KSTC: Keyphrase-driven Sentence Embedding and Task Independent Prompting for Filling Slot in the Generation of Theme Label

Sua Kim* Taeyoung Jeong* Seokyoung Hong* Seongjun Kim*
Jeongpil Lee Du-Seong Chang Myoung-Wan Koo
Department of Artificial Intelligence, Sogang University, Korea
{lightwsrld, john428, hsy960925, ksj12035, jplee, dschang, mwkoo}@sogang.ac.kr

Abstract

Intent discovery in task-oriented dialogue is typically cast as single-turn intent classification, leaving systems brittle when user goals fall outside predefined inventories. We reformulate the task as multi-turn zero-shot intent discovery and present KSTC, a framework that (i) embeds dialogue contexts, (ii) performs coarse clustering, (iii) generates a predicted theme label for each cluster, (iv) refines clusters using the Large Language Model (LLM) with the predicted theme label, and (v) relocates utterances according to user's preferences. Because generating informative predicted theme label is crucial during the LLM-driven cluster refinement process, we propose the Task Independent Slots (TIS), which generates effective theme label by extracting verb and noun slot–value.

Evaluated on DSTC12 Track2 dataset, KSTC took first place, improving clustering and labeling quality without in-domain supervision. Results show that leveraging conversational context and slot-guided LLM labeling yields domain-agnostic theme clusters that remain consistent under distributional shift. KSTC thus offers a scalable, label-free solution for real-world dialogue systems that must continuously surface novel user intents. The code will be available at https://github.com/sogang-isds/KSTC.

1 Introduction

In task-oriented dialogue systems deployed in real-world services, it is essential to extract user intent from conversations (Ni et al., 2022). As customer needs diversify and business environments continue to evolve, the field of intent discovery has emerged, which aims to identify user intents from utterance collections that are either unlabeled or only partially labeled (Liu et al., 2021; Zhang et al., 2021; Liang and Liao, 2023). However, most prior work on intent discovery focuses on single-turn utterances, emphasizing the development of clustering algorithms designed to learn user utterance representations aligned with clustering objectives (Yin et al., 2021; Park et al., 2024).

Recent research has increasingly focused on intent classification in multi-turn dialogues, where users' intentions gradually become evident throughout a conversation. Such research highlights the need for robust intent discovery methods that can adapt to the dynamic characteristics of dialogues and diverse application environments (Liu et al., 2024a,b).

DSTC12 Track 2[1], formulates theme detection as a joint clustering and theme labeling for the input utterances. According to the task definition, intents are mapped to a fixed set of predefined labels, whereas themes are user-facing outputs, such as those presented to call center analysts, and thus require more flexible and expressive representations that can be tailored to user preferences. In theme detection, individual users may demand fine-grained analysis of specific themes or, conversely, prefer high-level overviews, depending on their business goals. Therefore, enabling personalized theme labeling based on user preferences is a crucial requirement in this task. Furthermore, the DSTC12 Track 2 task involves theme detection in a zero-shot, domain-agnostic environment, where themes emerge progressively through multi-turn dialogue.

To address this, we propose **KSTC** (Keyphrase-driven Sentence embedding and Task independent prompting for filling slot in the Generation of theme label), as shown in Figure 1, a novel framework that incorporates user preferences, refines clusters effectively, and generates theme labels that are both semantically coherent and practically useful.

In Stage 1, we generate keyphrases from each

[1] https://github.com/amazon-science/dstc12-controllable-conversational-theme-detection.git

Proceedings of the Twelfth Dialog System Technology Challenge (DSTC12), pages 44–73
August 28, 2025 ©2025 Association for Computational Linguistics

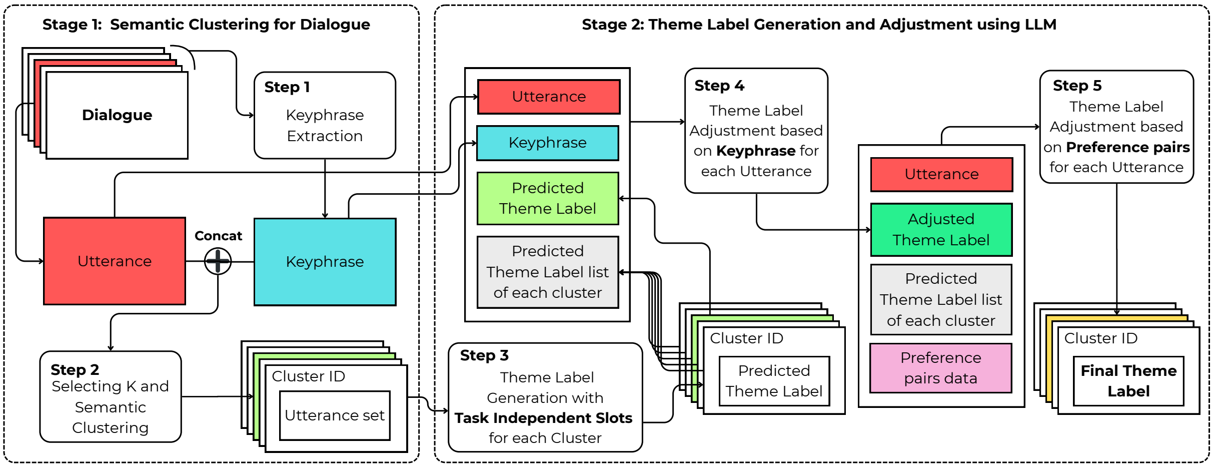

Stage 1: Semantic Clustering for Dialogue
Dialogue
Step 1
Keyphrase Extraction
Utterance
Concat
Keyphrase
Step 2
Selecting K and Semantic Clustering
Cluster ID
Utterance set
Stage 2: Theme Label Generation and Adjustment using LLM
Utterance
Keyphrase
Predicted Theme Label
Predicted Theme Label list of each cluster
Step 3
Theme Label Generation with Task Independent Slots for each Cluster
Step 4
Theme Label Adjustment based on Keyphrase for each Utterance
Cluster ID
Predicted Theme Label
Utterance
Adjusted Theme Label
Predicted Theme Label list of each cluster
Preference pairs data
Step 5
Theme Label Adjustment based on Preference pairs for each Utterance
Cluster ID
Final Theme Label

Figure 1: Overall Framework of KSTC. In Stage 1, we extracted keyphrases from each utterance and its surrounding context, and selected the most appropriate keyphrase using an LLM (Step 1). Each utterance was concatenated with its selected keyphrase and clustered accordingly (Step 2). In Stage 2, we generated theme labels for each cluster using Task Independent Slots (Step 3). We then refined the predicted theme labels by comparing them against the list of predicted labels within each cluster (Step 4). Finally, we generated the final theme label by adjusting the refined label based on preference pairs data (Step 5).

utterance and its context to extract conversational context in multi-turn dialogues. The keyphrases are embedded together with the corresponding utterance to perform semantic clustering. This process enables knowledge extension which cannot be fully represented by utterance-level embeddings, resulting in more semantically coherent and practical intent clusters.

In Stage 2, we utilize the Task Independent Slot methodology guided by LLM to extract key verbs and nouns associated with the intents of each utterance cluster. This enables the generation of effective predicted theme labels across diverse domains. We then use the predicted theme labels to refine the clusters through LLM based correction.

Our main contributions are as follows:

- **Semantic Clustering with Utterance and Keyphrase:** KSTC enhances semantic clustering by incorporating not only theme label annotated utterances but also up to three surrounding conversational turns. This allows for richer, more context aware clustering. To achieve this, we propose a semantic embedding method that leverages LLMs to extract keyphrases from the surrounding context of an utterance, which are then concatenated with the utterance prior to embedding.

- **Predicted Theme Label Generation for Initial Clusters:** To refine clusters effectively, we utilize the language understanding capabilities of LLMs to generate predicted theme labels for the initial clusters formed from semantic embeddings. These labels are created by filling Task Independent Slots using prompting technique, in which both fine-grained and broad semantic aspects of the cluster are captured.

- **Theme Label Adjustment Using Full Conversational Information:** We further improve the accuracy of the label by adjusting the theme label using not only the utterance's keyphrase context, but also the predicted theme labels of other groups and the current group's own label. This holistic use of conversational information enables more nuanced and accurate label refinement.

- **Incorporating Pre-defined Preferences:** Along with semantic information, KSTC also accounts for preference pairs data. These preferences allow for alternative clustering outcomes depending on user preference, even when the semantic content of the conversations is identical. This flexibility enables theme label adjustment to reflect both semantic structure and user's specific categorization needs.

2 Method

We use the NatCS (Gung et al., 2023) dataset introduced in DSTC12 Track 2, which consists of multiple dialogues, each composed of a sequence of utterances. A summary of the dataset statistics

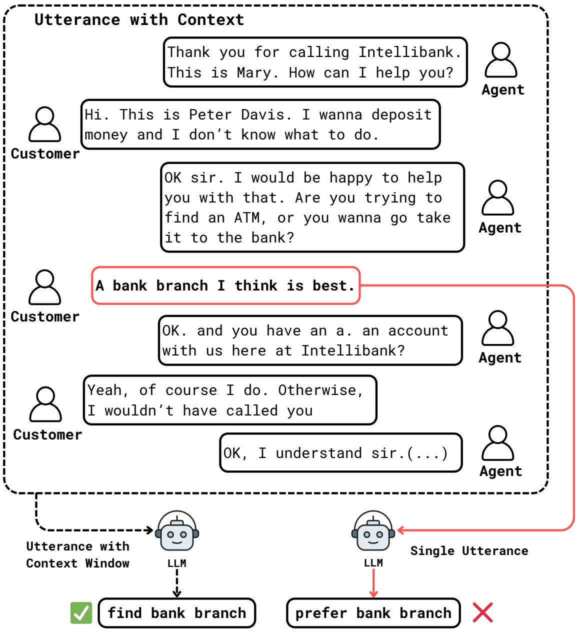

Utterance with Context
Thank you for calling Intellibank. This is Mary. How can I help you?
Agent
Hi. This is Peter Davis. I wanna deposit money and I don't know what to do.
Customer
OK sir. I would be happy to help you with that. Are you trying to find an ATM, or you wanna go take it to the bank?
Agent
A bank branch I think is best.
Customer
OK. and you have an a. an account with us here at Intellibank?
Agent
Yeah, of course I do. Otherwise, I wouldn't have called you
Customer
OK, I understand sir.(...)
Agent
Utterance with Context Window
LLM
LLM
Single Utterance
✅ find bank branch
prefer bank branch ✗

Figure 3: Comparison of keyphrases generated from single utterance and multi-turn context inputs

is presented in Appendix A. Among these, only a subset of utterances is annotated with theme labels. In this work, we focus exclusively on inferring theme labels for the annotated utterances.

Our overall framework consists of two stages, as illustrated in Figure 1. Here, Dialogue refers not to the entire conversation but to a localized context consisting of an annotated utterance and a few surrounding utterances within the same dialogue. Stage 1 performs semantic clustering by extracting keyphrases from each annotated utterance and its local context, followed by clustering based on these enriched semantic representations. Stage 2 generates theme labels for each cluster using an LLM, guided by the Task Independent Slot we designed, and then refines these labels at the utterance-level through additional LLM-based adjustments. We describe each step in detail in the following sections.

2.1 Step 1 : Keyphrase Extraction

Step 1 in Figure 1, illustrates the keyphrase extraction process. In natural language utterances, intent is often not explicitly stated but is instead contextually implicitly expressed. To address this, we generate keyphrases that aim to explicitly expose such contextually implicit intent, converting them into more interpretable and intuitive representations.

To improve the quality of the extracted keyphrases, we incorporate the surrounding dialogue context. Specifically, for each annotated utterance, we use its dialogue, which includes up to three preceding and three following utterances within the same dialogue. If the utterance appears near the beginning or end of a dialogue, fewer utterances may be included. To ensure thematic consistency, we exclude any surrounding utterances that are annotated with a different theme label.

