Active Generalized Category Discovery with Diverse LLM Feedback

Henry Peng Zou[1*], Siffi Singh[2], Yi Nian[2], Jianfeng He[2], Jason Cai[2], Saab Mansour[2], Hang Su[2],
[1]University of Illinois Chicago, [2]AWS AI Labs
pzou3@uic.edu

Abstract

Generalized Category Discovery (GCD) is a practical and challenging open-world task that aims to recognize both known and novel categories in unlabeled data using limited labeled data from known categories. Due to the lack of supervision, previous GCD methods face significant challenges, such as difficulty in rectifying errors for confusing instances, and inability to effectively uncover and leverage the semantic meanings of discovered clusters. Therefore, additional annotations are usually required for real-world applicability. However, human annotation is extremely costly and inefficient. To address these issues, we propose DeLFGCD, a unified framework for generalized category discovery that actively learns from diverse and collaborative LLM feedback. Our approach leverages three different types of LLM feedback to: (1) improve instance-level contrastive features, (2) generate category descriptions, and (3) align uncertain instances with LLM-selected category descriptions. Extensive experiments demonstrate the superior performance of DeLFGCD over state-of-the-art models across diverse datasets, metrics, and supervision settings. Code is available at GitHub.

1 Introduction

The success of many deep learning models often heavily depends on some ideal assumptions, such as the availability of large amounts of labeled data, and the closed-world setting where unlabeled data shares the same set of pre-defined categories as labeled data (Zhong et al., 2021b; Zhang et al., 2022; An et al., 2023; Zou and Caragea, 2023). However, these assumptions often do not hold in many real-world scenarios. For example, in customer service intent detection, new types of inquiries may emerge over time (Tang et al., 2023; Zhang et al., 2024). Similarly, e-commerce product categorization faces ongoing introduction of novel product types (Gong et al., 2023; Zou et al., 2024a,b). In this work, we try to lift these assumptions by considering a more realistic and challenging setting: *Generalized Category Discovery* (GCD) (Vaze et al., 2022).

[*]Work done as an intern at Amazon.

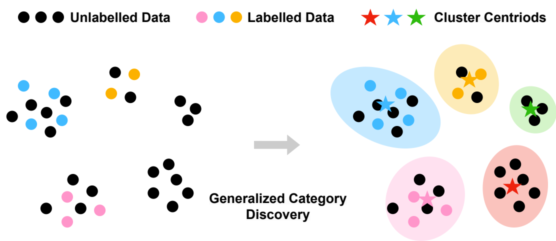

Figure 1: *Generalized Category Discovery* aims to automatically categorize unlabeled data by leveraging the information from a limited number of labeled data from known categories, while the unlabeled data may come from both known and novel categories.

As illustrated in Figure 1, GCD addresses a scenario where only a portion of the dataset is labeled and only a subset of categories is known. It aims to automatically categorize all unlabeled data, including instances from both known and novel categories, by leveraging information from a limited number of labeled instances (Vaze et al., 2022; Wen et al., 2023). This task is particularly relevant in realistic dynamic environments where new categories emerge over time, and manually labeling all data is impractical or prohibitively expensive (Ma et al., 2024a; Zhang et al., 2023b).

Previous work on GCD (Vaze et al., 2022; Pu et al., 2023) has primarily focused on applying contrastive learning to both labeled and unlabeled data to learn discriminative representations, followed by clustering methods like K-Means++ to discover both seen and novel categories. However, these approaches face inherent challenges: (1) Due to the lack of supervision, their models struggle to correct errors for confusing instances and categories. (2) Moreover, these methods often fail to uncover and leverage the semantic meanings of discovered

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics Volume 1: Long Papers, pages 7637–7658

clusters effectively.

To address the aforementioned issues and limitations, we propose DeLFGCD, a unified framework for generalized category discovery that actively learns from diverse and quality-enhanced LLM feedback. Our approach leverages LLMs in three effective ways: (1) Similar Instance Selection: We use LLMs to identify similar instances among ambiguous data points, refining embeddings through neighborhood contrastive learning (Zhong et al., 2021a). (2) Category Characterization: LLMs generate interpretable names and descriptions for newly discovered categories, making novel categories more accessible and meaningful. (3) Pseudo Category Selection and Alignment: We associate instance embeddings with LLM-selected category descriptions, fostering improved representation learning that considers category-instance relationships. By incorporating these diverse forms of LLM feedback, DeLFGCD achieves significant performance improvements over existing methods across multiple benchmark datasets. We also analyze the performance of DeLFGCD with varying numbers of known categories. Detailed ablation studies and analyses are provided to further understand each component and hyperparameter.

2 Related Work

Generalized Category Discovery. Generalized Category Discovery (GCD) (Vaze et al., 2022; Wen et al., 2023; An et al., 2024; Liang et al., 2024b) is a recently emerged task that addresses a realistic scenario where only limited labeled data is available and new categories may emerge. The goal is to automatically cluster all unlabeled data from both seen and novel categories (Bai et al., 2023; Zou et al., 2023). Pioneering works (Vaze et al., 2022; Pu et al., 2023) employ supervised and unsupervised contrastive learning to obtain discriminative embeddings, followed by clustering methods such as K-Means++. Wen et al. (2023); Bai et al. (2023) propose leveraging parametric classifiers and soft pseudo labels to mitigate model bias towards seen categories, thereby enhancing overall performance. Despite these advancements, traditional GCD methods that do not actively leverage additional human or LLM supervision signals (Zhang et al., 2021, 2022; Zhou et al., 2023; Zhang et al., 2023a; Liang and Liao, 2023; Raedt et al., 2023; Sung et al., 2023) still face significant challenges, including difficulty in rectifying errors for confusing instances and an inability to leverage the semantic meanings of discovered clusters (Ma et al., 2024a; An et al., 2024).

Active Learning and GCD. Active Learning (AL) (Ren et al., 2021; Ma et al., 2024b) aims to improve model performance by selecting and annotating a limited number of informative samples. Diverse sample selection strategies have been proposed, including uncertainty-based methods (Wang and Shang, 2014; Zhang et al., 2023b), diversity-based methods (Sener and Savarese, 2017; Ash et al., 2020) and hybrid methods (Agarwal et al., 2020; Huang et al., 2010). However, annotation in traditional AL is usually expensive and time-consuming (Cheng et al., 2023; Zhang et al., 2023b). Recent work in GCD, such as ALUP (An et al., 2024) and Loop (Liang et al., 2024b), has started to leverage LLMs feedback as a cost-effective alternative to human annotators to provide additional supervision signal for ambiguous data. However, these approaches utilize only a single type of LLM feedback in their model learning processes or do not account for feedback quality (An et al., 2024; Liang et al., 2024b,a). These limitations restrict their ability to effectively refine model predictions, particularly in ambiguous or challenging cases. To address these issues, we propose a unified framework for GCD that actively learns from diverse and quality-enhanced LLM feedback. To the best of our knowledge, this is the first work to consider both the diversity and quality of LLM feedback in the context of GCD. Table 12 provides a detailed comparison of our approach and closely related work in terms of LLM feedback diversity and quality.

3 DeLFGCD

3.1 Problem Formulation and Overview

Problem Formulation. Formally, assume we have an open-world dataset $\mathcal{D}$, comprising two subsets: a labeled set $\mathcal{D}_l = \{(x_i, y_i)\}_{i=1}^{N_l}$ contains only known categories, and an unlabeled set $\mathcal{D}_u = \{x_i\}_{i=1}^{N_u}$ contains both known and novel categories. *Generalized Category Discovery* (GCD) aims to accurately categorize all unlabeled data in D_u, having access to labels in $\mathcal{D}_l$. The total number of categories K is regarded as a known prior (Wen et al., 2023; An et al., 2024).

Overview. The main pipeline for DeLFGCD is shown in Figure 2. Both labeled data $\mathcal{D}_l =$

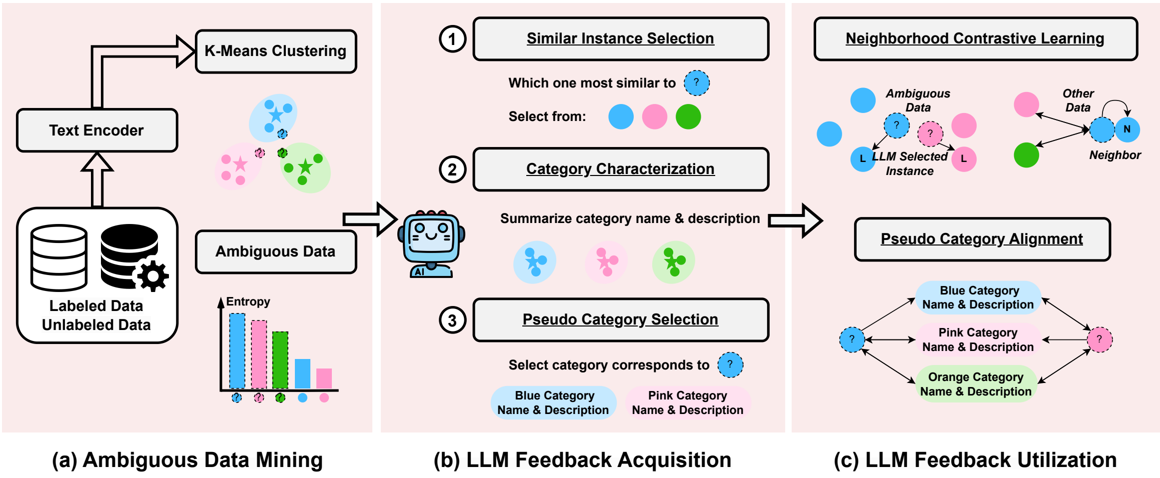

Figure 2: Pipeline of DeLFGCD. Both labeled and unlabeled data are first forwarded to a text encoder to extract features for K-Means++ clustering. Then we compute entropy and select instances with high entropy as ambiguous data to obtain LLM feedback for further refinement. Specifically, we query LLM to (1) select similar instances, (2) generate category descriptions and (3) assign pseudo categories to ambiguous data. Lastly, the three diverse feedback types are leveraged for model training via neighborhood contrastive learning and pseudo category alignment. During inference, we only utilize the text encoder and obtain final results via K-Means++ clustering on the extracted features. Illustration of the three types of LLM feedback with concrete examples is provided in Figure 3.

