Restaurant Menu Categorization at Scale: LLM-Guided Hybrid Clustering

Seemab Latif[1,2], Ashar Mehmood[2], Selim Turki[3], Huma Ameer[1], Ivan Gorban[3], Faysal Fateh[2], Muhammad Sabih[4],

[1] National University of Sciences and Technology (NUST), Islamabad, Pakistan
[2]Aawaz AI Tech (Pvt) Ltd. Islamabad, Pakistan
[3]Careem, Dubai, UAE, [4]Careem, Lahore, Pakistan

Correspondence: seemab.latif@seecs.edu.pk

Abstract

Inconsistent naming of menu items across merchants presents a major challenge for businesses that rely on large-scale menu item catalogs. It hinders downstream tasks like pricing analysis, menu item deduplication, and recommendations. To address this, we propose the Cross-Platform Semantic Alignment Framework (CPSAF), a hybrid approach that integrates DBSCAN-based clustering with SIGMA (Semantic Item Grouping and Menu Abstraction), a Large Language Model based refinement module. SIGMA employs in-context learning with a large language model to generate generic menu item names and categories. We evaluate our framework on a proprietary dataset comprising over 700,000 unique menu items. Experiments involve tuning DBSCAN parameters and applying SIGMA to refine clusters. The performance is assessed using both structural metrics i.e. cluster count, coverage and semantic metrics i.e. intra and inter-cluster similarity along with manual qualitative inspection. CPSAF improves intra-cluster similarity from 0.88 to 0.98 and reduces singleton clusters by 33%, demonstrating its effectiveness in recovering soft semantic drift.

1 Introduction

In a highly competitive and rapidly evolving ecosystem, restaurants increasingly seek to expand their menus and align offerings with consumer preferences. Traditionally, such decisions are based on manual market surveys and limited feasibility analysis. This mechanism is resource-intensive and lacks scalability. Therefore, data-driven approaches can help restaurant owners to make informed decisions. A persistent challenge in this domain is the absence of standardized naming conventions across menu item listings. For instance, menu items such as "Spicy Chicken Wrap" and "Zinger Wrap" may refer to identical products but are described differently across vendors or platforms. These inconsistencies hinder effective comparative analysis of menu data, limiting the ability to extract actionable insights. Addressing this problem through semantically informed techniques can enable consistency among the products and enable a wide range of applications i.e. data-driven recommendation systems, competitive pricing strategies, and targeted business expansion.

Researchers have employed various clustering algorithms that group the menu items based on lexical or embedding-based similarity (Gumelar et al., 2023), (Messaoudi et al., 2024). Although the methods are effective to some extent, they often fail to represent deeper semantic relationships, leading to fragmented or imprecise clusters. Recent advancements integrate Large Language Models (LLM) to improve clusters via prompting and few-shot learning (Huang and He, 2024), (Zhang et al., 2023), (Viswanathan et al., 2024), (Feng et al., 2024), (Lin et al., 2025). Despite these improvements, current approaches overlook subtle name variations that occur across restaurants. Consequently, they struggle to consistently align semantically equivalent menu items, limiting their utility in downstream analytical and recommendation tasks.

To address these challenges, we propose the Cross-Platform Semantic Alignment Framework (CPSAF), a hybrid pipeline that combines traditional clustering algorithms with LLM- driven refinement. The contributions of this study are as follows:

1. We propose CPSAF, a hybrid clustering framework that combines DBSCAN with LLM-based refinement to align semantically similar menu items across vendors.

2. We develop SIGMA (Semantic Item Grouping and Menu Abstraction), an in-context LLM pipeline that assigns category labels and

Proceedings of the 18th International Natural Language Generation Conference, pages 511–521
October 29–November 2, 2025.

generic names to improve semantic consistency.

3. We design a hierarchical assignment strategy to address soft semantic drift by clustering menu items with subtle lexical variations through multi-stage refinement.

2 Related Work

2.1 Clustering and De-duplication

Standardizing menu items across the restaurants is a challenging task. Gumelar et al. (Gumelar et al., 2023) employs K-means clustering algorithm, uncovering patterns that assist in managing varied items in the menu. Messaoudi et al. (Messaoudi et al., 2024) present a text-based approach for product clustering and recommendation in e-commerce. DBSCAN is used for product clustering, followed by a recommender system based on cosine similarity. Lastly, Latent Semantic Analysis is used for topic extraction from product description. This integration of different techniques forms a recommendation system which recommends similar products as per user preference. These studies demonstrate the potential of clustering algorithms, however, they do not capture semantic variations present in the menu items. Therefore, it leads to the challenge of fragmented clustering. For example, traditional clustering methods such as K-means or DBSCAN may incorrectly split semantically similar items like "Spicy Chicken Wrap" and "Zinger Wrap" into separate clusters because they rely heavily on lexical similarity rather than semantic meaning. Although these items refer to nearly identical menu offerings, minor variations in wording often lead to fragmented clusters, motivating the need for a more robust approach.

