Clustering Analysis of Food Recipe Ingredients: Dimensionality Reduction and Comparative Study

Thomas AUBOURG, Edgar DEMEUDE, Anh-Duy VU

October 30, 2025

Abstract

This research explores the hidden architecture of cooking by analyzing ingredient relationships across a massive database of 222,705 recipes. We first addressed data complexity by grouping thousands of similar ingredient names using text embeddings, then applied Truncated Singular Value Decomposition (SVD) to project high-dimensional ingredient co-occurrence patterns onto a compact 50-component latent space. A comparative study of four clustering algorithms revealed distinct performance profiles: HDBSCAN excelled in the raw co-occurrence space, identifying precise, coherent ingredient modules while robustly filtering noise. However, after applying TF-IDF weighting to emphasize distinctive ingredients over universal staples, DBSCAN emerged as superior, producing the most interpretable and culinarily relevant clusters. Further refinement using HDBSCAN with precomputed cosine distances and leaf cluster selection ultimately achieved the best balance of thematic coherence and culinary precision in the TF-IDF-weighted space, establishing it as the optimal configuration for advanced culinary data mining despite higher computational costs.

Contents

1 Introduction and Project Objective

This project aims to identify meaningful culinary ingredient groupings within a large-scale recipe dataset through comparative analysis of dimensionality reduction and clustering techniques. The goal is to move beyond simple co-occurrence counts to uncover the latent structure and functional ingredient modules that reflect real-world cooking patterns.

1.0.1 Project Repository

The complete source code, exploratory data analysis notebooks, and detailed methodology are publicly available on GitHub:

- GitHub Repository: https://github.com/Thomas-aub/Food_Mining

1.1 Dataset Description

The analysis utilizes two public datasets from Kaggle:

- Food.com Recipes and User Interactions
- Food.com Recipes with Search Terms and Tags

The core data structure is a high-dimensional, sparse ingredient co-occurrence matrix with $N_{\text{recipes}} = 222,705$ recipes and $N_{\text{ingredients}} = 13,360$ unique ingredients after initial cleaning. This represents an extremely high-dimensional feature space where each recipe is encoded by the presence or absence of thousands of possible ingredients, presenting significant challenges for clustering due to sparsity and the curse of dimensionality.

2 Methodology: Data Preprocessing

2.1 Ingredient Canonicalization: From Redundancy to Semantic Richness

Initial data cleaning involved merging raw datasets and standardizing entries. A critical preprocessing step was **semantic ingredient canonicalization**, essential for merging variations of the same ingredient and **reducing feature space redundancy**. This approach, which leverages **semantic proximity**, was crucial for effectively grouping terms such as "mozzarella cheese" and "fresh mozzarella" under a single canonical label.

The dataset initially contained **222,705 recipes** and **13,360 unique ingredient names** after the initial lexical cleaning.

2.1.1 Detailed Canonicalization Process

Table 1: Detailed Steps of Ingredient Canonicalization

Step	Enhanced Description	Role and Justification
Text Preprocessing	Ingredient names were normalized (lowercasing, accent removal), cleaned (noise, quantities), **lemmatized** using `WordNetLemmatizer`, and regional synonyms (ex: 'cilantro' $\rightarrow$ 'coriander') were substituted.	Addressed **basic morphological and spelling variations** to ensure minor textual differences did not skew the embedding process.
Embedding Generation	Text embeddings were created for each unique ingredient using the pre-trained SentenceTransformer model `all-MiniLM-L6-v2`, selected for its **computational efficiency and robustness** in capturing short-phrase semantic similarity.	Crucial for **moving beyond string matching** and incorporating the contextual meaning of ingredients.
Similarity Computation	A **cosine similarity matrix** was computed across all ingredient embeddings.	Cosine similarity is the standard measure for assessing the **orientational closeness** of semantic vectors.
Hierarchical Clustering	**Agglomerative Clustering** with **average linkage** was applied. A **distance threshold of 0.20** (equivalent to similarity ≥ 0.80) was empirically selected for strict, relevant grouping. The **canonical label** was the ingredient with the shortest name for improved readability.	This non-parametric method allowed iterative grouping based on semantic proximity, with precise threshold control.

2.1.2 Impact and Validation of Results

To illustrate the effectiveness of the semantic clustering, Table 2 provides examples of grouped ingredients and their canonical labels:

Table 2: Examples of Ingredient Canonicalization Clusters

Canonical Label	Grouped Ingredients
paper cup	paper baking cup, paper cup
garlic paste	garlic paste, garlic and red chile paste
beef stock	beef stock mix, beef stock, homemade beef stock, white beef stock
distilled vinegar	distilled white vinegar, white distilled vinegar, distilled vinegar, hot spiced white vinegar
stuffing mix	chicken flavor stuffing mix, chicken stuffing mix, chicken stove top stuffing mix, stove top lowsodium chicken stuffing mix, stove top stuffing mix, reducedsodium chicken flavor stuffing mix