We use an LLM to generate up to three candidate keyphrases for each dialogue. Subsequently, by verifying the candidate keyphrases, a single keyphrase that accurately reflects the core action of the utterance is selected. The prompt used for keyphrase generation is provided in Appendix D.

These context-aware keyphrases serve as intermediate semantic representations and are then used in both the clustering and labeling stages of our framework.

Figure 3 compares the keyphrases generated from single-utterance input versus those generated with multi-turn dialogue context (see Appendix B for additional examples). When the surrounding dialogue context (i.e., three preceding and following utterances) is provided (left), the LLM correctly generates "find bank branch", which accurately reflects the user's intent. In contrast, without the surrounding context (right), it generates "prefer bank branch", which fails to capture the intended meaning.

These results highlight the effectiveness of our context-aware approach. By leveraging additional conversational context, we are able to more accurately disambiguate intent and generate keyphrases that are both precise and semantically aligned.

2.2 Step 2 : Select K and Semantic Clustering

Step 2 in Figure 1 illustrates the semantic clustering process, which utilizes both the original utterances and the keyphrases extracted in the previous step.

To construct semantically rich representations for clustering, we first train an encoder following the ClusterLLM approach (Zhang et al., 2023), using the annotated utterances concatenated with the keyphrases as input. As suggested in their method, we repeat the training process for two iterations, which has been shown to improve the quality of semantic embeddings and enhance clustering performance.

During inference, we concatenate each annotated utterance with its corresponding keyphrase and encode the combined text using the trained encoder. This allows the resulting embedding to reflect both the original utterance and the enriched semantic intent captured by the keyphrase.

Once all utterance embeddings are obtained, we apply a clustering algorithm to group similar utterances. To determine the optimal number of clusters K, we adopt a hybrid strategy that combines both intrinsic and extrinsic evaluation signals. Specifically, we use the Silhouette score (Rousseeuw, 1987) and preference pairs data, which reflect user perspectives. The preference pairs consist of pairwise annotations indicating whether two utterances should belong to the same theme (Should-Link) or different themes (Cannot-Link). These preference pairs constitute part of the ground truth data and are provided to enable user-customized control over clustering granularity. An example of this dataset is provided in Appendix A.

Dataset	# of Clusters (Ground Truth)	# of Clusters (Predicted)
Banking	26	30
Finance	34	38
Insurance	27	38

Table 1: Number of clusters across the three datasets.

As input to this clustering step, we use the final keyphrase selected for each utterance. Each keyphrase is embedded using the 'text-embedding-3-large'(OpenAI, 2024), and K-Means clustering is performed for values of K ranging from 2 to 40. For each value of K, we compute a Combined Score (CS), defined as:

$$CS = w_{sil} \cdot S + w_{sl} \cdot \text{Acc}_{sl} + w_{cl} \cdot \text{Acc}_{cl} \quad (1)$$

Here, S denotes the silhouette score, reflecting both the number of clusters and the degree of intra-cluster cohesion. Acc_{sl} denotes the proportion of Should-Link pairs that were assigned to the same cluster, while Acc_{cl} represents the proportion of Cannot-Link pairs that were assigned to different clusters. In our experiments, the weights were set to $w_{sil} = 0.5$, $w_{sl} = 0.25$ and $w_{cl} = 0.25$. We select the value of K that achieves the highest Combined Score as the optimal number of clusters. Table 1 presents the selected number of clusters for each dataset. By leveraging semantically enriched keyphrases and user-driven constraints, our method enhances both the internal coherence and external validity of the resulting clusters.

Finally, we apply clustering algorithms (e.g., K-means, Agglomerative) to the utterance embeddings, using the optimal number of clusters selected based on the Combined Score. This clustering benefits from both surface-level features and the additional semantic cues introduced by the keyphrases.

2.3 Step 3 : Theme Label Generation with Task Independent Slots for Each Cluster

Step 3 in Figure 1 illustrates the methodology for generating theme labels for each cluster. To support this, we employ Task Independent Slots (TIS) that facilitate the extraction of task-related keywords from utterances in the same cluster. We guide the LLM with prompts to generate these slots, aiming to decompose tasks independently at a general domain level. Specifically, the LLM was guided by prompts to produce high-level action and conceptual categories commonly observed in real-world service conversations. The prompts used for generating these slots are provided in Appendix E. The generated slots consist of two complementary components that target distinct linguistic elements essential for intent identification: **Task Independent Verb Slots** and **Task Independent Noun Slots**.

The **Task Independent Verb Slots** define key action categories frequently observed in customer-agent dialogues, such as *require*, *request_info*, *cancel*, *confirm*, *update*, *inquire_issue*, and *recommend*. These verbs represent common types of user requests and interactions.

In contrast, the **Task Independent Noun Slots** encompass relevant entities and concepts pertinent to the tasks, including *product*, *service*, *account*, *schedule*, *personal_info*, *payment*, *status*, *issue*, *location*, *document*, and *indicator*.

For each cluster, we independently apply the Verb and Noun Slots to the aggregated utterances. We first analyze the semantic content of the utterances and extract verbs and nouns that correspond to the predefined slot categories. This procedure enables the identification of frequently occurring, slot-consistent verbs and nouns, facilitating an accurate characterization of the core actions and entities associated with the cluster's shared intent. To automate this process, we design a zero-shot prompt that enables an LLM to perform slots application and theme labeling.

Finally, we input the clustered utterances, the corresponding Verb and Noun Slots, and their extracted entities into the LLM to generate the final theme label for the cluster.

2.4 Step 4 : Theme Label Adjustment based on Keyphrase for each Utterance

Step 4 in Figure 1 illustrates the process of refining the predicted theme label for each utterance using additional semantic cues. For each utterance, we integrate the following information to facilitate adjustment: the utterance itself, its associated keyphrase, the initially predicted theme label inherited from its cluster, and the full set of theme labels predicted across all clusters.

The appropriateness of the assigned theme label with respect to the utterance's content and task context was evaluated using an LLM. If the label was deemed semantically appropriate, it was retained; otherwise, a more suitable label was selected from the list of predicted theme labels, considering the semantic alignment between the utterance, its keyphrase, and the available theme labels.

This verification and adjustment process aims to enable fine-grained, utterance-level theme labeling by leveraging the keyphrase as additional contextual information. A zero-shot prompt is employed to guide the LLM in assessing the correctness of labels and revising misassigned ones.

2.5 Step 5 : Theme Label Adjustment based on Preference pairs for each Utterance

Once keyphrase-based adjustments have been applied to all utterances, an additional adjustment step is conducted for utterances specified in the preference pairs data, as shown in step 5 in Figure 1. In the case of Should-Link, we consider not only direct pairs but also transitive relations among them. For example, if utterance u_j is in a Should-Link relation with u_k, and u_k is also in a Should-Link relation with u_l, then all three utterances u_j, u_k, u_l are expected to belong to the same cluster. Based on these transitive relationships, all connected utterances are assigned to the same group. For each group, candidate theme labels are collected from the keyphrase-adjusted clusters to which its member utterances belong. Then, we choose semantically consistent and representative theme labels using the LLM from among the candidate set. In the case of Cannot-Link, if a given pair of utterances is assigned to the same cluster, one of them must be reassigned. For example, for a Cannot-Link pair u_l and u_k, we consider the theme labels of their current clusters (after keyphrase-based adjustment), as well as those of other clusters to which neither utterance is currently assigned. Therefore, we identify utterances whose semantics are less aligned with the current theme label and select a more appropriate label by LLM from the candidate set. This process enables fine-grained cluster adjustment that faithfully reflects users' actual preferences.

This two-step adjustment process enables more accurate grouping by explicitly exposing contextual intent, particularly in cases where the original utterance lacks sufficient standalone information. The prompts used for steps 4 and 5 are detailed in Appendix F.

3 Experiments

3.1 Datasets

We evaluated our proposed method using the three development datasets and one test dataset provided by the organizers of DSTC12 Track 2. All four datasets are designed based on NatCS and were collected from four distinct domains: Banking, Finance, Insurance, and Travel. These datasets consist of multi-domain customer support dialogues between customers and agents.

3.2 Implementation detail

To compare performance with the number of clusters we selected, we used the ground truth number of clusters. This follows the convention used in prior studies (Zhang et al., 2023; Viswanathan et al., 2023; Liang et al., 2024).
For fine-tuning the embedding model, we used the AdamW optimizer with a batch size of 16. We used GPT-4o to generate and filter keyphrases and to generate theme labels. For clustering adjustment, we employed GPT-4.1 due to its overwhelming long context performance[2]. The full prompts are available in Appendix D.

3.3 Evaluation Metric

We focus on both the quality of clustering and the accuracy of label generation.

To evaluate the clustering quality, we compare the accuracy (ACC) and normalized mutual information (NMI) scores of our method with baselines.

For each cluster, the reference labels of its utterances will be compared to the label predicted for the cluster. We evaluate both semantic similarity and the inclusion of key terms using cosine similarity, ROUGE scores, and BARTScore.

Cosine similarity is a metric for measuring semantic similarity between the Sentence-BERT

[2] https://openai.com/index/gpt-4-1/

Method	Dataset					
	Banking		Finance		Insurance	
	NMI	ACC	NMI	ACC	NMI	ACC
Instructor (w/ KMeans) (Su et al., 2023)	65.32	54.79	65.18	51.48	56.57	43.42
Instructor (w/ Agglom.) (Su et al., 2023)	61.75	52.51	63.59	51.59	56.88	44.19
Instructor + Keyph. Clust. (w/ KMeans) (Su et al., 2023)	**74.29**	63.15	**75.01**	60.42	62.76	47.49
Instructor + Keyph. Clust. (w/ Agglom.) (Su et al., 2023)	73.88	**66.89**	74.15	**61.16**	**62.85**	**47.79**
CLUSTERLLM-I-iter (w/ KMeans)	75.75	60.82	76.66	60.64	63.12	50.50
CLUSTERLLM-I-iter (w/ Agglom.)	77.25	62.06	74.86	61.39	64.87	**52.29**
CLUSTERLLM-I-iter + Keyph. Clust. (w/ KMeans)	75.54	62.77	77.74	60.85	65.80	50.44
CLUSTERLLM-I-iter + Keyph. Clust. (w/ Agglom.)	**77.32**	**67.38**	**79.23**	**62.49**	**65.84**	52.06
KSTC	**81.68**	**78.34**	**81.91**	**63.94**	**70.24**	**57.31**

Table 2: Comparison of NMI and ACC across clustering methods and datasets. The KSTC results are based on CLUSTERLLM-I-iter with keyphrases using Agglomerative clustering. Best results are bolded.

(Reimers and Gurevych, 2019) embeddings of the reference and predicted labels. ROUGE scores (Lin, 2004) are N-gram overlap metrics that are effective for comparing short and concise sequences between the reference and predicted labels. Specifically, we compute ROUGE-1, ROUGE-2 and ROUGE-L scores. BARTScore (Yuan et al., 2021) is a metric designed to measure semantic similarity between a generated text and a reference text and is known to have a high correlation with human judgment. We use the pretrained bart-large-cnn[3] model, where higher score indicates greater semantic consistency between the two texts.

4 Results

4.1 Analysis of Stage 1 results

Table 2 presents a comparative analysis of the performance of various clustering algorithms under different conditions, measured by NMI and ACC.

Comparison of Initial Clustering Algorithms K-means exhibits high performance variability depending on the initialization of cluster centroids, whereas Agglomerative Clustering adopts a deterministic merging approach. Using the encoder trained with CLUSTERLLM-I-iter, Agglomerative Clustering demonstrated superior performance in all three datasets. In this setting, we used the Instructor-large as the pre-trained embedder. This can be interpreted as the trained embedder enhancing the merging criteria of Agglomerative Clustering, thereby better capturing the similarities among data points.

