RESEARCH ARTICLE OPEN ACCESS

A Comparative Study of Clustering Techniques on an Online Marketplace Dataset

Mrugani Kurtadikar, Deepak Singh
Adobe Inc Noida
Adobe Inc Noida
India

ABSTRACT
Online marketplaces often suffer from inconsistent categorization, affecting both discoverability and user experience. This study evaluates several clustering techniques to group items based on textual metadata, including name, summary, description, and keywords. The data is pre-processed using both TF-IDF and sentence embeddings, followed by using clustering algorithms such as K-Means, HDBSCAN, and Agglomerative Clustering. Performance and cluster effectiveness is evaluated using metrics like the Silhouette Score and Davies–Bouldin Index, alongside qualitative analysis of cluster cohesion. The findings suggest that sentence embeddings combined with HDBSCAN produce semantically meaningful clusters that align well with real-world categorical structures. This work provides insights into automated organization and classification of marketplace content.

Keywords:- Clustering Techniques

I. INTRODUCTION

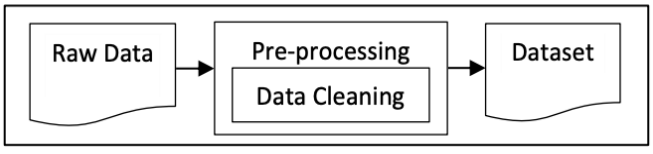
Online marketplaces, such as browser extension stores and creative tool platforms, contain hundreds to thousands of items that are often poorly categorized. Users face challenges in discovering relevant tools, especially as the volume of content increases. This paper examines the effectiveness of clustering algorithms in automatically grouping similar items based on textual metadata.

II. RELATED WORK

Clustering has been widely used for product segmentation, customer profiling, and document classification. Prior studies in retail domains have compared algorithms like K-Means, DBSCAN, and hierarchical clustering. However, few works focus on the clustering of text-heavy marketplace metadata, particularly in the context of plugins or extensions.

III. DATASET DESCRIPTION

We collected a dataset of approximately 300 items, each described by a title, summary, description, a set of keywords, and a binary flag indicating a specific feature. All text fields were concatenated and pre-processed through lowercasing, punctuation removal, and stop-word filtering [1].



IV. METHODOLOGY

A. *Text Vectorization*

To represent textual metadata numerically, we included two primary vectorization techniques: TF-IDF and sentence embeddings deduced from pre-trained transformer models.

For TF-IDF vectorization, we concatenated multiple text fields—including the name, summary, description, and keywords—into a single string for each sample. We also applied the TfidfVectorizer from Scikit-learn, limiting the feature space to the top 5,000 terms by frequency to balance expressiveness and computational effectiveness:

```
from sklearn.feature_extraction.text import TfidfVectorizer

texts = df['name'] + ' ' + df['summary'] + ' ' + df['description'] + ' ' + df['keywords']
tfidf = TfidfVectorizer(max_features=5000)
X_tfidf = tfidf.fit_transform(texts)
```

In addition, we employed sentence embeddings generated by the Sentence Transformer library, using the all-MiniLM-L6-v2 pre-trained model known for its compact size and strong semantic representation. This system transforms each concatenated text data into a dense vector embedding that captures contextual and semantic details beyond surface-level token frequencies[2]:

```
from sentence_transformers import SentenceTransformer

model = SentenceTransformer('all-MiniLM-L6-v2')
X_embed = model.encode(texts.tolist(), show_progress_bar=True)
```

B. *Clustering Algorithms*

We totally analysed three distinct clustering algorithms—K-Means, HDBSCAN, and Agglomerative Clustering—to analyse their effectiveness in grouping similar items. Each algorithm was applied independently to two types of feature representations: TF-IDF vectors, which capture term relevance based on word frequency, and embedding-based vectors, which render semantic relationships in a dense, low-dimensional space.

```
from sklearn.feature_extraction.text import TfidfVectorizer

texts = df['name'] + ' ' + df['summary'] + ' ' + df['description'] + ' ' + df['keywords'
tfidf = TfidfVectorizer(max_features=5000)
X_tfidf = tfidf.fit_transform(texts)
```

1) ***K-Means***: A centroid-grounded, that partitions data into **k** clusters by reducing the within-cluster sum of squares. It assumes global clusters and performs well when clusters are of similar size and density. We tested multiple k values to observe sensitivity to the number of clusters[6].

For K-Means, we fixed the number of clusters to 10 and ensured reproducibility through a fixed arbitrary seed:

```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=10, random_state=42)
labels = kmeans.fit_predict(X_embed)
```

2) ***HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise):*** A density-grounded algorithm that identifies clusters of varying shapes and densities without requiring the number of clusters as input. It builds a hierarchy of clusters and condenses it based on density stability, making it robust to noise and particularly effective for real-world, inversely distributed data [3].

```
import hdbscan

clusterer = hdbscan.HDBSCAN(min_cluster_size=5)
labels = clusterer.fit_predict(X_embed)
```

3) ***Agglomerative Clustering:*** A hierarchical approach that starts with each data point as its own cluster and merges the most analogous clusters iteratively based on a relation criterion (e.g., average, complete, or ward). This system is especially useful when the dataset exhibits a nested or tree-like cluster structure[7].

By applying each algorithm across both TF-IDF and embedding-grounded features, we aimed to assess how well each combination captured meaningful groupings and handled the underlying structure of the online business dataset.

Agglomerative Clustering was similarly evaluated (code omitted here for brevity), completing the comparative analysis

C. Evaluation Metrics

To quantitatively assess clustering performance, we employed the Silhouette Score[4] and the Davies–Bouldin Index[5]—two standard criteria that respectively measure cluster cohesion and separation as well as inter-cluster similarity.