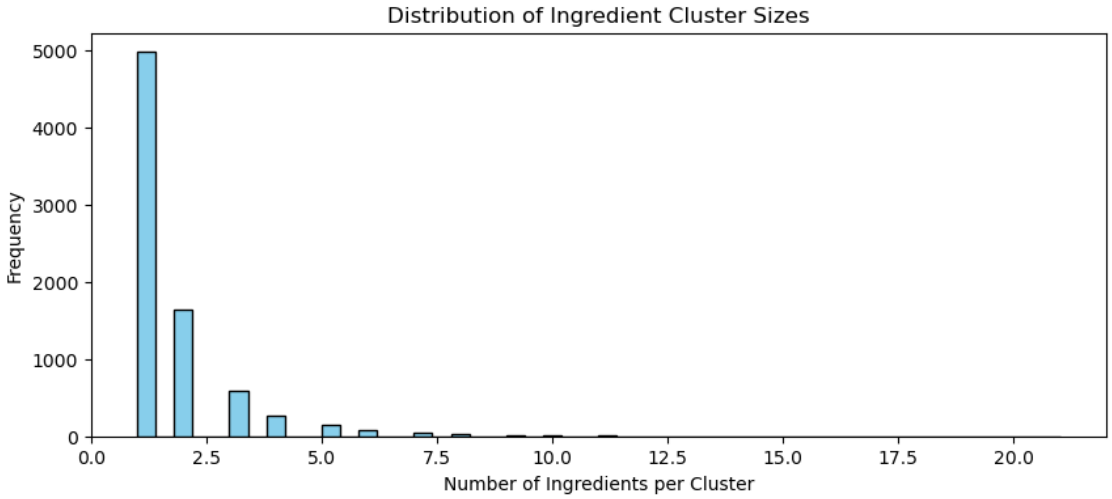

Figure 1: Distribution of ingredient cluster sizes following hierarchical clustering for canonicalization, showing successful merging of semantically close variants while preserving unique ingredients.

The distribution graph (**Figure 1**) reveals a strong asymmetry. The majority of clusters (nearly 5,000) are of size 1. This result is a **key indicator of the success of the selective approach**: it confirms the process succeeded in **merging only the semantically very close variants**, thus efficiently reducing redundancy while preserving the distinction and richness of most unique ingredients for the main analysis.

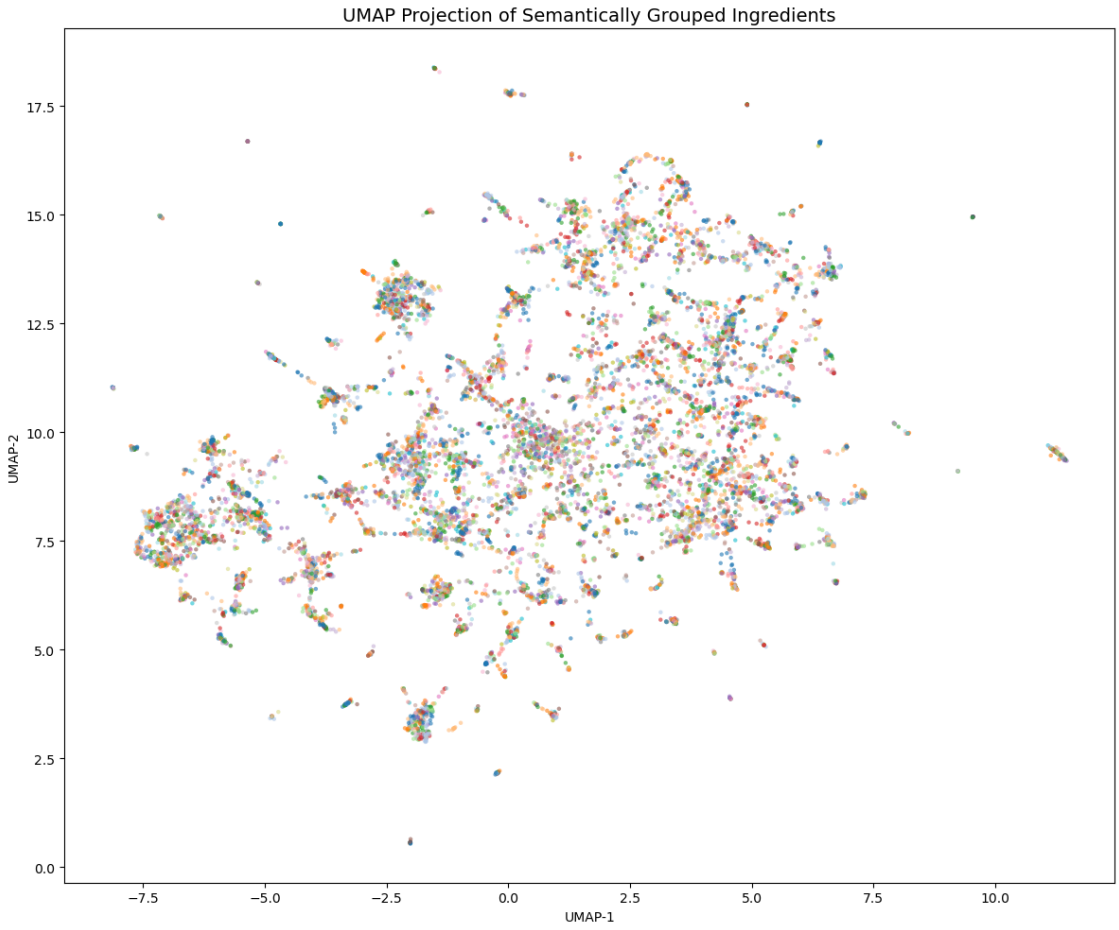

Figure 2: UMAP visualization of ingredient clusters derived from SentenceTransformer embeddings, confirming semantic proximity.

