from internet, it is likely they already have access to our evaluation datasets during training. For example, the topic mining datasets use StackExchange, Reddit and Arxiv tags as labels which is freely available online. However, as observed in Table 3, the performance of triplet prediction on these datasets are often far from perfect. Furthermore, the other datasets like Bank77 are synthesized datasets which is not accessible during training. FewRel is collected from Wikipedia corpus but their labels are not easily accessible. Similarly, GoEmo is collected from Reddit but the emotion labels are not accessible during training. Thus, dataset leakage is not a primary concern of this paper.

Dataset	Prompt
Bank77	Select the banking customer utterance that better corresponds with the Query in terms of intent.
CLINC(I)	Select the customer utterance that better corresponds with the Query in terms of intent.
FewRel	Select the example that better corresponds with the Query in terms of relation type.
FewNerd	Select the example that better corresponds with the Query in terms of entity type.
FewEvent	Select the example that better corresponds with the Query in terms of event type.
StackEx	Select the StackExchange question that better corresponds with the Query in terms of topic.
ArxivS2S	Select the Arxiv paper title that better corresponds with the Query in terms of domain.
GoEmo	Select the sentence that better corresponds with the Query in terms of emotion expressed.
Massive(I)	Select the user utterance that better corresponds with the Query in terms of intent.
MTOP(I)	Select the user utterance that better corresponds with the Query in terms of intent
Reddit	Select the Reddit question that better corresponds with the Query in terms of topic.
Massive(D)	Select the user utterance that better corresponds with the Query in terms of scenario.
MTOP(D)	Select the user utterance that better corresponds with the Query in terms of domain.
CLINC(D)	Select the customer utterance that better corresponds with the Query in terms of domain.

Table 11: Prefix of prompts for triplet task. Notice while we use different prompts for domain and intent (such as CLINC(I) and CLINC(D)) in our experiments, they might be used interchangeably.

Dataset	Prompt
Triplet Task	Select the banking customer utterance that better corresponds with the Query in terms of intent.
	<p>Query: Should i reinstall the payment app?</p> <p>Choice 1: I've received my card so now I need to know how to sync it to the app.</p> <p>Choice 2: Can I still use the app if I switched phones?</p> <p>Please respond with 'Choice 1' or 'Choice 2' without explanation.</p>
Pairwise Task	<p>[Example1]</p> <p>Sentence 1: I would like to see the source of my money.</p> <p>Sentence 2: My source of funds need verified.</p> <p>Yes. Because both intents are verify source of funds.</p>
	<p>[Example2]</p> <p>Sentence 1: Is there a fee for topping up</p> <p>Sentence 2: What are the top up charges for US cards?</p> <p>Yes. Because both intents are top up by card charge.</p>
	<p>[Example3]</p> <p>Sentence 1: Can I reactivate my lost card that I found this morning in my jacket pocket?</p> <p>Sentence 2: how to activate card?</p> <p>No. Because Sentence 1 has intent card linking and Sentence 2 has intent activate my card.</p>
	<p>[Example4]</p> <p>Sentence 1: What will I be charged for a physical card?</p> <p>Sentence 2: My card is about to expire and I need to know how much it costs and how long ...</p> <p>No. Because Sentence 1 has intent order physical card and Sentence 2 has intent card ...</p> <p>Determine whether the intents of two banking customer utterances below belong to the same intent category using above examples.</p> <p>Sentence 1: \$1 extra has been charged on my statement, why is that?</p> <p>Sentence 2: Will it automatically top-up if there isn't much money left?</p> <p>Please respond with 'Yes' or 'No' without explanation.</p>

Table 12: One example from Bank77 on both triplet task and pairwise task.