CLUSTERFUSION: Hybrid Clustering with Embedding Guidance and LLM Adaptation

Anonymous ACL submission

Abstract

Text clustering is a fundamental task in natural language processing, yet traditional clustering algorithms with pre-trained embeddings often struggle in domain-specific contexts without costly fine-tuning. Large language models (LLMs) provide strong contextual reasoning, yet prior work mainly uses them as auxiliary modules to refine embeddings or adjust cluster boundaries. We propose CLUSTERFUSION, a hybrid framework that instead treats the LLM as the clustering core, guided by lightweight embedding methods. The framework proceeds in three stages: embedding-guided subset partition, LLM-driven topic summarization, and LLM-based topic assignment. This design enables direct incorporation of domain knowledge and user preferences, fully leveraging the contextual adaptability of LLMs. Experiments on three public benchmarks and two new domain-specific datasets demonstrate that CLUSTERFUSION not only achieves state-of-the-art performance on standard tasks but also delivers substantial gains in specialized domains. To support future work, we release our newly constructed dataset and results on all benchmarks.[1]

1 Introduction

Text clustering underpins a wide range of NLP applications, including information retrieval and topic modeling. A common practice is to apply traditional clustering (MacQueen, 1967; Day and Edelsbrunner, 1984) on top of pre-trained embedding models (Muennighoff et al., 2022; Wang et al., 2022). While effective in generic benchmarks, clustering in real-world domains remains difficult: the task is inherently underspecified without explicit domain expertise such as user preference and domain knowledge (Caruana, 2013). Bridging this gap typically requires extensive fine-tuning of embeddings on large, domain-specific corpora, which is costly and often impractical.

Semi-supervised interactive clustering approaches attempt to inject domain expertise directly into the clustering process by collecting user feedback and shows significant improvement in performance (Basu et al., 2002, 2004; Bae et al., 2020). However, for large-scale datasets with many clusters, collecting sufficient human feedback is often impractical in both time and annotation cost.

Large language models (LLMs) offer a promising alternative for incorporating domain expertise into clustering. LLMs exhibit strong contextual understanding and in-context learning, enabling them to adapt to new tasks without explicit training. Recent work has explored leveraging LLMs to improve traditional clustering pipelines (De Raedt et al., 2023; Zhang et al., 2023; Wang et al., 2023; Viswanathan et al., 2024; Feng et al., 2024). In these approaches, the backbone remains an embedding-based clustering algorithm, while the LLM serves as an auxiliary module that enriches representations or adjusts local boundaries.

An alternative perspective is to treat the LLM itself as the clustering core. Conceptually, this can be decomposed into two stages: (i) topic summarization, where the LLM proposes a set of candidate topics by abstracting the corpus, and (ii) topic assignment, where the LLM revisits each record and assigns it to one of the proposed topics to form clusters. LLMs have already shown strong performance in zero-shot and few-shot summarization and classification through prompting (Ouyang et al., 2022; Achiam et al., 2023), suggesting they are capable of both stages. Importantly, this paradigm also offers a straightforward mechanism to incorporate user preference and domain knowledge: prompts can specify what kind of clusters should be formed (e.g., whether ambiguous utterances such as "yes,"

[1] [Disabled GitHub link to comply with anonymized review requirements]

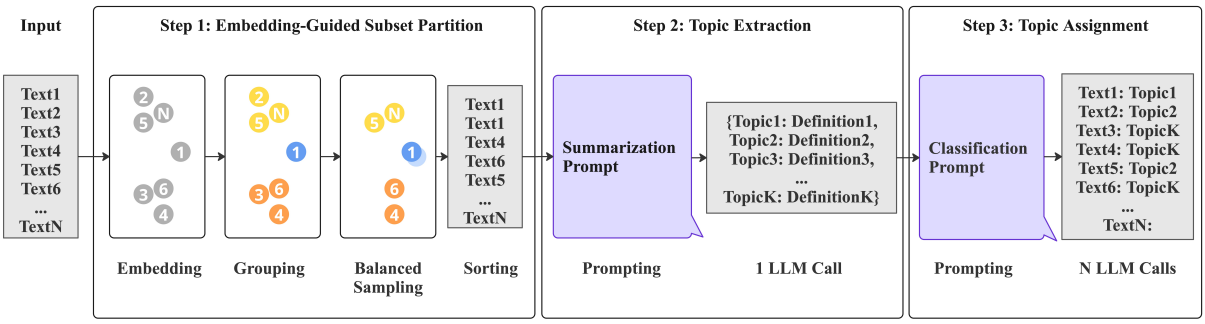

Figure 1: Overview of the CLUSTERFUSION framework

"no," and "unsure" should be separated or grouped) and how individual records with domain terminology should be interpreted.

We propose CLUSTERFUSION, a hybrid clustering framework that rethinks the relationship between traditional clustering and LLMs. Unlike prior work that uses LLMs to enhance traditional embedding-based clustering, we use traditional embedding to enhance LLM-based clustering. The pipeline proceeds in three steps: (1) embedding-guided subset partition, where embedding-based methods partition the corpus and select representative, ordered exemplars; (2) topic extraction, where an LLM generates candidate cluster topics from these exemplars; and (3) topic assignment, where each record is classified into one of the topics via a second LLM prompt.