Performance Analysis of KSTC's Clustering We define the final KSTC method by applying keyphrase and preference adjustments after embedding the clustering method with the highest performance among existing approaches, CLUSTERLLM-I-iter+Keyph. Clust. (w/ Agglom.). KSTC achieves the highest performance in both NMI and ACC metrics. This improvement is attributed to the effective correction of ambiguous cluster boundaries when based solely on utterances and keyphrases, through the predicted theme labels generated by Task Independent Slots. In other words, our method integrates not only semantic information from the text but also information derived from external knowledge, enabling a more precise understanding of the intrinsic data structure and the formation of accurate clusters.

Effectiveness of Keyphrase Utilization Combining keyphrases extracted from conversational context with the previously introduced Agglomerative clustering, Table 2 demonstrates that the CLUSTERLLM-I-iter+Keyph. Clust. (w/ Agglom.) approach consistently achieves superior performance across various datasets. Specifically, compared to CLUSTERLLM-I-iter Clust. (w/ Agglom.), the keyphrase-enhanced model achieves an average improvement of 2.7% in ACC and 6.1% in NMI across all datasets. Furthermore, we employed t-SNE (Van der Maaten and Hinton, 2008) for visualization, as illustrated in Appendix G, our keyphrase-enhanced clustering method separates clusters more distinctly. Consequently, we propose CLUSTERLLM-I-iter+Keyph. Clust. (w/ Agglom.) as the initial clustering for KSTC. This indicates that keyphrases, which capture

[3] https://huggingface.co/facebook/bart-large-cnn

DataSet	#Clusters	Clustering Algorithm	Theme Label Generation	Evaluation Metric										
				Clustering		Cosine	BART	Rouge-1		Rouge-2		Rouge-L		Avg.
				NMI	ACC	Similarity	Score	Recall	Precision	Recall	Precision	Recall	Precision	Cosine Rouge
Banking		Baseline		0.5984	0.5149	0.5579	-6.5217	0.4208	0.3959	0.1525	0.1629	0.4181	0.3946	0.3575
	K=26	Initial clustering of KSTC	Baseline	0.7732	0.6738	0.6324	-5.2352	0.5391	0.5446	0.2563	0.2229	0.5272	0.5268	0.4642
		Initial clustering of KSTC	TIS	0.7778	0.6916	0.7098	-4.7616	0.6081	0.6556	0.2948	0.2888	0.5912	0.6334	0.5403
		KSTC (TIS)		**0.8213**	0.7301	0.7289	-4.6786	0.6309	**0.6802**	0.3044	0.2971	0.6133	**0.6570**	0.5588
	K=30	Initial clustering of KSTC	Baseline	0.7692	0.645	0.6316	-5.4410	0.5090	0.5216	0.1950	0.1840	0.4967	0.5036	0.4345
		Initial clustering of KSTC	TIS	0.7906	0.7252	0.7280	-4.6786	0.6196	0.6570	0.3350	0.3447	0.6028	0.6348	0.5603
		KSTC (TIS)		0.8192	**0.7570**	**0.7452**	**-4.6070**	**0.6393**	0.6780	**0.3448**	**0.3555**	**0.6218**	0.6549	**0.5771**
Finance		Baseline		0.6218	0.4979	0.5398	-6.3143	0.4717	0.4286	0.2387	0.2084	0.4417	0.3895	0.3883
	K=34	Initial clustering of KSTC	Baseline	0.7923	0.6249	0.5986	-5.4484	0.4884	0.5202	0.2773	0.2877	0.4829	0.5129	0.4526
		Initial clustering of KSTC	TIS	0.7923	0.6249	0.6918	-4.2393	0.6716	0.6601	0.4623	0.4316	0.6716	0.6601	0.6070
		KSTC (TIS)		0.8222	0.6481	0.6997	-4.1861	0.6820	0.6699	0.4701	0.4387	0.6820	0.6699	0.6160
	K=38	Initial clustering of KSTC	Baseline	0.7914	0.6377	0.6091	-5.4215	0.4967	0.5182	0.2771	0.2841	0.4907	0.5099	0.4551
		Initial clustering of KSTC	TIS	0.7954	0.6441	0.6951	-4.1185	0.7043	0.6681	**0.4812**	0.4487	0.7043	0.6681	0.6243
		KSTC (TIS)		**0.8302**	**0.6771**	**0.7022**	**-4.0637**	**0.7109**	**0.6987**	**0.4812**	**0.4531**	**0.7109**	**0.6987**	**0.6365**
Insurance		Baseline		0.5173	0.3930	0.4221	-7.0239	0.2673	0.2294	0.1062	0.0703	0.2607	0.2223	0.2255
	K=27	Initial clustering of KSTC	Baseline	0.6564	0.5206	0.4433	-6.5038	0.3343	0.3235	0.1153	0.0946	0.3264	0.3156	0.2790
		Initial clustering of KSTC	TIS	0.6595	0.5206	0.4807	-5.4413	0.4248	0.3798	0.1341	0.0965	0.4258	0.3748	0.3309
		KSTC (TIS)		0.7123	**0.5882**	0.5042	-5.4325	0.4248	0.3863	0.1365	0.1002	0.4257	0.3818	0.3371
	K=38	Initial clustering of KSTC	Baseline	0.6733	0.5379	0.4592	-6.2450	0.3684	0.3251	0.1306	0.0985	0.3574	0.3142	0.2933
		Initial clustering of KSTC	TIS	0.6722	0.5349	0.5128	-5.3022	0.4574	0.4072	0.1658	0.1285	0.4456	0.4004	0.3597
		KSTC (TIS)		**0.7254**	0.5746	**0.5331**	**-5.2376**	**0.4780**	**0.4225**	**0.1717**	**0.1352**	**0.4595**	**0.4116**	**0.3731**

Table 3: Labeling performance comparison on the NATCS datasets. The best clustering result (ClusterLLM-I-iter with keyphrases and Agglomerative clustering) is used as the initial clustering for the KSTC.

the core information of a dialogue, serve as salient features that enhance cluster cohesion and contribute to improved clustering performance. This also suggests that keyphrases can further enhance clustering performance, even within an already optimized embedding space.

4.2 Analysis of Stage 2 results

Table 3 presents a comparative analysis of the KSTC's final clustering and label per on the Banking, Finance and Insurance datasets, following LLM-based adjustment.

- **Baseline**: It extracts utterance embeddings from Sentence-Transformers and performs K-Means clustering in the resulting embedding space. The number of clusters is set to the ground-truth value. Subsequently, the Mistral-7B-Instruct model is used to generate a single theme label for each cluster based on all utterances it contains.

Analysis of LLM Adjustment Performance and Label Generation Methods of KSTC

Following theme labeling on the initial clustering results, KSTC, which incorporates LLM-based adjustment and cluster refinement, achieves consistent performance improvements across all three datasets, significantly outperforming the baseline. These improvements are observed consistently across both clustering and labeling evaluation metrics.

Method	Preference	Banking	Finance	Insurance
Initial clustering of KSTC	Should-Link	32.93%	41.62%	12.90%
	Cannot-Link	32.93%	41.62%	12.90%
keyphrase-based adjustment	Should-Link	52.44%	43.93%	19.35%
	Cannot-Link	84.15%	84.39%	87.3%
KSTC	Should-Link	99.39%	98.84%	96.13%
	Cannot-Link	98.78%	100%	99.21%

Table 4: Preference-satisfaction ratio

As shown in Table 4, the proposed method enables fine-grained adjustments of complex and nuanced user intent representations through keyphrase-based contextual adjustment and preference-based adjustment. Table 4 reports the preference-satisfaction ratio, computed as the number of satisfied preference pairs divided by the total number of preference pairs (higher is better).

We attribute the improvement in clustering and labeling quality through LLM adjustment to more accurate predictions in theme label generation. As shown in Table 3, when using the value of K determined based on the Combined Score for initial clustering, followed by theme label generation using Task Independent Slots, performance improves over using the ground-truth K in terms of average cosine similarity and ROUGE scores, with improvements of 1.8%p in Banking, 1.7%p in Finance, and 3.6%p in Insurance datasets, respectively. To analyze the source of the performance gains, we examined, for each value of K, the degree to which utterances within a single cluster shared the same theme label (Cluster Purity). Using the proposed method, the proportion of perfectly pure clusters (100% Purity)

Dataset	Utterance	Predicted Theme Label	Theme Label (Ground Truth)
Finance	Also, what are your your hours at at at at the branch over there on on Baker Street?	get branch location/hours	get branch location/hours
	Yes, I'm trying to find out what I owe for my credit card.	check credit card balance	check credit card balance
	I need to find out what my net income is from January to June of this year.	get net income	get net income
	Thank you, I just, I'm looking for some. A line of credit, perhaps.	apply for line of credit	apply for line of credit
	Yes, so I was wondering if you could tell me the current CPI, please?	request consumer price index	get consumer price index

Table 5: Comparison between predicted theme labels and ground truth theme labels in Finance Dataset.

	Clustering		Theme Label accuracy							Theme Label Style			
Method	**ACC**	**NMI**	**Rouge-1**	**Rouge-2**	**Rouge-L**	**Cosine Similarity**	**BertScore Precision**	**Recall**	**F1**	**LLM-as-a-Judge Section1**	**Section2**	**Average**	**Avg. Overall**
(Team C) Ours	**0.6797**	**0.7039**	**0.4522**	**0.2381**	**0.4510**	**0.6991**	**0.9502**	**0.9469**	**0.9471**	**1.0000**	**0.9948**	**0.9974**	**0.7550**
(Team D)	0.5176	0.4771	0.3457	0.2131	0.3427	0.5593	0.9252	0.9148	0.9191	0.8039	0.7660	0.7850	0.6308
(Team E)	0.3582	0.4773	0.4228	0.1650	0.4122	0.6248	0.9385	0.9284	0.9327	0.9346	0.9569	0.9458	0.6748

Table 6: Official results for test submissions by DSTC12 Track2, Automatic evaluation

	Per-Utterance Functional					Per-Cluster Structural		Per-Cluster-Functional	
Method	**Semantic Relevance**	**Analytical Utility**	**Granularity**	**Actionability**	**Domain Relevance**	**Conciseness Word Choice**	**Grammatical Structure**	**Thematic Distinctiveness**	**Avg. Overall**
(Team C) Ours	**0.8967**	**0.8275**	**0.4784**	**0.7477**	**0.9882**	**1.0000**	**1.000**	**0.9111**	**0.8562**
(Team D)	0.6876	0.6366	0.2641	0.6026	0.9425	0.9167	0.6667	0.9091	0.7032
(Team E)	0.8627	0.5464	0.2248	0.5451	0.9111	0.9365	0.9365	0.7834	0.7183

Table 7: Official results for test submissions by DSTC12 Track2, Human evaluation

relative to the total number of clusters increased by 2.31%p in Banking, 0.8%p in Finance, and 8.38%p in Insurance. In terms of utterance counts, the number of utterances contained in perfectly pure clusters grew by 57 in Banking, 15 in Finance, and 58 in Insurance. The analysis is provided in Appendix H. These findings suggest that our approach improves overall clustering quality. Moreover, high-purity clusters with their strong topical coherence create favorable conditions for the subsequent LLM-based automatic labeling stage, leading to more accurate and reliable theme generation.

Our labeling method, Task Independent Slots, prioritizes the selection of core verbs and objects within the cluster and employs the LLM to generate more appropriate theme label expressions, thereby capturing finer details. This demonstrates that the high quality of initial labeling contributes to the overall improvement in final clustering and labeling performance. Table 5 substantiates these gains: each predicted label (i) removes superfluous words, (ii) appears as an event-centered verb phrase, (iii) strikes the right balance between being actionable and sufficiently general, and (iv) is informative enough to narrow downstream resolution steps—while almost matching the gold label for sampled utterance in the finance dataset. Additional examples for Banking and Insurance are provided in Appendix I.

Test Data Results

Tables 6 and 7 are the official results of the test submission by the participants. This includes both human evaluation and LLM-based evaluation. Our method, denoted as Team C, is the model ranking first.