$\{(x_i, y_i)\}_{i=1}^{N_l}$ and unlabeled data $\mathcal{D}_u = \{x_i\}_{i=1}^{N_u}$ are first forwarded to a backbone encoder $f(\cdot)$ to extract initial features $h_i = f(x_i)$. These features are then passed through a projection head $g(\cdot)$ to learn contrastive features $z_i = g(h_i)$. We mine ambiguous data and leverage diverse LLM feedback to: (1) refine their contrastive features, (2) generate category descriptions, and (3) align ambiguous instances with LLM-selected category descriptions. During inference, we only utilize the backbone and obtain final results via K-Means++ clustering on the post-backbone features. Next, we explain each main component and how we acquire and utilize each type of LLM feedback in detail.

3.2 LLM Feedback 1: Similar Instance Selection and Utilization

Following (Zhang et al., 2023b; Liang et al., 2024b; An et al., 2024), we mine ambiguous data and potential positive candidates to query LLM for similar instance selection and refine their embeddings to be more contrastive.

Ambiguous Data Mining. We mine ambiguous instances based on entropy. Specifically, we first extract features from the backbone and perform clustering via K-Means++. The soft assignment of each sample x_i to each cluster k is calculated with the commonly used Student's t-distribution (Xie

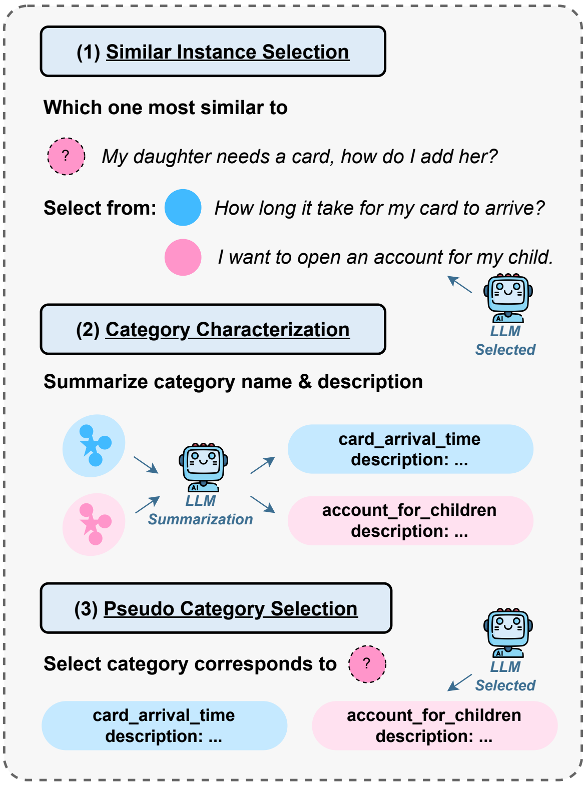

Figure 3: Illustration of three different types of LLM feedback utilized in DeLFGCD. Illustration of the whole pipeline is provided in Figure 2.

et al., 2016) as:

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

where h_i is the extracted feature, μ_k is the cluster center from K-Means++, α is the degree of freedom in the Student's t-distribution. We then compute entropy to measure the uncertainty for each sample:

$$\mathcal{H}_i = -\sum_{k=1}^{K} p_{ik} \log p_{ik} \quad (2)$$

The data with the highest entropy is regarded as ambiguous data and is used to form the query set:

$$\mathcal{Q} = \{x_i | \mathcal{H}_i \in top_v(\mathcal{H})\} \quad (3)$$

where v is a hyperparameter that determines the total amount of samples to query LLMs.

Similar Instance Selection. For each ambiguous data, we randomly sample M instances from each of its closest M clusters as candidates. We then query the LLM to choose the instance most similar to the ambiguous data as the *positive* instance for the following neighborhood contrastive learning. The corresponding prompt and examples are provided in Figure 6. For data other than ambiguous data, we do not query the LLM but randomly sample an instance from its k-nearest neighbors and regard it as the *positive* instance.

Neighborhood Contrastive Learning. Lastly, we finetune our model to refine embeddings to be more contrastive via the neighborhood contrastive learning loss (Zhong et al., 2021a):

$$\mathcal{L}^{\text{ncl}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{\left(z_i^T z_{p_i}/\tau\right)}}{\sum_{j \neq i} e^{\left(z_i^T z_j/\tau\right)}} \quad (4)$$

where p_i is the LLM-selected *positive* instance for the mined ambiguous data or the sampled positive neighbor for other data, z_j is the embedding of other in-batch data, τ is the temperature parameter.

3.3 LLM Feedback 2: Category Characterization

Upon identifying the ambiguous data in the previous step, we can query the LLM to obtain pseudo category labels for these samples and use them to enhance model performance. However, since the category labels of newly emerged categories remain unknown, it is infeasible to request LLMs to directly assign the selected data to unknown categories. Thus we propose ***Category Characterization*** to characterize the novel categories and enable LLM to assign pseudo category labels. Specifically: (1) After obtaining the K-Means++ clustering results on post-backbone features, we first select top-k samples closest to each cluster centroid as representative data. (2) Then for each cluster, we use its representative data to query LLM to generate a pair of cluster name and description. The corresponding prompt and example are provided in Figure 7. Comprehensive analyses are conducted to measure the performance of category characterization and the effect of different representative data selection strategies, which are presented in Appendix E. Notably, using only 10 representatives, we achieve 77% coverage and 71% semantic matching score.

3.4 LLM Feedback 3: Pseudo Category Selection and Alignment

Previous LLM-assisted GCD approaches (Liang et al., 2024b; An et al., 2024) only leverage instance-level relationship for model optimization. Here we take category-instance relationship into account and leverage category-instance LLM feedback to align ambiguous instances with LLM-selected positive category names & descriptions.

Specifically, for each selected ambiguous instance, we query the LLM to identify the most similar category name & description from those generated in the previous category characterization step. This selected category becomes the positive example, while all other unselected categories serve as negative examples. The corresponding prompt and example are provided in Figure 8. Then we use a contrastive loss to align the embedding of queried ambiguous data with the embedding of the selected positive category name & description:

$$\mathcal{L}^{align} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{\left(z_i^T d_p/\tau\right)}}{\sum_{j \neq i} e^{(z_i, d_j/\tau)}} \quad (5)$$

where d_p is the embedding of the selected positive category name & description, d_j is the embedding of other unselected negative category names & descriptions, and $sim(\cdot,\cdot)$ represents the cosine similarity function.

As a result, the overall loss that leverages labeled data and LLM feedback is formulated as:

$$\mathcal{L} = \mathcal{L}^{ce} + \mathcal{L}^{ncl} + \lambda \mathcal{L}^{align} \quad (6)$$

where λ is the weight of the alignment loss. Note that during model training, we incorporate the supervised cross-entropy loss $\mathcal{L}^{ce}$ on labeled data to enhance model learning from known categories.

