LSHC: LLM-GUIDED SEMANTIC HIERARCHICAL CLUSTERING FOR IMBALANCED AND MULTI-GRANULAR TEXT DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Text clustering is a popular research topic that faces challenges such as multi-level granularity, cluster size imbalance, and local structure ambiguity in the embedding space. We propose LLM-Guided Semantic Hierarchical Clustering (LSHC), a bottom-up framework that combines embedding-based text clustering with semantic judgment from a large language model (LLM). LSHC constructs fine-grained initial clusters, which can preserve small semantic clusters. This makes the framework particularly suitable for imbalanced data. Then, an LLM is used to determine an appropriate optimization operation: MERGE, KEEP, or RECLUSTER. MERGE combines semantically similar micro-clusters, KEEP preserves distinct neighbor clusters, and RECLUSTER prevents error assignments from propagating to higher semantic levels by reorganizing micro-clusters with mixed semantics. Iteratively conducting these three operations constructs a semantic hierarchy and can be terminated according to the desired cluster granularity. Experiments on six benchmarks show that LSHC achieves notable improvements on several highly imbalanced datasets, particularly MASSIVE-Intent and MTOP-Intent.

1 INTRODUCTION

Text clustering aims to assign unsupervised text to semantically consistent clusters. It is a fundamental component of many natural language processing applications, including intent discovery, topic analysis, data organization, and recommendation systems (Dai et al., 2024; Leiber et al., 2024). With the rapid development of pretrained language models, current embedding methods can provide meaningful semantic representations for text datasets, making it possible to apply traditional clustering algorithms directly in their embedding space (Ethayarajh, 2019; Fan et al., 2020). However, real-world text datasets have more complicated structures that affect the effectiveness of traditional clustering methods. In particular, cluster sizes can be highly imbalanced, assignment errors can be propagated to higher semantic levels, and the appropriate semantic granularity is commonly unknown in advance (Sun et al., 2024; Gong et al., 2025). These challenges make it difficult for a single clustering result to represent the potential semantic structure of a dataset.

Cluster imbalance is particularly challenging for text clustering (Lu et al., 2019; Alex, 2025; Altalhan et al., 2025). Distance-based clustering algorithms may struggle when cluster sizes are imbalanced. Some small but semantically meaningful clusters may be absorbed into larger clusters. This issue is especially important in real-life applications (Boussougou et al., 2025; Dong et al., 2025). For intent discovery, some intents may contain only a small part of the samples but still represent a meaningful semantic category. In this case, we propose a bottom-up hierarchical clustering framework. It first assigns fine-grained micro-clusters and then aggregates them based on their semantic similarity. This framework ensures that meaningful small clusters can be preserved in an independent branch of the hierarchy, instead of being immediately absorbed by a dominant cluster. Therefore, it is suitable for exploring semantic structures under heterogeneous and imbalanced cluster sizes (Lu et al., 2024).

Nevertheless, the current agglomerative hierarchical clustering method introduces another problem: the bottom clustering errors will be propagated to higher levels. Once two inaccurate bottom micro-clusters are merged, the incorrect samples remain collected in all subsequent levels of the hierarchy.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

These errors will propagate upward and progressively contaminate higher-level semantic structures. To address this issue, we introduce a reclustering operation to the hierarchical aggregation process. Before conducting a merge operation, the framework evaluates representative samples from both the center and boundary regions of the candidate pairwise clusters. If the two clusters exhibit consistent semantics, they can be merged directly. On the contrary, if their samples have mixed or inconsistent semantic structures, the corresponding samples are reclustered. Reclustering serves as an error-correction mechanism: it can reorganize previously misassigned samples before the hierarchy continues, preventing local clustering errors from being permanently preserved in subsequent semantic levels.

Large language models (LLMs) play an important role in recent text clustering methods by providing strong semantic reasoning capabilities (Bystroński et al., 2025; Hong et al., 2025; Arık et al., 2026). However, directly using LLMs to assign every sample to construct clusters can induce enormous inference latency and token consumption. Our proposed framework separates semantic decision-making from numerical clustering. Embedding-based clustering algorithms are responsible for computationally intensive processes such as similarity estimation, candidate selection, merging, and reclustering (Ethayarajh, 2019; Fan et al., 2020; Li et al., 2020). The LLM is only applied to determine which clustering operation should be performed, rather than directly assigning samples. Specifically, representative samples near the cluster centers stand for the dominant semantics of the cluster, while selected boundary samples present potentially ambiguous or overlapping structures. The LLM evaluates this semantic evidence and determines the appropriate operation. This design concentrates expensive and time-consuming language-model reasoning on a small number of decision cases, leaving the massive assignment process to efficient embedding-based cluster algorithms, significantly reducing token usage and inference time.

Moreover, the proposed framework naturally constructs multiple levels of semantic granularity (Hanley & Durumeric, 2025; Li et al., 2025). Textual semantics often allow multiple valid levels of granularity. For example, fine-grained intents such as "turn a device on" and "turn a device off" may be assigned to distinct clusters at a lower level, while these two intents can also be aggregated to a broader "device control" category at a higher level. A single clustering result presents only one such granularity and typically requires the desired number of clusters to be determined in advance. In contrast, our bottom-up framework progressively aggregates semantic clusters, producing a hierarchy ranging from fine-grained intents to coarse semantic domains. The variety of hierarchical structures can be regarded as different valid semantic clusters of the same dataset. Therefore, downstream applications are able to select an appropriate granularity according to their requirements.

Based on these considerations, we propose an LLM-guided semantic hierarchical text clustering framework that combines the efficiency of embedding-based clustering with the semantic reasoning capability of LLMs. It treats LLMs as a lightweight semantic controller that determines how the clustering structure should be optimized. Through hierarchical aggregation, small semantic clusters can be well-separated in cluster-imbalanced datasets. The recluster operation can correct potential local clustering errors before they propagate to higher levels. The final hierarchical clustering result provides a representation of semantic structures at multi-level granularities.