5 Conclusion

We propose KSTC, a clustering and theme labeling framework that operates in unseen intent scenarios and exhibits robust domain adaptability. Our method enhances clustering performance by leveraging keyphrases extracted from conversational context, enabling the generation of semantically fine-grained theme labels using the Task Independent Slots. This approach facilitates high quality label creation even in practical datasets that require complex and nuanced intent understanding. Moreover, KSTC offers flexibility that reflects pre-defined user preferences. Experimental results demonstrate that LLM-based cluster refinement consistently improves both clustering and labeling performance across all three datasets. In addition, the effectiveness of our method was demonstrated by ranking first in both automatic and human evaluations in DSTC12 Track 2.

The domain independent performance of KSTC in this zero-shot setting is expected to significantly contribute to intent analysis in real-world industrial applications.

6 Limitations

KSTC generates informative predicted theme labels for each cluster using Task Independent Slots, and effectively performed clustering refinement based on this information, achieving significant performance improvements across multi-turn intent discovery datasets. However, our method is currently applicable only to datasets where each utterance is annotated with explicit intent labels. Future research should focus on developing an algorithm that can first determine whether an intent exists within a dialogue.

Acknowledgements

This work was supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00621, RS-2022-II220621, Development of artificial intelli- gence technology that provides dialog-based multi- modal explainability)

References

Grant Anderson, Emma Hart, Dimitra Gkatzia, and Ian Beaver. 2024. An open intent discovery evaluation framework. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 760–769, Kyoto, Japan. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

James Gung, Emily Moeng, Wesley Rose, Arshit Gupta, Yi Zhang, and Saab Mansour. 2023. Natcs: Eliciting natural customer support dialogues. *Preprint*, arXiv:2305.03007.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 1373–1378. Association for Computational Linguistics (ACL).

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. 2024. Synergizing large language models and pre-trained smaller models for conversational intent discovery. In *Findings of the Association for Computational Linguistics: ACL 2024*, Bangkok, Thailand.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Junhua Liu, Yong Keat Tan, Bin Fu, and Kwan Hui Lim. 2024a. Intent-aware dialogue generation and multi-task contrastive learning for multi-turn intent classification. *Preprint*, arXiv:2411.14252.

Junhua Liu, Yong Keat Tan, Bin Fu, and Kwan Hui Lim. 2024b. Lara: Linguistic-adaptive retrieval-augmentation for multi-turn intent classification. *Preprint*, arXiv:2403.16504.

Pengfei Liu, Youzhang Ning, King Keung Wu, Kun Li, and Helen Meng. 2021. Open intent discovery through unsupervised semantic clustering and dependency parsing. *Preprint*, arXiv:2104.12114.

Jinjie Ni, Tom Young, Vlad Pandelea, Fuzhao Xue, and Erik Cambria. 2022. Recent advances in deep learning based dialogue systems: A systematic survey. *Preprint*, arXiv:2105.04387.

OpenAI. 2024. Chatgpt. `https://platform.openai.com/docs/guides/embeddings`. Accessed: 2025-06-02.

Jeiyoon Park, Yoonna Jang, Chanhee Lee, and Heuiseok Lim. 2024. Analysis of utterance embeddings and clustering methods related to intent induction for task-oriented dialogue. *Preprint*, arXiv:2212.02021.

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

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. 2023. One embedder, any task: Instruction-finetuned text embeddings. *Preprint*, arXiv:2212.09741.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Vijay Viswanathan, Kiril Gashteovski, Carolin Lawrence, Tongshuang Wu, and Graham Neubig. 2023. Large language models enable few-shot clustering. *Preprint*, arXiv:2307.00524.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Hui Yin, Xiangyu Song, Shuiqiao Yang, Guangyan Huang, and Jianxin Li. 2021. Representation learning for short text clustering. *Preprint*, arXiv:2109.09894.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Preprint*, arXiv:2106.11520.

Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lyu. 2021. Discovering new intents with deep aligned clustering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14365–14373.

Yuwei Zhang, Zihan Wang, and Jingbo Shang. 2023. Clusterllm: Large language models as a guide for text clustering. *Preprint*, arXiv:2305.14871.

A Dataset Statistics

Dataset	# of Dialogues	Avg. Words per Turn	# of Intent-labeled Utterances	# of Intents	# of Domains
Banking	980	59.6 ± 23.1	1634	26	1
Finance	3000	65.6 ± 22.4	1723	34	1
Insurance	954	70.6 ± 19.2	1333	27	1

Table 8: Dataset statistics

A summary of NATCS dataset statistics is shown in Table 8. "Avg. Words per Turn" indicates the average number of words per dialogue turn (mean ± std.).

Dataset	should_link	cannot_link
Banking	164	164
Finance	173	173
Insurance	155	126

Table 9: Preference data statistics

The number of preference pairs dataset for each domain can be found in Table 9.
For example, in the case of Should-Link, if the user's preferences suggest that the utterances "I gotta get my grandma some money." and "We just transfer the first because I need to close the account..." should belong to the same theme, other similar utterances would be associated with a single theme that semantically unifies the meanings of those utterances "make external wire transfer" or a close paraphrase of it. On the other hand, in the case of Cannot-Link, if the preferences indicate that "I want to change my email" and "I want to update my personal information" should not belong to the same theme, then the corresponding themes, "update email" and "update personal info", along with their associated utterance clusters, should remain separate.

B Extract Keyphrase & Illustrative Examples

We conducted experiments to extract keyphrases from multi-turn dialogue contexts by setting the context window size to 1, 3, and 5, respectively. In each experiment, the context window determines how many utterances before and after the theme-labeded utterance are taken into account. Examples are shown below.

Context Window = 1

Input Data:
"Theme_label: first, could you give me my balance perhaps? Maybe I can figure it out that way."

Ground-truth theme label:
"label_1": "check account balance",
"label_2": "check account balance"

Output Data:
keyphrase: "check account balance, view account balance, get balance information"
final_keyphrase: "get balance information"

Context Window = 3

Input Data:
"That's not a problem.",
"Take your time.",
"OK, Sundown. OK, that works. OK. Now, what you said there was a transaction you were concerned about?",
"Theme_label: first, could you give me my balance perhaps? Maybe I can figure it out that way.",
"OK, it looks like you've got two thousand six hundred forty-three dollars and twenty-eight cents.",
"OK. Oh, man. I'm not sure where that is actually what the problem is. could you give me the last date of my transaction and the dollar amount?"

Ground-truth theme label:
"label_1": "check account balance",
"label_2": "check account balance"

Output Data:
keyphrase: "check balance, recent transaction details, transaction date and amount"
final_keyphrase: "check balance"

DataSet	#Clusters	Clustering Algorithm	Theme Label Generation	Evaluation Metric										
				Clustering		Cosine	BART	Rouge-1		Rouge-2		Rouge-L		Avg.
				NMI	ACC	Similarity	Score	Recall	Precision	Recall	Precision	Recall	Precision	Cosine Rouge
Banking	K=30	Initial clustering of KSTC	Action-Object Pairs	0.7688	0.6753	0.5219	-5.4862	0.3827	0.5529	0.115	0.115	0.3827	0.5529	0.3747
			CoT	0.7821	0.7001	0.6819	-5.6382	0.5907	0.6030	0.2267	0.2371	0.5722	0.5760	0.4982
			CoT + Few-shot	0.7791	0.6916	0.6799	-5.6073	0.5968	0.5777	0.2349	0.2363	0.5807	0.5563	0.4946
			TIS	**0.7906**	**0.7252**	**0.7280**	**-4.6786**	**0.6196**	**0.6570**	**0.3350**	**0.3447**	**0.6028**	**0.6348**	**0.5603**
Finance	K=38	Initial clustering of KSTC	Action-Object Pairs	0.7815	0.5994	0.4675	-5.6427	0.33	0.5533	0.0065	0.0197	0.33	0.5533	0.3229
			CoT	**0.7954**	**0.6441**	0.6631	-5.1056	0.6543	0.6150	0.4151	0.3672	0.6543	0.6150	0.5692
			CoT + Few-shot	**0.7954**	**0.6441**	0.6626	-5.0778	0.6520	0.6028	0.4276	0.3756	0.6520	0.6028	0.5679
			TIS	**0.7954**	**0.6441**	**0.6951**	**-4.1662**	**0.7043**	**0.6681**	**0.4812**	**0.4487**	**0.7043**	**0.6681**	**0.6243**
Insurance	K=38	Initial clustering of KSTC	Action-Object Pairs	0.6777	0.5611	0.3803	-6.2104	0.27	0.4017	0.1282	0.174	0.27	0.4017	0.2894
			CoT	0.6733	0.5379	0.5079	-6.0002	0.4276	0.3713	**0.1918**	**0.1499**	0.4256	0.3638	0.3483
			CoT + Few-shot	**0.6739**	**0.5386**	0.5066	-6.0164	0.4401	0.3595	0.1584	0.1139	0.4381	0.3520	0.3384
			TIS	0.6722	0.5349	**0.5128**	**-5.2990**	**0.4574**	**0.4072**	0.1658	0.1285	**0.4456**	**0.4004**	**0.3597**

Table 10: Labeling performance comparison on the NatCS datasets. The best clustering result (ClusterLLM-I-iter with keyphrases and Agglomerative clustering) is used as the initial clustering for the KSTC.

Context Window = 5

Input Data:
"OK, one more security question. what street did you grow up on?",
"Oh, dear now you're making me think. You know, if I remember it correctly, it was on. Oh, hell. See, I told you this has me all worked up. I don't know what ugh Gosh. It's five thirteen Sundown Avenue.",
"That's not a problem."
"Take your time."
"OK, Sundown. OK, that works. OK. Now, what you said there was a transaction you were concerned about?"
"Theme_label: first, could you give me my balance perhaps? Maybe I can figure it out that way."
"OK, it looks like you've got two thousand six hundred forty-three dollars and twenty-eight cents."
"OK. Oh, man. I'm not sure where that is actually what the problem is. could you give me the last date of my transaction and the dollar amount?"
"It looks it would've been forty-seven dollars eighty-three cents on September twenty-sixth."
"Hmm that doesn't ring any bells. OK."

Ground-truth theme label:
"label_1": "check account balance",
"label_2": "check account balance"

Output Data:
keyphrase: "inquire about recent transactions"
final_keyphrase: "inquire about recent transactions"

When the context window is set to 1, the model focuses solely on the target utterance. As a result, it successfully captures the general theme (e.g., balance inquiry) but fails to identify more detailed aspects of the user's request. With a context window of 3, the surrounding utterances are considered, allowing the model to extract keyphrases that better reflect the user's actual intent. These results are more aligned with the ground-truth theme labels. And when the context window is increased to 5, the broader context often includes utterance segments where the theme shifts. This can lead to keyphrases that diverge from the user's intended goal.

These results suggest that appropriate context window settings are crucial for extracting contextually aligned keyphrases in multi-turn dialogue. In particular, using only a single utterance may lead to information sparsity, while overly large context windows may harm topic consistency.

C Label Generation Prompt Ablation Study

Table 10 summarizes the performance of different label generation strategies using LLMs, specifically examining the effects of Action-Object Pairs, Chain-of-Thought (CoT), CoT + Few-shot, and the Task Independent Slots. As shown in the table, the Task Indepenent Slots consistently outperformed the other approaches across all datasets in terms of average Cosine Similarity and ROUGE scores for theme label generation. The BARTScore was also highest when using Task Independent Slots.

Importantly, the reported performance reflects the results *prior to applying any additional theme label reassignment using LLMs*. In other words, the evaluation is based solely on the labels initially generated by each prompting strategy, without any post-hoc refinement or correction.

Action-Object pairs. (Anderson et al., 2024; Liu et al., 2021) extract ACTION-OBJECT pairs from utterances within each cluster using the direct object rule of the spaCy dependency parser (Honnibal and Johnson, 2015). They use the most frequent ACTION-OBJECT pair within each cluster as the cluster label.