The **UMAP visualization (Figure 2)** confirms the process's effectiveness. Multiple clusters are tightly grouped, demonstrating that the hierarchical clustering successfully **captured and organized the complex semantics** of the ingredients.

Quantitative Result: The process resulted in a **reduction of the ingredient vocabulary from 13,360 unique ingredients to 7,832 canonical labels**, representing a significant 41.4% **reduction**. This decrease minimizes the **dimensionality** of the dataset, improving the robustness and interpretability of subsequent models. The cleaned output was saved as `recipes_cleaned.csv` for all subsequent analyses.

3 Dimensionality Reduction

3.1 Method Selection Rationale

We selected linear methods (PCA and Truncated SVD) over non-linear techniques (t-SNE, UMAP) based on several critical considerations related to our data characteristics and analysis objectives. Our ingredient matrix is extremely high-dimensional and sparse, making linear approaches computationally efficient and capable of handling this large, sparse structure while providing interpretable components that capture the main axes of ingredient variation.

Linear techniques like PCA and Truncated SVD preserve global relationships and ingredient co-occurrence patterns essential for clustering broad culinary themes and staple ingredient groups. In contrast, t-SNE and UMAP are optimized for creating visually intuitive, low-dimensional (usually two- or three-dimensional) representations that preserve local structure for visualization purposes. While these non-linear methods can, in principle, capture more global relationships when using high perplexity (t-SNE) or large neighborhood sizes (UMAP), such configurations greatly increase computational and memory requirements. Given the scale of our dataset with tens of thousands of features, such configurations would be prohibitively expensive to compute.

Additionally, the output dimensions of t-SNE and UMAP are not directly interpretable for ingredient analysis, limiting their usefulness for downstream clustering beyond visualization. Since our main objective was dimensionality reduction for clustering rather than visualization, we prioritized methods that preserve broader structure, provide interpretable components, and remain computationally tractable.

3.2 Truncated SVD Implementation

Truncated Singular Value Decomposition was selected as the primary dimensionality reduction method due to its direct handling of sparse matrices. Unlike standard PCA, Truncated SVD does not require data centering, making it particularly suitable for our sparse ingredient co-occurrence data.

The implementation details were as follows:

- **Input Dimension:** 7,832 features (ingredients)
- **Target Dimension:** $N_{\text{COMPONENTS}} = 50$, resulting in a transformed matrix $\mathbf{X}_{\text{reduced}}$ with shape $7,832 \times 50$
- **Explained Variance:** The 50 components captured 37.0% of the total variability in the ingredient co-occurrence space, providing a strong trade-off between compression and information preservation
- **Post-processing:** L2-normalization was applied to the reduced feature matrix to ensure that the magnitude of ingredient frequency did not dominate similarity computation in the clustering phase, allowing algorithms to focus on co-occurrence patterns rather than absolute counts

The technique was implemented using the scikit-learn library, which allows efficient computation directly on sparse matrices. This level of compression provided sufficient information preservation while dramatically reducing computational complexity for subsequent clustering analyses.

3.3 PCA as Validation Benchmark

For comparison and validation purposes, Principal Component Analysis (PCA) was also applied on the same dataset. PCA was implemented through the scikit-learn framework, which standardizes the data and computes principal components via an underlying singular value decomposition. Similar to Truncated SVD, the dimensionality was reduced from 7,832 to 50 components.

The PCA-transformed data were also L2-normalized before clustering, ensuring consistency across dimensionality reduction techniques. While PCA requires the data to be dense (involving an implicit conversion step), it served as a reliable benchmark for evaluating whether the Truncated SVD results accurately represented the main variance structure of the dataset. After validation, Truncated SVD was retained for all subsequent analyses due to superior scalability with sparse data.

4 Clustering Analysis: Pre-TF-IDF Results

4.1 Clustering Methods

Four distinct clustering algorithms were applied to the 50-dimensional SVD-reduced space to identify functional ingredient modules:

- **MiniBatchKMeans:** A partitioning baseline method that optimizes cluster assignments by minimizing within-cluster variance based on squared Euclidean distances from cluster centroids. This approach tends to find spherical and roughly equally sized clusters and does not explicitly model noise or outliers. Tested across cluster counts k from 20 to 100.
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** A density-based method capable of identifying clusters of arbitrary shape and distinguishing noise points without requiring the number of clusters to be specified beforehand. Key parameters include epsilon (ε), which sets the radius for neighborhood density estimation, and `min_samples`, defining the minimum number of points to form a dense region. Experiments explored ε values in the range 0.5 to 0.7 with fixed `min_samples=10`.
- **Hierarchical (Agglomerative) Clustering:** Produces a dendrogram illustrating progressive merging of ingredient clusters from fine to coarse granularity. This method does not require pre-specification of cluster numbers and reveals nested, multi-scale structures. Different linkage criteria (ward, complete, average, single) were explored for $k = 20$–40 to understand the effect of cluster proximity definitions.
- **HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise):** An advanced density-based method that combines hierarchical approaches with density estimation. Unlike DBSCAN, which requires a global density parameter and struggles with clusters of differing densities, HDBSCAN automatically adjusts to find clusters at multiple density levels. The algorithm calculates mutual reachability distances incorporating local density, builds a minimum spanning tree, constructs a cluster hierarchy, and condenses it based on cluster stability metrics. Tested with minimum cluster sizes (`min_cluster_size`) from 5 to 25.