2.2 Semantic Labeling and Refinement Using Large Language Models

Researchers are utilizing the LLMs capabilities for the data refinement and enhancement tasks. Ka,ath et al. (Kamath et al., 2024) used LLM to generate key hotel accommodation insights from descriptions and reviews, with human evaluation assessing output quality and identifying hallucinations and areas for improvement. Huang and He (Huang and He, 2024) transform text clustering into a classification task via LLMs. They use prompting techniques to generate potential labels for a dataset and assign the labels to each sample. Similarly, Zhang et al. (Zhang et al., 2023) propose ClusterLLM, a framework that uses LLMs to guide text clustering through triplet-based prompts. Viswanathan et al. (Viswanathan et al., 2024) also explores the potential of LLMs using few-shot learning for clustering scenarios. Their study demonstrates the improvement in clusters quality by integrating LLMs with expert feedback. Researchers are leveraging the LLMs' ability to perform name normalization. Brinkmann et al. (Brinkmann et al., 2024) have utilized LLMs to expand and generalize the menu item's title into a standard form in e-commerce data extraction tasks.

Recent frameworks like k-LLMmeans generate textual cluster centroids via LLMs, improving semantic cohesion compared to basic k-means (Diaz-Rodriguez, 2025). Another study refines clustering by reassigning edge cases through a three-step process (Feng et al., 2024). It begins with K-means clustering, followed by forming superpoints from outliers, which are then clustered using agglomerative methods. Finally, an LLM acts as a semantic oracle to reassign edge cases based on cluster fit. Combining traditional embedding techniques with LLMs, Lin et al. (Lin et al., 2025) proposed Selection and Pooling with LLMs (SPILL). It is a domain-adaptive method that addresses the challenge of domain generalization by using LLMs to refine clusters. In a different domain, LLM-CER (Fu et al., 2025) explores LLM-guided in-context clustering for entity resolution, achieving high accuracy highlighting LLM clustering's potential in structured, high-precision tasks. The KCluster method (Wei et al., 2025) applies an LLM to derive a similarity metric between textual items before clustering and generates descriptive cluster labels, demonstrating a versatile use of LLMs for cluster formation in question banks. Furthermore, Petukhova et al. (Petukhova et al., 2025) empirically demonstrate that embeddings from LLMs outperform traditional embedding-based clustering in purity and silhouette.

Although these methodologies illustrate the potential of LLMs in refining clustering outputs, they do not focus on capturing soft semantic drifts that are subtle differences in the meaning between text menu items. For instance, "Spicy Chicken Wrap" vs. "Zinger Wrap" share a core identity, they differ in preparation style or brand-specific naming. Such variations often coexist, and models may split similar menu items into separate clusters, undermining semantic coherence.

Statistic	Count
Total Brands	4000
Total Merchants	8,000
Total Menu Items	80,000,000
Total Unique Menu Items	700,000

Table 1: Summary of Proprietary Dataset (Approximate Values)

3 Dataset

3.1 Proprietary Dataset

This study uses a proprietary dataset provided by a leading ride-hailing and delivery platform operating in the Middle East and South Asia. The dataset includes information on food brand names, merchants, menu items, menu item descriptions, and unique identifiers for menu items and brands. It contains more than 8 million menu items from more than 4,000 restaurants/merchants, which are preprocessed into around 700,000 distinct menu items. Table 1 summarizes the key statistics of the raw dataset. This rich dataset sets the foundation for proposing the hybrid approach for clustering analysis.

3.2 Data Standardization

Identifying Unique Brands and Merchants: We identify that brands have multiple merchants, and merchants have similar menu items, therefore, we group the menu items together to make unique and distinct menu items.

Filtering Out Inactive Merchants: We ensure that the dataset includes only relevant and active merchants.

Grouping Data by Brand: After cleaning the dataset, we grouped the remaining data by brand.