Empirical results show that CLUSTERFUSION matches or surpasses state-of-the-art methods on standard benchmarks and yields substantial improvements on newly constructed domain-specific datasets. To foster further research, we release our dataset and results on all benchmarks.

2 CLUSTERFUSION Framework

As stated above, CLUSTERFUSION treats the LLM as the clustering core by decomposing clustering into two tasks: (i) topic summarization, where the LLM abstracts the corpus into a set of candidate topics, and (ii) topic assignment, where each record is assigned to one of the proposed topics. While LLMs are capable of performing both tasks, a fundamental bottleneck lies in the limited context window, which is particularly restrictive for summarization on large datasets. Topic assignment is less affected by this limitation, since it only requires one record per call. To mitigate this bottleneck, we introduce an embedding-guided subset partition stage that carefully selects and organizes the input for summarization. This stage consists of two substeps—*sampling* and *sorting*—which we describe in detail below.

An overview of the framework is shown in Figure 1, and the pseudocode is provided in Algorithm 1.

2.1 Embedding-Guided Subset Partition

We leverage embedding-based methods to transform the input dataset into a manageable, representative, and well-organized sample. This stage consists of two steps:

Grouping and Balanced Sampling. Given a dataset $D_N = \{x_1, x_2, \ldots, x_N\}$ with N records, we first compute embedding representations:

$$Z = \{z_1, z_2, \ldots, z_N\} = \text{Embedder}(D_N) \quad (1)$$

where $z_i \in \mathbb{R}^d$ is the embedding vector for record x_i.

We then apply a traditional clustering algorithm such as KMeans to partition the embeddings into M groups:

$$G = \{G_1, G_2, \ldots, G_M\} = \text{Group}(Z, M) \quad (2)$$

where each group $G_m = \{(x_j, z_j) : j \in \mathcal{I}_m\}$ contains records and their embeddings assigned to group m.

To construct a representative sample D_S of size S, we perform balanced sampling by drawing $\lfloor S/M \rfloor$ samples from each group. For groups with sufficient samples ($|G_m| \geq \lfloor S/M \rfloor$), we sample without replacement to ensure diversity. For smaller groups ($|G_m| < \lfloor S/M \rfloor$), we sample with replacement to meet the target sample size:

$$D_S = \bigcup_{m=1}^{M} \text{Sample}(G_m, \lfloor S/M \rfloor) \quad (3)$$

Sampling is necessary to respect the LLM's context window limit, while balanced sampling amplifies the signal from low-frequency clusters and ensures that minority groups remain represented.

Sorting. To improve coherence in the LLM input, we arrange the sampled records $D_S = \{x_{s_1}, x_{s_2}, \ldots, x_{s_S}\}$ before concatenation. We evaluate two sorting strategies:

(i) Cluster-based ordering: Records are sorted by their predicted group index:

$$\pi_{\text{cluster}} = \arg\text{sort}(\{\text{group}(x_{s_i})\}_{i=1}^{S}) \quad (4)$$

(ii) Similarity-based ordering: Records are sorted by their embedding similarity (cosine or Euclidean distance) to the first sampled record:

$$\pi_{\text{sim}} = \arg\text{sort}(\{\text{sim}(z_{s_1}, z_{s_i})\}_{i=1}^{S}) \quad (5)$$

where $\text{sim}(\cdot,\cdot)$ denotes a similarity measure (e.g., cosine similarity or Euclidean distance).

The final sorted sample is then $D'_S = \{x_{s_{\pi(1)}}, x_{s_{\pi(2)}}, \ldots, x_{s_{\pi(S)}}\}$, where $\pi \in \{\pi_{\text{cluster}}, \pi_{\text{sim}}\}$. From our experiments, such pre-organization improves readability and reduces fragmentation, enabling the LLM to generate more consistent and higher-quality topic summaries.

2.2 Summarization: Topic Extraction

The sampled and sorted records are concatenated into a prompt for the LLM, which is instructed to abstract the corpus into a concise set of candidate topics along with brief definitions. These topics serve as the proposed cluster labels. This step leverages the LLM's strong contextual reasoning abilities, and also enables the flexible incorporation of domain knowledge or user preferences directly into the prompt (e.g., specifying whether ambiguous utterances should be merged into one category or separated into distinct clusters). The complete prompt templates and domain-specific examples are provided in Appendix A.

2.3 Classification: Topic Assignment

Each record in the original dataset is then independently assigned to one of the candidate topics through a separate LLM call. The model is constrained to choose from the topic list produced in the *Topic Extraction* step; if it outputs an invalid label, the call will be rerun. The full prompt template for this classification step is also detailed in Appendix A.