The main contributions of this work are summarized as follows: 1. Imbalance-capable hierarchical semantic clustering. We develop a bottom-up framework in which small but semantically consistent clusters can be preserved in multiple levels of granularity. 2. RECLUSTER as assignment error correction. We introduce a reclustering operation that can reorganize semantically mixed candidate clusters before further aggregation. 3. LLM guidance with lower inference cost. Instead of asking an LLM to cluster the complete dataset, we use it only to determine the appropriate operation for clustering optimization. 4. Multi-granularity semantic structure discovery. This framework allows the same dataset to be represented at multiple levels of granularity, from fine-grained intents to broader semantic domains.

2 Related Work

Recent studies have explored the integration of large language models (LLMs) into text clustering. The capabilities of LLMs can help embedding-based clustering algorithms with semantic awareness. While directly using LLMs to cluster every sample can achieve decent clustering performance, an

emerging direction treats LLMs as semantic supervisors that provide additional constraints, refine uncertain assignments, or construct more representative cluster semantics.

LLM-guided clustering. ClusterLLM (Zhang et al., 2023) is a valuable framework that uses an instruction-tuned LLM to guide text clustering without requiring access to the internal embeddings. It queries the LLM with reasonably selected triplet questions for semantic supervision that serve as supervision for fine-tuning a smaller embedding model. It further employs pairwise queries to determine an appropriate clustering granularity from the cluster hierarchy. More recently, LLMEdgeRefine (Feng et al., 2024) focuses on incorrect assignments near the cluster boundaries. It identifies edge points that are easy to assign incorrectly. Then, it introduces LLM-based refinement to iteratively improve their cluster assignments. The method additionally constructs super-points to reduce the influence of outliers, allowing clustering methods to utilize semantic information provided by the LLM. This strategy demonstrates the potential of applying expensive LLM reasoning to ambiguous samples instead of querying the entire dataset.

LLMs for cluster representation and representation learning. Another direction of research applies LLM semantics to cluster-level representations. *Summaries as Centroids for Interpretable and Scalable Text Clustering* (Diaz-Rodriguez, 2026) proposes k-NLPmeans and k-LLMmeans, which replace traditional numerical centroids with textual summaries of cluster contents. These summaries serve as an interpretable semantic category, while the clustering process still occurs in the embedding space. LLMs have also been used to improve the latent space before clustering. PPLL (Zhang & Li, 2025) first employs an LLM to generate mini-cluster labels and then constructs positive pairs according to these labels. A BYOL-based objective is then used to train the embedding model without constructing negative pairs. The final learned latent representations are clustered by k-means.

Relation to our method. Our work follows the general principle of using LLMs as semantic judges rather than a sample-level clustering engine. Compared with existing approaches, we first develop embedding-based hierarchical clustering to obtain candidate cluster structures. For each selected pair of clusters, selected representative samples from both the cluster centers and their boundary regions are provided to the LLM. It only determines an appropriate operation: MERGE, KEEP, or RECLUSTER. In this case, the language-model reasoning is concentrated on a small number of cluster-level judgments, while traditional clustering methods remain responsible for assigning the entire dataset.

3 METHOD

The proposed LLM-guided semantic hierarchical text clustering framework progressively constructs semantic structures from fine-grained clusters to coarse-grained clusters. Our framework separates numerical clustering from semantic decision-making. Embedding-based methods are responsible for reclustering and updating clusters, while the LLM is only used to determine how a selected pair of clusters should be processed.

Given an unsupervised text dataset

$$\mathcal{D} = \{x_1, x_2, \ldots, x_N\}, \tag{1}$$

a text embedding method $f_\theta(\cdot)$ first maps each sample into a d-dimensional embedding space:

$$\mathbf{z}_i = f_\theta(x_i), \qquad \mathbf{z}_i \in \mathbb{R}^d. \tag{2}$$

After obtaining an initial fine-grained clustering, the proposed method iteratively selects neighboring cluster pairs, collects representative samples from both clusters in each pair, and queries an LLM to determine one of three operations:

$$o_{ij} \in \{\texttt{MERGE}, \texttt{KEEP}, \texttt{RECLUSTER}\}. \tag{3}$$

The selected operation is then executed using an embedding-based clustering method. This procedure is repeated until a stopping criterion is satisfied. Since clusters are progressively aggregated from fine-grained structures, the hierarchical clustering can naturally provide different levels of semantic granularity.

3.1 HIERARCHICAL CLUSTERING SETTING

The clustering is conducted in a bottom-up manner. An embedding-based clustering algorithm first generates an initial fine-grained partition: $\mathcal{P}^{(0)} = \{C_1, C_2, \ldots, C_{K_0}\}$, where K_0 denotes the initial number of clusters. For each cluster C_i, its centroid is computed as

$$\boldsymbol{\mu}_i = \frac{1}{|C_i|} \sum_{x_n \in C_i} \mathbf{z}_n. \tag{4}$$

At iteration t, the current partition is represented as $\mathcal{P}^{(t)} = \{C_1^{(t)}, C_2^{(t)}, \ldots, C_{K_t}^{(t)}\}$, where K_t is the current number of clusters. Each iteration has four major steps: 1. identify a neighboring cluster pair; 2. select representative samples from the pair; 3. query the LLM to determine the operation; 4. execute the operation and update the clustering structure.

Through repeated iterations, the initial fine-grained partition is progressively transformed into higher-level semantic structures.

3.2 CLUSTER PAIR SELECTION

Evaluating every possible pair of clusters using an LLM would require $\mathcal{O}(K_t^2)$ semantic comparisons. Instead, we first use the embedding space to identify geometrically close cluster pairs.

For two clusters C_i and C_j, their similarity is calculated by the cosine similarity between their centroids:$s_{ij} = \cos(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)$.

The available cluster pairs are ranked according to their centroid similarity. The most similar available pair is evaluated first:

$$(C_i, C_j) = \arg \max_{(C_p, C_q) \notin \mathcal{R}_{reject}} s_{pq}, \tag{5}$$

where $\mathcal{R}_{reject}$ denotes the set of cluster pairs that have already received a `KEEP` decision under the current clustering state. When a pair receives `KEEP`, it is temporarily excluded from subsequent candidate selection. The algorithm therefore evaluates the next most similar available pair rather than repeatedly selecting the same rejected pair.