Chain-of-Thought (CoT). Following the method proposed by Wei et al. (2023), this approach structures prompts such that the LLM performs step-by-step reasoning over the set of utterances to infer labels. This incremental reasoning process allows the model to generate appropriate labels even in zero-shot settings.

Few-shot. The few-shot setting, inspired by Brown et al. (2020), augments the CoT prompt with several example labels to guide the LLM in labeling clusters. While this approach tends to enhance labeling consistency, it is highly sensitive to the choice and composition of the examples, potentially introducing domain bias based on the examples provided.

D Prompt for Generating Keyphrases

Prompt for Generating Keyphrases in Banking

```
#Objective#
I am trying to cluster online banking-related queries based on whether they express the same intent.
For each dialogue, generate keyphrases ##that describe the utterance marked with a Theme_label's main intent or request##,
with a maximum of 3 keyphrases.
Keyphrases must:
- Be highly relevant to online banking domain.
- Focus on a **single main intent** per phrase.
- Be closely related to each other within the utterance's context.
The output must be in the form of <Key phrase example>, not full sentences.

<Key phrase example>
- update phone/email/address
- request email
- find atm
- report notice
- update personal info
</Key phrase example>

#utterance#
{utterances}
```

Table 11: Prompt for generating keyphrases in the Banking Dataset.

Prompt for Generating Keyphrases in Finance

```
#Objective#
I am trying to cluster finance-related queries based on whether they express the same intent.
For each dialogue, generate keyphrases ##that describe the utterance marked with a Theme_label's main intent or request##,
with a maximum of 3 keyphrases.
Keyphrases must:
- Be highly relevant to finance domain.
- Focus on a **single main intent** per phrase.
- Be closely related to each other within the utterance's context.
The output must be in the form of <Key phrase example>, not full sentences.

<Key phrase example>
- update phone/email/address
- request email
- get account info
- currency exchange rates
- update personal info
</Key phrase example>

#utterance#
{utterances}
```

Table 12: Prompt for generating keyphrases in the Finance Dataset.

Prompt for Generating Keyphrases in Insurance

#Objective#
I am trying to cluster insurance-related queries based on whether they express the same intent.
For each dialogue, generate keyphrases ##that describe the utterance marked with a Theme_label's main intent or request##, with a maximum of 3 keyphrases.
Keyphrases must:
- Be highly relevant to insurance domain.
- Focus on a **single main intent** per phrase.
- Be closely related to each other within the utterance's context.

The output must be in the form of <Key phrase example>, not full sentences.

<Key phrase example>
- update address
- create account
- change password/security question
- get pet insurance
- update personal info

</Key phrase example>

#utterance#
{utterances}

Table 13: Prompt for generating keyphrases in the Insurance Dataset.

Prompt for Filtering Keyphrases

#Objective#
Output one keyphrase that best describes ##the main request or intent from the utterances marked with a Theme_label##.
Must focus on the main action indicated by the Theme_label, not additional preferences or conditions. (ex: cuisine type, seating preferences, location)
Must select one keyphrase from the Keyphrases list.

#utterance#
{utterances}

#Keyphrases#
{keyphrases}
{format_instructions}

Table 14: Prompt for filtering keyphrases.

E Prompt for Generate Theme Label

E.1 Prompt for Generating Task Independent Slots

Prompt for Generating Task Independent Verb Slots

<task>
You're helping design a standardized **verb-based intent schema** for Dialogue State Tracking (DST) and intent classification across multiple domains.

Each slot name should represent a high-level **action or intention** that users commonly express during task-oriented conversations.

Please follow these guidelines:

1. Focus on **general categories of user actions or intentions**, not specific tasks. For example, use broad actions like "request" or "confirm", not specific activities like "book a flight" or "reset password".
2. Each slot name should be domain-agnostic and reusable across different sectors.
3. Cover a wide range of commonly expressed **user goals, requests, or dialogue functions** in real-world service conversations.

Now generate 10–15 such **generalized verb slot names** along with a **brief description** for each that explains its meaning and use case.

Format:

- slot_name_1: short description
- slot_name_2: short description
...
</task>

Table 15: Prompt for generating Task Independent Verb Slots

Prompt for Generating Task Independent Noun Slots

<task> You're helping design a standardized **entity-based slot schema** for Dialogue State Tracking (DST) and intent classification across multiple domains.

Each slot name should represent a high-level **conceptual category** of entities that users commonly refer to during task-oriented conversations.

Please follow these guidelines:

1. Focus on **abstract concepts or categories**, not specific instances. For example, use general terms like "document" or "location", not "passport" or "branch office".
2. Each slot name should be domain-agnostic and reusable across different sectors.
3. Cover a wide range of commonly referenced **objects, targets, or informational elements** in real-world dialogue tasks.

Now generate 10–15 such **generalized entity slot names** along with a **brief description** for each that explains its meaning and use case.

Format:

- slot_name_1: short description
- slot_name_2: short description
...
</task>

Table 16: Prompt for generating Task Independent Noun Slots

E.2 Prompt for Task Independent Verb Slots

Prompt for Task Independent Verb Slots

<Context>
You are assisting in building a Dialogue State Tracking (DST) system for the domain domain.
You are given utterances that express one intent enclosed in <Utterances> tags.
You are given a schema of generalized intent slots derived from verb groupings. The schema is enclosed in <Schema> tags.
</Context>

<Schema>
- require: The user is asking for a certain request or application.
- request_info: The user is asking for information or clarification about a product, service, or process.
- cancel: The user wants to cancel a service, request, or reservation.
- confirm: The user is verifying the correctness or status of a particular detail or action.
- update: The user wants to modify or refresh existing information or settings.
- inquire_issue: The user is reporting or inquiring about a problem, error, or complaint.
- recommend: The user is seeking advice or a suggestion for the best option.
</Schema>

<Objective>
Analyze the user utterances below and guess user's intent.
Then read <Schema> and determine which generalized intent slots from the <Schema> are relevant.
For each relevant slot, extract up to **three concise action verbs or verb phrases** that best represent the user's intent.
When you extract the verb, **you must follow both <Style> and <Caution> below**

Only extract **verbs or verb phrases** that meet all the following criteria:
- The verb must describe the **user's final goal**, NOT the object or topic.
- Use only the **base form** of the verb (e.g., "check", not "checking" or "checked").
- Avoid vague or speculative verbs unless they clearly reflect intent.

If a slot is **not relevant to the utterances or not useful for DST**, assign it a value of None.
However, **at least one slot must contain a valid verb or verb phrase** — do not return all None.
Always return **all five slots as keys in the JSON**, even if their value is None.
I will give you bunch of tip if you do great, let's think step by step.
</Objective>

<Style>
Use precise and concise verb phrases that clearly express intent.
If the user's action is directly stated, extract that exact verb or phrase.
If the intent is implicit or paraphrased, infer the most representative verb based on meaning.
</Style>

<Audience>
This output will be used by developers and researchers working on an LLM-based DST system.
They will use your output to evaluate whether the model correctly understands and generalizes user intent.
</Audience>

<Caution>
1. The verb have to make sense when the subject is 'user'.
Example:
utterance : Can you tell me about information?
correct verb : (user wants to) get (information)
incorrect verb : (You) tell (me about information)

2. The verb phrase must describe a class of EVENTS. **Do not** use states, entities properties, claims.
Example:
learn [event] vs. know [state]
redeem [event] vs. redemption[entity]
complain [event] vs. angry [property]
report defect [event] vs. product is defective [claim]
</Caution>

<Response Format>
Provide your answer strictly in the following JSON format:
{{
"request_info": [...],
"cancel": [...],
"require": [...],
"confirm": [...],
"update": [...],
"inquire_issue": [...],
"recommend": [...],
}}
</Response Format>

<Utterances>
utterances
</Utterances>

Now return the verb slot-value pairs as described above.

Table 17: Prompt for Task Independent Verb Slots

E.3 Prompt for Task Independent Noun Slots

Prompt for Task Independent Noun Slots

```
<Context>
You are assisting in building a Dialogue State Tracking (DST) system for the domain domain.
You are given utterances that express one intent enclosed in <Utterances> tags.
You are given a schema of generalized entity slots derived from semantic groupings. The schema is enclosed in <Schema> tags.
</Context>

<Schema>
- product: The product discussed or requested by the user.
- service: The service requested by the user.
- account: An account, subscription, or contract relevant to the user’s service.
- schedule: Any time-based request or item such as a date, time, or appointment.
- personal_info: Personal identification details like name, contact number, or address.
- payment: Payment-related information such as method, status.
- status: The progress or result of a request, task, or application.
- issue: A technical or service-related problem the user is experiencing.
- location: A physical place relevant to the conversation (e.g., branch, region).
- document: An official document or form related to the user’s intent.
- indicator: The indicator showing or measuring the condition or level of something.
</Schema>

<Objective>
Analyze the user utterances below and guess user’s intent.
Then read <Schema> and determine which generalized intent slots from the <Schema> are relevant.
For each relevant slot, extract up to **three concise nouns or noun phrases** that **BEST REPRESENTS the user’s INTENT**.
When you extract the nouns, **you must follow both <Style> and <Caution> below**

Only extract **nouns or noun phrases** that meet all the following criteria:
- The noun must describe the **user’s final goal**.
- Extract **only noun phrases or named entities** — do not include verbs, adjectives, or statements.
- Avoid vague or overly generic terms like "thing".
- If you want to use verbal noun, do not use it, **use the noun which means same instead**.
- **Do not** include article, pronoun and possessive.
- Use expressions found in the utterances which represents intent.

If a slot is **not relevant to the utterances or not useful for DST**, assign it a value of None.
However, **at least one slot must contain a valid noun or noun phrase** — do not return all None.
Always return **all five slots as keys in the JSON**, even if their value is None.
I will give you bunch of tip if you do great, let’s think step by step.
</Objective>

<Style>
Use clean, specific noun phrases.
Use lowercase unless the phrase is a proper noun.
Use real phrases from the utterances whenever possible.
</Style>

<Caution>
Do not extract exact noun for personal_info and location.
Example:
utterance : My name is Andy.
correct noun : name
incorrect noun : Andy
</Caution>

<Response Format>
Provide your answer strictly in the following **JSON format**:
{{
"product": [...],
"service": [...],
"account": [...],
"schedule": [...],
"personal_info": [...],
"payment": [...],
"status": [...],
"issue": [...],
"location": [...],
"document": [...],
"indicator": [...]
}}
</Response Format>

<Utterances>
utterances
</Utterances>

Now return the extracted entity slot-value pairs as described above.
```

Table 18: Prompt for Task Independent Noun Slots

E.4 Prompt for Generating Theme Label for each Cluster

E.4.1 Prompt for Generating Theme Label for each Cluster by Chain of Thought

Prompt for Generating Theme Label by Chain of Thought in NatCS

<task>
You are an expert call center assistant. You will be given a set of utterances in <utterances> </utterances> tags, each one on a new line.

The utterances are part of call center conversations between the customer and the support agent in the **{domain}** domain.

Your task is to generate a short label describing the theme of all the given utterances.
The label should capture the **customer's intended action** in the call and be written in a clear, standardized format.
The label should be a **verb phrase** starting with a base-form verb.

—

<guidance>
Output your response in the following way:
<theme_label_explanation>Your short step-by-step explanation behind the theme</theme_label_explanation>
<theme_label>Your final theme label</theme_label>
</guidance>
</task>

<utterances>
{utterances}
</utterances>

Table 19: Prompt for generating theme label by Chain of Thought in NatCS

E.4.2 Prompt for Generating Theme Label by Chain of Thought and Few Shot

Prompt for Generating Theme Label by Chain of Thought and Few Shot in NatCS

<task>
You are an expert call center assistant. You will be given a set of utterances in <utterances> </utterances> tags, each one on a new line.

The utterances are part of call center conversations between the customer and the support agent in the **{domain}** domain.

Your task is to generate a short label describing the theme of all the given utterances.
The label should capture the **customer's intended action** in the call and be written in a clear, standardized format.
The label should be a **verb phrase** starting with a base-form verb.