4.2 Parameter Sweep Results

A comprehensive parameter search was performed for each clustering method to identify optimal configurations on the reduced ingredient space (7,832 ingredients $\times$ 50 dimensions via Truncated SVD, explained variance 46%).

For K-Means, cluster counts ranging from 20 to 100 were tested, with the highest silhouette scores near 0.071 at $k = 30$–100 and stable Davies-Bouldin indices (2.57–3.15). DBSCAN was evaluated over multiple epsilon values, with the best silhouette score (0.26) at $\varepsilon = 0.5$ but with high detected noise (86.7%) and moderate cluster compactness. HDBSCAN, varied by minimum cluster size (5 to 25), systematically achieved the top silhouette scores (around 0.32–0.33) and tightest Davies-Bouldin indices ($\sim$1.13–1.23), but at the cost of labeling over 75% of data as noise, especially with smaller minimum cluster sizes. Hierarchical agglomerative clustering, using ward, average, complete, and single linkage for $k = 20$–40, produced clusters efficiently but with lower silhouette scores (mostly < 0.04) and less compact groupings.

The comparison revealed that HDBSCAN with smaller minimum cluster sizes (`mcs=15` or 20) yielded the best cluster compactness (silhouette $\approx$ 0.32–0.33), lowest Davies-Bouldin indices (< 1.2), and detected the highest proportion of data as noise (up to 85%), revealing fine-grained, well-separated ingredient clusters but excluding most outliers. DBSCAN with $\varepsilon = 0.5$ similarly identified dense ingredient groupings (silhouette 0.26), though at slightly reduced quality and with substantial noise (86%). K-Means, across cluster counts $k = 30$–100, consistently partitioned all recipes with moderate silhouette ($\sim$0.07) and compactness, suitable for broader culinary categories rather than detailed definitions.

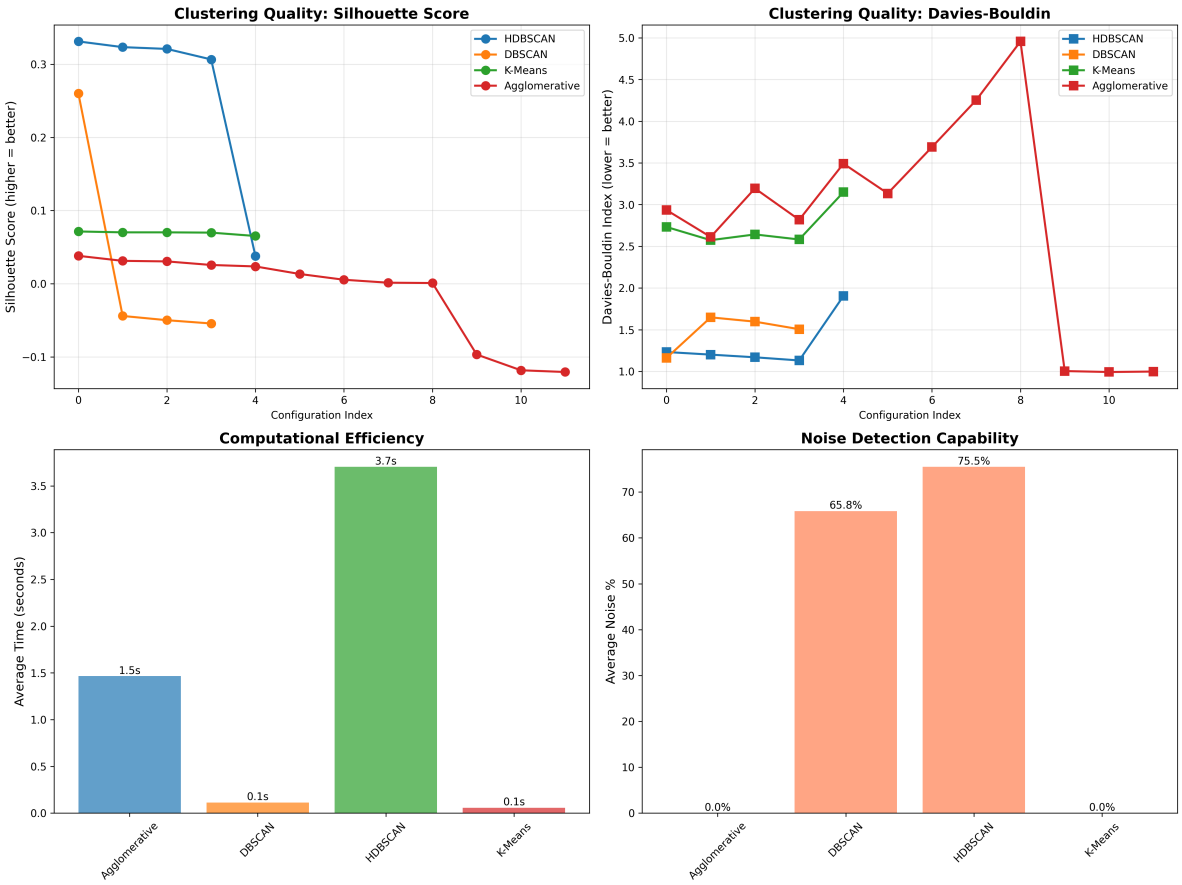

Figure 3: Quantitative comparison showing HDBSCAN's superior silhouette scores and cluster compactness before TF-IDF transformation.