3.3 REPRESENTATIVE CANDIDATE SELECTION

Centroid similarity identifies which pair of clusters should be compared but does not provide evidence to determine their semantic relationship. Collecting all samples from a pair of clusters to the LLM would also result in unnecessary token and inference costs. Therefore, we collect a small set of informative samples to represent each cluster.

For a selected pair (C_i, C_j), four candidate sets are constructed:

$$\mathcal{E}_{ij} = \{I_i, I_j, B_{i \to j}, B_{j \to i}\}. \tag{6}$$

Here, I_i and I_j represent the internal semantics of these two clusters, while $B_{i \to j}$ and $B_{j \to i}$ represent potentially ambiguous samples near the cluster boundary.

For each cluster, we select k samples closest to its centroid. For cluster C_i and cluster C_j ,

$$I_i = \mathrm{TopK}_{x \in C_i} \cos(\mathbf{z}_x, \boldsymbol{\mu}_i), I_j = \mathrm{TopK}_{x \in C_j} \cos(\mathbf{z}_x, \boldsymbol{\mu}_j). \tag{7}$$

These samples represent the dominant semantics of the corresponding clusters.

We additionally select samples that are far from their own cluster center but close to the center of the other cluster. For $x \in C_i$, we define its cross-cluster tendency toward C_j as

$$r(x; i \to j) = \cos(\mathbf{z}_x, \boldsymbol{\mu}_j) - \cos(\mathbf{z}_x, \boldsymbol{\mu}_i). \tag{8}$$

A larger value indicates that the sample is more similar to the neighboring cluster than to its current cluster. We therefore select

$$B_{i \to j} = \mathrm{TopK}_{x \in C_i} r(x; i \to j), B_{j \to i} = \mathrm{TopK}_{x \in C_j} r(x; j \to i). \tag{9}$$

The four candidate sets provide complementary semantic evidence:

$$\underbrace{I_i, I_j}_{\text{core semantics}} \quad + \quad \underbrace{B_{i\to j}, B_{j\to i}}_{\text{boundary semantics}} . \tag{10}$$

Only the original texts corresponding to these selected samples are provided to the LLM.

3.4 LLM-Guided Operation Selection

Given the representative candidates, the LLM determines which operation should be performed:

$$o_{ij} = g_{\text{LLM}} \left(I_i, I_j, B_{i\to j}, B_{j\to i} \right), \tag{11}$$

The LLM does not assign each sample to clusters. Instead, it works as a semantic operation selector. The decision considers both the dominant semantics represented by the inside candidates and the possible semantic mixture represented by the boundary candidates. This design restricts LLM reasoning to a local operation decision, while sample-level clustering is conducted by an embedding-based clustering method.

3.5 Hierarchical Operations

Based on the LLM decision, one of three operations is applied to the selected pair.

MERGE. The `MERGE` operation is selected when the representative samples indicate that C_i and C_j correspond to the same semantic concept: $\mathcal{S}(C_i) = \mathcal{S}(C_j) = \mathcal{S}(B_{i\to j}) = \mathcal{S}(B_{j\to i})$.

The two clusters are replaced by their union: $C_{\text{new}} = C_i \cup C_j$.

The partition is updated as $\mathcal{P}^{(t+1)} = \left(\mathcal{P}^{(t)} \setminus \{C_i, C_j\}\right) \cup \{C_{\text{new}}\}$, and $K_{t+1} = K_t - 1$.

The centroid of the new cluster and its similarities to the remaining clusters are then updated.

KEEP. The `KEEP` operation is selected when C_i and C_j represent different semantic concepts and their current separation is appropriate: $\mathcal{S}(C_i) = \mathcal{S}(B_{i\to j}), \mathcal{S}(C_j) = \mathcal{S}(B_{j\to i})$, and $\mathcal{S}(C_i) \neq \mathcal{S}(C_j)$.

No further process is applied to this pair of clusters: $\mathcal{P}^{(t+1)} = \mathcal{P}^{(t)}$.

The pair is instead recorded in the rejected-pair set: $\mathcal{R}_{reject} \leftarrow \mathcal{R}_{reject} \cup \{(C_i, C_j)\}$.

The algorithm then evaluates the next most similar pair. This mechanism prevents the workflow from repeatedly selecting the same nearest pair after the LLM has determined that this pair of clusters should remain separate.

RECLUSTER. The `RECLUSTER` operation is selected when the candidate samples indicate that the current partition between C_i and C_j is semantically mixed or contains potentially incorrect assignments: $\mathcal{S}(C_i) \neq \mathcal{S}(B_{i\to j}), \mathcal{S}(C_j) \neq \mathcal{S}(B_{j\to i})$

The two clusters are first combined into a local sample set: $U_{ij} = C_i \cup C_j$. An embedding-based clustering algorithm is then reapplied locally: $\mathcal{Q}_{ij} = \text{Cluster}\left(U_{ij}\right)$, where $\mathcal{Q}_{ij} = \{C'_1, C'_2, \ldots, C'_r\}$. The original pair of clusters is replaced by the new local clusters:

$$\mathcal{P}^{(t+1)} = \left(\mathcal{P}^{(t)} \setminus \{C_i, C_j\}\right) \cup \mathcal{Q}_{ij}. \tag{12}$$

Compared with `MERGE`, the design of `RECLUSTER` is not to reduce the number of clusters. Instead, it serves as a local purification mechanism that allows incorrectly assigned samples to be reorganized between clusters before hierarchical aggregation.

3.6 Iterative Hierarchical Construction

After an operation is executed, the clustering structure is updated, and the next iteration begins. For `MERGE` and `RECLUSTER`, the affected cluster centroids and pairwise similarities are recalculated because the global cluster structure has changed. Previously recorded `KEEP` rejected-pair sets are also invalidated.

For KEEP, the partition remains unchanged and the evaluated pair is added to $\mathcal{R}_{reject}$. The algorithm subsequently selects the next most similar available pair according to Eq. 5.

The overall iterative process can therefore be summarized as Pair Selection, Candidate Selection, LLM Decision, and Update. As MERGE operations progressively combine similar fine-grained clusters, the clustering structure generated throughout this process naturally represents different levels of semantic granularity.