KCR	Methods	CLINC ACC	CLINC ARI	CLINC NMI	BANKING ACC	BANKING ARI	BANKING NMI	StackOverflow ACC	StackOverflow ARI	StackOverflow NMI	Average
5%	GCD (CVPR 2022)	83.29	76.77	93.22	21.17	9.35	43.41	17.00	3.42	14.57	40.24
	SimGCD (ICCV 2023)	83.24	75.89	92.79	25.62	12.67	47.46	18.50	6.49	17.91	42.29
	Loop (ACL 2024)	84.89	77.43	93.26	21.56	10.24	44.77	18.80	5.76	17.54	41.58
	DeLFGCD (Ours)	**88.18**	**82.40**	**94.94**	**30.94**	**18.32**	**54.05**	**22.30**	**8.32**	**21.25**	**46.74**
10%	GCD (CVPR 2022)	82.04	75.95	93.33	59.09	46.34	76.22	75.40	56.01	72.66	70.78
	SimGCD (ICCV 2023)	84.71	77.08	93.27	60.03	47.80	76.53	77.10	57.70	72.30	71.84
	Loop (ACL 2024)	84.89	78.12	93.52	64.97	53.05	79.14	80.50	**62.97**	75.98	74.79
	DeLFGCD (Ours)	**88.71**	**83.29**	**95.21**	**67.99**	**57.30**	**82.23**	**82.40**	62.81	**79.67**	**77.73**
25%	DeepAligned (AAAI 2021)	74.07	64.63	88.97	49.08	37.62	70.50	54.50	37.96	50.86	58.69
	MTP-CLNN (ACL 2022)	83.26	76.20	93.17	65.06	52.91	80.04	74.70	54.80	73.35	72.61
	GCD (CVPR 2022)	82.31	75.45	92.94	69.64	58.30	82.17	81.60	65.90	78.76	76.34
	ProbNID (ACL 2023)	71.56	63.25	89.21	55.75	44.25	74.37	54.10	38.10	53.70	60.48
	USNID (TKDE 2023)	83.12	77.95	94.17	65.85	56.53	81.94	75.76	65.45	74.91	75.08
	SimGCD (ICCV 2023)	84.44	77.53	93.44	69.55	57.86	81.71	79.80	65.19	79.09	76.51
	CsePL (EMNLP 2023)	86.16	79.65	94.07	71.06	60.36	83.22	79.47	64.92	74.88	77.09
	ALUP (NAACL 2024)	88.40	82.44	94.84	74.61	62.64	84.06	82.20	64.54	76.58	78.92
	Loop (ACL 2024)	86.58	80.67	94.38	71.40	60.95	83.37	82.20	66.29	79.10	78.33
	DeLFGCD (Ours)	**91.51**	**87.07**	**96.27**	**76.98**	**66.00**	**85.62**	**84.10**	**71.01**	**80.90**	**82.16**
50%	DeepAligned (AAAI 2021)	80.70	72.56	91.59	59.38	47.95	76.67	74.52	57.62	68.28	69.92
	MTP-CLNN (ACL 2022)	86.18	80.17	94.30	70.97	60.17	83.42	80.36	62.24	76.66	77.16
	GCD (CVPR 2022)	86.53	81.06	94.60	74.42	63.83	84.84	85.60	72.20	80.12	80.36
	ProbNID (ACL 2023)	82.62	75.27	92.72	63.02	50.42	77.95	73.20	62.46	74.54	72.47
	USNID (TKDE 2023)	87.22	82.87	95.45	73.27	63.77	85.05	82.06	71.63	78.77	80.01
	SimGCD (ICCV 2023)	87.24	81.65	94.83	74.42	64.17	85.08	82.00	70.67	80.44	80.06
	CsePL (EMNLP 2023)	88.66	83.14	95.09	76.94	66.66	85.65	85.68	71.99	80.28	81.57
	ALUP (NAACL 2024)	90.53	84.84	95.97	79.45	68.78	86.79	86.70	73.85	81.45	83.15
	Loop (ACL 2024)	90.98	85.15	95.59	75.06	65.70	85.43	85.90	72.45	80.56	81.87
	DeLFGCD (Ours)	**94.53**	**90.79**	**97.12**	**80.26**	**70.40**	**87.65**	**89.40**	**78.92**	**85.04**	**86.01**

Table 1: Main results of DeLFGCD compared to baseline methods across different datasets and known category ratios (KCR). DeLFGCD outperforms both standard GCD approaches and the latest LLM-based work Loop (An et al., 2024), showing significant improvements especially on the challenging BANKING dataset and with limited known categories. Performance gains are observed across most KCRs, metrics, and datasets.

4 Experiments

4.1 Experimental Setup

Dataset and Metrics. We evaluate DeLFGCD on three standard generalized category discovery benchmarks: BANKING (Casanueva et al., 2020), CLINC (Larson et al., 2019) and StackOverflow (Xu et al., 2015). We use the same training, validation, and testing splits as previous work (Liang et al., 2024b; An et al., 2024). Descriptions, statistics and setup of all used datasets are provided in Appendix A. Following (Lin et al., 2019; Zhang et al., 2022; Liang et al., 2024b), we adopt the following three metrics for evaluation: Clustering Accuracy (ACC), Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI). The specific definitions are provided in Appendix B.

Baselines. We compare our model with two types of baselines: (1) Recent SOTA GCD method with LLM feedback: Loop (An et al., 2024), ALUP (Liang et al., 2024b). (2) Various standard GCD approaches without leveraging LLM feedback: GCD (Vaze et al., 2022), SimGCD (Wen et al., 2023), DeepAligned (Zhang et al., 2021), MTP-CLNN (Zhang et al., 2022), ProbNID (Zhou et al., 2023), USNID (Zhang et al., 2023a), CsePL (Liang and Liao, 2023). *gpt-4o-mini* is used as the default LLM to acquire LLM feedback.

Implementation Details. Following (An et al., 2024), we take the BERT-based-uncased model (Wolf et al., 2020) as our base model and pretrain it first on both labeled and unlabeled data from the current dataset using cross-entropy loss and masked language modeling loss (Devlin et al., 2019). We use the *[CLS]* token as initial text features for clustering. For contrastive learning, we employ a two-layer MLP to project the 768-d initial features into a 128-d space. A complete list of default hyper-parameters is provided in Appendix C. To reduce the computing and query cost, we follow the practice of (An et al., 2024): mine ambiguous data and update the query set every 5 epochs, and repeat the described approach 5 times.

4.2 Main Results

We summarize our main results under different known category ratios in Table 1 and describe our key findings below:

Compare with standard GCD approaches. It can be observed that our DeLFGCD consistently achieves stronger performance than standard GCD approaches. For example, on the most challenging BANKING dataset with 25% known category ratio, DeLFGCD significantly outperforms CsePL (Liang and Liao, 2023) by 5.35%/7.42%, and SimGCD (Wen et al., 2023) by 7.07%/9.54% in terms of ACC/ARI. These substantial improvements validate the effectiveness of our overall pipeline of active generalized category discovery from diverse LLM feedback.

Compare with state-of-the-art LLM-based GCD. DeLFGCD outperforms the recent state-of-the-art GCD method with LLM feedback, Loop (An et al., 2024) and ALUP (Liang et al., 2024b), across most settings and datasets. This verifies our motivation that multiple diverse feedback can be leveraged to effectively boost GCD model performance. Noteworthy is that, DeLFGCD can offer remarkable improvements over Loop even with extremely limited known categories: +9.38% ACC on BANKING and +3.5% ACC on StackOverflow with 5% known category ratio. Note that we re-ran the released code for GCD, SimGCD, and Loop to obtain their results, while the results for other baselines were retrieved from (Liang et al., 2024b).

Superior Overall Performance. The performance gain of DeLFGCD is generally maintained across different known category ratios, different metrics and datasets, showing the robustness of our approach. Furthermore, on the CLINC dataset, DeLFGCD achieves over 90% ACC with 25% KCR, demonstrating its potential for high-accuracy category discovery in these domains.

Datasets	BANK	CLINC	STACK	Average
Instance-Instance LLM Feedback: Similar Instance Selection				
Random Selection	0.037	0.070	0.060	0.056
Semantic Nearest	0.273	0.377	0.243	0.298
Naive LLM Selection	0.337	0.510	0.423	0.423
w. In-Context Demon	0.433	0.543	0.483	0.487
w. Filtering	0.466	0.562	0.563	0.530

Sample Types	Hardest Samples		Random Samples	
Datasets	BANK	CLINC	BANK	CLINC
Cluster-Instance LLM Feedback: Category Selection				
Naive LLM Selection	0.517	0.757	0.707	0.807
w. In-Context Demon	0.530	0.797	0.717	0.860
w. Filtering	0.557	0.859	0.847	0.943

Table 2: LLM feedback quality investigation (accuracy). Naively prompting LLM, as done in previous work, yields unsatisfactory results, though still much higher than random and semantic nearest selection. LLM feedback quality can be greatly enhanced with in-context demonstrations and filtering. More details in Section 5.

5 Ablation Study and Analysis

5.1 LLM Feedback Quality Investigation

In this section, we present our pilot investigation into the quality of diverse LLM feedback and introduce simple strategies to enhance its quality. We conduct experiments on three standard GCD datasets: BANKING (Casanueva et al., 2020), CLINC (Larson et al., 2019), and StackOverflow (Xu et al., 2015). Accuracy is used as the evaluation metric, calculated by comparing ground-truth answers with predictions. To better reflect real-world use cases, we evaluate performance on the 300 most challenging or ambiguous samples from each dataset. (Details on ambiguous data mining are provided in Section 3.2).