Algorithm 1 CLUSTERFUSION Framework

Input: Dataset D_N; number of clusters K
Hyperparameters: Number of groups M; total sample size S
Output: Cluster assignments for all $x \in D_N$

Step 1: Embedding-Guided Subset Partition
$Z \leftarrow \text{Embedder}(D_N)$
$G \leftarrow \text{Group}(Z, M)$
$D_S \leftarrow \emptyset$
for $m = 1 \rightarrow M$ **do**
  replace_flag $\leftarrow$ True if $|G_m| < S/M$ else False
  $D_S \leftarrow D_S \cup \text{Sample}(G_m, S/M, \text{replace_flag})$
end for
$D_S \leftarrow \text{Sort}(D_S)$
Step 2: Topic Summarization
$T \leftarrow \text{LLM.summarize}(D_S, K)$
Step 3: Topic Assignment
for all $x \in D_N$ **do**
  $t \leftarrow \text{LLM.classify}(x, T)$
  assign $x \mapsto t$
end for
return clustering $\{x \mapsto t\}_{x \in D_N}$

3 Experimental Setting

3.1 Datasets

We evaluate CLUSTERFUSION on three widely used clustering benchmarks and two newly constructed domain-specific datasets. The generic benchmarks allow comparison with prior work, while the domain-specific corpora test the model's ability to adapt to specialized contexts. The existing generic benchmarks are:

- **Bank77** (Casanueva et al., 2020): The Bank77 dataset contains 3,080 user queries directed to an online banking assistant, annotated into 77 intent categories.
- **CLINC** (Larson et al., 2019): The CLINC dataset contains 4,500 user queries from a task-oriented dialog with 150 intent classes, after removing "out-of-scope" queries (Zhang et al., 2023). In addition, we found that a small amount of labels were ambiguous or inconsistent, and we performed further label cleaning.[2]
- **Tweet** (Yin and Wang, 2016): The Tweet dataset contains 2,472 social media posts annotated into 89 intent categories.

New domain-specific benchmarks:[3]

[2] Correction notes are available at: [Disabled GitHub link to comply with anonymized review requirements.].

[3] During data cleaning, we removed all personally identifiable information (PII), including usernames, profile links, and user IDs, as well as hateful or offensive content for both

Dataset	#Samples	#Clusters
Bank77	3,080	77
CLINC	4,500	150
Tweet	2,472	89
OpenAI Codex (new)	406	26
Adobe Lightroom (new)	6,410	40

Table 1: Summary of datasets used in our experiments

- **OpenAI Codex**: We constructed a new dataset of 406 YouTube comments from the OpenAI Codex announcement video[4] before June, 2025.The dataset reflects a highly technical domain and contains emerging terminology. Comments were manually annotated into 26 categories capturing user reactions, technical questions, and broader discussions.

- **Adobe Lightroom**: We collected 6,410 customer reviews for Adobe Lightroom iOS, annotated into 40 categories. This dataset captures domain-specific terminology and nuanced user feedback.

Table 1 summarizes the datasets' statistics.

3.2 Metrics

Following prior work (Zhang et al., 2021), we evaluate the quality of the induced clusters against ground-truth labels using normalized mutual information (NMI) and accuracy. Accuracy is computed by identifying the optimal alignment between predicted clusters and ground-truth labels via the Hungarian algorithm (Kuhn, 1955). These metrics are widely adopted in clustering evaluation as they capture both distributional similarity and label-level correspondence.

3.3 Hyper-Parameters and Design Choices

Following prior work, we assume the number of clusters K is known. During the embedding-guided subset partition stage, we set the number of groups M to $2K$, providing more diverse partitions for sampling. In general, choosing $M > K$ is recommended as it enhances diversity in the sampled records. In the summarization stage, we then explicitly prompt the LLM to generate exactly K topics.

To compare effectively with these approaches, we use the same base encoders reported for SCCL, ClusterLLM, Keyphrase Clustering and LLMEdgeRefine: INSTRUCTOR (Su et al., 2022) for Bank77 and CLINC, and DistilBERT[5] (Sanh et al., 2019; Reimers and Gurevych, 2019) for Tweet. For our newly constructed domain-specific datasets, we adopt the INSTRUCTOR model.

For the LLM component, we employ GPT-4o (Hurst et al., 2024) as the backbone model for both topic summarization and assignment.

4 Result

We evaluate CLUSTERFUSION on three widely used benchmarks (Bank77, CLINC, Tweet) and two fresh domain-specific datasets (OpenAI Codex, Adobe Lightroom). We compare CLUSTERFUSION against strong baselines and state-of-the-art methods, including DSE (Zhou et al., 2022), PCK-Means, LLM Correction, and Keyphrase Clustering (Viswanathan et al., 2024), SCCL (Zhang et al., 2021), ClusterLLM (Zhang et al., 2023), and LLMEdgeRefine (Feng et al., 2024).