The hierarchical iteration terminates when any of three stopping conditions is satisfied.

Maximum iterations. To bound the computational cost, the procedure terminates when $t \geq T_{\max}$, where $T_{\max}$ denotes the maximum number of iterations.

Top-m KEEP decisions. If the most similar cluster pairs are consistently judged to represent different semantic concepts, the current clustering result can be regarded as a stable semantic level. We therefore terminate the procedure when the top m candidate cluster pairs all receive KEEP decisions. We use $m = 5$ by default: $o_{p_1 q_1} = o_{p_2 q_2} = \cdots = o_{p_5 q_5} = \texttt{KEEP}$.

This provides a semantic stopping signal without introducing any additional manually tuned similarity threshold.

Desired number of clusters. When a predefined number of clusters K_{target} is specified, the procedure terminates once $K_t \leq K_{\text{target}}$.

This allows the framework to generate a clustering result at a desired semantic granularity. Overall, the stopping condition is satisfied if any one of these three conditions is met.

Algorithm 1 summarizes the complete framework.

4 Experimental results

Datasets. We evaluate the proposed framework on six text clustering datasets: MASSIVE-Intent, MASSIVE-Domain, MTOP-Intent, MTOP-Domain, FewEvent, and GoEmotions. The dataset statistics are summarized in Table 3. These six datasets present slight to extremely severe class imbalance, with imbalance ratios ranging from 2.3 to 1616.0. MTOP-Intent represents the most extreme setting, containing 102 classes with an imbalance ratio of 1616.0. These datasets therefore provide suitable benchmarks for evaluating our method under imbalanced clustering scenarios, from fine-grained intent levels to coarse-grained domain levels.

Embedding Models. To evaluate the effectiveness of the proposed framework in different embedding methods, we employ three sentence embedding models: Gemma (Team et al., 2025), INSTRUCTOR (Su et al., 2023), and E5 (Wang et al., 2024). This setting also allows us to distinguish whether improvements are introduced by the clustering framework or the embedding model.

Baselines. We compare our method with both traditional clustering algorithms and four recent LLM-guided text clustering approaches: ClusterLLM, LLMEdgeRefine, k-NLPMeans, and PPLL. For traditional baselines, we employ k-means and spectral clustering, and evaluate both methods with three embedding models, resulting in six embedding-based baseline configurations.

LLM Configuration. We employ Qwen3-8B (Zhang et al., 2025) with its thinking/reasoning mode disabled for all LLM-based decisions in our framework. Compared with approaches that apply an LLM to directly cluster the entire dataset, our method only requires the LLM to determine the appropriate operation for a selected pair of clusters from a small set of representative samples. The prompt of the LLM is presented in Appendix A.1

Initialization and Candidate Sampling. We use k-means to construct the initial fine-grained clustering, which serves as the starting point for the further bottom-up clustering process. The initial number of clusters is set to twice the target number of clusters. For each cluster in a selected pair, we extract three samples closest to its cluster center to represent its inner semantics and three boundary samples that are far from their own center and close to another cluster center.

Recluster. We use spectral clustering for the RECLUSTER operation, because the samples within two ambiguous clusters may form complex or manifold-like structures that cannot be adequately separated according to their distances to the centroids. The number of clusters is decided by the largest gap in the first 10 eigenvalues.

Target Cluster Number and Stopping Criteria. Text clustering is unsupervised, and our framework does not require ground-truth labels to perform its cluster operations. However, in order to evaluate external clustering metrics, we set the target number of clusters to the number of ground-truth labels for each dataset. The target cluster number is only one of the stopping conditions rather than a mandatory requirement for executing the framework. Thus, the method may terminate before reaching the predefined target if the clustering structure is already stable according to the other two criteria.

4.1 RESULTS

For quantitative evaluation, we employ Clustering Accuracy (ACC) and Normalized Mutual Information (NMI). The Hungarian algorithm is used to find the one-to-one mapping between predicted clusters and ground-truth labels for accuracy measurement, and NMI measures shared information between true classes and clusters.

Table 1: Clustering performance on benchmark datasets.

Method	Encoder	Massive-I		Massive-D		MTOP-I		MTOP-D		Few-Event		GoEmotions	
		ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
k-means	E5	53.53	71.45	52.96	61.27	33.95	71.19	82.40	85.72	40.72	63.28	21.28	19.80
	INSTRUCTOR	54.24	70.44	57.26	62.89	31.44	66.71	90.99	84.72	38.44	61.06	21.94	17.40
	Gemma	68.19	82.01	75.45	79.57	51.48	83.36	**98.59**	**96.83**	38.21	64.03	31.85	32.95
Spectral	E5	55.31	69.59	63.82	62.24	33.49	67.22	84.86	83.08	57.53	65.27	21.75	19.61
	INSTRUCTOR	53.93	69.97	63.89	66.73	31.60	65.53	85.91	78.82	57.55	64.17	25.71	20.57
	Gemma	72.39	81.67	61.84	70.72	52.74	82.55	68.60	82.31	**58.81**	68.92	33.28	30.89
	ClusterLLM	60.69	77.64	60.85	68.67	35.04	73.83	92.13	89.23	50.60	**73.89**	26.75	23.89
	LLMEdge	63.42	76.66	63.05	68.67	46.00	72.92	92.89	88.19	–	–	34.76	29.74
	k-NLPmeans	56.70	73.90	63.20	70.60	–	–	–	–	–	–	24.20	22.30
	PPLL	61.21	75.72	62.37	68.39	40.10	75.14	91.74	88.67	–	–	**39.82**	**35.00**
LSHC	E5	57.70	71.41	53.73	59.63	58.50	75.43	58.66	72.64	56.05	66.63	16.73	21.46
	INSTRUCTOR	57.63	71.05	58.34	61.52	54.10	69.94	69.77	74.64	46.75	63.07	17.24	13.33
	Gemma	**74.68**	**83.00**	**75.99**	**79.80**	**87.57**	**92.44**	84.93	91.37	54.96	66.74	32.41	27.20

Table 1 reports the clustering performance of the proposed method against traditional clustering algorithms and recent LLM-guided text clustering baselines. Overall, the results demonstrate that clustering performance is strongly affected by both the embedding space and the clustering framework. In particular, the LSHC framework combined with Gemma embeddings achieves strong performance on several challenging and severely imbalanced datasets. However, its advantage is not obvious on datasets where the initial embedding space already presents a relatively simple clustering structure.