Table 2 summarizes the results of our investigation into two types of LLM feedback quality: (1) Instance-Instance LLM Feedback: Similar Instance Selection (An et al., 2024). For each ambiguous instance, we randomly sample M instances from each of its closest M clusters as candidates and then query the LLM to select the instance most similar to the ambiguous data as the positive instance. (2) Cluster-Instance LLM Feedback: Category Selection. For each ambiguous instance, we query the LLM to identify the most similar category from a given list of candidate categories. Our findings show that naively prompting the LLM, as done in previous work (Liang et al., 2024b; An et al., 2024), leads to unsatisfactory results, achieving an average accuracy of only 0.487 for similar instance selection. While this is still much higher than random

	CLINC			BANKING			Stackoverflow			
Method	**ACC**	**ARI**	**NMI**	**ACC**	**ARI**	**NMI**	**ACC**	**ARI**	**NMI**	**Average**
GCDLLMs	91.51	87.07	96.27	76.98	66.00	85.62	84.10	71.01	80.90	82.16
GCDLLMs w/o Filtering & Demo	91.02	86.32	96.06	76.14	65.60	85.45	83.40	68.93	80.10	81.45
- Cross-Entropy Loss	89.73	84.72	95.57	72.21	62.22	84.12	82.30	67.25	79.13	79.69
- Neighborhood Contrastive Learning	79.20	71.90	92.32	67.40	56.13	81.77	72.40	56.81	71.53	72.16
- Instance-Instance LLM Feedback	88.04	83.90	95.54	71.95	63.44	84.78	80.80	65.34	78.31	79.12
- Cluster-Instance LLM Feedback	88.18	82.98	95.00	72.56	62.23	83.86	79.60	62.71	76.90	78.22

Table 3: Ablation study of the different components of DeLFGCD. Each component contributes to the final performance, with Cluster-Instance LLM Feedback and Neighborhood Contrastive Learning being particularly impactful. Instance-Level LLM Feedback refers to Similar Instance Selection. Cluster-Instance LLM Feedback refers to both Category Characterization and Pseudo Category Selection and Alignment.

		CLINC			BANKING			Stackoverflow			
LLMs	**Cost**	**ACC**	**ARI**	**NMI**	**ACC**	**ARI**	**NMI**	**ACC**	**ARI**	**NMI**	**Average**
gpt-3.5-turbo	$0.5/ $1.5	89.29	84.47	95.54	73.38	64.01	85.20	81.40	68.91	80.24	80.27
gpt-4o-mini	$0.15 / $0.6	91.02	86.32	96.06	76.14	65.60	85.45	83.40	68.93	80.10	81.45
gpt-4o	$2.5 / $10	90.89	86.36	96.12	76.27	65.36	85.44	82.40	67.20	79.45	81.05
DeepSeek-V3	$1.25 / $1.25	91.91	87.28	96.28	76.07	65.93	85.65	83.40	69.20	80.02	81.75
Qwen-2.5-72B	$1.2 / $1.2	91.16	86.90	96.15	74.90	64.61	85.27	82.30	67.42	79.39	80.90
Llama-3.3-70B	$0.88 / $0.88	90.84	86.19	96.09	74.38	64.43	85.17	80.70	66.99	79.17	80.44
Qwen-2.5-7B	$0.3 / $0.3	90.13	85.82	95.89	73.21	62.89	84.22	80.60	66.62	79.31	79.85
Llama-3.2-3B	$0.06 / $0.06	88.80	83.07	94.89	72.60	62.00	84.27	80.10	66.50	78.90	79.01

Table 4: Different variants of LLMs. Open-source models, particularly *DeepSeek-V3*, can compete with closed-source alternatives. *gpt-4o-mini* offers a good balance between cost and performance, achieving the second-highest scores at a relatively low cost. Cost: pricing per 1M input tokens and output tokens.

selection or semantic nearest-neighbor selection and can aid model learning, improving feedback quality is more beneficial for model training. To address this, we adopt two simple strategies to enhance LLM feedback quality and mitigate noise. When querying the LLM, we incorporate in-context demonstrations from known categories and ask the LLM not only to provide an answer but also to output its confidence in the response, and we then filter out low-confidence answers. As shown in Table 2, these strategies substantially improve LLM feedback quality for both types of feedback.

5.2 Effectiveness of Each Component

To show the effectiveness of each component in DeLFGCD, we measure the performance after removing different parts across the three benchmarking datasets with 25% known category ratio in Table 3. We observe that the performance decreases after stripping each component, suggesting that all components in DeLFGCD contribute to the final performance. The performance of DeLFGCD drops most significantly after removing Neighborhood Contrastive Learning across all three datasets. This justifies the importance of leveraging LLMs to identify similar instances among ambiguous data points and refining embeddings through neighborhood contrastive learning. Besides, it can be seen that in-context demonstration and filtering low-confidence LLM feedback indeed help boost model performance, which aligns with our investigation and findings in Section 5.

Interestingly, compared to the Instance-Instance LLM Feedback - Similar Instance Selection, model performance drops more after removing Cluster-Instance LLM Feedback on most datasets and metrics. Specifically, removing Cluster-Instance LLM Feedback results in a 3.23% decrease in ACC, while removing Instance-Instance LLM Feedback leads to a 2.33% decrease. This finding underscores the benefits of aligning instance embeddings with corresponding LLM-selected category descriptions, which fosters improved representation learning by considering category-instance relationships. Not surprisingly, the removal of Cross-Entropy Loss also leads to a noticeable performance drop, particularly on the BANKING and StackOverflow datasets. This validates the importance of supervised learning on labeled data to enhance model

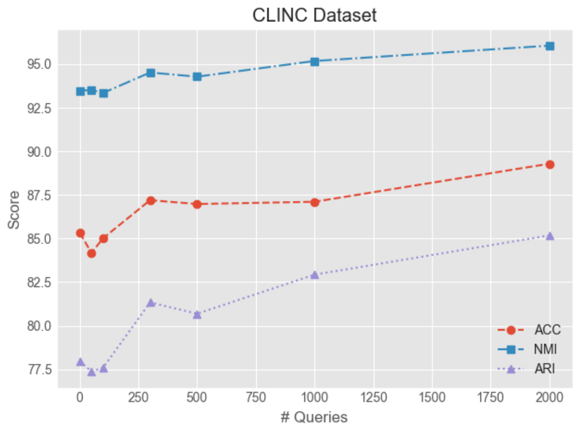

Figure 4: Influence of the number of query samples. Increasing the number of query samples generally leads to better performance.

Dataset	CLINC			BANKING		
Clustering Algorithm	ACC	NMI	ARI	ACC	NMI	ARI
K-Means++	91.16	95.85	85.99	75.88	85.62	65.51
HDBSCAN	90.53	95.79	85.51	75.36	85.17	64.36
GMM	89.82	95.67	84.92	75.03	85.49	64.50

Table 5: Results with other clustering algorithms.

Epoch	All	Head	Middle	Tail	Known	Novel
# Instances Range	168∼32	168∼134	133∼106	104∼32	-	-
0	52.44	56.20	45.3	55.56	67.5	47.5
5	65.13	71.5	60	63.98	78.16	60.86
10	71.98	72.6	75.6	68.06	82.37	68.58
15	74.87	75.8	76.5	72.5	83.68	71.98
20	75.06	71.6	75.4	77.96	82.63	72.59
25	76.14	79.1	76.2	73.33	80.92	74.57

Table 6: Detailed breakdown of accuracy results on different types of classes on imbalanced dataset.

performance on known categories and establish a strong foundation for subsequent LLM-enhanced learning processes. These results demonstrate that each component of DeLFGCD plays a crucial role in its overall performance.

5.3 Different Variants of LLMs

We investigate the performance impact of using different variants of LLMs in Table 4. Note that the pricing for GPT-series models is sourced from OpenAI[1], while the pricing for all open-source models is sourced from Together.AI[2]. It can be observed that DeepSeek-V3 achieves the highest overall average performance (81.75%), closely followed by the closed-source gpt-4o-mini (81.45%) and gpt-4o (81.05%), demonstrating that open-source models can compete with proprietary alternatives. Notably, gpt-4o-mini offers the best balance between cost and effectiveness, delivering competitive performance at a significantly lower cost ($0.15 / $0.6). In dataset-specific trends, DeepSeek-V3 excels in CLINC with the highest accuracy (91.91%), while gpt-4o-mini leads in BANKING (76.14% ACC), and both DeepSeek-V3 and gpt-4o-mini perform best on Stackoverflow (83.40% ACC). These results suggest that while OpenAI's closed-source models remain strong, open-source alternatives like DeepSeek-V3 are closing the gap and providing viable, high-performing options.

5.4 Influence of Query Sample Number

We study the influence of the number of query samples in Figure 4. We can observe that increasing the number of query samples generally leads to better performance, as more LLM feedback signals are available. For instance, on the CLINC dataset, the ARI increases from 77.5% to 85.0% as the number of query samples increases from 0 to 2000, and the trend shows that more performance gain can be obtained with more query samples. Yet, the performance gain starts to saturate on the BANKING dataset as the number of query samples reaches 500, we hypothesize this is because the BANKING dataset is more challenging and distinguishing ambiguous samples and categories becomes increasingly difficult as the number of samples increases.

5.5 Results with Other Clustering Algorithms

Our method is also compatible with other common clustering algorithms such as GMM (Gaussian Mixuture Models) or clustering algorithms with automatic K (category number) estimation (e.g., DBSCAN, HDBSCAN). We have added experiments using HDBSCAN and GMM in our DeLFGCD method. The results in Table 5 show that all three clustering algorithms achieve similar performance, and our method with K-Means++ achieves competitive performance, likely due to the refined representation space learned via LLM feedback and contrastive objectives.