4.1 Comparison of Effectiveness

Quantitative Comparison. Across all five datasets, CLUSTERFUSION consistently achieves the highest accuracy among state-of-the-art methods, with improvements that are statistically significant in Table 2. The gains are especially pronounced on domain-specific datasets Table 3. On OpenAI Codex, accuracy improves from 44.5 (best baseline) to 66.0, a relative increase of 48%. In terms of NMI, CLUSTERFUSION matches or closely tracks state-of-the-art results on Bank77 and CLINC, achieves the best score on Tweet (91.4), and delivers substantial boosts on domain-specific datasets: the improvement is around 29% on OpenAI Codex (from 56.1 to 72.6) and Adobe Lightroom (from 46.8 to 60.0).

Qualitative Analysis. The significant quantitative gains of CLUSTERFUSION arise from its flexibility in incorporating both domain knowledge and user preferences through prompting: (i) domain expertise can be injected into both summarization and assignment prompts to improve terminology understanding, and (ii) user preferences about which clusters should or should not be formed can be specified during the summarization stage. Table 4 illustrates these advantages with examples from the OpenAI Codex dataset.

datasets. These steps ensure compliance with ethical standards.

[4] https://www.youtube.com/watch?v=FUq9qRwrDrI

[5] Following ClusterLLM and Keyphrase Clustering, we used a version of DistilBERT finetuned for sentence similarity classification.

Method	Bank77		CLINC		Tweet	
	Acc	NMI	Acc	NMI	Acc	NMI
KMeans	65.2±0.0	83.3±0.0	80.9±0.0	92.5±0.0	59.1±0.0	80.6±0.0
Agglomerative	64.0±0.0	81.7±0.0	77.7±0.0	91.5±0.0	57.5±0.0	80.6±0.0
DSE (w/ KMeans)	49.2±0.0	71.0±0.0	62.1±0.0	85.5±0.0	50.8±0.0	78.4±0.0
PCKMeans	59.6±0.0	79.6±0.0	79.6±0.0	92.1±0.0	65.3±0.0	85.1±0.0
LLM Correction	64.1	81.9	77.8	91.3	59.0	81.5
Keyphrase Clust. (w/ KMeans)	65.3±0.0	82.4±0.0	79.4±0.0	92.6±0.0	62.0±0.0	83.8±0.0
Keyphrase Clust. (w/ Agglom.)	66.4±0.0	**84.1±0.0**	81.3±0.0	93.1±0.0	69.8±0.0	86.6±0.0
SCCL	–	–	80.9	92.9	78.2	89.2
ClusterLLM	71.2	–	83.8	94.0	–	–
LLMEdgeRefine	–	–	86.8	**94.9**	–	–
CLUSTERFUSION (Unsorted)	62.7±1.2	80.8±0.5	77.6±1.7	91.2±0.8	71.6±1.8	86.1±1.0
CLUSTERFUSION (KMeans order)	66.3±3.2	82.9±0.9	83.8±3.9	93.9±1.0	84.2±3.9	90.9±2.5
CLUSTERFUSION (Cosine order)	**68.3±2.9**	81.4±0.9	**87.3±1.0**	93.5±0.4	**87.1±8.6**	**91.4±4.5**
CLUSTERFUSION *(Oracle order)*	79.4±1.3	87.2±0.8	90.3±2.6	95.4±0.6	91.3±3.9	94.7±2.0

Table 2: Clustering performance on three public benchmarks. Where applicable, standard deviations are obtained by running clustering 5 times with different seeds. For state-of-the-art methods' performance, the displayed 0.0 standard deviations in the report were nonzero but below 0.05 in all experiments.For distance sorting, we tested both Cosine and Euclidean sorting; Cosine performed better in most cases and aligns with common practice, so we only include its result. Oracle order denotes sorting records by their ground-truth topics, which is not achievable in practice but serves as an upper bound illustrating the potential effect of ordering.

Method	OpenAI Codex		Adobe Lightroom	
	Acc	NMI	Acc	NMI
KMeans	41.7±0.0	55.5±0.0	30.7±0.0	18.8±0.0
Agglomerative Clust.	44.5±0.0	56.1±0.0	16.3±0.0	18.5±0.0
Keyphrase Clust. (w/ KMeans)	40.3±2.5	52.7±1.4	71.5±0.3	46.8±0.6
Keyphrase Clust. (w/ Agglom.)	35.8±1.8	48.9±2.0	71.3±0.4	46.0±0.6
CLUSTERFUSION (Unsorted)	61.8±2.9	70.5±0.9	81.9±2.3	56.7±5.2
CLUSTERFUSION (KMeans order)	63.6±5.0	71.5±3.5	82.8±1.2	58.0±2.6
CLUSTERFUSION (Cosine order)	**66.0±4.6**	**72.6±1.8**	**83.5±1.9**	**60.0±3.2**
CLUSTERFUSION *(Oracle order)*	69.9±3.9	75.4±2.9	84.2±2.1	61.6±2.6

Table 3: Clustering performance on two domain-specific datasets. Standard deviations are obtained by running clustering 5 times with different seeds. For distance sorting, we tested both Cosine and Euclidean sorting; Cosine performed better in most cases and aligns with common practice, so we only include its result.