Comparison with traditional clustering methods. Compared with k-means and spectral clustering, the proposed method shows improvements on several datasets, particularly with Gemma embeddings. On MASSIVE-I, LSHC with Gemma achieves an ACC of 74.68 and an NMI of 83.00, outperforming k-means and spectral clustering. A similar pattern can be observed on MTOP-I, where the improvement is obviously larger. These results suggest that directly applying a traditional clustering algorithm to the embedding space may be insufficient in highly imbalanced semantic structures. The iterative cluster-level refinement of our method can effectively reorganize such initial clusters. The results on MASSIVE-D further demonstrate this advantage. LSHC with Gemma embeddings achieves 75.99 ACC and 79.80 NMI, slightly improving over k-means-Gemma and spectral clustering.

Comparison with LLM-guided text clustering methods. The proposed method is also competitive with recent LLM-assisted clustering approaches, including ClusterLLM, LLMEdge, k-NLPMeans, and PPLL. The most notable result is observed on MTOP-I, where LSHC with Gemma embedding reaches 87.57 ACC and 92.44 NMI, and obviously outperforms the reported results of ClusterLLM, LLMEdge, and PPLL. On MASSIVE-I, LSHC also outperforms ClusterLLM,

LLMEdge, k-NLPMeans, and PPLL. On MASSIVE-D, the proposed method obtains 75.99/79.80, again outperforming the reported results of the four recent baselines. For all evaluated datasets, MTOP-I has the most severe imbalance ratio (1616.0), while MASSIVE-I also exhibits noticeable imbalance (57.8). The particularly large improvements on these datasets are consistent with the imbalance motivation of our bottom-up framework: relatively small semantic groups can be preserved and progressively merged rather than being immediately absorbed into dominant clusters. Moreover, the RECLUSTER operation provides an opportunity to correct locally mixed clusters before they propagate to higher levels of the hierarchy.

Results on MTOP-D, FewEvent, and GoEmotions. The proposed method does not uniformly outperform the baselines on all datasets. This suggests that when the embedding space already forms well-separated, centroid-friendly clusters, additional hierarchical refinement may provide limited benefit and may even RECLUSTER an already effective cluster.

4.2 Ablation Study

Table 2 presents the ablation study investigating the contribution of the RECLUSTER operation by comparing the LSHC with a *merge-only* variant on MTOP-I and MASSIVE-I. Overall, introducing the RECLUSTER operation improves clustering performance for most embedding combinations. It demonstrates that the MERGE operation is insufficient to correct errors from the initial clustering.

The improvement is particularly obvious on MTOP-I. With Gemma embeddings, introducing RECLUSTER increases ACC from 74.94 to 87.57 and NMI from 88.67 to 92.44, corresponding to improvements of 12.63 and 3.77 percentage points, respectively. E5 embeddings also present similar results. For INSTRUCTOR, although NMI slightly decreases from 70.36 to 69.94, ACC improves from 48.24 to 54.10. These results indicate that RECLUSTER is particularly beneficial for MTOP-I, which exhibits the most severe cluster imbalance in the evaluated datasets. A similar trend is observed on MASSIVE-I. RECLUSTER improves the ACC for all three embedding models. These improvements suggest that the benefit of the RECLUSTER operation is not restricted to a particular latent space.

These observations support the motivation for introducing the RECLUSTER operation. In a merge-only bottom-up procedure, incorrectly mixed samples within existing clusters cannot be corrected. MERGE can only combine clusters without modifying their internal assignments, resulting in errors propagated to higher levels of the hierarchy. In contrast, the RECLUSTER operation reorganizes overlapped cluster pairs using spectral clustering, allowing incorrectly assigned samples to be reclustered before further merging. Overall, the ablation results demonstrate that local reclustering provides an effective error-correction mechanism for the hierarchical clustering process, with particular improvements on the severely imbalanced MTOP-I dataset.

5 Case Study

To provide a more intuitive understanding of how the LSHC framework iteratively refines the clustering structure, we conduct a case study on a subset extracted from the MASSIVE-Intent dataset. We randomly select samples belonging to 10 ground-truth intent labels and initialize the clustering process with $K_{\text{init}} = 15$ clusters. The target number of clusters is set to $K_{\text{target}} = 10$ for evaluation. Starting from this over-clustering result, the proposed method reaches the target number after seven cluster operations. The entire procedure contains five MERGE operations, one KEEP operation, and one RECLUSTER operation, demonstrating how these operations collaboratively construct the final clustering.

The first two iterations demonstrate the basic bottom-up merging behavior. In the first operation, clusters 0 and 10 are identified as representing the same general intent of *playing music*. Similarly, clusters 8 and 12 are merged in the second operation because both primarily describe *playing or resuming audiobook*. These examples show that the framework can iteratively aggregate micro-clusters with the over-cluster initialization. The next iteration illustrates that neighboring clusters are not always merged. During the third cluster-pair decision, clusters 0 and 11 are selected as candidates, but the LLM assigns a KEEP operation. Cluster 0 mainly focuses on playing music playlists and modifying them, while cluster 11 expresses a preference for specific music genres. Although both clusters belong to the broader music domain and are therefore close in the embedding space,

they are classified into different fine-grained intents. The KEEP decision prevents these semantically distinct clusters from being incorrectly combined.

The case study also demonstrates why a merge-only strategy is insufficient. For clusters 0 and 8, the representative samples represent music playlists and audiobooks, respectively. However, their boundary samples contain mixed semantics, such as "add this song", "resume playback". In this case, the LLM suggests RECLUSTER. All samples from the two clusters are then reclustered using spectral clustering.