5.6 Impact of class imbalance

This section adds a detailed breakdown of accuracy results on different kinds of classes on the imbalanced dataset BANKING (Known Category Ratio = 25%) across different training epochs. Head/Tail denotes the head/tail one-third of classes with the most/least training instances, ranging from 168~134/104~32. We can see that LLM feedback

[1]https://platform.openai.com/docs/pricing
[2]https://www.together.ai/pricing

Dataset	Input Tokens (M)	Output Tokens (M)	Total Tokens (M)	Total Cost ($)	Query Time (min)
BANKING	3.63178	0.05924	3.69102	0.58031	12.32
CLINC	5.81621	0.06585	0.58821	0.91194	14.80
StackOverflow	1.50342	0.05270	1.55612	0.25713	11.00

Table 7: Computational cost on three evaluated datasets.

	BANKING			CLINC		
Run	ACC	NMI	ARI	ACC	NMI	ARI
1	76.98	85.62	66.00	91.51	96.27	87.07
2	75.88	85.62	65.51	91.16	95.85	85.99
3	75.88	84.99	64.59	90.44	95.85	85.64
Avg	76.25	85.41	65.37	91.04	95.99	86.23
Std	**0.635**	**0.364**	**0.716**	**0.546**	**0.242**	**0.745**

Table 8: Mean performance and standard deviation.

can effectively improve the performance of both Head and Tail classes throughout the training process. More interestingly, we observe that the performance improvement on novel classes (+26.07%) is more obvious than the performance improvement on known classes (+14.42%), especially in the later training phase, as more ambiguous data can be mined from novel classes and can receive LLM feedback for refinement.

5.7 Computational Cost

In response, we have added a new table summarizing the actual token usage, total $ cost, and wall-clock LLM query time for DeLFGCD across all datasets. These measurements were collected using gpt-4o-mini under the default query budget of 500 ambiguous samples. As shown below, a full run requires 1.55M–5.88M tokens depending on dataset, which corresponds to $0.25-$0.91 per training run, and all LLM querying completes in under 15 minutes. These results demonstrate that DeLFGCD is extremely cost-efficient and practical for real-world deployment, with total cost less than $1 per run when using gpt-4o-mini.

5.8 Robustness Results

To demonstrate the robustness of our methods, we ran our method three times on both the BANKING and CLINC benchmarks and report the mean and standard deviation below. The low standard deviations (e.g., ACC standard deviation less than 0.64 on banking dataset) demonstrate that our approach yields robust performance across runs and still consistently achieves SOTA performance despite the inherent noise and stochasticity of LLM feedback.

	BANKING	CLINC
GPT4o (KCR=100%)	70.67	80.67
w. In-Context Demonstration	71.67	86.00
DeLFGCD (KCR=25%)	76.98	91.51
DeLFGCD (KCR=50%)	80.26	94.53

Table 9: Comparison with LLM-native baselines.

5.9 Comparison with LLM-native Baselines

In this section, we have added GPT-4o and GPT-4o with 10 in-context demonstrations as two additional LLM-native baselines, and the accuracy results are shown in the table below. Even when we provide all category names and corresponding in-context examples to GPT-4o (KCR = 100%), our DeLFGCD still consistently delivers improvements over these baselines while assuming only KCR = 25% and 50%, demonstrating the effectiveness of our method in combining both SLM training and LLM feedback.

6 Conclusion

This paper introduces DeLFGCD, a holistic framework that leverages diverse and quality-enhanced LLM feedback for generalized category discovery. Our approach addresses key limitations of existing methods, including insufficient supervision, lack of self-correction mechanisms for ambiguous data, and underutilization of semantic meanings in discovered categories. We integrate both instance-level and cluster-level LLM feedback into a contrastive learning framework, while aligning the embeddings of ambiguous instances with LLM-generated and selected category descriptions. Using three real-world datasets, we report new state-of-the-art results with DeLFGCD over existing methods across various supervision setups. We provide comprehensive ablation studies and analyses to understand each component in our framework.

Limitations

Our current framework is designed for textual data, which limits its applicability to other domains. In the future, we plan to extend it to vision and multimodal domains, exploring learning from multimodal large language models. Additionally, when using external LLMs, data privacy and security remain critical concerns that require ongoing vigilance. Leveraging open-source models such as DeepSeek, LLaMa could help mitigate these risks.

References

Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. 2020. Contextual diversity for active learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 137–153. Springer.

Wenbin An, Wenkai Shi, Feng Tian, Haonan Lin, QianYing Wang, Yaqiang Wu, Mingxiang Cai, Luyan Wang, Yan Chen, Haiping Zhu, and Ping Chen. 2024. Generalized category discovery with large language models in the loop. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 8653–8665, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Wenbin An, Feng Tian, Qinghua Zheng, Wei Ding, QianYing Wang, and Ping Chen. 2023. Generalized category discovery with decoupled prototypical network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12527–12535.

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2020. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*.

Jianhong Bai, Zuozhu Liu, Hualiang Wang, Ruizhe Chen, Lianrui Mu, Xiaomeng Li, Joey Tianyi Zhou, YANG FENG, Jian Wu, and Haoji Hu. 2023. Towards distribution-agnostic generalized category discovery. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Qinyuan Cheng, Xiaogui Yang, Tianxiang Sun, Linyang Li, and Xipeng Qiu. 2023. Improving contrastive learning of sentence embeddings from AI feedback. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11122–11138, Toronto, Canada. Association for Computational Linguistics.

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.

Shumin Deng, Ningyu Zhang, Luoqiu Li, Chen Hui, Tou Huaixiao, Mosha Chen, Fei Huang, and Huajun Chen. 2021. Ontoed: Low-resource event detection with ontology embedding. 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 2828–2839.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Shansan Gong, Zelin Zhou, Shuo Wang, Fengjiao Chen, Xiujie Song, Xuezhi Cao, Yunsen Xian, and Kenny Zhu. 2023. Transferable and efficient: Unifying dynamic multi-domain product categorization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 476–486, Toronto, Canada. Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809.

Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. 2010. Active learning by querying informative and representative examples. *Advances in neural information processing systems*, 23.

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)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Jinggui Liang and Lizi Liao. 2023. ClusterPrompt: Cluster semantic enhanced prompt learning for new intent discovery. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10468–10481, Singapore. Association for Computational Linguistics.

Jinggui Liang, Lizi Liao, Hao Fei, and Jing Jiang. 2024a. Synergizing large language models and pre-trained smaller models for conversational intent discovery. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14133–14147, Bangkok, Thailand. Association for Computational Linguistics.

Jinggui Liang, Lizi Liao, Hao Fei, Bobo Li, and Jing Jiang. 2024b. Actively learn from LLMs with uncertainty propagation for generalized category discovery. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7845–7858, Mexico City, Mexico. Association for Computational Linguistics.

Ting-En Lin, Hua Xu, and Hanlei Zhang. 2019. Discovering new intents via constrained deep adaptive clustering with cluster refinement. *ArXiv*, abs/1911.08891.

Shijie Ma, Fei Zhu, Zhun Zhong, Xu-Yao Zhang, and Cheng-Lin Liu. 2024a. Active generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16890–16900.

Shijie Ma, Fei Zhu, Zhun Zhong, Xu-Yao Zhang, and Cheng-Lin Liu. 2024b. Active generalized category discovery. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16890–16900.

Nan Pu, Zhun Zhong, and Nicu Sebe. 2023. Dynamic conceptional contrastive learning for generalized category discovery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7579–7588.

Maarten De Raedt, Fréderic Godin, Thomas Demeester, and Chris Develder. 2023. Idas: Intent discovery with abstractive summarization. *ArXiv*, abs/2305.19783.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. 2021. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40.

Ozan Sener and Silvio Savarese. 2017. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*.

Mujeen Sung, James Gung, Elman Mansimov, Nikolaos Pappas, Raphael Shu, Salvatore Romeo, Yi Zhang, and Vittorio Castelli. 2023. Pre-training intent-aware encoders for zero- and few-shot intent classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10433–10442, Singapore. Association for Computational Linguistics.

Yu-Chien Tang, Wei-Yao Wang, An-Zi Yen, and Wen-Chih Peng. 2023. RSVP: Customer intent detection via agent response contrastive and generative pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10400–10412, Singapore. Association for Computational Linguistics.

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. 2022. Generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7492–7501.

Dan Wang and Yi Shang. 2014. A new active labeling method for deep learning. In *2014 International joint conference on neural networks (IJCNN)*, pages 112–119. IEEE.

Xin Wen, Bingchen Zhao, and Xiaojuan Qi. 2023. Parametric classification for generalized category discovery: A baseline study. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16590–16600.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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

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

Feng Zhang, Wei Chen, Fei Ding, Meng Gao, Tengjiao Wang, Jiahui Yao, and Jiabin Zheng. 2024. From discrimination to generation: Low-resource intent detection with language model instruction tuning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 10167–10183, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

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

Hanlei Zhang, Hua Xu, Xin Wang, Fei Long, and Kai Gao. 2023a. A clustering framework for unsupervised and semi-supervised new intent discovery. *IEEE Transactions on Knowledge and Data Engineering*.