After selecting optimal configurations, detailed cluster information was extracted. For HDBSCAN (`min_cluster_size=15`, `min_samples=2`, $\varepsilon = 0.05$), 41 meaningful clusters were identified with a noise proportion of 85.3% and a high silhouette score of 0.323, reflecting dense, coherent ingredient subsets such as distinct Italian cheeses and baking mixes. K-Means ($k = 100$) rapidly produced 100 clusters with zero noise, capturing broad ingredient groupings suitable for exploratory analysis, evidenced by a consistent silhouette score of 0.066. Agglomerative clustering (ward, $k = 40$) yielded 40 interpretable hierarchical clusters with no noise but lower clustering cohesion (silhouette $\sim$0.036).

4.3 Visual and Qualitative Analysis

UMAP visualizations delivered an intuitive summary of how each clustering method partitions the ingredient space before TF-IDF, highlighting both cluster structure and noise handling.

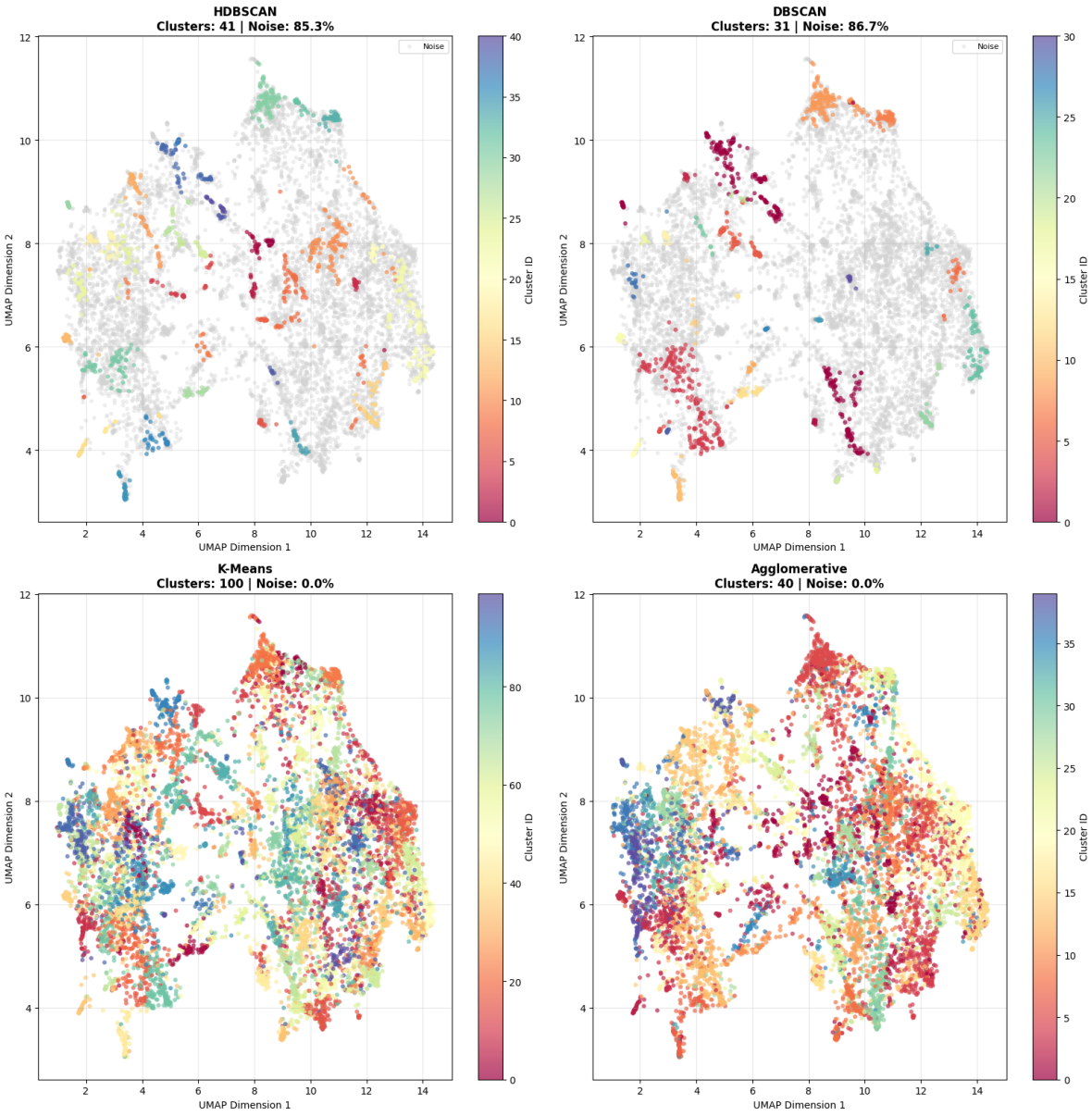

Figure 4: UMAP visualization highlighting the stark contrast between density-based methods (top row: compact clusters, high noise) and partitioning methods (bottom row: full coverage, diffuse boundaries).

HDBSCAN (top left) produces finely separated, well-localized clusters in the UMAP space, but with a vast majority of points colored in gray and marked as noise (85.3%). Valid clusters are usually compact and clearly defined, confirming the algorithm's strength in extracting dense, specific ingredient groupings and confidently filtering outliers or rare items.

DBSCAN (top right) reveals a similar trend: prominent compact clusters appear, but a high noise percentage (86.7%) leaves large regions of the space as gray noise. This further illustrates DBSCAN's density-based nature—regions of high density are grouped effectively, but sparser connections are designated as noise.