The rest of the operations continue to enhance semantic consistency. Clusters 7 and 14 are merged because both express intents related to setting or waking up at a specific time. Clusters 11 and 14 are merged based on their playing-music semantics. Clusters 4 and 6 are merged because both are related to playing games. After these operations, the number of clusters decreased from the initial 15 to the target number of 10. The details of the case study are shown in Figure 1. The t-SNE figures of clusters in each iteration are presented in Figures 2 to 8.

6 CONCLUSION

In this work, we presented an LLM-guided semantic hierarchical clustering framework. It is designed particularly to handle imbalanced text data and support clustering at different levels of semantic granularity. Instead of applying LLMs to directly assign samples, our framework introduces them only to determine the appropriate operation between selected candidate pairs of clusters. Based on representatives collected from center and boundary samples, three operations: MERGE, KEEP, and RECLUSTER are used to iteratively construct and refine the clustering structure. In particular, the RECLUSTER operation provides a purification mechanism to correct local error assignments. This lightweight use of the LLM also helps to reduce token usage and inference time.

Experiments on six text clustering benchmarks with different embedding models demonstrate the effectiveness of LSHC, with obvious improvements on severely imbalanced datasets such as MASSIVE-I and MTOP-I. The ablation study further shows the importance of RECLUSTER operation compared with a merge-only strategy. Moreover, the case study illustrates how the three operations interact during the clustering process: semantically consistent clusters are iteratively merged, meaningful fine-grained clusters are preserved, and ambiguous boundary samples can be locally reassigned. These results demonstrate the potential of combining traditional clustering algorithms with lightweight LLM-based semantic judgment. The LLM serves as a decision-maker rather than directly clustering the entire dataset.

However, the effectiveness of the LSHC framework is highly dependent on the semantic quality of the embedding space, and the improvements are not uniform for all datasets and embeddings. Future work should explore more robust initialization and stopping conditions, further reduce the number of required LLM decisions, and automatically identify the meaningful semantic granularity without a target number of clusters.

AI USE STATEMENT

In this work, Generative AI tools assisted with manuscript editing, and we used generative AI tools as an LLM judge in our proposed method. We have not used generative AI tools for other tasks that require disclosure. Additionally, we used generative AI tools to assist with coding. We have reviewed all AI-assisted work. We take responsibility for the final text, mathematical claims, and empirical reporting.

REPRODUCIBILITY STATEMENT

We release code to reproduce all results in this paper at https://github.com/research-paper-code/LSHC

REFERENCES

Suja A Alex. Imbalanced data learning using smote and deep learning architecture with optimized features. *Neural Computing and Applications*, 37(2):967–984, 2025.

Manahel Altalhan, Abdulmohsen Algarni, and Monia Turki-Hadj Alouane. Imbalanced data problem in machine learning: A review. *IEEE access*, 13:13686–13699, 2025.

Ahmet Okan Arık, Gizem Parlayandemir, and Serra Çelik. Llm-based data augmentation for text classification on imbalanced datasets: A case study on fake news detection. *Egyptian Informatics Journal*, 33:100886, 2026.

Milandu Keith Moussavou Boussougou, Prince Hamandawana, and Dong-Joo Park. Enhancing voice phishing detection using multilingual back-translation and smote: an empirical study. *IEEE Access*, 13:37946–37965, 2025.

Mateusz Bystroński, Mikołaj Hołysz, Grzegorz Piotrowski, Nitesh V Chawla, and Tomasz Kajdanowicz. Smotext: Smote meets large language models. *arXiv preprint arXiv:2505.13434*, 2025.

Hao Dai, Yang Liu, Peng Su, Hecheng Cai, Shudong Huang, and Jiancheng Lv. Multi-view clustering by inter-cluster connectivity guided reward. In *Forty-first International Conference on Machine Learning*, 2024.

Jairo Diaz-Rodriguez. Summaries as centroids for interpretable and scalable text clustering. In *International Conference on Learning Representations*, volume 2026, pp. 43607–43630, 2026.

Zijian Dong, Yilei Wu, Chongyao Chen, Yingtian Zou, Yichi Zhang, and Juan Helen Zhou. Improve representation for imbalanced regression through geometric constraints. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5082–5091. IEEE, 2025.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 55–65, 2019.

Lu Fan, Guangfeng Yan, Qimai Li, Han Liu, Xiaotong Zhang, Albert YS Lam, and Xiao-Ming Wu. Unknown intent detection using gaussian mixture model with an application to zero-shot intent classification. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 1050–1060, 2020.

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

Ming Gong, Yuqing Zhou, and Yan Ma. Rism clustering algorithm based on relative density and inter-cluster connectivity degree. *Scientific Reports*, 15(1):41506, 2025.

Hans William Alexander Hanley and Zakir Durumeric. Hierarchical level-wise news article clustering via multilingual matryoshka embeddings. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2476–2492, 2025.

Mengze Hong, Wailing Ng, Chen Jason Zhang, Yuanfeng Song, and Di Jiang. Dial-in llm: Human-aligned llm-in-the-loop intent clustering for customer service dialogues. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5896–5911, 2025.

Collin Leiber, Niklas Strauß, Matthias Schubert, and Thomas Seidl. Dying clusters is all you need-deep clustering with an unknown number of clusters. In *2024 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 726–733. IEEE, 2024.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 9119–9130, 2020.

Yanhua Li, Xiaocao Ouyang, Chaofan Pan, Jie Zhang, Sen Zhao, Shuyin Xia, Xin Yang, Guoyin Wang, and Tianrui Li. Multi-granularity open intent classification via adaptive granular-ball decision boundary. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 24512–24520, 2025.

Yang Lu, Yiu-Ming Cheung, and Yuan Yan Tang. Bayes imbalance impact index: A measure of class imbalanced data set for classification problem. *IEEE transactions on neural networks and learning systems*, 31(9):3525–3539, 2019.

Yuyin Lu, Hegang Chen, Pengbo Mao, Yanghui Rao, Haoran Xie, Fu Lee Wang, and Qing Li. Self-supervised topic taxonomy discovery in the box embedding space. *Transactions of the Association for Computational Linguistics*, 12:1401–1416, 2024.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 1102–1121, 2023.