Yuwei Zhang, Zihan Wang, and Jingbo Shang. 2023b. ClusterLLM: Large language models as a guide for text clustering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13903–13920, Singapore. Association for Computational Linguistics.

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.

Zhun Zhong, Enrico Fini, Subhankar Roy, Zhiming Luo, Elisa Ricci, and Nicu Sebe. 2021a. Neighborhood contrastive learning for novel class discovery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10867–10875.

Zhun Zhong, Linchao Zhu, Zhiming Luo, Shaozi Li, Yi Yang, and Nicu Sebe. 2021b. Openmix: Reviving known knowledge for discovering novel visual categories in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9462–9470.

Yunhua Zhou, Guofeng Quan, and Xipeng Qiu. 2023. A probabilistic framework for discovering new intents. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3771–3784, Toronto, Canada. Association for Computational Linguistics.

Henry Zou and Cornelia Caragea. 2023. JointMatch: A unified approach for diverse and collaborative pseudo-labeling to semi-supervised text classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7290–7301, Singapore. Association for Computational Linguistics.

Henry Zou, Vinay Samuel, Yue Zhou, Weizhi Zhang, Liancheng Fang, Zihe Song, Philip Yu, and Cornelia Caragea. 2024a. ImplicitAVE: An open-source dataset and multimodal LLMs benchmark for implicit attribute value extraction. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 338–354, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Henry Zou, Gavin Yu, Ziwei Fan, Dan Bu, Han Liu, Peng Dai, Dongmei Jia, and Cornelia Caragea. 2024b. EIVEN: Efficient implicit attribute value extraction using multimodal LLM. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 453–463, Mexico City, Mexico. Association for Computational Linguistics.

Henry Zou, Yue Zhou, Weizhi Zhang, and Cornelia Caragea. 2023. DeCrisisMB: Debiased semi-supervised learning for crisis tweet classification via memory bank. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6104–6115, Singapore. Association for Computational Linguistics.

A Datasets

In this section, we provide descriptions, statistics and setup for all evaluated datasets.

Dataset Descriptions & Statistics BANKING (Casanueva et al., 2020) is a dataset of online banking queries labeled with 77 fine-grained customer intents, such as card arrival, card payment not recognized, pending cash withdrawal and pending card payment. StackOverflow (Xu et al., 2015) consists of technical questions covering a wide range of programming languages, frameworks, and software tools, and annotated with labels such as MATLAB, WordPress, Apache and Bash. CLINC (Larson et al., 2019) is a popular dataset for multi-domain intent detection. It contains 150 labels and covers diverse domains such as travel, utility and work. The statistics of these datasets are summarized in Table 10.

Dataset Setup Following (Liang et al., 2024b), we randomly select a specified ratio {5%, 10%, 25%, 50%} of all categories as known categories, denoted as known category ratio (KCR). For each known category, 10% of the data is selected to form the labeled dataset $\mathcal{D}_l$, while the remaining samples constitute the unlabeled dataset $\mathcal{D}_u$. Our ablation studies in the main text use a default KCR of 25%, while those in the appendix use a default KCR of 10%.

Dataset	Domain	# Categoreis	# Samples
BANKING	Banking	77	13,083
CLINC	Multi-Domain	150	22,500
StackOverflow	Programming	20	22,000

Table 10: Dataset statistics.

B Evaluation Metrics

We utilize three standard evaluation metrics to evaluate the GCD performance: ACC, ARI, and NMI. The accuracy metric (ACC) evaluates how well the predicted cluster assignments align with the ground-truth labels, and is formulated as:

$$ACC = \frac{\sum_{i=1}^{N} \mathbb{1}_{y_i = map(\hat{y}_i)}}{N}$$

where $\hat{y}_i$ represents the predicted label and y_i denotes the ground-truth label for each sample x_i. $map(\cdot)$ is a mapping function that uses the Hungarian algorithm (Kuhn, 1955) to establish a one-to-one correspondence between predicted labels $\hat{y}_i$ and their ground-truth counterparts y_i.

The Adjusted Rand Index (ARI) evaluates clustering quality by examining pairwise relationships between predicted and ground-truth cluster assignments. ARI can be calculated as:

$$ARI = \frac{\sum_{i,j}\binom{n_{ij}}{2} - [\sum_i \binom{u_i}{2}\sum_j \binom{v_j}{2}]/\binom{N}{2}}{\frac{1}{2}[\sum_i \binom{u_i}{2} + \sum_j \binom{v_j}{2}] - [\sum_i \binom{u_i}{2}\sum_j \binom{v_j}{2}]/\binom{N}{2}}$$

where $u_i = \sum_j n_{i,j}$ and $v_j = \sum_i n_{i,j}$, N represents the total sample count, and $n_{i,j}$ is the number of samples simultaneously belonging to the i^{th} predicted cluster and j^{th} ground-truth cluster.

The Normalized Mutual Information (NMI) metric measures the consistency between predicted and ground-truth clustering results by quantifying their mutual information. NMI is defined as:

$$NMI(\hat{y}, y) = \frac{2 \cdot I(\hat{y}, y)}{H(\hat{y}) + H(y)}$$

where $\hat{y}$ and y represent the predicted and ground-truth label sets respectively. The mutual information between these sets is denoted by $I(\hat{y}, y)$, while $H(\cdot)$ represents the entropy function.

C Hyperparameters

A complete list of default hyperparameters on all evaluated datasets is provided in Table 11.

	BANKING	CLINC	StackOverflow
# Query Samples v	500		
Similar Instance Selection Options M	5		
# Representatives for Category Characterization	10		
Pseudo Category Selection Candidate Ratio	0.5		
Batch Size	80		
Learning Rate	1e-05		
# Training Epochs	25		
Query Set Update Epoch Interval	5		
Alignment Wieght λ	{0.05, 0.1}		
Degree of Freedom α	1		
Temperature τ	0.07		
k-Nearest Neighbors k	50	50	500

Table 11: Complete list of default hyperparameters on BANKING, CLINC, StackOverflow datasets.

D Additional Studies and Results

D.1 Different Variants of Prompts

To investigate the impact of different prompt components on the performance of DeLFGCD, we conducted experiments with various prompt variants of Category Characterization, as shown in Table 13. Notably, we also add demonstrations of

Component	Others	ALUP	Loop	Ours
Diversity				
Instance-Instance Feedback ①	✗	✓	✓	✓
Cluster-Instance Feedback ②	✗	✗	✗	✓
Cluster-Level Feedback ③	✗	✗	✓	✓
(Learning w. Feedback ③)	✗	✗	✗	✓
Quality				
Feedback Quality Investigation	✗	✗	✗	✓
Quality Enhancement	✗	✗	✗	✓
Feedback Post-Filtering	✗	✗	✗	✓

Table 12: Comparison of closely related work in GCD. *ALUP* (Liang et al., 2024b) and *Loop* (An et al., 2024) use only *one* type of LLM feedback for *model learning* and *ignore feedback quality*. Other work in GCD does not leverage LLM feedback. We are the first to consider both LLM feedback diversity and quality in GCD.

some known category names in the full prompt. Our analysis reveals several key findings: (1) The full DeLFGCD prompt, which includes demonstrations, category names and descriptions, consistently achieves the best performance across all metrics on both the CLINC and Stackoverflow datasets. (2) Removing demonstrations from the prompt leads to a noticeable decrease in performance, particularly in ACC and ARI metrics. This suggests that providing examples helps the LLM better understand the task and generate more accurate category characterizations. (3) Omitting the category name generation results in the most significant performance drop on the CLINC dataset, indicating that concise category labels are particularly important for this dataset. (4) For the Stackoverflow dataset, removing the category description has a smaller impact compared to removing names or demonstrations, suggesting that category names might be more crucial for this technical domain. These results underscore the importance of carefully designed prompts in leveraging LLMs for generalized category discovery. The combination of demonstrations, category names, and descriptions in our prompts contributes to the overall effectiveness of DeLFGCD across different datasets.

	CLINC			Stackoverflow		
Prompt Variants	ACC	NMI	ARI	ACC	NMI	ARI
DeLFGCD	**87.69**	**95.02**	**82.27**	**82.40**	**79.67**	62.81
w.o. name	85.78	93.99	79.65	79.50	76.66	61.97
w.o. description	86.13	94.14	79.68	81.00	76.88	**63.22**
w.o. demonstration	86.22	94.55	80.82	79.40	76.63	61.76

Table 13: Different variants of prompts in LLM Category Characterization. Full prompt with demonstrations, category names, and descriptions achieves the best results.

D.2 Influence of Alignment Weight

In Table 14, we show the influence of alignment weight λ on model performance. We observe that the choice of alignment weight greatly influences the model's performance across different datasets. For the CLINC dataset, we observe a substantial improvement in performance as the alignment weight increases from 0 to 0.05, with ACC rising by 3.6 percentage points (from 89.64% to 93.24%) and ARI improving by 5.62 percentage points (from 83.82% to 89.44%). Similarly, for the Stackoverflow dataset, we see notable gains, particularly in ARI, which increases by 4.12 percentage points (from 74.20% to 78.32%) when the alignment weight is set to 0.05.