Domain Knowledge Alignment. The corpus includes emerging AI terminology such as *Anthropic, Claude, DeepSeek*, which embedding-based methods struggle to interpret semantically. Rows 1-4 of Table 4 show KMeans lacking contextual understanding of the new terminologies: it groups "Anthropic is in trouble" (competitor mention) with "so bad" (general criticism) because their embeddings are both semantically negative, and groups "use Deepseek model to get rid of cost" with use-case requests ("automate my mortgage") because both are financial related. Keyphrase improves over KMeans but still produces less precise groupings. In contrast, CLUSTERFUSION leverages LLM-based contextual understanding to correctly separate competitor mentions, general criticism, profit model discussion, and feature requests.

User Preference Alignment. In the Codex announcement dataset, feature-related feedback is more informative, while comments about presentation staging and filming are less relevant and should be grouped into a single cluster. Rows 5-8 of Table 4 show comments about presentation aesthetics unrelated to product features, such as salad bowl, presenter's beard, and MacBook corners. CLUSTERFUSION effectively groups these into one cluster ("Presenter Criticism and Observations"). Keyphrase fragments them across separate clusters, even though distinctions like strawberries vs. salad are irrelevant for product reception. This fragmentation wastes cluster slots that could represent substantive feedback. Consolidating presentation comments into a unified, low-priority category allows more clusters for technical discussions, competitor comparisons, and feature requests.

4.2 Ablation Study: Ordering Matters

During our experiments, we observed that exemplar ordering plays a critical role in the quality of topic summarization. As shown in Table 3,

Comment	Ground-truth	KMeans	Keyphrase	CLUSTERFUSION
Domain Knowledge: Understand Emerging Tech Terminology				
"Anthropic is in trouble"	Competition With Competitors	Pessimistic Expectations	Impact of AI on Employment and Software Development	Competitors Emergence
"so bad"	General Criticism	Pessimistic Expectations	General Criticism	General Criticism
"someone just needs to come up with a way to run 670b deepseek models, so we can get rid of the cost"	Profit Model Discussion	Cost and lack of local implementation	Competitor solutions outperforming current offerings	Better solutions provided by competitors
"Please automate my mortgage and bills"	Future Use Case	Cost and lack of local implementation	Technology Ethics, Privacy, and Practical Applications	End-user Feature Requests
User Preference: Consolidate Non-Feature Related Comments				
"Are those strawberries and broccoli in the bowl?"	Feedback On The Presenter And Presentation Style	Table items and food discussions	Strawberry-related inquiries	Presenter Criticism and Observations
"that random salad decoration lol"	Feedback On The Presenter And Presentation Style	Table items and food discussions	Video Production and Presentation Critiques	Presenter Criticism and Observations
"Interesting beard choice"	Feedback On The Presenter And Presentation Style	Excitement and Enthusiasm	Presenter Appearance and Personality Observations	Presenter Criticism and Observations
"omg he hit the corner on the Mac on the table, hahahaha"	Feedback On The Presenter And Presentation Style	Cursor Tool and Presentation Observations	Humorous observations about MacBook setup and sticker usage	Presenter Criticism and Observations

Table 4: Qualitative examples from the OpenAI Codex dataset.

even the unsorted baseline already outperforms traditional clustering methods. However, performance improves consistently when records are pre-organized—whether by cluster assignments (e.g., KMeans) or by embedding similarity (e.g., cosine distance)—as illustrated in Figure 2 and Figure 3. On the Tweet dataset, for instance, accuracy rises by 22% (from 71.6 to 87.1) solely due to improved ordering.

One explanation for the impact of ordering lies in how transformer-based LLMs handle long contexts. While self-attention allows tokens to interact globally, empirical studies show that attention effectiveness decays with distance, making it harder to integrate scattered signals (Press et al., 2021; Liu et al., 2023). Disorganized inputs force the model to attend across heterogeneous content, increasing instability in topic proposals. Pre-organizing records into semantically coherent groups reduces this burden and allows the LLM to better exploit its contextual reasoning, yielding better clusters.

To further contextualize these gains, we include an oracle ordering where records are sorted by ground-truth topics. While infeasible in real applications, this upper bound reveals the latent potential of ordering. The consistent gap between heuristic and oracle orderings suggests that improved ordering heuristics could unlock further performance improvements. These findings underscore ordering not merely as an implementation detail, but as a central mechanism for stabilizing LLM outputs and maximizing clustering quality.

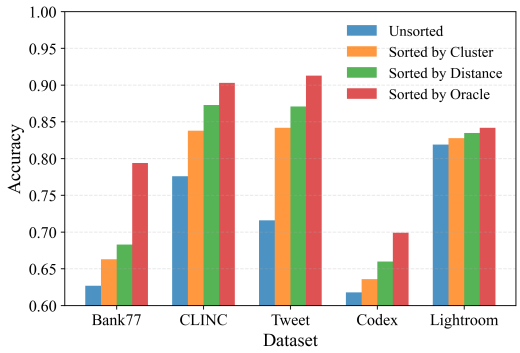

Figure 2: Clustering performance (Accuracy) under different ordering strategies across five datasets.