4 Cross-Platform Semantic Alignment Framework (CPSAF)

This study proposes a novel hybrid pipeline that combines a clustering algorithm and LLM refinement to standardized menu item representations. Its algorithm is given in Algorithm 1. The proposed framework unfolds in two core stages:

4.1 LLM-driven Refinement on Cluster Chunks

We present SIGMA (Semantic Item Grouping and Menu Abstraction), a structured pipeline powered by LLMs with in-context learning. It enriches menu item data by augmenting names and descriptions with category labels and generic names. The process begins with manually curating a vocabulary of category names (denoted as C in Algorithm 1) to ensure consistent and interpretable classification. Menu items are then clustered into semantically coherent chunks using embedding-based similarity. This reduces the complexity of prompt construction and allows the LLM to operate within contextually relevant groups. For each cluster, prompts are constructed using representative examples to enable in-context learning of how menu item descriptions map to predefined categories. Using these prompts, the LLM assigns a category to each menu item, drawing on its language understanding to align the menu items with the curated ontology. Following categorization, the model is prompted again, this time to generate a small set of candidate generic names that abstract the core concept of each menu item, stripping away brand-specific or overly de-

Algorithm 1: CPSAF: Cross-Platform Semantic Alignment Framework

Input: Menu items $M = \{m_1, \ldots, m_n\}$, embedding function E, LLM API, category list $\mathcal{C}$
Output: Refined semantic clusters

Compute embeddings $V = \{E(m) \mid m \in M\}$;
InitialClusters $\leftarrow$ DBSCAN$(V, \epsilon, \text{min_samples})$;
foreach *cluster* C_i *in InitialClusters* **do**
  Construct few-shot prompt from examples in $\mathcal{C}$;
  foreach *item* $m \in C_i$ **do**
    m.category $\leftarrow$ LLM_Assign_Category(m);
    m.generic_name $\leftarrow$ LLM_Generate_Generic_Name(m);
foreach *unclustered item* u **do**
  if $\exists C_j$ *such that cosine_sim*$(u, C_j) \geq \tau$ **then**
    Assign u to C_j;
  else
    Create new cluster using u.category embedding;
return FinalClusters;

tailed language. Finally, the LLM selects or refines the most appropriate generic name using guided prompting. The result is a semantically enriched dataset in which each menu item is described by a standardized category and a concise generalized name.

4.2 Hierarchical Assignment of Clusters

The menu item clusters are generated using DBSCAN in a three-stage refinement process on generic names and categories Figure 1. In Stage I, DBSCAN clusters the menu items based on their concatenated category and menu item names, forming initial semantically coherent groups. In Stage II, unclustered menu items are matched to existing clusters using a 90% threshold on embedding-based semantic similarity of generic name and menu item name. In Stage III, any remaining menu items are clustered using semantic embeddings of their categories. This hierarchical approach ensures similar menu items (e.g., "Spicy Chicken Sandwich" and "Hot Chicken Burger") are grouped together.

5 Experimental Design

Our evaluation framework measures cluster quality based on structural and semantic properties, as well as computation time. We first analyze baseline unsupervised models K-Means, Agglomerative, Hierarchical DBSCAN (HDBSCAN), and DBSCAN focusing on runtime and metrics in Table 3. We then enhance the clustering results of a high-performing, low-compute model using our proposed SIGMA approach. The final evaluation combines quantitative metrics with manual qualitative validation for a robust assessment of cluster performance.

5.1 Clustering Baseline Experiments

We perform clustering experiments using traditional unsupervised methods. First, we apply K-Means clustering on our 700,000 menu items embedding dataset, experimenting with various values of k determined by the elbow method. However, K-Means assigns identical or highly similar menu items to different clusters, due to its reliance on global distance minimization without semantic awareness (Petukhova et al., 2025). Then we experiment with Agglomerative Clustering and HDBSCAN. Agglomerative clustering shows poor scalability as its bottom-up approach requires computing and updating a full pairwise distance matrix, which becomes computationally intensive and memory-heavy at scale. This results in substantial slowdowns and makes it impractical for datasets of our size.

Algorithm	Time (sec)	Time (min)
K-Means	1,521	25.4
Agglomerative	65,006	18.1 hours
HDBSCAN	394,000	109.4 hours
DBSCAN	**192**	**3.2**

Table 2: Average Computation Time of Clustering Algorithms on 0.7 Mil Menu Items. Computation time is averaged over 5 runs.

HDBSCAN, while appealing for its ability to handle variable-density clusters and noise, also under performs due to its dependency on complex graph-based operations. It constructs a minimum spanning tree and computes cluster hierarchies, which introduces significant overhead at this data scale. Moreover, the model tends to fragment semantically similar menu items across multiple small clusters, reducing its effectiveness for our use case where preserving semantic cohesion within clusters is essential. The computation time for the four clustering models is given in Table 2.

Finally, we apply DBSCAN, experimenting with various values of ϵ to identify clusters based on the local density in the embedding space. While DBSCAN effectively detects dense groups of similar menu items, it also results in significant fragmentation. In our experiments, it generates more than 65,000 clusters containing more than one menu item, and when including singleton clusters, the total exceeds 200,000. This high degree of fragmentation reduces the semantic coherence of the clusters and limits their practical utility. A key challenge is the variation in naming conventions across brands and different descriptions for semantically similar menu items often result in these menu items being placed in separate clusters. This outcome highlights the limitations of relying solely on raw embedding distances and underscores the need for a refinement process that can capture fine-grained semantic similarities more effectively.

