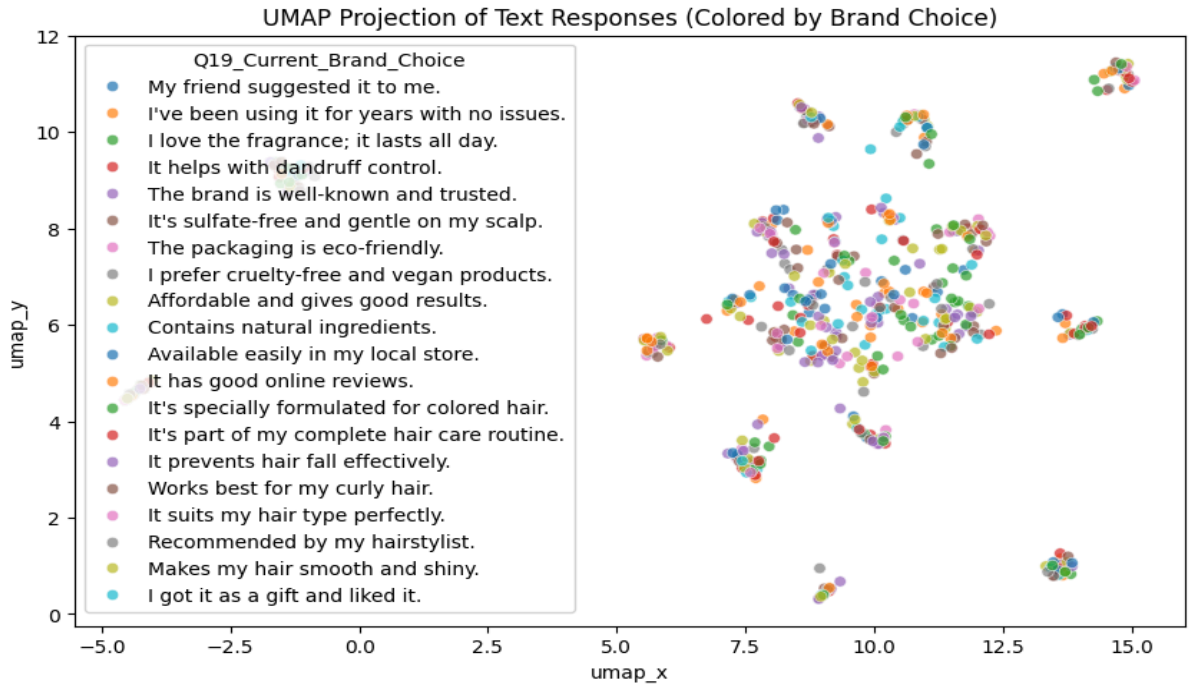
Section 16: Text Summarization & Clustering