4.3 Cost Performance

While clustering accuracy is critical, practical deployment also requires methods to be cost-efficient. Unlike representation-learning approaches such as SCCL (Zhang et al., 2021) or DSE (Zhou et al., 2022), CLUSTERFUSION is training-free: it requires no fine-tuning or GPU-intensive optimization, relying only on one-off clustering and prompting.

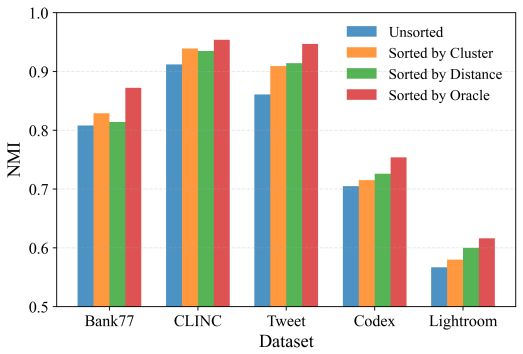

Figure 3: Clustering performance (NMI) under different ordering strategies across five datasets.

Compared to other training-free, LLM-based methods, CLUSTERFUSION incurs a cost profile similar to the Keyphrase method (Viswanathan et al., 2024). Our additional summarization step introduces a small overhead, which is controlled by the chosen sample size. Similar to the Keyphrase method, our dominant cost arises from topic assignment, which scales linearly with dataset size. Crucially, at comparable cost, CLUSTERFUSION delivers substantially higher accuracy, demonstrating superior cost-effectiveness.

Figure 4 visualizes the cost-accuracy trade-off between CLUSTERFUSION and the Keyphrase baseline across all five datasets. Each point represents one dataset, and the dashed lines connect the same dataset under both methods. The plot reveals a consistent pattern: CLUSTERFUSION points are positioned higher (better accuracy) at nearly identical costs. The vertical separation is particularly pronounced for domain-specific datasets such as OpenAI Codex, where CLUSTERFUSION achieves accuracy improvements of 63.8% with less than $0.02 total cost increase.

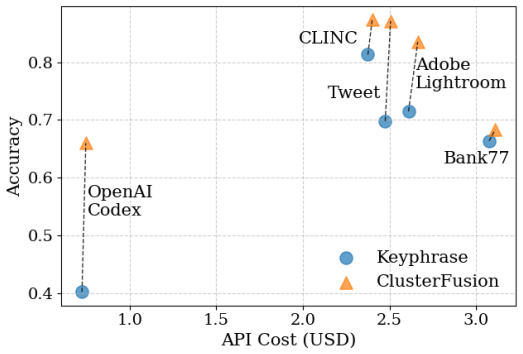

Figure 4: Cost-accuracy comparison between CLUSTERFUSION and Keyphrase. API price is based on OpenAI's GPT-4o pricing (April 2025).

5 Related Work

Clustering has long been a central problem in machine learning. A common practice is to apply classical clustering algorithms such as KMeans or agglomerative clustering (MacQueen, 1967; Day and Edelsbrunner, 1984) on top of pre-trained embeddings (Muennighoff et al., 2022; Wang et al., 2022).

However, the task is inherently underspecified without explicit domain expertise such as user preference and domain knowledge (Caruana, 2013). Bridging this gap often requires fine-tuning embeddings on large, domain-specific corpora, coupling representation learning with clustering objectives. Zhou et al. (2024) categorize such deep clustering research into multi-stage pipelines (Huang et al., 2014; Tao et al., 2021), iterative refinement (Yang et al., 2016; Chang et al., 2017; Caron et al., 2018; Van Gansbeke et al., 2020; Niu et al., 2020, 2022), generative frameworks (Dilokthanakul et al., 2016), and joint optimization (Xie et al., 2016; Hadifar et al., 2019; Zhang et al., 2021). While effective, these approaches demand heavy training, hyperparameter tuning, and substantial computational resources.

Semi-supervised clustering introduces limited external guidance to inject domain expertise, such as cluster seeds (Basu et al., 2002), pairwise constraints (Basu et al., 2004; Zhang et al., 2019), feature-level feedback (Dasgupta and Ng, 2010), or interactive operations like split–merge and lock–refine (Coden et al., 2017; Awasthi et al., 2017). Although these methods can yield more meaningful partitions, they typically require tens or hundreds of human–model interactions (Coden et al., 2017), making them impractical at scale.

More recently, large language models (LLMs) have opened new directions for clustering by leveraging their strong contextual understanding and in-context learning. LLMs have been applied to tasks such as abstractive summarization (De Raedt et al., 2023), semantic relation judgments (Zhang et al., 2023), explanation generation (Wang et al., 2023), keyphrase expansion (Viswanathan et al., 2024), and edge refinement (Feng et al., 2024). In all these approaches, however, the backbone remains an embedding-based clustering algorithm, with the LLM acting only as an auxiliary module to enrich representations or refine boundaries.