5.2 Refinement through the SIGMA Approach

To overcome the challenges posed by purely unsupervised clustering, we propose SIGMA, a hybrid

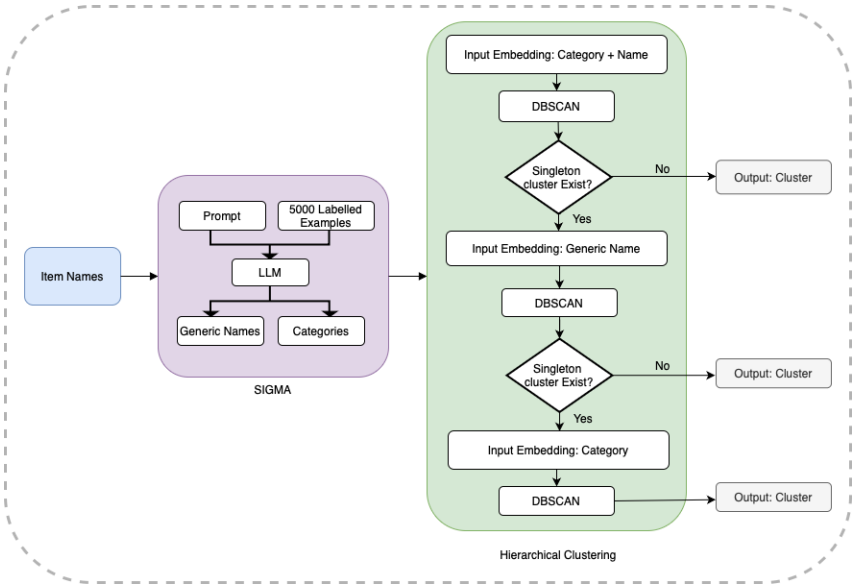


Figure 1: Hierarchical Assignment of Clusters using DBSCAN

refinement framework leveraging LLMs. After initial clustering with DBSCAN, SIGMA enhances cluster quality through LLM-driven semantic abstraction. We test the SIGMA approach with two LLMs, *Gemini 1.5 Flash*[1] and *DeepSeek R1*[2] 7B models for semantic name generation and refinement. For our usecase Google Gemini 1.5 Flash performed better.

5.3 Evaluation Metrics

We assess clustering quality and refinement effectiveness using four metrics: intra-cluster similarity, inter-cluster similarity, number of clusters, and cluster coverage. Intra-cluster similarity measures the average cosine similarity among menu items within the same cluster, while inter-cluster similarity captures the cosine similarity between cluster centroids. The number of clusters reflects total clusters formed, excluding noise (i.e., single-item clusters), and cluster coverage indicates the percentage of menu items assigned to any cluster. We categorize these into primary quality metrics and secondary structural metrics, as detailed in Table 3. Primary quality metrics assess semantic cohesion and separation of clusters, while secondary structural metrics provide insight into clustering density and data coverage. Manual qualitative verification of the sample clusters is also carried out to ensure semantic coherence.

Table 4 presents the evaluation results of various unsupervised clustering methods. Among them, DBSCAN produced the most cohesive clusters, exhibiting high intra-cluster similarity. In contrast, K-Means and Agglomerative Clustering showed better inter-cluster separation but lower internal cohesion. Unlike K-Means and Agglomerative, which force full data partitioning, DBSCAN and HDBSCAN can identify and exclude noise, making them more resilient to outliers and irregular data structures. On the basis of these findings, DBSCAN is selected for further experimentation because of its balance of cohesion and robustness. HDBSCAN is excluded due to high computational cost and lower cohesion, while K-Means and Agglomerative Clustering are discarded for producing semantically inconsistent clusters.

Primary Quality Metrics	Secondary Structural Metrics
Intra-cluster Similarity	Number of Clusters
Inter-cluster Similarity	Cluster Coverage

Table 3: Evaluation Metrics Categorization

5.4 Experimental Setup

We perform clustering experiments on a proprietary dataset containing 700,000 unique menu items. Each menu item is represented through text embeddings generated from its name using Google Gemini "text-embedding-004". The hyperparam-

[1]https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/1-5-flash

[2]https://api-docs.deepseek.com/news/news250120

	Primary Quality Metrics		Secondary Structural Metrics	
Algorithm	**Intra-cluster Similarity**	**Inter-cluster Similarity**	**Number of Clusters**	**Cluster Coverage**
K-Means	0.7133	0.0037	32,000	100%
Agglomerative	0.6990	**0.0034**	32,000	100%
HDBSCAN	0.9211	0.0040	70,000*, 200,000@	49%
DBSCAN	**0.9995**	0.0117	65,000 *, 200,000@	32%