Also, we generated two-dimensional visualizations using Uniform Manifold Approximation and Projection (UMAP) to qualitatively evaluate cluster interpretability.

Sample metric calculation snippet:

```
from sklearn.metrics import silhouette_score, davies_bouldin_score

print("Silhouette Score:", silhouette_score(X_embed, labels))
print("Davies-Bouldin Index:", davies_bouldin_score(X_embed, labels))
```

HDBScan grounded result clusters:

The following cluster in Fig. 1 presents the clusters generated using the HDBSCAN algorithm. These results highlight its capability to detect dense clusters while effectively filtering out non-relevant and outliers.

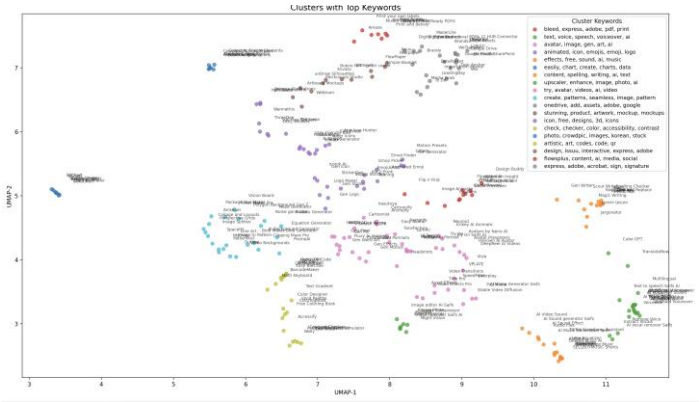


Fig. 1

K-Means grounded result clusters:

The clusters shown below in Fig. 2 were generated using the K-Means algorithm. This system partitions the data into a fixed number of clusters, often resulting in well-separated, compact clusters.

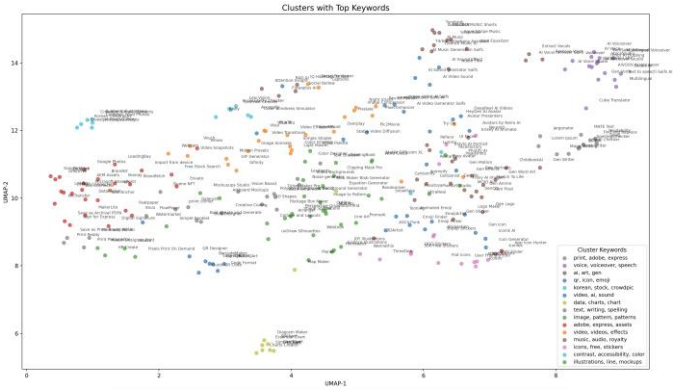


Fig.2

V. METHODOLOGY

Quantitative Evaluation of Cluster Quality:

Clustering approaches exercising sentence embeddings consistently outperformed those grounded on TF-IDF representations in terms of Silhouette scores, with the performance gap most pronounced when employing HDBSCAN. Specially, K-Means clustering demonstrated better results on higher-dimensional embedding spaces, whereas TF-IDF vectors were better suited for Agglomerative Clustering. These findings highlight a significant interaction between the clustering algorithm and the feature representation, underscoring the importance of selecting appropriate combinations for optimal performance.

Qualitative Assessment of Cluster Interpretability:

Clusters generated from sentence embeddings displayed strong semantic coherence, naturally grouping into intuitive categories such as "Accessibility Checkers," "Media Enhancers," and "Data Visualisation." In contrast, clusters derived from TF-IDF vectors exhibited considerable keyword overlap but lacked meaningful semantic relationship, resulting in groupings that were less interpretable and less aligned with user expectations.

Discussion:

Sentence-transformer embeddings provide a semantically rich feature space that facilitates more effective clustering. Density-based methods, exemplified by HDBSCAN, offer distinct advantages including flexibility in cluster shape and inherent robustness to noise, without requiring prior specification of the number of clusters. This is corroborated quantitatively by Silhouette scores: HDBSCAN achieved a notably high score of 0.5826 on UMAP-reduced embeddings, reflecting strong intra-cluster cohesion and clear inter-cluster separation. Conversely, K-Means consistently yielded low Silhouette scores (~0.1042) on both original and UMAP-transformed embeddings, suggesting that its assumptions of spherical clusters and uniform density are ill-suited to the complex, semantically rich nature of text-derived embeddings. Collectively, these results emphasize the suitability of density-based clustering methods for organizing marketplace data, where cluster shapes are irregular and density distributions are heterogeneous.

VI. CONCLUSIONS

This study demonstrates the efficacy of clustering techniques in organizing online marketplace items based on textual metadata. The combination of embedding-based vectorization with density-based clustering algorithms such as HDBSCAN produces more coherent and practically useful groupings compared to traditional approaches. Future research directions include extending the methodology to support multilingual datasets and conducting evaluations at larger scales to further validate and refine the approach.

REFERENCES

[1] Ulfa, A. H., Maulidina, S. N., & Chandra, H. (2021). Product clustering analysis on the marketplace using K-means approach (case study: Shopee). Asian Journal of Science and Engineering, 1(2), 73–78.

[2] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (pp. 3982–3992).

[3] Campello, R. J. G. B., Moulavi, D., & Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 160–172). Springer.

[4] Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1(2), 224–227. https://doi.org/10.1109/TPAMI.1979.4766909

[5] Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

[6] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

[7] Murtagh, F., & Contreras, P. (2012). Algorithms for hierarchical clustering: An overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2(1), 86–97. https://doi.org/10.1002/widm.53