In contrast, we explore a complementary perspective: the core clusters in our method are de-

fined by the LLM itself. Our method leverages embeddings not as the clustering backbone, but as a pre-organizing guide that structures the input data before LLM inference. This design allows the LLM to fully exploit its contextual reasoning ability for topic induction and assignment, while preserving the scalability and stability benefits of traditional clustering, as demonstrated by our empirical results.

6 Conclusion

We introduced CLUSTERFUSION, a hybrid framework that leverages embedding-based methods to pre-organize data and LLMs to perform clustering. Experiments across five datasets show that CLUSTERFUSION not only outperforms existing state-of-the-art methods on standard benchmarks, but also delivers substantial gains on domain-specific datasets, where adaptability is critical. Our analysis further highlights the role of ordering in enhancing the outputs. By treating the LLM as the clustering core rather than a supporting module, CLUSTERFUSION provides a training-free, cost-efficient, simple-designed, and easily adaptable solution for clustering across diverse domains.

Limitations

While CLUSTERFUSION achieves strong performance, several limitations remain. First, for very large datasets, context length constraints may still require a divide-and-conquer strategy, though our sampling and ordering design could be naturally extended and integrated to batch settings. Second, the framework assumes prior knowledge of the number of clusters, which may not always be realistic. Future work could explore strategies for automatically estimating or adapting the number of clusters.

Ethical Considerations

Data Privacy. We use public benchmarks (BANK77, CLINIC, Tweet) and two newly constructed ones based on public Adobe Lightroom and Codex reviews. All sources are public and free of personally identifiable information (PII), and comply with their respective terms of use. No sensitive or proprietary data is involved.

Bias in LLMs. Our method uses GPT-4o (Hurst et al., 2024) for topic summarization and assignment, which may reflect biases from its training data. While our balanced sampling strategy helps mitigate this, we acknowledge potential societal or linguistic bias. Future work could explore fairness-aware prompting or bias auditing.

Reproducibility. We document each step of our pipeline in detail. Grouping uses open-source libraries (such as KMeans), and LLM-based steps rely on modular, prompt-based methods. We also release the datasets and raw experimental results to support reproducibility.

Environmental Impact. To reduce compute cost, we limit LLM usage through lightweight grouping and representative sampling. Nonetheless, we recognize the environmental footprint of LLM inference and encourage future work on more efficient alternatives.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Pranjal Awasthi, Maria Florina Balcan, and Konstantin Voevodski. 2017. Local algorithms for interactive clustering. *Journal of Machine Learning Research*, 18(3):1–35.

Juhee Bae, Tove Helldin, Maria Riveiro, Sławomir Nowaczyk, Mohamed-Rafik Bouguelia, and Göran Falkman. 2020. Interactive clustering: A comprehensive review. *ACM Computing Surveys (CSUR)*, 53(1):1–39.

Sugato Basu, Arindam Banerjee, and Raymond J Mooney. 2002. Semi-supervised clustering by seeding. In *Proceedings of the nineteenth international conference on machine learning*, pages 27–34.

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

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

Rich Caruana. 2013. Clustering: probably approximately useless? In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, CIKM '13, page 1259–1260, New York, NY, USA. Association for Computing Machinery.

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

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

Anni Coden, Marina Danilevsky, Daniel Gruhl, Linda Kato, and Meena Nagarajan. 2017. A method to accelerate human in the loop clustering. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 237–245. SIAM.

Sajib Dasgupta and Vincent Ng. 2010. Which clustering do you want? inducing your ideal clustering with minimal feedback. *Journal of Artificial Intelligence Research*, 39:581–632.

William HE Day and Herbert Edelsbrunner. 1984. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification*, 1(1):7–24.

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

Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. 2016. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*.

Zijin Feng, Luyang Lin, Lingzhi Wang, Hong Cheng, and Kam-Fai Wong. 2024. Llmedgerefine: Enhancing text clustering with llm-based boundary point refinement. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18455–18462.

Amir Hadifar, Lucas Sterckx, Thomas Demeester, and Chris Develder. 2019. A self-training approach for short text clustering. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 194–199.

Peihao Huang, Yan Huang, Wei Wang, and Liang Wang. 2014. Deep embedding network for clustering. In *2014 22nd International conference on pattern recognition*, pages 1532–1537. IEEE.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

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 1 others. 2019. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Arjun Mitra, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 11:157–173.

J. B. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Chuang Niu, Hongming Shan, and Ge Wang. 2022. Spice: Semantic pseudo-labeling for image clustering. *IEEE Transactions on Image Processing*, 31:7264–7278.

Chuang Niu, Jun Zhang, Ge Wang, and Jimin Liang. 2020. Gatcluster: Self-supervised gaussian-attention network for image clustering. In *European Conference on Computer Vision*, pages 735–751. Springer.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ofir Press, Noah A. Smith, and Mike Levy. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 280–291.

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

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, and Tao Yu. 2022. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*.

Yaling Tao, Kentaro Takagi, and Kouta Nakata. 2021. Clustering-friendly representation learning via instance discrimination and feature decorrelation. *arXiv preprint arXiv:2106.00131*.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. 2020. Scan: Learning to classify images without labels. In *European conference on computer vision*, pages 268–285. Springer.