Table 4: Evaluation Results of Four Unsupervised Clustering Methods on 700,000 Menu Items Embeddings. Note: These values are an average of 5 runs and rounded off. *without single menu item clusters, @with single menu item clusters

eters for the final DBSCAN clustering are; ϵ = 0.15, min samples: 1. These parameter values are selected through empirical testing, balancing the trade-off between cluster granularity and semantic coherence measured via intra-cluster similarity. All experiments are conducted on a system running Ubuntu 20.04.6, Intel i7-8700 processor with 12 threads running at 4.6 GHz, an NVIDIA GeForce RTX 2080 Ti and 64 GB of RAM.

6 Results and Analysis

6.1 Preliminary DBSCAN Evaluation and Parameter Tuning

We extensively evaluated DBSCAN to determine optimal parameters and establish baseline performance. Using 25k and 82k subsets, we analyzed the behavior of DBSCAN's with varying ϵ values. Lower ϵ produced finer but sparse clusters, while higher ϵ improved coverage at the cost of semantic precision (Table A .7). For the 82k set, $\epsilon = 0.15$ offered the best balance, yielding 8,962 clusters and clustering 46,693 menu items. Final evaluation on the full dataset with name-only embeddings confirmed $\epsilon = 0.15$ as optimal, producing 61,538 clusters, clustering 466,131 menu items, and achieving the highest intra-cluster similarity of 0.88 (Table A .8).

6.2 Clustering Performance with CPSAF

Table 5 summarizes key metrics and compares baseline DBSCAN with our proposed CPSAF framework. Initial DBSCAN clustering forms more than 62,000 multi-item clusters and achieves 77% coverage of more than 700,000 unique menu items. However, it leads to excessive fragmentation, with 33% singleton clusters and more than 88% clusters that contain fewer than 10 menu items. In contrast, the hybrid SIGMA refinement reduced the number of clusters to approximately 11,000 and achieved 100% coverage, ensuring that all menu items are semantically grouped.

The proposed methodology outperforms DBSCAN in all evaluation metrics. Intra-cluster similarity improves from 0.88 to 0.98, showing enhanced semantic coherence. While inter-cluster similarity decreases from 0.79 to 0.77, indicating better separation. These improvements stem from CPSAF's ability to normalize textual inconsistencies through SIGMA and assign semantically similar menu items in three stages through generic naming and category embeddings.

6.3 Analysis

The preliminary evaluations of the subsets and the complete dataset determine the optimal ϵ value and the embedding configuration. As shown in Table A .7 and Table A .8, lower ϵ values produced finer-grained clusters with higher intra-cluster similarity, but suffered from reduced coverage. In contrast, higher values improve coverage at the cost of semantic precision. However, even with this configuration, DBSCAN struggles with fragmentation, producing over 62,000 clusters and leaving nearly 23% of menu items ungrouped.

The CPSAF framework, which extends DBSCAN with SIGMA-based refinement, significantly improves these limitations. The hybrid approach retains larger clusters while improving coherence across small and mid-size clusters, providing a more balanced and semantically aligned distribution. The transition from baseline DBSCAN to the hybrid CPSAF approach results in clusters that are not only structurally compact but also semanti-

Metric	DBSCAN	DBSCAN + SIGMA
Total Unique Items	601,259	601,259
Cluster Coverage (%)	77%	**100%** ↑
Number of Clusters	62,135	**10,834** ↓
Singleton Clusters (%)	33%	**0%** ↓
Clusters $<$ 10 Items	55,741	**5,187** ↓
Clusters 10–50 Items	5,135	4,508
Clusters 50–100 Items	731	661
Clusters $>$ 100 Items	528	478
Intra-cluster Similarity	0.88	**0.98** ↑
Inter-cluster Similarity	0.79	**0.77** ↓

Table 5: Clustering Metrics Comparison: DBSCAN vs. Hybrid (DBSCAN+ SIGMA)