Li Sun, Zhenhao Huang, Hao Peng, Yujie Wang, Chunyang Liu, and Philip S Yu. Lsenet: Lorentz structural entropy neural network for deep graph clustering. *arXiv preprint arXiv:2405.11801*, 2024.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*, 2024.

XiaoTong Zhang and Ying Li. Improving clustering with positive pairs generated from llm-driven labels. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12213–12229, 2025.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

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

A APPENDIX

A.1 PROMPT FOR LLM

Prompt 1: Input for Qwen Model

You are evaluating two clusters of text samples.

INNER samples are nearest to a cluster center and represent its core meaning.

OUTER samples are farthest from a cluster center, close to another cluster center, and represent its boundary semantics and possible heterogeneity.

Choose exactly one operation:

MERGE: both clusters, including their inner and outer samples, express essentially the same semantic intent.

KEEP: the clusters are related but contain a meaningful, coherent distinction that should be preserved. Fine-grained distinctions such as ON/OFF, SEND/RECEIVE, CREATE/DELETE, or BEFORE/AFTER can require KEEP.

RECLUSTER: the current assignments appear mixed, inconsistent, or incorrectly partitioned, so the samples should be repartitioned locally.

Judge semantic meaning rather than lexical similarity. Return JSON only.

CLUSTER A

INNER:

A_inner_samples

OUTER:

A_boundary_samples

CLUSTER B

INNER:

B_inner_samples

OUTER:

B_boundary_samples

{{

”operation”: ”MERGE — KEEP — RECLUSTER”,

”reason”: ”brief semantic explanation”

}} ”””

A.2 CASE STUDY

"iteration": 0, "cluster_i": 0, "cluster_j": 10, "operation": "MERGE", "reason": "Both clusters center around the semantic intent of playing music, with inner samples focusing on general play commands and outer samples involving specific songs, playlists, or additional context. The distinction between clusters is not semantically meaningful and does not warrant separate categorization."

"iteration": 1, "cluster_i": 8, "cluster_j": 12, "operation": "MERGE", "reason": "Both clusters center around the semantic intent of playing or resuming an audiobook, with inner samples focusing on resuming playback and outer samples showing related but broader actions like pausing or playing different titles."

"iteration": 2, "cluster_i": 0, "cluster_j": 11, "operation": "KEEP", "reason": "Cluster A focuses on playing music playlists and modifying them, while Cluster B expresses a preference for specific music genres. The distinction between playlist manipulation and genre preference is meaningful and should be preserved."

"iteration": 2, "cluster_i": 7, "cluster_j": 14, "operation": "MERGE", "reason": "Both clusters revolve around the core intent of setting or waking up at a specific time, with inner samples expressing the primary action and outer samples showing variations or related but less direct expressions."

"iteration": 3, "cluster_i": 0, "cluster_j": 8, "operation": "RECLUSTER", "reason": "The clusters are mixed in semantic intent, with Cluster A focusing on music playlists and Cluster B on audiobooks, but outer samples in both clusters include actions that span both domains (e.g., 'add this song', 'resume playback'). This suggests a need for local repartitioning to better capture distinct semantic boundaries."

"iteration": 4, "cluster_i": 11, "cluster_j": 14, "operation": "MERGE", "reason": "Both clusters center around the intent of playing music, with INNER samples expressing a general desire to play music and OUTER samples showing variations in specific requests or additional actions related to playback."

"iteration": 5, "cluster_i": 4, "cluster_j": 6, "operation": "MERGE", "reason": "Both clusters revolve around the concept of playing or engaging in games, with INNER samples focusing on the act of playing and OUTER samples extending to related activities or variations of gameplay. The semantic intent is essentially the same."

Figure 1: Output of the case study in each iteration



Figure 2: ”iteration”: 0, ”cluster_i”: 0, ”cluster_j”: 10, ”operation”: ”MERGE”



Figure 3: ”iteration”: 1, ”clusterı”: 8, ”cluster_j”: 12, ”operation”: ”MERGE”



Figure 4: ”iteration”: 2, ”cluster_i”: 0, ”cluster_j”: 11, ”operation”: ”KEEP”



Figure 5: ”iteration”: 2, ”cluster_i”: 7, ”cluster_j”: 14, ”operation”: ”MERGE”

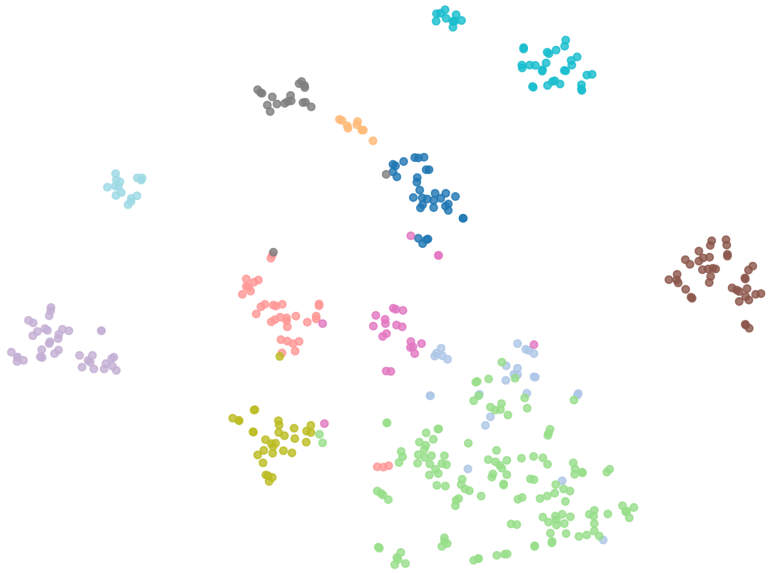

Figure 6: ”iteration”: 3, ”cluster_i”: 0, ”cluster_j”: 8, ”operation”: ”RECLUSTER”