Vijay Viswanathan, Kiril Gashteovski, Kiril Gashteovski, Carolin Lawrence, Tongshuang Wu, and Graham Neubig. 2024. Large language models enable few-shot clustering. *Transactions of the Association for Computational Linguistics*, 12:321–333.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Zihan Wang, Jingbo Shang, and Ruiqi Zhong. 2023. Goal-driven explainable clustering via language descriptions. *arXiv preprint arXiv:2305.13749*.

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

Jianwei Yang, Devi Parikh, and Dhruv Batra. 2016. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5147–5156.

Jianhua Yin and Jianyong Wang. 2016. A model-based approach for text clustering with outlier detection. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 625–636. IEEE.

Dejiao Zhang, Feng Nan, Xiaokai Wei, Shangwen Li, Henghui Zhu, Kathleen McKeown, Ramesh Nallapati, Andrew Arnold, and Bing Xiang. 2021. Supporting clustering with contrastive learning. *arXiv preprint arXiv:2103.12953*.

Hongjing Zhang, Sugato Basu, and Ian Davidson. 2019. A framework for deep constrained clustering-algorithms and advances. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 57–72. Springer.

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

Sheng Zhou, Hongjia Xu, Zhuonan Zheng, Jiawei Chen, Zhao Li, Jiajun Bu, Jia Wu, Xin Wang, Wenwu Zhu, and Martin Ester. 2024. A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. *ACM Computing Surveys*, 57(3):1–38.

Zhihan Zhou, Dejiao Zhang, Wei Xiao, Nicholas Dingwall, Xiaofei Ma, Andrew O Arnold, and Bing Xiang. 2022. Learning dialogue representations from consecutive utterances. *arXiv preprint arXiv:2205.13568*.

A LLM Prompts

This appendix provides the complete prompt templates used in our CLUSTERFUSION framework for both topic summarization and topic assignment stages.

A.1 Topic Summarization Prompt

The following prompt template is used to extract candidate topics from the sampled and sorted records. The prompt follows a two-message structure:

> **System message:** *You are an intelligent assistant skilled in summarizing and extracting insights.*
>
> **User message:** *You are now tasked with reviewing records related to {feature_context}. Each record is separated by ";" symbols.*
>
> *Records: {concatenated_records}*
>
> *Your goal is to extract the key topics from these records. For each identified topic, please provide a brief explanation along with examples. The total number of topics you extract should be exactly {num_topics}, not more than {num_topics} and not fewer than {num_topics}.*
>
> *The result should be returned in JSON format, where each key represents an index, and the corresponding value is a dictionary with:*
>
> - *A topic name as the key, and*
> - *A description and some examples as the value.*
>
> *{extra_guidance if extra_guidance else ""}*

where `feature_context` specifies the domain or context of the dataset, `concatenated_records` contains the sampled and sorted records from D'_S (as generated in Equation 1-5) concatenated with ";" separators, `num_topics` is the target number of clusters K, and the optional `extra_guidance` placeholder can be used to provide additional user preference when needed.

A.1.1 Domain-Specific Adaptations

For domain-specific datasets, we provide task-specific context and preferences to guide topic formation. Below we show the adaptations for the OpenAI Codex dataset as an illustrative example:

Feature Context (Domain Knowledge):

> *OpenAI Codex feature announcement video posted on YouTube. Comments are from viewers responding to the demo by Fouad Matin and Romain Huet, showing Codex CLI working with OpenAI models including o3, o4-mini, and GPT-4.1. In this domain, Manus, Aider, Deepseek, Claude (by Anthropic), Cursor, Warp are competitors; Gemini and Deepseek are non-OpenAI models. There are also certain YouTube channels*

making tech tutorial/review videos, such as Fireship and Lucas Montano. Notably, the recorded demo had some staging details: a salad bowl placed on the table, the interface displayed in light mode, and a smiley-face sticker covering the Apple logo on the MacBook.

Extra Guidance (User Preferences):

Special instructions for topic formation:

- *If the comment is not related to codex feature but related to presentation or presenters (such as salad bowl, MacBook sticker), consolidate them into one cluster*

These adaptations directly enable the qualitative improvements shown in Table 4, where domain terminology is correctly interpreted and non-substantive feedback is consolidated. The complete prompts for all datasets will be released on GitHub to support reproducibility.

A.2 Topic Assignment Prompt

For classifying each record into one of the extracted topics, we use the following two-message prompt structure:

System message: *You are a helpful assistant, that can help me label each record into topics.*

User message: *Following records is about {feature_context}. Please classify the record into one of the following topics, which are represented as a dictionary. Its keys are the names of the topics and values are the descriptions of the topic: {topic_def_dict}*

Comment: {comment}

Return the result in JSON format with the following format: key 'topic', with value as the picked topic;

where `topic_def_dict` contains the topics and descriptions generated in the summarization stage, and `comment` is the individual record being classified.