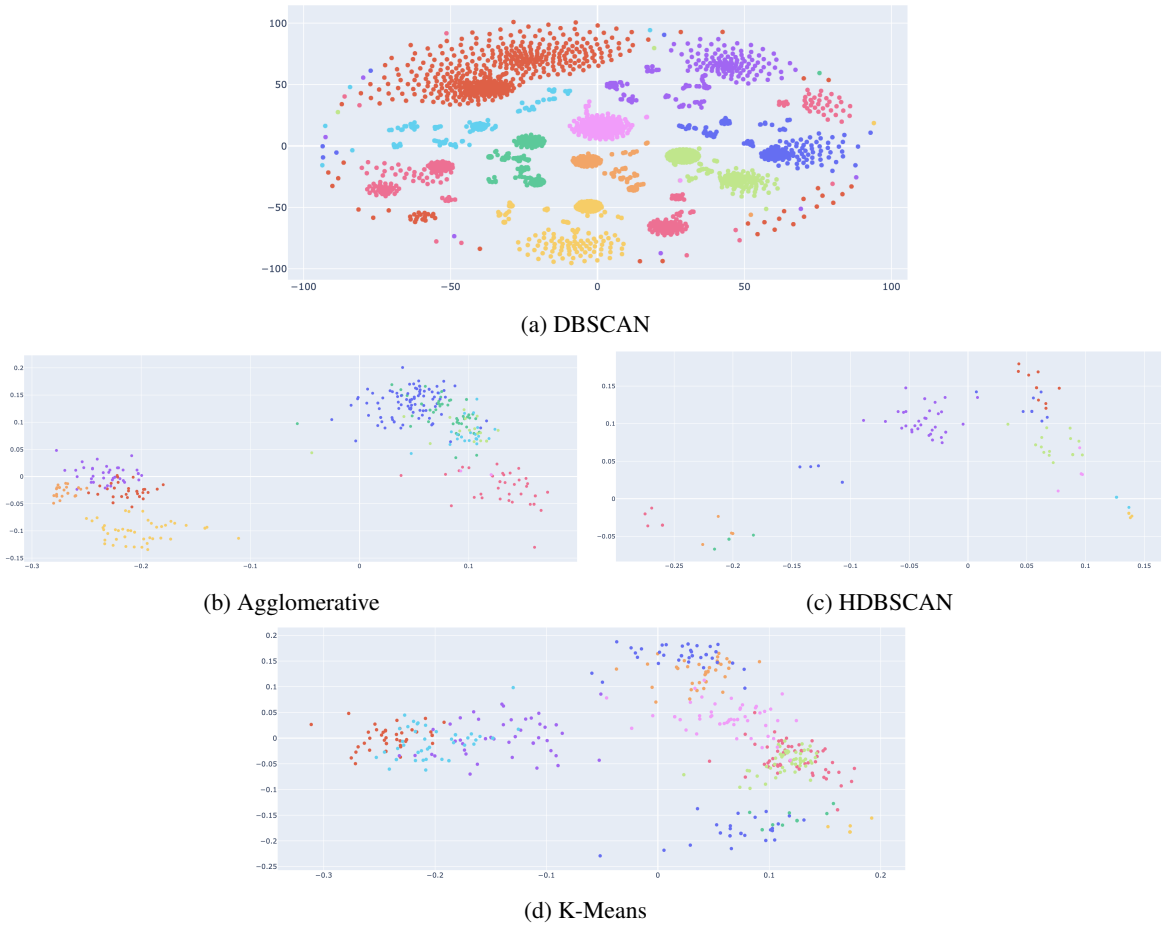
cally consistent. We further validate this configuration on a diverse set of long, keyword-overlapping dish names. Qualitative inspection confirms that items with partial lexical overlap but differing core semantics—for instance, *"Almond Butter"* vs. *"Almond Milk Chia Seed Pudding"*, *"Avocado Mix with Ashta"* vs. *"Avocado Salmon Salad"*, and *"Fried Chicken"* vs. *"Fried Chicken Roll"* are consistently placed in distinct clusters unless embedding similarity indicated strong contextual alignment. This illustrates the framework's ability to separate similar yet semantically divergent items, while still grouping related variants when contextual signals are strong. This makes the framework highly applicable for downstream tasks given in the Appendix B.

6.4 Clustering Results Visualizations and Analysis

To assess the effectiveness of clustering in semantically aligning menu items, we visualized the outputs of DBSCAN, Agglomerative Clustering, HDBSCAN, and K-Means using t-SNE projections of menu item embeddings (Figure 2). DBSCAN produced the most semantically coherent clusters, capturing irregular shapes and effectively handling noise. Furthermore, more than 30% of all LLM-generated categories are manually checked for correctness and semantic appropriateness. In addition, approximately 80% of all clusters are manually reviewed to ensure that the items grouped together represent the same or closely related menu offerings. Agglomerative clustering formed balanced but overlapping clusters, often merging distinct concepts due to hierarchical linkage. HDBSCAN improved robustness but tended to over-prune, discarding many valid menu items as noise. K-Means created clean, spherical clusters but suffered from semantic fragmentation. These results highlight the need for a flexible clustering approach. The expressiveness of DBSCAN in modeling dense yet irregular semantic spaces justifies its selection as the foundational step in our CPSAF framework.