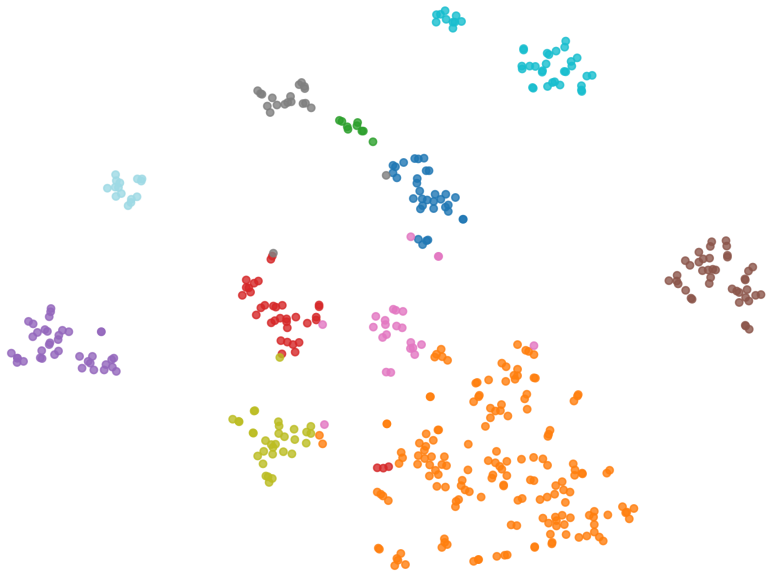

Figure 7: ”iteration”: 4, ”cluster_i”: 11, ”cluster_j”: 14, ”operation”: ”MERGE”

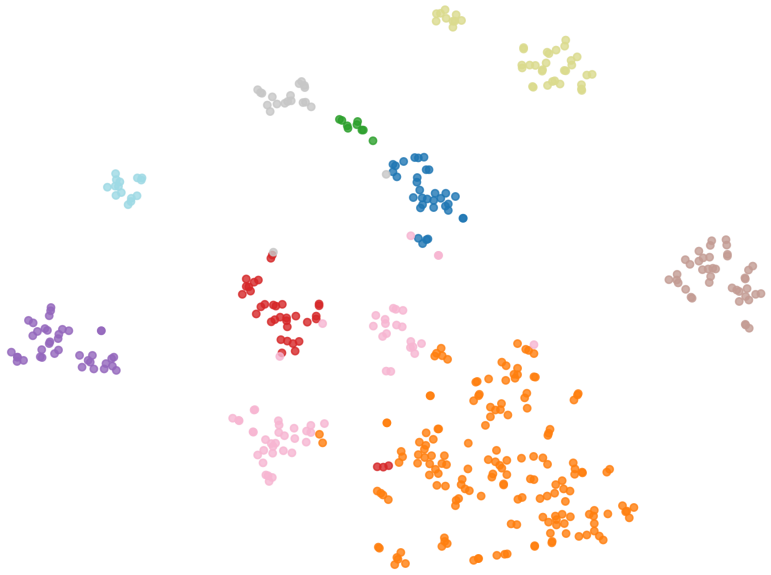

Figure 8: ”iteration”: 5, ”cluster_i”: 4, ”cluster_j”: 6, ”operation”: ”MERGE”

A.3 ABLATION FIGURE

Table 2: Ablation study of the merge-only and recluster strategies.

Method	Encoder	MTOP-I		Massive-I	
		ACC	NMI	ACC	NMI
Merge only	E5	51.89	73.85	54.20	70.61
	INSTRUCTOR	48.24	70.36	55.72	70.10
	Gemma	74.94	88.67	70.91	82.71
LSHC	E5	58.50	75.43	57.70	71.41
	INSTRUCTOR	54.10	69.94	57.63	71.05
	Gemma	**87.57**	**92.44**	**74.68**	**83.00**

A.4 STATISTICS OF THE BENCHMARK DATASETS

Table 3: Statistics of the benchmark datasets.

Dataset	# Samples	# Labels	Imbalance Ratio
Massive-I	11,510	59	57.8
Massive-D	11,510	18	8.0
MTOP-I	15,638	102	1616.0
MTOP-D	15,638	11	2.3
Few-Event	18,969	34	33.6
GoEmotions	23,485	27	69.4

A.5 ALGORITHM

Algorithm 1 LSHC

Require: Dataset $\mathcal{D}$, encoder f_θ, initial clustering method, target cluster number K_{target}, maximum iterations $T_{\max}$, representative size k
Ensure: Hierarchical clustering results $\mathcal{H}$

Embed samples $Z \leftarrow f_\theta(\mathcal{D})$
Generate initial partition $\mathcal{P}^{(0)}$
$\mathcal{H} \leftarrow \{\mathcal{P}^{(0)}\}$
$\mathcal{R}_{reject} \leftarrow \emptyset$
$t \leftarrow 0$
while $t < T_{\max}$ **do**
  if $|\mathcal{P}^{(t)}| \leq K_{\text{target}}$ **then**
    break
  end if
  Compute and update cluster centroids
  Rank candidate cluster pairs by centroid similarity
  if top m available pairs have all received `KEEP` **then**
    break
  end if
  Select most similar pair $(C_i, C_j) \notin \mathcal{R}_{reject}$
  Select inside candidates I_i and I_j
  Select boundary candidates $B_{i\to j}$ and $B_{j\to i}$
  $o_{ij} \leftarrow g_{\text{LLM}}(I_i, I_j, B_{i\to j}, B_{j\to i})$
  if $o_{ij} =$ `MERGE` **then**
    $C_{\text{new}} \leftarrow C_i \cup C_j$
    Replace C_i, C_j with C_{new}
    Update centroids similarities
    Empty rejection set in $\mathcal{R}_{reject}$
  else if $o_{ij} =$ `RECLUSTER` **then**
    $U_{ij} \leftarrow C_i \cup C_j$
    $\mathcal{Q}_{ij} \leftarrow \text{Cluster}(U_{ij})$
    Replace C_i, C_j with $\mathcal{Q}_{ij}$
    Update centroids similarities
    Empty rejection set in $\mathcal{R}_{reject}$
  else
    $\mathcal{R}_{reject} \leftarrow \mathcal{R}_{reject} \cup \{(C_i, C_j)\}$
  end if
  Store the current clustering in $\mathcal{H}$
  $t \leftarrow t + 1$
end while
return $\mathcal{H}$