Customer Search Query Clustering Using Machine Learning

Saja Moussa

September 29, 2025

Abstract

This report presents the ongoing work on clustering and analyzing customer search queries using machine learning techniques. The primary goal is to provide insights into customer demand through embeddings, clustering, and analytics dashboards. This first part focuses on dataset collection, preprocessing, and translation steps applied to a multilingual shopping queries dataset from the Amazon ESCI Challenge. The dataset contains 277,044 queries in English, Japanese, and Spanish. Preprocessing includes deduplication, cleaning, and translation to English, resulting in a reduced dataset suitable for downstream embedding and clustering tasks.

1 Introduction

Customer search queries are a valuable source of information for understanding customer intent and preferences. Clustering and analyzing these queries can help identify trends, optimize product search, and support decision-making. This project applies state-of-the-art NLP techniques, including Sentence-BERT embeddings and Transformers from Hugging Face, to cluster multilingual shopping queries and generate actionable insights.

2 Data Collection

The dataset used in this project comes from the Amazon ESCI Challenge, containing 277,044 shopping queries across three languages:

- English (US): 210,679 queries
- Japanese (JP): 35,399 queries
- Spanish (ES): 30,966 queries

The dataset includes four columns: query text, product ID, example ID, and locale. For the purpose of clustering queries, only the text of the queries is required.

3 Data Preprocessing and Translation

Preprocessing is a crucial step for preparing the dataset for machine learning tasks. The following ETL pipeline was applied:

1. **Extraction:** Loading the dataset into a Pandas DataFrame.
2. **Column Removal:** Dropping unnecessary columns (`product_id` and `example_id`).
3. **Duplicate and Null Removal:** Removing duplicate queries and null values.
4. **Translation:** Translating Spanish and Japanese queries to English using Hugging Face models `Helsinki-NLP/opus-mt-es-en` and `Helsinki-NLP/opus-mt-ja-en`.
5. **Loading:** Saving the cleaned and translated dataset to a new CSV file for further analysis.

4 Comparative Analysis of Original and Translated Dataset

The table below summarizes the dataset before and after preprocessing:

Table 1: Dataset Statistics Before and After Preprocessing

Metric	Original Dataset	Translated Dataset
Number of Queries	277,044	14,534
Number of Columns	4	1
Duplicate Queries	262,493	0
Unique Queries	14,551	14,534
Vocabulary Size	14,387	14,003
Mean Query Length (words)	3.23	3.47
Max Query Length (words)	29	160

4.1 Observations

- Deduplication and translation reduced the dataset size by 94.75%.
- Vocabulary decreased slightly (2.67%) after normalization to English.
- Translation enables downstream ML models to process queries in a single language.
- The cleaned dataset now contains unique queries ready for embedding generation, clustering, and analytics.

5 Embedding Generation

Once the dataset was cleaned and normalized into English, the next step was to represent each query in a numerical vector space using pre-trained Sentence-BERT embeddings.

5.1 Model Selection

We selected the `all-MiniLM-L6-v2` model, a lightweight and widely used variant of Sentence-BERT, which produces 384-dimensional dense embeddings. This model balances speed and performance, making it well-suited for large-scale query datasets.

Each query was converted into an embedding vector, resulting in a matrix of shape:

$$(NumberofQueries) \times 384$$

This representation captures the semantic similarity between queries: similar queries are mapped to nearby vectors in the embedding space.

5.2 Verification

To ensure quality:

- The embedding matrix was checked for dimensional consistency.
- No NaNs or empty vectors were detected.
- A random sample of vectors was inspected to confirm correct output.

6 Dimensionality Reduction for Visualization

Since embeddings exist in a high-dimensional space (384 dimensions), dimensionality reduction techniques were applied to project the embeddings into lower dimensions for visualization and clustering.

6.1 Method

- **UMAP (Uniform Manifold Approximation and Projection)** was chosen to reduce the embeddings to 50 dimensions, balancing computational efficiency with structural preservation.
- For visualization purposes, a 2D projection of a sample of queries was also created.