1. Cluster Assignments

- Table with: Respondent_ID, Question_ID, Raw_Text, Cluster_Label



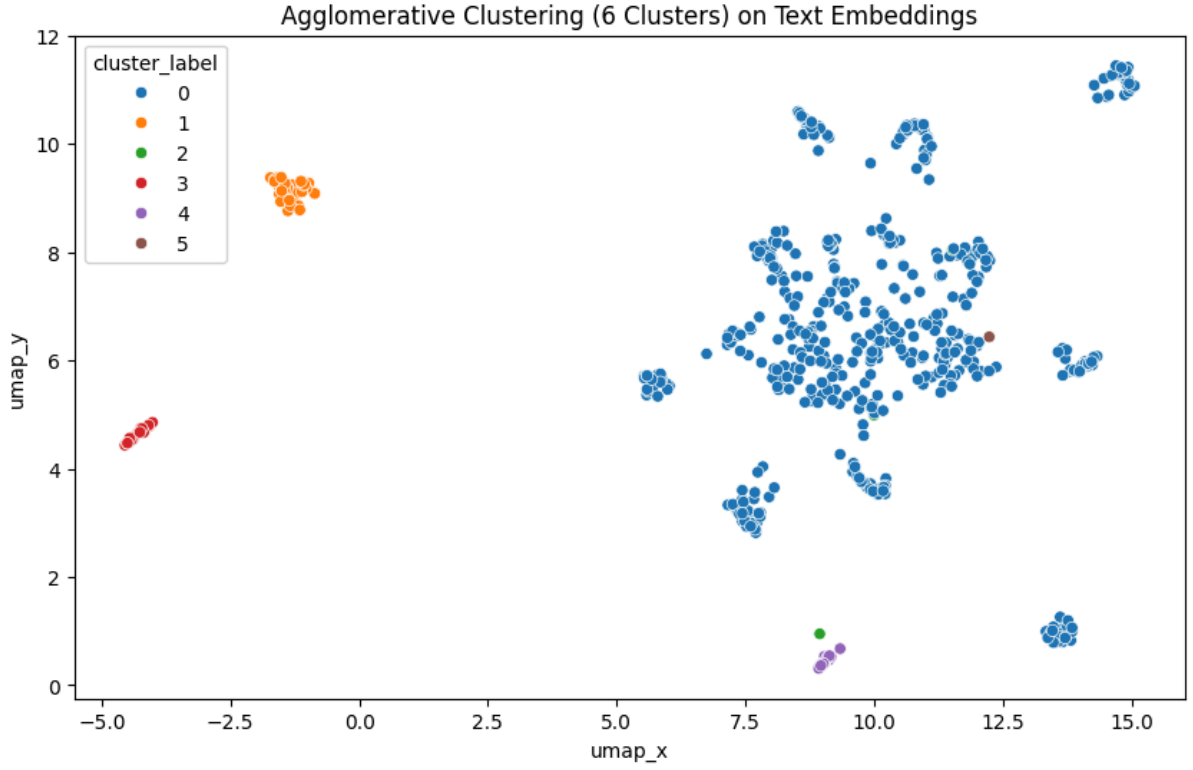


Chart Interpretation & Insights

Chart 1: UMAP Projection of Text Responses (Colored by Brand Choice)

This chart visualizes individual responses projected into 2D space using UMAP (Uniform Manifold Approximation and Projection), colored by brand mention.

What it shows:

- Each dot represents a unique consumer text response to the open-ended brand choice question.
- Different colors represent different brand preferences.
- Clustered areas indicate semantically similar responses, while isolated dots represent more unique or niche sentiments.

Insights:

- There's a clear central dense cluster with a wide mix of brand choices, suggesting common brand perceptions or features (e.g., fragrance, dandruff control, affordability).
- Several distinct clusters indicate specific brand attributes (e.g., "eco-friendly", "sulfate-free", "recommended by hairstylist").
- UMAP effectively separates shared consumer language and brand-specific language, enabling clearer brand-driven narratives.

Chart 2: Agglomerative Clustering (6 Clusters) on Text Embeddings

This chart groups the same UMAP embeddings into 6 semantic clusters using hierarchical clustering.

What it shows:

- Each point (consumer response) is colored by cluster label (0 to 5), derived from semantic similarity using agglomerative clustering.
- Cluster 0 dominates (514 responses), while clusters 1 to 5 represent more specific or niche themes.

Cluster-Level Interpretation & Themes

Cluster	Count	Key Themes	Interpretation
0	514	General hair care satisfaction, popular brands, affordability, trust, dandruff, fragrance	Mainstream perception — largest segment; reflects dominant brand equity
1	38	Packaging design, eco-friendliness, ease of use, leak-proof, scalp comfort	Niche focus on packaging experience and usability
2	3	Sulfate-free, gentle on hair, gifting in sachets	Highly niche cluster focused on mild, giftable, sachet-based products
3	26	Hair fall reduction, dandruff, UV protection, chemical-free, squeeze bottles	Functional benefit cluster — addresses hair/scalp issues and format utility
4	18	Luxury, eco-conscious, curl definition, keratin/biotin, frizz control, packaging comfort	Premium hair care and styling — defined curls, strength, frizz management
5	1	Travel-friendly, eco-friendly, dandruff control, natural shine	Isolated but insightful — sustainability and performance combined

---

Strategic Insights from Clustering

1. Cluster 0 represents the dominant consumer voice. It includes the largest group and gives insights into the general perception and baseline product expectations.

2. Clusters 1, 3, and 4 highlight emerging needs and whitespace opportunities:
    - Cluster 1 focuses on packaging functionality and eco-design.
    - Cluster 3 emphasizes treatment-based solutions such as dandruff control, UV protection, and usability.
    - Cluster 4 reflects styling and nourishment needs (e.g., keratin, biotin, curl definition).
3. Cluster 5, though small, shows an emerging consumer segment interested in travel convenience and eco-friendly packaging.

Evaluation Metrics

Clustering Evaluation Metrics – Insights

1. Cophenetic Correlation Coefficient: 0.6158

What it is:

- This measures how well the dendrogram (produced by hierarchical clustering) preserves the pairwise distances between the original data points.
- Range: 0 to 1 (closer to 1 is better).

Insight:

- A score of **0.6158** indicates a **moderately good fit** between the hierarchical cluster structure and the original distances in the embedding space.
- It means that the clustering is **reasonably faithful** in preserving the relative distances between consumer text responses.
- The hierarchical relationships among clusters are **trustworthy enough** to interpret macro-level themes (e.g., packaging vs. hair fall control), but not perfectly granular.

---

2. Silhouette Score (n=6): 0.1218

What it is:

- This measures **how well each point fits within its cluster**, comparing intra-cluster cohesion vs. inter-cluster separation.
- Range: -1 to 1
    - **> 0.5:** Strong clustering
    - **0.25–0.5:** Reasonable
    - **< 0.25:** Weak structure or overlapping clusters
    - **< 0:** Poor clustering

Insight:

- A score of **0.1218** suggests **weak clustering structure**.
- There's **significant overlap** between the semantic meaning of the clusters, which is expected in **text data** where themes (e.g., "eco-friendly" and "travel-friendly") often intersect.
- Despite weak silhouette score, **semantic interpretability** (from summaries you provided) shows that clusters are **still meaningful** for thematic analysis — especially clusters 1 to 4.