K-Means (bottom left), with zero noise handling, assigns every ingredient to a cluster, resulting in a more uniform coloring across the space. There is far greater dispersal of cluster assignments, and boundaries between clusters are often diffuse, reflecting the algorithm's tendency to partition the space into evenly sized, non-overlapping regions regardless of underlying density.

Agglomerative (bottom right) shows a similar fully colored landscape due to its all-inclusive approach. Clusters may follow prominent axes or gradients, but transitions between cluster regions are often smooth rather than sharply bounded.

The clustering results reveal a diverse spectrum of ingredient groupings, closely reflecting culinary realities and thematic relationships. DBSCAN exemplifies strong thematic coherence: Cluster 0 (288 ingredients) groups classic Mexican-inspired components—cilantro, lime juice, chili powder, salsa, black beans, avocado, jalapeno—showing clear separation of regional flavors. Cluster 3 (209) is a typical dessert baking set: cream cheese, whipped cream, confectioner's sugar, chocolate chips, and cocoa, demonstrating DBSCAN's ability to extract tight ingredient families.

HDBSCAN refines these granular groupings. Cluster 21 (75) assembles Italian staples: parmesan, mozzarella, pasta sauce, ricotta, pepperoni. Cluster 8 (78) isolates universal condiments: garlic powder, ketchup, mustard, hot sauce. Other clusters focus on seasonal baking (28: various cake mixes and frostings) or specialized diet ingredients. HDBSCAN's noise filtering enhances clarity, often removing single-use items or rare edge ingredients.

K-Means produces broader, inclusive clusters. Cluster 21 (264) captures general Asian cuisine basics—soy sauce, ginger, sesame oil, rice vinegar. Cluster 32 (97) blends pizza and pasta toppings: mozzarella, Italian seasoning, sausage, lasagna. Thus, K-Means partitions ingredients into comprehensive culinary categories, prioritizing completeness over niche focus.

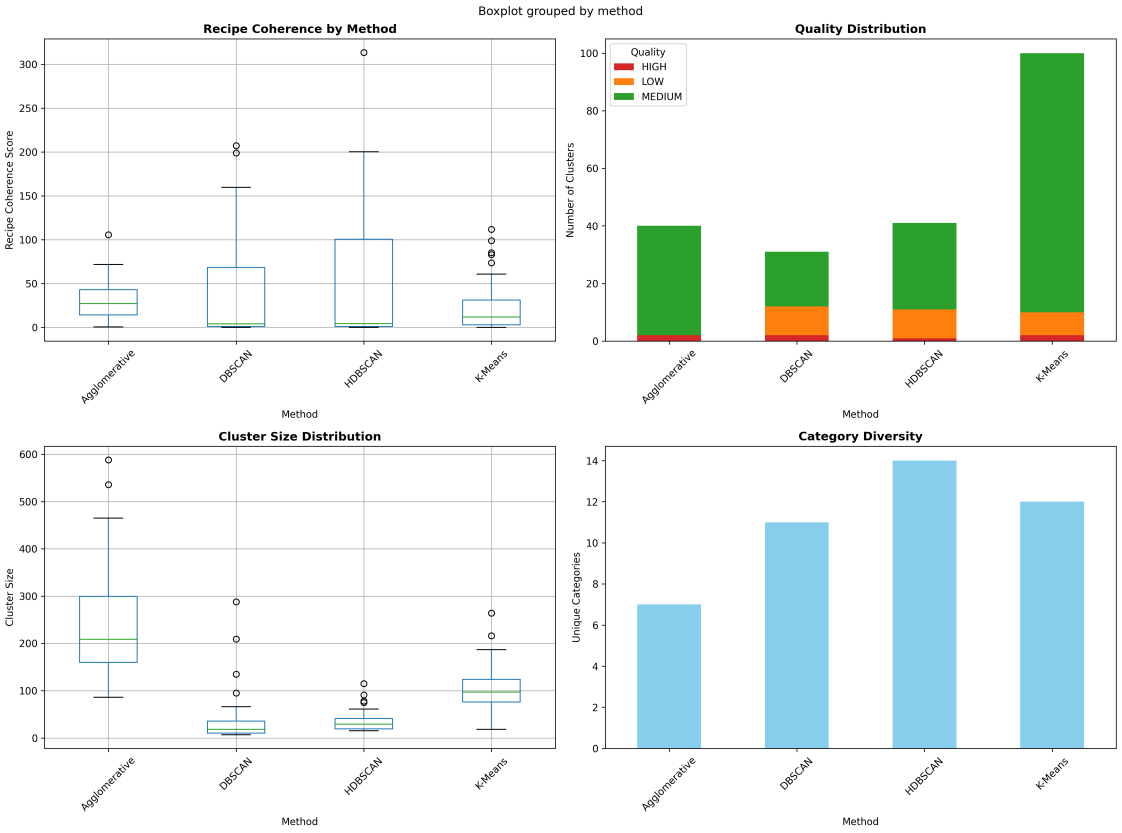

Figure 5: Cluster quality distribution showing HDBSCAN's superior proportion of high-quality clusters with broadest culinary theme coverage before TF-IDF.