Nevertheless, setting the alignment weight as 0.05, 0.1 can generally offer us good performance improvements. Beyond this range, the performance boost tends to decline, with a particularly sharp drop observed when the weight is set to 1. This suggests that while the alignment between instance embeddings and category descriptions is crucial for improving model performance, it needs to be carefully balanced with other learning objectives. Too much emphasis on alignment (e.g., weight of 1) can lead to overfitting to the LLM-generated descriptions, potentially at the expense of other important features learned from the data.

	CLINC			Stackoverflow		
Alignment Weight	ACC	NMI	ARI	ACC	NMI	ARI
0	89.64	95.46	83.82	86.90	81.59	74.20
0.001	91.24	95.89	85.84	87.30	82.18	74.79
0.05	**93.24**	**96.90**	**89.44**	**89.00**	**84.43**	**78.32**
0.1	92.93	96.87	89.27	88.00	83.02	76.54
0.5	91.07	96.63	87.43	86.60	81.45	74.00
1	82.49	93.68	76.74	87.20	82.30	74.67

Table 14: Influence of alignment weight with 50% known category ratio. Setting the alignment weight as 0.05, 0.1 can generally offer us good performance improvements.

D.3 Effect of Candidate Ratio

We now analyze the effect of candidate ratio in Cluster-Instance LLM Feedback in Table 15. Candidate ratio refers to the number of candidate categories provided for pseudo category selection over the total number of categories in the dataset. We observe that including more candidates can generally improve the model performance on the BANKING dataset as a small number of candidates may potentially omit the most relevant category. However, the performance gets slightly decreased on the CLINC dataset when increasing the ratio from 0.75 to 1, we hypothesize this is because CLINC contains a much larger number of clusters (150) and incorporating all of their descriptions results in a huge prompt length, leading to decreased LLM pseudo category selection performance as the LLM may be overwhelmed by the large number of options.

	BANKING			CLINC		
Candidate Ratio	ACC	NMI	ARI	ACC	NMI	ARI
0.1	67.56	81.51	55.77	84.49	94.01	79.33
0.25	67.56	82.24	57.33	87.87	94.98	82.51
0.5	67.60	82.40	58.38	87.42	94.30	80.55
0.75	68.02	82.57	58.16	87.78	94.32	81.25
1	69.29	82.49	58.37	86.80	94.51	81.13

Table 15: Effect of candidate ratio in Cluster-Instance LLM Feedback. Candidate Ratio: the number of candidate categories provided for pseudo category selection over the total number of categories in the dataset.

D.4 Generalizability Results

To show the generalizability of DeLFGCD, we add experiments on four datasets with other types of data, and the accuracy results are shown in the table below. DeLFGCD consistently delivers improvements over the most recent SOTA method, Loop (An et al., 2024), showing that our approach also works well on datasets with other types of data and domains. We provide short descriptions of the added datasets below: (1) GoEmotions (Demszky et al., 2020) is a dataset of Reddit comments labeled with 27 emotions, such as amusement, fear, and gratitude. (2) Empathetic Dialogues (Rashkin et al., 2019) consists of conversations between a speaker and listener and is labeled with 32 fine-grained emotions. (3) FewEvent (Deng et al., 2021) is a few-shot event extraction dataset annotated with 34 event classes. (4) FewRel (Han et al., 2018) is a relation classification dataset consisting of labeled sentence-level relation instances across 64 relation types.

	EmpatheticDialogues		GoEmotions		FewRel		FewEvent	
# Classes	**32**		**27**		**64**		**34**	
Methods	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
Loop (ACL 2024)	36.40	44.85	32.77	33.73	60.94	75.41	42.80	62.36
DeLFGCD (Ours)	**44.05**	**49.88**	**37.57**	**37.14**	**65.35**	**77.07**	**44.75**	**65.43**

Table 16: Results on other types of data and domains.

D.5 Representative Failure Case

In this section, we provide a representative failure case below, which highlights common errors such as ambiguous cluster overlaps and mismatched category names:

"Given the following utterances and demonstrations of some known category names, return a category name and a short category description to summarize the common intent of these utterances in the format (Category Name: [category_name], Description: [description]) without explanation.

Demonstrations of Some Known Category Names:['failed_transfer', 'card_acceptance', 'activate_my_card', 'pending_card_payment', 'pending_top_up', 'transfer_not_received_by_recipient', 'reverted_card_payment?']

Utterance 1: how do i use a card to top up? Utterance 2: can i top up my card with other cards? Utterance 3: is it possible to top up by card? Utterance 4: if i want to topup by card, how do i do it? Utterance 5: am i able to add money into my account using my american express?

LLM Generated Category Name and Description: (Category Name: top_up_by_card, Description: Inquiries about using a card, particularly American Express, to add funds to an account.)

Query Ground Truth Labels:
['topping_up_by_card',
'supported_cards_and_currencies',
'topping_up_by_card',
'supported_cards_and_currencies',
'supported_cards_and_currencies']"

E Analyses on Category Characterization

This section provides comprehensive analyses to measure the performance of category characterization, and the effect of different representative data selection strategies.

Evaluation Metrics. We evaluate the performance of category characterization with the following four metrics, each of which aims to answer different questions or aspects of cluster interpretation: (1) Coverage Score: Do the interpreted/characterized clusters cover all ground-truth categories? (2) Uniformity Score: How evenly do the interpreted clusters cover all ground-truth categories if not covering all ground-truth categories? (3) SeMatching Score: How well does the generated category name & description match the ground-truth ground-truth categories in terms of semantics similarity? (4) Informative Score: An overall metric that considers both semantic similarity and uniformity between the interpreted clusters and ground-truth categories. The specific implementation of these metrics is provided in Figure 5.

The effect of different sampling strategies. We investigate three different sampling strategies for category characterization: (i) *Random*: randomly select n instances from each cluster as representatives; (ii) *Nearest to Center*: For each cluster, select the n instances that are nearest to the K-Means++ cluster center as representatives; (iii) *Sub-KMeans Centroids*: For each cluster, we first perform another K-Means++ clustering and produce n sub-level cluster centroids, which as used as representatives of the original cluster. Table 17 summarizes the evaluation results with these sampling strategies and different numbers of representative samples. Not surprisingly, both *Nearest to Center* and *Sub-KMeans Centroids* sampling strategies perform much better than *Random* sampling. Generally speaking, *Nearest to Center* can achieve slightly better performance than *Sub-KMeans Centroids*. We hypothesize the reason for this is that compared to *Sub-KMeans Centroids*, instances closest to the cluster center are most representative of the cluster and have more coherent semantic meaning, which makes it easier for LLM to produce more accurate and non-overlapping cluster summarization. Furthermore, we observe that including more representatives improves both Coverage and Uniformity Scores, while slightly decreasing the SeMatching Score. This is because having more representatives helps generate unique summarization, but also introduces more noise to make it hard to produce semantically accurate and consistent category descriptions.

Performance on different label settings. Table 18 summarizes the evaluation results with varying known category ratios and different numbers of labeled data for the *Nearest to Center* sampling strategy. It can be observed that increasing the known category ratio leads to consistently better performance, as more demonstrations of known category names can be provided in the prompt and thus LLM can generate more unique and accurate category names and descriptions. Besides, we can see that as more labeled data is added to known categories, most evaluation scores tend to first increase and then saturate. The SeMatching Score stays roughly the same, indicating that a few number of instances nearest to cluster centers is sufficient to produce semantically similar category names and descriptions with the ground-truth ones.

Performance with different LLMs. We now analyze the impact and costs of using different LLMs, or more specifically, different GPT models. Table 19 demonstrates the results. Interestingly, more expensive models, such as *gpt-4* and *gpt-4-turbo*, do necessarily not perform better than cheaper models, such as *gpt-3.5-turbo* and *gpt-4o-mini*, in category characterization. Besides, we can observe that *gpt-4o-mini* performs worse than *gpt-4o* with 1, 10 representatives in Coverage, Uniformity and Informative Scores, but achieves better SeMatching Score and on-par overall performance with 100 representatives. Furthermore, while being 100~200 times cheaper than *gpt-4*, *gpt-4o-mini* achieves on-par or better performances in most evaluated scores and settings. The competitive performance and extremely low price of *gpt-4o-mini* render it a good choice to be considered for category characterization.

F LLM Feedback Prompts & Examples

This section provides all the prompts we used to acquire the three diverse LLM feedback and corresponding examples: (i) Similar Instance Selection - Figure 6. (ii) Category Characterization - Figure 7. (iii) Pseudo Category Selection - Figure 8. Besides, we obtain LLM confidence scores by appending the following sentence in prompts: "Please also show your confidence by providing a probability between 0 and 1.".