—

To help you understand the expected format, here are **example labels from a different domain (Travel)**:

- book flight ticket
- cancel hotel reservation
- change travel date
- request seat upgrade
- check baggage policy
- report lost luggage
- confirm airport pickup
- reschedule connecting flight
- apply travel insurance
- inquire visa requirement

—

<guidance>
Output your response in the following way:
<theme_label_explanation>Your short step-by-step explanation behind the theme</theme_label_explanation>
<theme_label>Your final theme label</theme_label>
</guidance>
</task>

<utterances>
{utterances}
</utterances>

Table 20: Prompt for generating theme labels by Chain of Thought and Few Shot in NatCS

E.4.3 Prompt for Generating Theme Labels by Task Independent Slots

Prompt for Generating Theme Labels by Task Independent Slots NATCS

<task>
You are an expert call center assistant. You will be given a set of utterances in <utterances> </utterances> tags, each one on a new line.
You will be given a set of verb candidates in <Verb Candidates> </Verb Candidates> tags, each one on a new line.
You will be given a set of entity candidates in <Entity Candidates> </Entity Candidates> tags, each one on a new line.

The utterances are part of call center conversations between the customer and the support agent in the **{domain}** domain.
Your task is to generate a short label describing the theme of all the given utterances.
The label should capture the **customer's intended action** in the call and be written in a clear, standardized format.
Use a set of verb and entity candidates if necessary.
The label should be a **verb phrase** starting with a base-form verb.

—

To help you understand the expected format, here are **example labels from a different domain (Travel)**:

- book flight ticket
- cancel hotel reservation
- change travel date
- request seat upgrade
- check baggage policy
- report lost luggage
- confirm airport pickup
- reschedule connecting flight
- apply travel insurance
- inquire visa

Strict Rules:
- The final theme label MUST NOT include any slot names such as "request_info", "inquire_issue", etc.
- You MUST select actual verbs and noun phrases that naturally appear in user language, not schema keys.
- Only use candidate expressions (e.g., "check") from the given sets — not their slot names.
- The theme label should be understandable to a human without knowing the underlying schema.

<guidance>
Output your response in the following way:
<theme_label_explanation>Your short step-by-step explanation behind the theme</theme_label$_{e}xplanation >$
<theme_label>Your final theme label</theme_label>
</guidance>
</task>

<utterances>
{utterances}
</utterances>

<Verb Candidates By Slot>
verb_dict
</Verb Candidates By Slot>

<Entity Candidates By Slot>
{entity_dict}
</Entity Candidates By Slot>

Table 21: Prompt for generating theme labels by Task Independent Slots in NATCS

F Prompt for Adujustment

F.1 Prompt for Theme Label Adjustment based on Keyphrase for each Utterance

Prompt for Theme Label Adjustment based on Keyphrase for each Utterance

```
You are tasked with determining the most appropriate label for a given Utterance.
When choose the label, focus on the require or intent of the utterance.
Keyphrase reflects the context of the dialogue and generated to capture the requir or intent, Use keyphrase as a reference to decide the label.
Instructions:

- Domain: "{domain}"
- Utterance: "{utterance}"
- Keyphrase: "{keyphrase}"
- Current label: "{predicted_label}"
- Candidates:
{candidates}

Choose the most appropriate label from the candidates.
Even if there are labels similar to the current lable, the Current label already captures the intent well, you must keep it.
Respond with the label only.
If none of the candidates are appropriate, respond with 'None'.
{format_instructions}
```

Table 22: Prompt for theme label adjustment based on Keyphrase for each utterance

F.2 Prompt for Theme Label Adjustment based on Preference pairs for each Utterance

Prompt for Theme Label Adjustment based on Should-Link for each Utterance

```
<task>
You will be given a set of utterances in <utterances> </utterances> tags, each one on a new line.
You will be given a set of Label candidates in <Label Candidates> </Label Candidates> tags, each one on a new line.

The utterances are part of call center conversations between the customer and the support agent in the **{domain}** domain.

Your task is to choose a label describing the theme of all the given utterances.
Use a set of set of utterances and Label Candidates.
Do not modify Label Candidates. Just choose a Label.

<guidance>
Output your response in the following way:
<theme_label_explanation>Your short step-by-step explanation behind the theme</theme_label_explanation>
<theme_label>Your final theme label</theme_label>
</guidance>
</task>

<utterances>
{utterances}
</utterances>

<Label Candidates>
{Current_Cluster_Labels}
</Label Candidates>
```

Table 23: Prompt for theme label adjustment based on Should-Link for each utterance

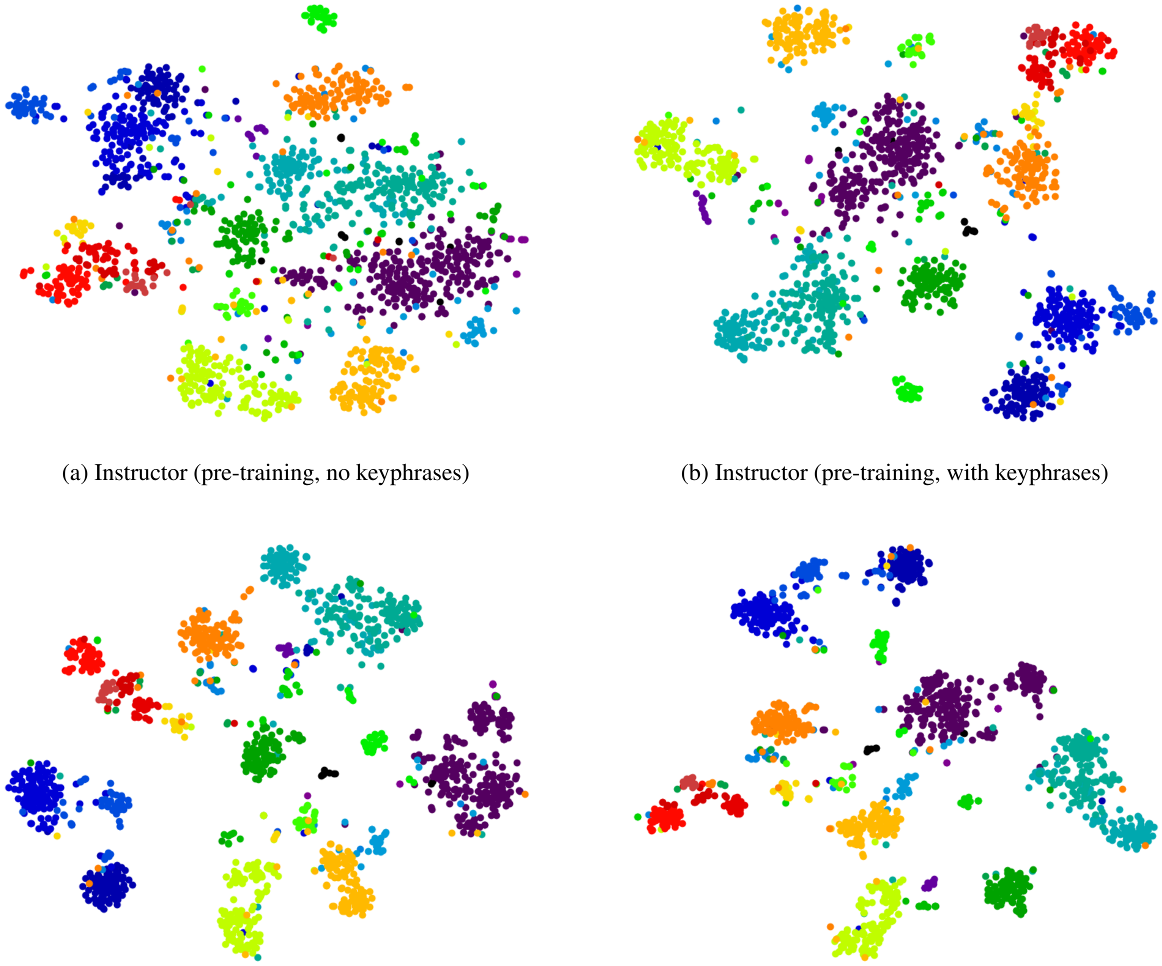
Prompt for Theme Label Adjustment based on Cannot-Link for each Utterance

```
<task>
You will be given a set of A utterances in <A_utterances> </A_utterances> tags, each one on a new line.
You will be given a set of B utterances in <B_utterances> </B_utterances> tags, each one on a new line.
You will be given a set of Cluster Label in <Cluster_Labels> </Cluster_Labels> tags, each one on a new line.
You will be given a set of Changed Cluster Label Candidates in <Label_Candidates> </Label_Candidates> tags, each one on a new line.

The utterances are part of call center conversations between the customer and the support agent in the **{domain}** domain.

Your task is to choose the utterance between A utterance and B utterance that is less aligned with the Cluster Label.
If you choose A utterance, you return just "A" else is "B".
Additionally, based on the selected utterances, you choose a group of candidate cluster labels that best match the current selected utterance among the Changed Cluster Label Candidates.

Use a set of set of utterances and Label Candidates.
Do not modify Changed Cluster Label Candidates. Just choose a Label.

<guidance>
Output your response in the following way:
<selected_utterance>Selected utterance</selected_utterance>
<theme_label_explanation>Your short step-by-step explanation behind the theme</theme_label_explanation>
<theme_label>Your final theme label</theme_label>
</guidance>
</task>

<A_utterances>
{A_utterances}
</A_utterances>

<B_utterances>
{B_utterances}
</B_utterances>

<Cluster_Labels>
{Cluster_Labels}
</Cluster_Labels>

<Label_Candidates>
{Label_Candidates}
</Label_Candidates>
```

Table 24: Prompt for theme label adjustment based on Cannot-Link for each utterance

G Impact of Keyphrases on Embedding Structure (t-SNE)

G.1 Banking Dataset



(a) Instructor (pre-training, no keyphrases)

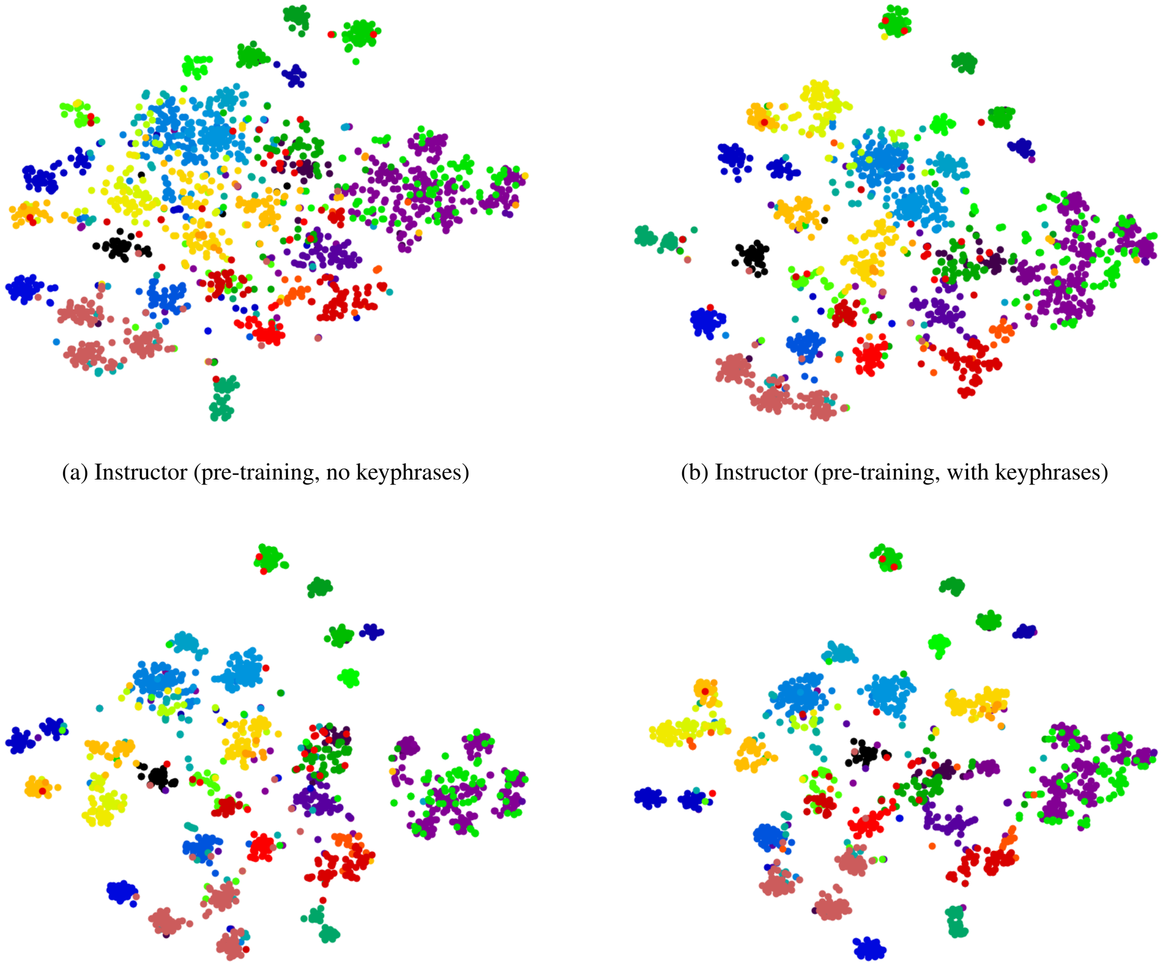
(b) Instructor (pre-training, with keyphrases)