Hand-labeling of clusters using keyword mapping allowed for semi-automated ingredient category assignment. Recipe coherence was computed as the ratio of recipes containing at least two ingredients from the cluster to the total ingredients in that cluster—a direct measure of practical, real-world overlap. HDBSCAN not only produces highly coherent clusters but also covers the broadest spectrum of culinary themes, validating its capacity for nuanced analysis and wide applicability.

Pre-TF-IDF Conclusion: While both DBSCAN and HDBSCAN perform strongly for culinary profile discovery, HDBSCAN works slightly better in the raw co-occurrence space, offering enhanced clarity and detail for ingredient relationships and specialized recipe identification, making it the optimal choice for downstream culinary analytics.

5 TF-IDF Enhancement

5.1 Motivation for Feature Re-Weighting

The pre-TF-IDF analysis was conducted on the raw ingredient co-occurrence matrix (post-SVD reduction). A known limitation of this approach is the disproportionate influence of ubiquitous staples such as salt, water, and pepper, which occur in a very high number of recipes but rarely define the unique cuisine or dish type. These staples can dilute the distinctiveness of co-occurrence patterns, potentially blurring the boundaries of functional ingredient modules and causing diverse recipes to be grouped primarily by their shared common ingredients rather than their distinctive culinary characteristics.

5.2 TF-IDF Implementation Methodology

To address this limitation, Term Frequency-Inverse Document Frequency (TF-IDF) transformation was applied to the ingredient matrix. The TF-IDF approach assigns lower weights to universal staples (high document frequency) and higher weights to distinctive, recipe-defining ingredients (low document frequency):

1. Applied the TF-IDF transformer to the original sparse ingredient co-occurrence matrix
2. Applied Truncated SVD to the new TF-IDF-weighted matrix to generate a new 50-component latent space
3. Re-ran the comparative clustering analysis, focusing specifically on density-based methods to determine if the resulting clusters show improved thematic specificity and culinary clarity
4. Created cluster documents by aggregating ingredients in recipes belonging to each cluster
5. Applied TF-IDF vectorizer to compute refined ingredient importance scores within clusters
6. Extracted top TF-IDF scoring ingredients per cluster to identify key defining ingredients
7. Performed cluster purity analysis using metrics like exclusivity, internal coherence, cuisine purity, semantic density, and size penalty

The expected outcome was that the TF-IDF-weighted space would produce clusters that are less generic, highlighting more specific culinary relationships and niche ingredient pairings.

5.3 Post-TF-IDF Results

5.3.1 K-Means and Agglomerative Performance

Both K-Means and Agglomerative clustering produced broad, inclusive clusters with moderate to low coherence and purity scores in the TF-IDF space. K-Means clusters remained broad with many ingredients grouped into relatively large clusters, with coherence and purity scores moderate to low due to the method's emphasis on proximity to centroids rather than strict density. This resulted in clusters with less tight culinary themes, often blending multiple cuisines or recipe types together.

Agglomerative clustering created interpretable hierarchies good for broad ingredient category groupings. However, internal coherence and exclusivity were generally low compared to density methods. It captured super-categories like herbs, spices, or baking staples but struggled with fine, functionally distinct culinary modules post-TF-IDF. Both methods' reliance on centroid or linkage distances rather than density resulted in blended culinary themes.

5.3.2 DBSCAN Performance in TF-IDF Space

DBSCAN showed remarkable improvement in the TF-IDF-weighted space, demonstrating several key strengths:

- Clusters were generally tighter and more exclusive, with higher purity grades in many cases

- Exclusivity scores were often above 0.7, indicating that ingredients in clusters rarely appeared mixed with ingredients from other clusters
- Internal coherence scores were relatively high, suggesting strong co-occurrence relationships within clusters
- Clusters reflected meaningful culinary themes such as baking, Indian, Mexican, and various regional groups
- Cluster sizes tended to be moderate, balancing specificity and coverage
- The TF-IDF transformation emphasized distinctive ingredients, leading to more focused and interpretable culinary clusters

DBSCAN's fixed neighborhood radius better captured dense, distinctive ingredient groupings emphasized by TF-IDF weighting, making it highly effective for refined culinary data mining in this transformed space.

5.3.3 HDBSCAN Initial Performance with Euclidean Distance

The initial clustering results using HDBSCAN with the Euclidean metric for the SVD-reduced TF-IDF ingredient embeddings did not perform as well as expected. This configuration struggled primarily because Euclidean distance in a reduced-dimensional space can obscure meaningful similarities between ingredients. Ingredients that are semantically close might not be spatially close in Euclidean terms after projection, leading to poor cluster cohesion and over-segmentation.

Specifically, HDBSCAN with Euclidean distance showed:

- Lower cluster purity and exclusivity scores compared to DBSCAN in the TF-IDF space
- Some clusters with high internal coherence but also showing fragmentation or smaller cluster sizes
- The method's hierarchical approach sometimes leading to over-segmentation in the TF-IDF space
- Cuisine purity and semantic density scores suggesting reasonable thematic grouping, but often less robust than DBSCAN's clear, larger clusters
- Reduced coverage and coherence metrics post-TF-IDF

Intermediate Conclusion: DBSCAN outperformed HDBSCAN with Euclidean distance in cluster purity, exclusivity, and culinary coherence after TF-IDF transformation, likely because DBSCAN's fixed neighborhood radius better captures dense, distinctive ingredient groupings emphasized by TF-IDF.