Sampling Strategy	# Representatives	Coverage	Uniformity	SeMatching	Informative
Random	1	0.19	0.54	0.70	0.38
	3	**0.26**	**0.60**	0.68	**0.41**
	5	0.21	0.50	**0.71**	0.35
	10	0.23	0.56	0.67	0.37
	20	0.25	0.56	0.65	0.37
	50	0.23	0.54	0.65	0.35
	100	0.23	0.52	0.64	0.33
Nearest to Center	1	0.47	0.71	**0.70**	0.50
	3	0.62	0.81	0.69	0.56
	5	0.58	0.79	**0.70**	0.55
	10	0.58	0.81	**0.70**	0.56
	20	0.61	0.82	0.69	0.56
	50	0.57	0.82	0.68	0.56
	100	**0.65**	**0.87**	0.68	**0.59**
Sub-KMeans Centroids	1	0.45	0.71	**0.70**	0.49
	3	0.55	0.79	0.66	0.52
	5	0.62	0.82	0.66	0.54
	10	0.57	0.81	0.67	0.54
	20	0.60	0.84	0.66	0.55
	50	0.60	0.84	0.66	0.55
	100	**0.65**	**0.87**	0.65	**0.56**

Table 17: Effect of different sampling strategies and number of representative samples.

Known Category Ratio	Labeled Shot	Coverage	Uniformity	SeMatching	Informative
0.1	5	0.45	0.75	0.63	0.47
	10	0.58	0.80	**0.70**	0.56
	20	0.62	0.83	0.69	0.57
	50	**0.68**	**0.87**	0.69	**0.60**
	100	0.64	0.86	**0.67**	0.58
0.25	5	0.52	0.79	0.70	0.56
	10	0.66	0.87	0.69	0.60
	20	**0.70**	**0.89**	0.70	**0.62**
	50	0.69	0.88	0.70	**0.62**
	100	0.66	0.87	**0.71**	0.61
0.5	5	0.73	0.90	0.71	0.64
	10	0.77	0.92	0.71	0.66
	20	0.79	0.92	0.71	0.66
	50	**0.81**	**0.93**	**0.72**	**0.67**
	100	0.78	0.92	0.71	0.66

Table 18: Performance on different known category ratios and different numbers of labeled data.

# Representatives	LLMs	Cost	Coverage	Uniformity	SeMatching	Informative
1	gpt-3.5-turbo	$1/ $2	0.47	0.75	**0.70**	0.52
	gpt-4o-mini	$0.15 / $0.6	0.48	0.72	**0.70**	0.50
	gpt-4o	$5 / $15	**0.61**	**0.86**	0.66	**0.57**
	gpt-4-turbo	$10 / $30	0.45	0.74	0.65	0.48
	gpt-4	$30 / $60	0.45	0.71	**0.70**	0.49
10	gpt-3.5-turbo	$1/ $2	0.64	0.86	0.65	0.55
	gpt-4o-mini	$0.15 / $0.6	0.58	0.80	**0.70**	0.56
	gpt-4o	$5 / $15	**0.66**	**0.88**	0.67	**0.59**
	gpt-4-turbo	$10 / $30	0.62	0.86	0.64	0.55
	gpt-4	$30 / $60	0.65	0.86	0.65	0.56
100	gpt-3.5-turbo	$1/ $2	0.58	0.84	0.64	0.53
	gpt-4o-mini	$0.15 / $0.6	**0.66**	**0.87**	**0.67**	**0.58**
	gpt-4o	$5 / $15	0.61	0.86	0.66	0.56
	gpt-4-turbo	$10 / $30	0.62	0.86	0.60	0.52
	gpt-4	$30 / $60	0.64	**0.87**	0.61	0.53

Table 19: Performance with different LLMs.

```python
def evaluate_category_characterization(predictions, references):
    ## Compute Similarity Matrix and Matching Index
    """Compute the similarity between each cluster and ground truth category
    and match each cluster to the most similar ground-truth category."""
    # Load the Sentence Transformer model
    model = SentenceTransformer('sentence-transformers/all-MiniLM-L6-v2')

    # Compute embeddings for both lists of sentences
    prediction_embeddings = model.encode(predictions, convert_to_tensor=True)
    reference_embeddings = model.encode(references, convert_to_tensor=True)

    # Compute pairwise cosine similarity
    similarity_matrix = util.pytorch_cos_sim(prediction_embeddings, reference_embeddings)
    similarity_matrix = similarity_matrix.cpu().numpy()

    # return the index of the maximum value in each row
    max_indices = np.argmax(similarity_matrix, axis=1).tolist()
    print('Similarity Matrix: ', similarity_matrix)
    print('Matching Index: ', max_indices)

    ## Compute Diverse Metrics
    K = len(references)
    # Coverage Score: The percentage of unique interpreted category in the list
    coverage_score = len(set(max_indices)) / K
    print('Coverage Score: ', coverage_score)

    # Uniformity Score: how evenly the list covers all the items, max score is 1
    counts = [max_indices.count(i) for i in range(K)] # num of times each cateogry matched
    ratio = [count / len(max_indices) for count in counts] # ratio of each category matched
    uniformity_score = -sum([r * np.log(r) for r in ratio if r > 0]) / np.log(K) # entropy
    print('Uniformity Score: ', uniformity_score)

    # Semantic Matching Score: how well the list matches references regarding semantic similarity
    max_scores = np.max(similarity_matrix, axis=1)
    semantic_matching_score = np.mean(max_scores)
    print('Semantic Matching Score: ', semantic_matching_score)

    # Informativeness Score: consider both the semantic matching score and the uniformity score
    informativeness_score = semantic_matching_score * uniformity_score
    print('Informativeness Score: ', informativeness_score)

    return coverage_score, uniformity_score, semantic_matching_score, informativeness_score
```

Figure 5: Implementation of the four evaluation metrics for category characterization: Coverage Score, Uniformity Score, SeMatching Score and Informative Score.

Prompt & Example 1:

Select the customer utterance that better corresponds with the Query in terms of intent. Please respond in the format 'Choice [number]' without explanation, e.g., 'Choice 1', 'Choice 2', etc.

Query: is the cash withdrawal going to show?

Choice 1: how long will my atm withdrawal be pending?

Choice 2: when will my deposit post to my account?

Choice 3: how can i get a refund for an item i purchased but has not yet arrived?

Choice 4: why on earth is the exchange rate so bad? can i get a rate closer to the actual interbank rate?

Choice 5: i would like to cancel a purchase.

LLM Selected Similar Instance:

Choice 1

Prompt & Example 2:

Select the customer utterance that better corresponds with the Query in terms of intent. Please respond in the format 'Choice [number]' without explanation, e.g., 'Choice 1', 'Choice 2', etc.

Query: daughter needs card, how do i add her

Choice 1: how do i receive more physical cards

Choice 2: how long should i expect to wait for my card to arrive?

Choice 3: i need to transfer to my account and cant.

Choice 4: i want to open an account for my child.

Choice 5: i just got a new card how do i get it to start working?

LLM Selected Similar Instance:

Choice 4

Figure 6: LLM Feedback 1: Similar Instance Selection Prompt and Example. Click here to return to Section 3.2.

Prompt & Example:

Given the following utterances and demonstrations of some known category names, return a category name and a short category description to **summarize the common intent of these utterances** in the format (Category Name: [category_name], Description: [description]) without explanation.

Demonstrations of Some Known Category Names:

['automatic_top_up', 'failed_transfer', 'card_acceptance', 'activate_my_card', 'pending_card_payment', 'pending_top_up', 'transfer_not_received_by_recipient', 'reverted_card_payment?']

Utterance 1: the exchange rate was wrong in the foreign country i got cash in.

Utterance 2: i purchased something while traveling, and the exchange rate applied was wrong.

Utterance 3: when i received my cash, the exchange rate was wrong.

Utterance 4: the currency exchange rate was wrong for a purchase i made.

Utterance 5: i purchased something abroad, and the exchange rate that was applied was wrong.

...

LLM Generated Category Name and Description:

(Category Name: exchange_rate_issue, Description: Issues related to incorrect or wrong exchange rates applied during foreign transactions or currency exchanges.)

Query Ground Truth Labels:

['wrong_exchange_rate_for_cash_withdrawal', 'card_payment_wrong_exchange_rate', 'wrong_exchange_rate_for_cash_withdrawal', 'card_payment_wrong_exchange_rate', 'card_payment_wrong_exchange_rate']

Figure 7: LLM Feedback 2: Category Characterization Prompt and Example. Click here to return to Section 3.3.

Prompt & Example:

Select the category that better corresponds with the Query in terms of intent. Please respond in the format 'Choice [number]' without explanation, e.g., 'Choice 1', 'Choice 2', etc.

Query: help, i need to top up my account. where do i send a check?

Choice 1: (Category Name: transfer_into_account, Description: Inquiries about transferring or adding money to an account, particularly through bank transfers.)

Choice 2: (Category Name: topping_up_by_cash_or_cheque, Description: Inquiries about the process of adding funds to an account using a cheque.)

Choice 3: (Category Name: account_for_children, Description: Inquiries about opening or setting up accounts for children.)

Choice 4: (Category Name: funds_source_inquiry, Description: Inquiries about the origin or source of funds in the account.)

Choice 5: (Category Name: balance_not_updated_after_bank_transfer, Description: Inquiries about the account balance

...

LLM Selected Cluster Description:

Choice 2: (Category Name: topping_up_by_cash_or_cheque, Description: Inquiries about the process of adding funds to an account using a cheque.)

Figure 8: LLM Feedback 3: Pseudo Category Selection Prompt and Example. Click here to return to Section 3.4.