Combined Interpretation

Metric	Value	Interpretation
Cophenetic Correlation	0.6158	Structure of clusters moderately reflects actual text similarities
Silhouette Score (6 clusters)	0.1218	Clusters are not tightly separated; themes overlap, typical in text clustering

2. **Analysis Notebook**
    - Code and comments for embedding, clustering, and summarization
    - Parameter settings (distance metric, linkage method)

https://colab.research.google.com/drive/18H6BxSEtdJiG19SZo6dnLsH2uL0OXTDE#scrollTo=NY6srAoRzzuH

Description

Section 16: Text Summarization & Clustering

Scope: Group open-ended survey responses into coherent themes and summarize them into concise executive-level insights, supported by evaluation metrics.

Objectives

- Cluster open-ended responses from selected questions into thematic groups.
- Generate 2–3 sentence summaries per cluster using advanced NLP summarization.
- Validate cluster fidelity using silhouette scores and cophenetic correlation.

Analysis Tasks

Task	Details	Method

1. Data Preparation	- Use preprocessed responses from Q19–Q21, Q25–Q26, Q28–Q29, Q33, Q40–Q41. - Use the cleaned & lemmatized text.	`python import pandas as pd df = pd.read_csv("open_ended_data_cleaned.csv") responses = df["Clean_Text"].tolist()`
2. Sentence Embedding	- Generate sentence embeddings using Sentence-BERT. - Optional: Reduce dimensionality using UMAP.	`python from sentence_transformers import SentenceTransformer from umap import UMAP model = SentenceTransformer("all-MiniLM-L6-v2") embeddings = model.encode(responses, show_progress_bar=True) reducer = UMAP(n_components=10, random_state=42) reduced_embeddings = reducer.fit_transform(embeddings)`
3. Clustering	- Perform hierarchical agglomerative clustering. - Set number of clusters (e.g., 5–7).	`python from sklearn.cluster import AgglomerativeClustering cluster_model = AgglomerativeClustering(n_clusters=6, affinity="euclidean", linkage="ward") df["Cluster_Label"] = cluster_model.fit_predict(reduced_embeddings)`
4. Cluster Evaluation	- Evaluate quality using silhouette score. - Assess dendrogram	`python from sklearn.metrics import silhouette_score from scipy.cluster.hierarchy import linkage, cophenet from scipy.spatial.distance import pdist sil_score = silhouette_score(reduced_embeddings,`

	fidelity with cophenetic coefficient.	`df["Cluster_Label"]) print("Silhouette Score:", sil_score) Z = linkage(reduced_embeddings, method="ward") coph_corr, _ = cophenet(Z, pdist(reduced_embeddings)) print("Cophenetic Correlation:", coph_corr)`
5. Cluster Summarization	- Use a transformer-based summarizer (BART/T5) for each cluster. - Summarize with 2–3 sentences.	`python from transformers import pipeline summarizer = pipeline("summarization", model="facebook/bart-large-cnn") cluster_summaries = {} for cluster_id in df["Cluster_Label"].unique(): cluster_texts = df[df["Cluster_Label"] == cluster_id]["Clean_Text"].tolist() joined_text = " ".join(cluster_texts)[:1024] # truncate to fit model limits summary = summarizer(joined_text, max_length=60, min_length=20, do_sample=False)[0]["summary_text"] cluster_summaries[cluster_id] = summary`
6. Thematic Validation	- Select 2–3 sample responses per cluster. - Assign descriptive labels to clusters.	`python cluster_samples = {} for cluster_id in df["Cluster_Label"].unique(): sample_texts = df[df["Cluster_Label"] == cluster_id]["Raw_Text"].head(3).tolist() cluster_samples[cluster_id] = sample_texts # Add manual labels during report creation`

Deliverables

Cluster Assignments

- Table format:
  `Respondent_ID`, `Question_ID`, `Raw_Text`, `Cluster_Label`
  Export:

```
python df.to_csv("cluster_assignments.csv", index=False)
```

Cluster Summaries

- 5–7 paragraphs (1 per cluster) summarizing thematic content.
- Example verbatims for each cluster:
    - `cluster_samples[cluster_id]` for manual review and selection.

Evaluation Metrics

- **Silhouette Score:** Measure of intra-cluster cohesion.
- **Cophenetic Correlation Coefficient:** Dendrogram fidelity to original distances.

Analysis Notebook

- Includes all code:
    - Sentence embeddings
    - Dimensionality reduction
    - Clustering
    - Summarization
    - Evaluation metrics
- Parameter Settings:
    - Sentence-BERT: `all-MiniLM-L6-v2`
    - UMAP: `n_components=10`
    - Clustering: `n_clusters=6`, `linkage=ward`
    - Summarizer: `facebook/bart-large-cnn`, `max_length=60`

Presentation-Ready Slide Snippets

- 1 slide per cluster:
    - **Cluster Label** (e.g., “Fragrance & Sensory Appeal”)
    - **Summary Paragraph** (2–3 lines)
    - **Representative Quotes** (2–3 verbatims)