5.4 HDBSCAN Configuration Refinement

To address HDBSCAN's limitations in the TF-IDF space, the configuration was substantially revised based on the understanding that Euclidean distance was not well-suited for the transformed data:

- **Distance Metric:** Switched from Euclidean to precomputed cosine distance matrix. Cosine distance, which measures angular difference between vectors, is often better suited for TF-IDF-weighted data because it captures similarity in ingredient usage patterns regardless of magnitude differences.
- **Cluster Selection:** Changed to `cluster_selection_method='leaf'` for finer-grained, well-separated clusters. The leaf selection method focuses on finer-grained, well-separated cluster leaves in the hierarchical tree, improving thematic specificity.
- **Rationale:** The cosine similarity metric better respects the semantic structure of TF-IDF ingredient embeddings, as it is invariant to vector magnitude and focuses on the pattern of ingredient presence rather than absolute frequencies.

This revised configuration yielded noticeably improved cluster quality and interpretability. Clusters became larger, more coherent, and better aligned with culinary categories, as reflected in higher purity scores:

- Larger, more coherent clusters aligned with culinary categories
- Higher purity and exclusivity scores surpassing both Euclidean-HDBSCAN and DBSCAN
- Rich TF-IDF ingredient profiles capturing distinct themes such as well-defined Indian spice blend clusters (including ingredients like turmeric, cumin, coriander, garam masala, and mustard seeds) or cohesive French baking essentials clusters (butter, flour, sugar, vanilla, and cream)
- Superior performance in capturing semantic relationships in TF-IDF-transformed ingredient embeddings
- Better balance between cluster precision, coverage, and thematic clarity

This contrasts sharply with the fragmented and less coherent clusters observed using Euclidean distance, where such specialized culinary groups were intermixed or diluted by less relevant ingredients. The precomputed cosine HDBSCAN configuration ultimately achieved better precision, thematic coherence, and analytical utility, solidifying its role as the primary clustering method in the TF-IDF-enhanced ingredient space.

6 Final Evaluation and Trade-offs

6.1 Computational Considerations

The optimal HDBSCAN configuration (precomputed cosine distances with leaf cluster selection), while providing superior clustering quality and thematic coherence, comes with higher computational costs that must be acknowledged:

- Computing the full pairwise cosine distance matrix has quadratic complexity $O(n^2)$, causing increased memory usage and longer processing times, especially on large ingredient sets
- Hierarchical clustering with leaf selection involves more complex computations than flat clustering methods like DBSCAN or K-Means
- The approach demands more computing resources and careful optimization strategies

However, for applications requiring the most precise and interpretable culinary clusters, this computational trade-off is justified. The enhanced cluster quality, thematic specificity, and culinary relevance provided by this configuration make it worthwhile for advanced culinary analytics applications.

6.2 Performance Summary

Table 3: Clustering Method Performance Comparison Across Analysis Phases

Phase	**Method**	**Silhouette**	**Key Strength**
Pre-TF-IDF	HDBSCAN	0.32–0.33	Fine-grained modules
	DBSCAN	0.26	Strong themes
	K-Means	0.07	Full coverage
	Agglomerative	0.04	Hierarchical structure
Post-TF-IDF	DBSCAN	High	Best standard config
	HDBSCAN (Euclidean)	Low	Over-segmentation
	HDBSCAN (Cosine)	Highest	Optimal precision

7 Conclusion

The methodology progression from raw co-occurrence analysis through TF-IDF weighting to optimized distance metrics demonstrates a crucial insight: no single clustering approach universally excels across all data representations. Optimal performance depends critically on both data representation (raw co-occurrence vs. TF-IDF-weighted) and algorithm configuration (Euclidean vs. cosine distance, standard vs. leaf cluster selection). This finding has important implications for culinary data mining and similar high-dimensional sparse data analysis tasks.

For advanced culinary applications requiring precise, interpretable ingredient modules—such as recipe recommendation systems, ingredient substitution engines, and regional cuisine profiling tools—the cosine-distance HDBSCAN configuration represents the optimal approach for this dataset. Future work could explore adaptive distance metrics that automatically adjust to data characteristics, investigate hierarchical TF-IDF weighting schemes that assign different importance levels across ingredient categories, and apply these validated clustering methodologies to practical culinary applications to assess their real-world impact on recipe discovery and culinary innovation.

8 Author Contributions

All authors contributed to the overall methodology design, interpretation of results, and the preparation of this manuscript. The work distribution among the contributors was as follows:

- **Edgar Demeude:** Responsible for dataset acquisition, preprocessing, and semantic embedding generation.
- **Thomas Aubourg:** Conducted preprocessing refinement, dimensionality reduction (PCA and Truncated SVD), and implemented the K-Means and Hierarchical clustering analyses.
- **Anh-Duy Vu:** Performed the remaining analyses including DBSCAN and HDBSCAN experiments, TF-IDF transformation, cosine-distance refinement, visualization, and comprehensive result interpretation.

References

[1] The scikit-learn developers. "Demo of HDBSCAN clustering algorithm." *scikit-learn* documentation, 2025. https://scikit-learn.org/stable/auto_examples/cluster/plot_hdbscan.html

[2] Rémy Cazabet. *Master IA/DS : Data Mining*. Teaching web page. https://cazabetremy.fr/Teaching/DSIA/DM.html

[3] McInnes, L. and Healy, J. "Understanding UMAP." Pair-Code blog, 2021. https://pair-code.github.io/understanding-umap/

[4] Pre-trained sentence-transformers model https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2