(c) CLUSTERLLM-I-iter (post-training, no keyphrases)

(d) CLUSTERLLM-I-iter (post-training, with keyphrases)

Figure 4: t-SNE Visualization of Embeddings on Banking Dataset

G.2 Finance Dataset



(a) Instructor (pre-training, no keyphrases)

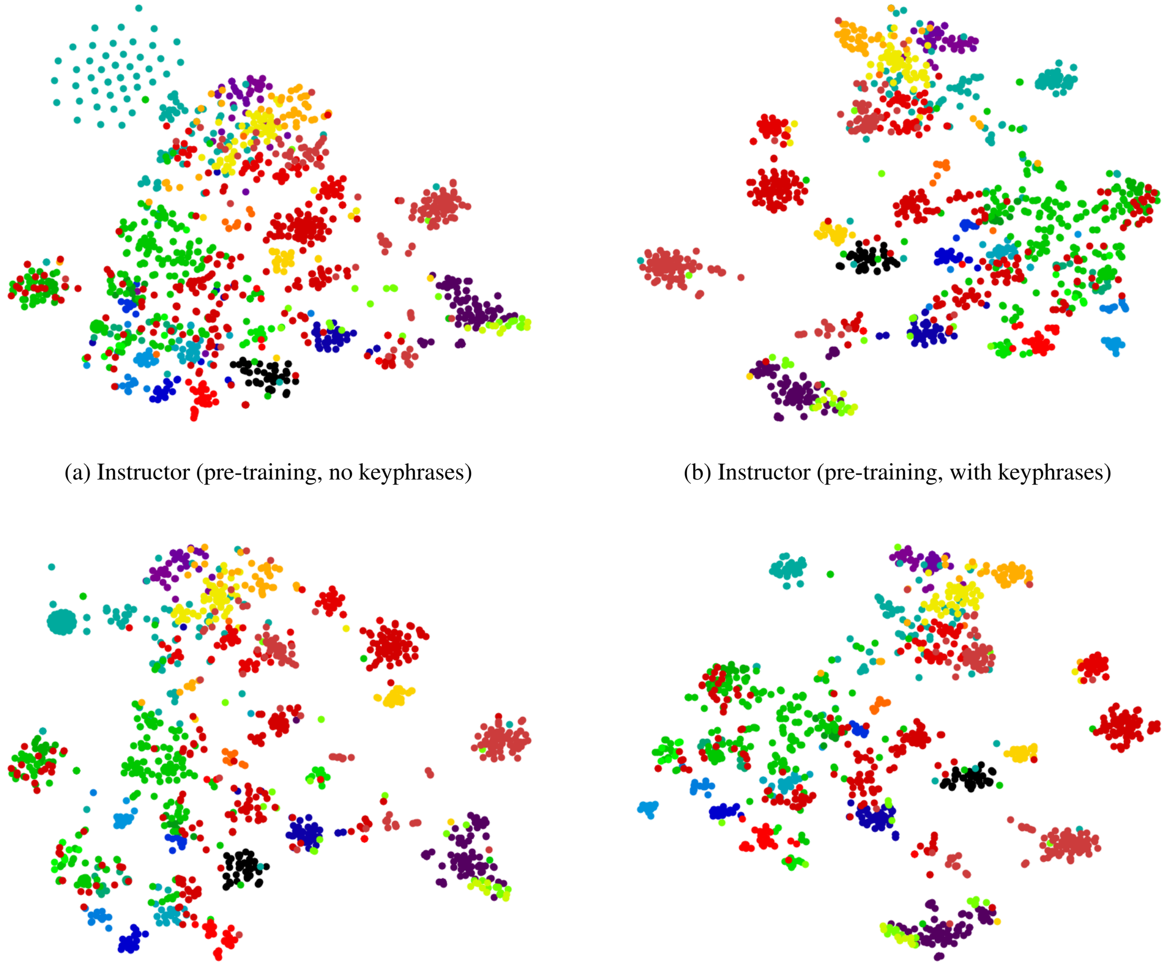
(b) Instructor (pre-training, with keyphrases)

(c) ClusterLLM-I-iter (post-training, no keyphrases)

(d) ClusterLLM-I-iter (post-training, with keyphrases)

Figure 5: t-SNE Visualization of Embeddings on Finance Dataset

G.3 Insurance Dataset



(a) Instructor (pre-training, no keyphrases)

(b) Instructor (pre-training, with keyphrases)

(c) CLUSTERLLM-I-iter (post-training, no keyphrases)

(d) CLUSTERLLM-I-iter (post-training, with keyphrases)

Figure 6: t-SNE Visualization of Embeddings on Insurance Dataset

H Cluster Purity and Num of Utterances

H.1 Banking Dataset

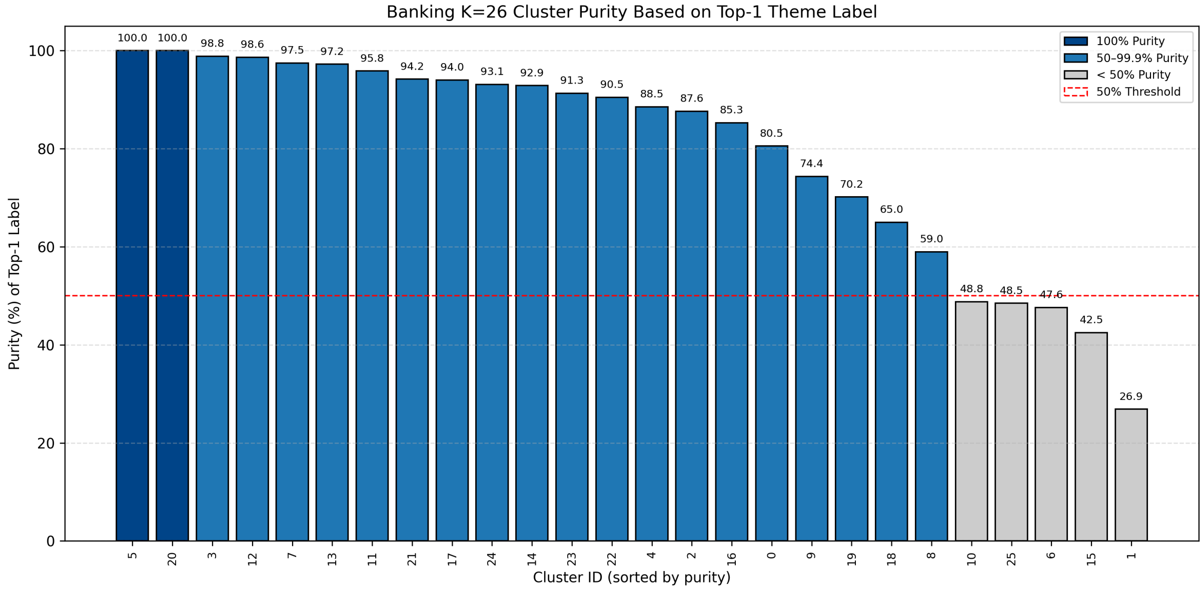


Figure 7: Histogram of cluster-purity on the Banking Dataset (K=26). Two clusters achieve 100% purity, whereas five clusters have purity below 50%.

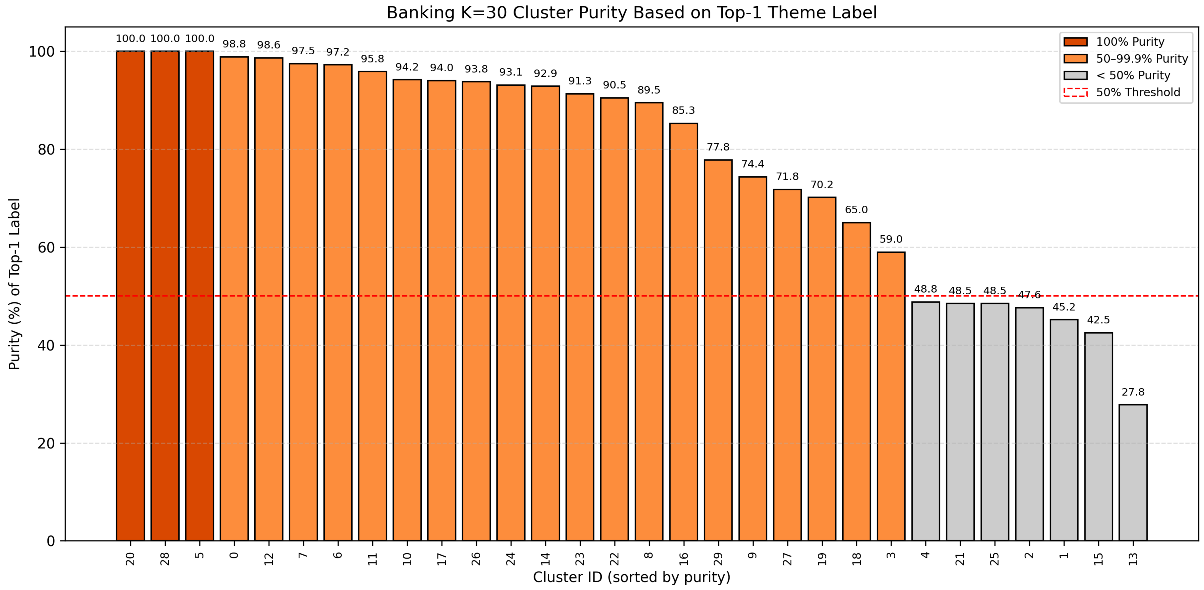


Figure 8: Histogram of cluster-purity on the Banking Dataset (K=30). Three clusters achieve 100% purity, whereas seven clusters have purity below 50%.