7 Ablation Study

To assess the contribution of individual components within the CPSAF framework, we conducted an ablation study by selectively removing or modifying key modules and measuring their impact on clustering performance. As shown in Table 6, removing the SIGMA refinement module significantly reduced intra-cluster similarity (from 0.98 to 0.88) and cluster coverage (from 100% to 77%), indicating the central role of LLM-based semantic enrichment. Excluding the generic name generation step led to moderate degradation (intra-cluster similarity of 0.92), underscoring its contribution to semantic coherence. Finally, omitting Stage III (final reassignment) of the hierarchical clustering pipeline resulted in a less pronounced but measurable decline in both cohesion and coverage. These results validate the effectiveness of each component in improving the semantic quality and structural compactness of clusters.



(a) DBSCAN

(b) Agglomerative

(c) HDBSCAN

(d) K-Means

Figure 2: t-SNE projections of clustered menu embeddings using four algorithms. DBSCAN exhibits irregular but semantically coherent clusters; K-Means and Agglomerative form compact, often globular groupings; HDBSCAN selectively ignores outliers for higher precision.

Component Removed	Intra-cluster Similarity	% Coverage
Full CPSAF (No components removed)	**0.98**	**100%**
SIGMA (LLM refinement removed)	0.88	77%
Generic Names	0.92	85%
Stage III Final reassignment	0.94	90%

Table 6: Ablation Study Results for CPSAF Components, Averaged over 5 runs.

8 Conclusion

This paper addresses the challenge of identifying inconsistencies in menu item data and transforming them into actionable insights for business growth on food delivery platforms. We propose CPSAF, a Cross-Platform Semantic Alignment Framework, which integrates DBSCAN-based clustering with LLM-powered refinement (SIGMA) to enhance cluster accuracy, cohesion, and coverage. The framework incorporates a three-stage hierarchical assignment strategy to resolve textual inconsistencies between brands and merchants. We conducted extensive experiments across four clustering algorithms, varying dataset sizes and configurations to determine optimal DBSCAN parameters. It is followed by full-scale evaluations of 700,000 unique menu items. Our hybrid approach achieves 100% cluster coverage, intra-cluster similarity of 0.98, and significantly lowers fragmentation compared to baseline DBSCAN. The analysis confirms that CPSAF produces structurally compact and semantically coherent clusters. These refined clusters

enable downstream applications such as pricing analysis, brand recommendations, and merchant-level targeting. In future work, our goal is to extend our system to support advanced features that benefit both customers and merchants. On the customer side, we plan to incorporate context-aware notifications, hyper-personalized recommendations, and natural language chatbot integrations to enhance user engagement and satisfaction. For merchants, our future efforts will focus on leveraging data-driven insights for menu optimization, targeted marketing, and strategic business intelligence, including geo-specific trends and competitive benchmarking.

Code Availability

The implementation of CPSAF and SIGMA used in this paper is available at GitHub Repository.

Limitations

Although the proposed CPSAF framework improves the quality of clustering using LLM-based refinement, its effectiveness is limited when ground truth category labels are unavailable, making the evaluation dependent on internal metrics and manual inspection. Furthermore, the proprietary dataset used for training and testing restricts external reproducibility, and no experiments are provided on public benchmarks.

Ethical Consideration

We confirm that the menu item data used in our experiments originates from a proprietary, anonymized dataset provided by a commercial industry platform and does not contain any personally identifiable information. In addition, all sensitive content was excluded during preprocessing. Based on the above, there are no ethical considerations in this paper.

References

Alexander Brinkmann, Nick Baumann, and Christian Bizer. 2024. Using llms for the extraction and normalization of product attribute values. In *Advances in Databases and Information Systems: 28th European Conference, ADBIS 2024, Bayonne, France, August 28–31, 2024, Proceedings*, page 217–230, Berlin, Heidelberg. Springer-Verlag.

Jairo Diaz-Rodriguez. 2025. k-llmmeans: Scalable, stable, and interpretable text clustering via llm-based centroids. *Preprint*, arXiv:2502.09667.

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.

Jiajie Fu, Haitong Tang, Arijit Khan, Sharad Mehrotra, Xiangyu Ke, and Yunjun Gao. 2025. In-context clustering-based entity resolution with large language models: A design space exploration. *Preprint*, arXiv:2506.02509.

Lasmi Lasmini Gumelar, Ruuhwan Ruuhwan, and Missi Hikmatyar. 2023. Unveiling culinary patterns: Implementation of k-means clustering algorithm on food products in cafes. *INNOVATICS: Innovation in Research of Informatics*, 5(2).