7 Clustering of Queries

Clustering was performed in two hierarchical stages.

7.1 Stage 1: HDBSCAN

- HDBSCAN was applied to group semantically similar queries into fine-grained clusters.
- A total of 77 clusters were identified, with 3,178 queries labeled as noise (i.e., not belonging to any cluster).
- This step created small, coherent clusters representing niche topics.

7.2 Stage 2: K-Means on Clusters

- The HDBSCAN clusters were further grouped using K-Means into 20 higher-level "super-clusters."
- This hierarchical structure allowed queries → clusters → super-clusters.
- Each super-cluster was assigned a representative name by extracting top keywords from its queries, ignoring stopwords.

8 Cluster Examples

Table 2 shows selected examples of super-clusters, their keywords, and sample queries.

Table 2: Examples of Identified Super-Clusters

Super-Cluster	**Top Keywords**	**Sample Queries**
Projector	projector, screen, epson, inch, stand	planetary projector; holographic projector 3d
Gifts	gifts, wall, women, stickers, ring	christmas decorations; decals; silver ring of ley woman
Paper	paper, printer, poster, ink, toilet	poster big lebowski; professional photograph printer
Chair	bed, chair, rack, organizer, storage	2 compact beds; anti-fatigue kitchen mat lemons
Funko	funko, pop, gladiator, crystal	funko gladiator; funko pop dark crystal
Hair	hair, coffee, mask, oil, air	anti-aging facial oil; accessories apple pencil
Women	women, men, case, shoes, iphone	grey woman coats; overcoat ma'am
Light	light, led, glasses, fan, curtain	color led light ribbon; motorcycle headlamp

9 Findings and Observations

- Clustering revealed 77 fine-grained clusters grouped into 20 super-clusters.
- Large thematic groups emerged, such as **Women**, **Hair**, **Light**, and **Gifts**, each covering hundreds to thousands of queries.
- Smaller but focused clusters emerged around specific products such as **Funko**, **Projectors**, or **Magnetic holders**.
- Noise points represented unusual or ambiguous queries not fitting into any cluster.

10 Analytics Dashboard

To support interactive exploration and business-friendly visualization of the clustering results, an analytics dashboard was developed using `Streamlit`. This tool enables real-time inspection of customer queries at different levels of granularity and provides actionable insights into customer demand patterns.

10.1 Dashboard Features

The dashboard consists of four main tabs:

- **Tab 1 - Overview:** Displays global metrics such as total queries, number of clusters, and noise points. Includes a horizontal bar chart showing query distribution by super cluster, a histogram of cluster size distribution, and a summary statistics table. Charts dynamically adapt to the current selection (all clusters vs. a specific super cluster).
- **Tab 2 - Super Clusters:** Provides an interactive summary table of all super clusters. Selecting a cluster reveals detailed statistics, sample queries, and a pie chart showing sub-cluster distribution.
- **Tab 3 - Sub Clusters:** Offers heatmap visualizations of sub-cluster sizes across super clusters and interactive tools for exploring relationships between clusters.
- **Tab 4 - Word Clouds:** Generates word clouds for all queries or for specific clusters. Includes side-by-side comparisons of two super clusters and adaptive options for current selections.

10.2 Visual Feedback and Interactivity

- **Metrics Section:** Adjusts automatically to show either global or cluster-specific statistics. For example, when a super cluster is selected, the dashboard displays its size instead of the overall dataset.
- **Dynamic Indicators:** Blue info boxes and titles clarify what data is currently being displayed.
- **Advanced Visualizations:** When viewing all clusters, a treemap shows hierarchical relationships between super clusters and sub-clusters, alongside a bar chart of the top 15 sub-clusters. When a specific cluster is selected, the dashboard shows a horizontal bar chart of its sub-clusters and a detailed breakdown with sample queries.
- **Error Handling:** The system provides clear feedback in edge cases, ensuring robustness when clusters contain very few queries or noise.