H.2 Finance Dataset

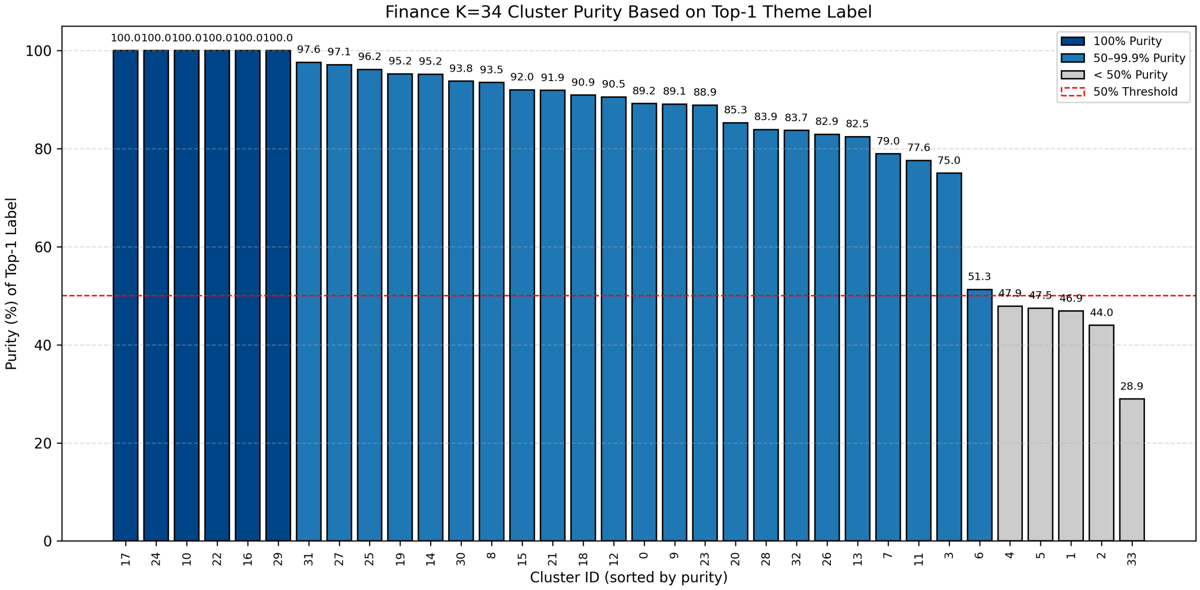


Figure 9: Histogram of cluster-purity on the Finance Dataset (K=34). Six clusters achieve 100% purity, whereas five clusters have purity below 50%.

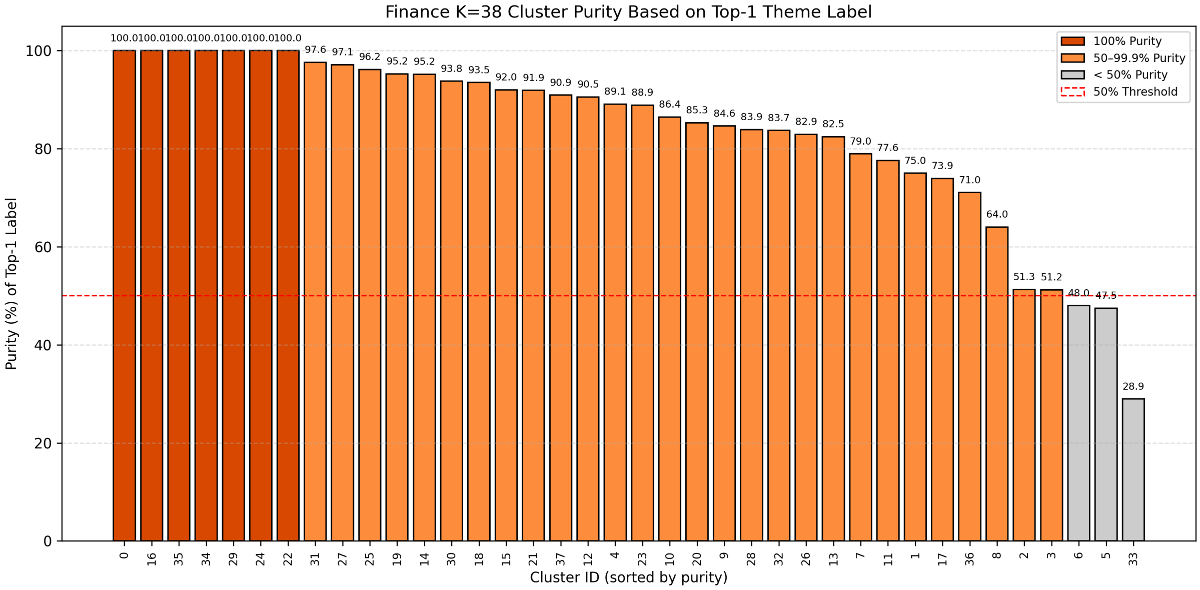


Figure 10: Histogram of cluster-purity on the Finance Dataset (K=38). Seven clusters achieve 100% purity, whereas three clusters have purity below 50%.

H.3 Insurance Dataset

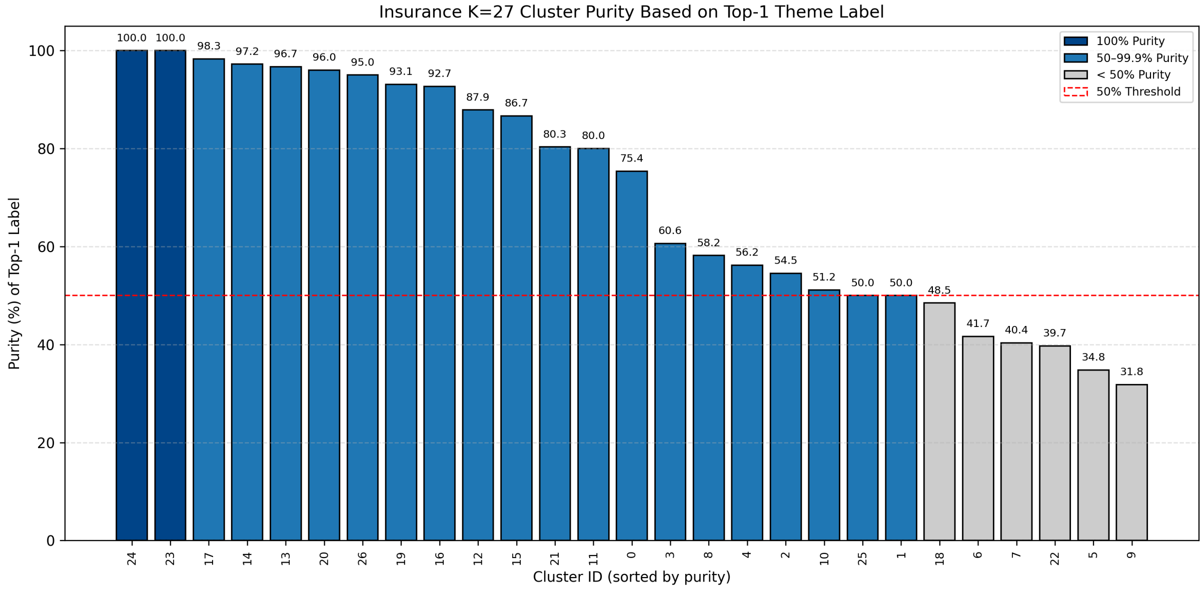


Figure 11: Histogram of cluster-purity on the Insurance Dataset (K=27). Two clusters achieve 100% purity, whereas six clusters have purity below 50%.

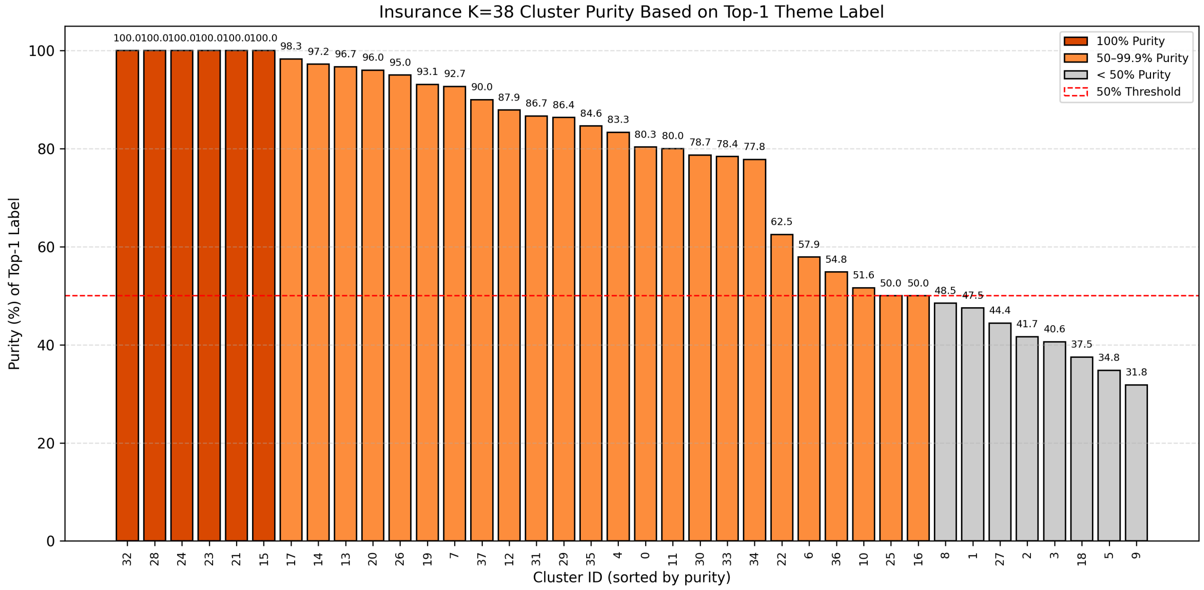


Figure 12: Histogram of cluster-purity on the Insurance Dataset (K=38). Six clusters achieve 100% purity, whereas eight clusters have purity below 50%.

H.4 Num of Utterances

Dataset	#Clusters	Cluster ID	Theme Label	#Utterances	Total Utterances
Banking	K=26	5	perform operations with limits	32	74
		20	request/check status of charge dispute	42	
	K=30	5	perform operations with limits	32	131
		20	request/check status of charge dispute	42	
		28	make external wire transfer	57	
Finance	K=34	10	get branch location/hours	42	223
		16	ask about correspondence	42	
		17	get consumer price index	28	
		22	change account or card pin	40	
		24	update phone/email/address	38	
		29	cancel/order check	33	
	K=38	0	get branch location/hours	42	238
		16	ask about correspondence	42	
		22	change account or card pin	40	
		24	update phone/email/address	38	
		29	cancel/order check	33	
		34	schedule appointment	15	
		35	get consumer price index	28	
Insurance	K=27	23	find agent	14	56
		24	get billing info	42	
	K=38	15	get branch location/hours	13	114
		21	file automobile claim/report accident	20	
		23	find agent	14	
		24	get billing info	42	
		28	get plan info	9	
		32	get plan info	16	

Table 25: Number of utterances contained in 100%-purity clusters.

I Predicted Theme Labels

Dataset	Utterance	Predicted Theme Label	Theme Label (Ground Truth)
Banking	first, could you give me my balance perhaps? Maybe I can figure it out that way.	check account balance	check account balance
	OK thanks. I really just need an ATM.	find atm	find atm
	yeah actually I was thinking of opening up a savings account.	open bank account	open bank account
	I need it transferred to my new checking account.	make wire transfer	make external wire transfer
	Oh I was wondering where your nearest branch location is?	find nearest branch	find branch
Finance	Also, what are your your hours at at at at the branch over there on on Baker Street?	get branch location/hours	get branch location/hours
	Yes, I'm trying to find out what I owe for my credit card.	check credit card balance	check credit card balance
	I need to find out what my net income is from January to June of this year.	get net income	get net income
	Thank you, I just, I'm looking for some. A line of credit, perhaps.	apply for line of credit	apply for line of credit
	Yes, so I was wondering if you could tell me the current CPI, please?	request consumer price index	get consumer price index
Insurance	Marian Wright here, Timothy. I was trying to pay my insurance online, and it did not confirm the submit.	check payment status	check payment status
	I have had an incident in my garage workshop.	file poperty claim	file poperty claim
	Yes, I was billed twice this month, and I need to see what's going on.	report billing issue	report billing issue
	Yes and my ex husband knows that so I would like to change it.	change security question	change password/security question
	Yes, I definitely need to speak to a supervisor. This is is craziness thing I have ever heard!	request call back	request callback

Table 26: Predicted theme labels in NatCS Dataset.