Chen Huang and Guoxiu He. 2024. Text clustering as classification with llms. *arXiv preprint arXiv:2410.00927*.

Srinivas Ramesh Kamath, Fahime Same, and Saad Mahamood. 2024. Generating hotel highlights from unstructured text using LLMs. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 280–288, Tokyo, Japan. Association for Computational Linguistics.

I-Fan Lin, Faegheh Hasibi, and Suzan Verberne. 2025. Spill: Domain-adaptive intent clustering based on selection and pooling with large language models. *arXiv preprint arXiv:2503.15351*.

Fayçal Messaoudi, Manal Loukili, and Raouya El Youbi. 2024. A text-based approach for product clustering and recommendation in e-commerce using machine learning. In *Internet of Things and Big Data Analytics for a Green Environment*, pages 124–137. Chapman and Hall/CRC.

Alina Petukhova, Joao P. Matos-Carvalho, and Nuno Fachada. 2025. Text clustering with large language model embeddings. *International Journal of Cognitive Computing in Engineering*, 6:100–108.

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.

Yumou Wei, Paulo Carvalho, and John Stamper. 2025. Kcluster: An llm-based clustering approach to knowledge component discovery. *Preprint*, arXiv:2505.06469.

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

A Finding the ϵ Values for DBSCAN

We conducted a comprehensive evaluation of DBSCAN to identify optimal hyperparameters and establish baseline performance. By experimenting on 25k and 82k subsets, we analyzed the algorithm's sensitivity to different values of ϵ. Smaller ϵ values produced more granular but sparser clusters, whereas larger ϵ values increased coverage but compromised semantic precision 7.

B Operational Extensions of CPSAF

The coherent clusters generated by the Cross Platform Semantic Alignment Framework (CPSAF) serve as a strong foundation for a range of downstream applications. This work extends into two key areas: Comparative Pricing Analysis and Brand Recommendation. These applications offer practical insights for businesses especially restaurants and food service platforms looking to refine their menus, adjust pricing strategies, or explore alternative branding options. By utilizing CPSAF's structured outputs, businesses can make more informed, data-driven decisions in competitive market environments.

B.1 Comparative Pricing Analysis

CPSAF's structured data analysis capabilities can be extended to comparative pricing analysis, enabling the aggregation and comparison of pricing data for similar products across sources or time periods. When applied to such datasets, CPSAF can uncover pricing patterns, detect variations, and highlight anomalies. For instance, it can analyze how different retailers price menu items within the same product line, offering insights into competitive strategies and pricing behavior. This application demonstrates CPSAF's ability to work with cross-sectional data and generate actionable pricing metrics. It can incorporate diverse data inputs such as competitor price listings, historical sales records, and promotional discounts to assess relative pricing positions. Analysts can use CPSAF to explore metrics like average price gaps, the frequency of price changes, or the identification of outlier pricing. This use case illustrates how CPSAF can be adapted to explore market pricing dynamics, showcasing its flexibility beyond its original scope.

B.2 Brand Recommendation Framework

CPSAF can also be extended to brand recommendation scenarios, where it analyze consumer preferences, product attributes, and purchase histories to suggest alternative brands that fulfill similar needs. Its analytical capabilities support the evaluation of brand similarities by comparing product features, quality ratings, pricing tiers, and user feedback. For example, CPSAF can cluster products based on shared brand characteristics and identify comparable brands within those groups. This approach enables the generation of brand suggestions such as recommending alternatives when a preferred brand is unavailable demonstrating the framework's ability to handle complex relational data. This scenario highlights CPSAF's flexibility in addressing broader recommendation tasks beyond its core functionality, reinforcing its potential for diverse, data-driven applications.

Test	Dataset	ϵ	Number of Clusters	Number of Items Clustered
1	25k	0.20	1,011	8,523
2	25k	0.25	2,713	10,106
3	25k	0.30	2,790	14,013
4	25k	0.32	2,621	15,649
5	82k	0.13	8,878	43,799
6	**82k**	**0.15**	**8,962**	**46,693**
7	82k	0.17	8,777	50,350
8	82k	0.20	9,184	27,815
9	82k	0.25	9,044	41,060
10	82k	0.30	7,955	54,113
11	82k	0.32	7,281	59,279

Table 7: DBSCAN Subset Evaluation across ϵ Values and Dataset Sizes, Averaged over 5 runs.

ε	Clusters	Items	Intra-cluster Similarity	Inter-cluster Similarity
0.17	56,060	482,724	0.85	0.45
0.16	58,651	470,001	0.86	0.43
0.15	**61,538**	**466,131**	**0.88**	**0.40**

Table 8: DBSCAN Full Dataset Evaluation, Averaged over 5 runs.