Mining the words of Springfield

A computational journey into the dialogues of The Simpsons, through the use of text mining techniques.

Alessandro Bosi, 837381
Martina Pantò, 901346

1. Introduction

This project presents a comprehensive linguistic and thematic analysis of *The Simpsons*, one of the longest-running animated TV series.
By examining **over *60,000* cleaned dialogue lines** from **more than *700* episodes**, we explore character-specific language use, conversational patterns, and recurring themes through advanced text mining techniques. Methods such as ***lemmatization***, ***sentiment classification, topic modeling, and clustering*** highlight unique linguistic features of characters and reveal the narrative and emotional nuances embedded in their dialogues. The work demonstrates the power of computational linguistics to analyze cultural phenomena and provides a framework for further research in dialogue-based media.

2. Preprocessing

Lemmatization

We applied the *WordNetLemmatizer* to convert each token to its base form, in order to fill the new column *final_text*, which represents the cleaned and normalized version of each dialogue line.

Dialogue segmentation

Since each conversation in the original dataset was separated by a completely empty line, we used this pattern to assign a unique *dialogue index* to each group of consecutive lines. We then removed all empty lines.

Normalization

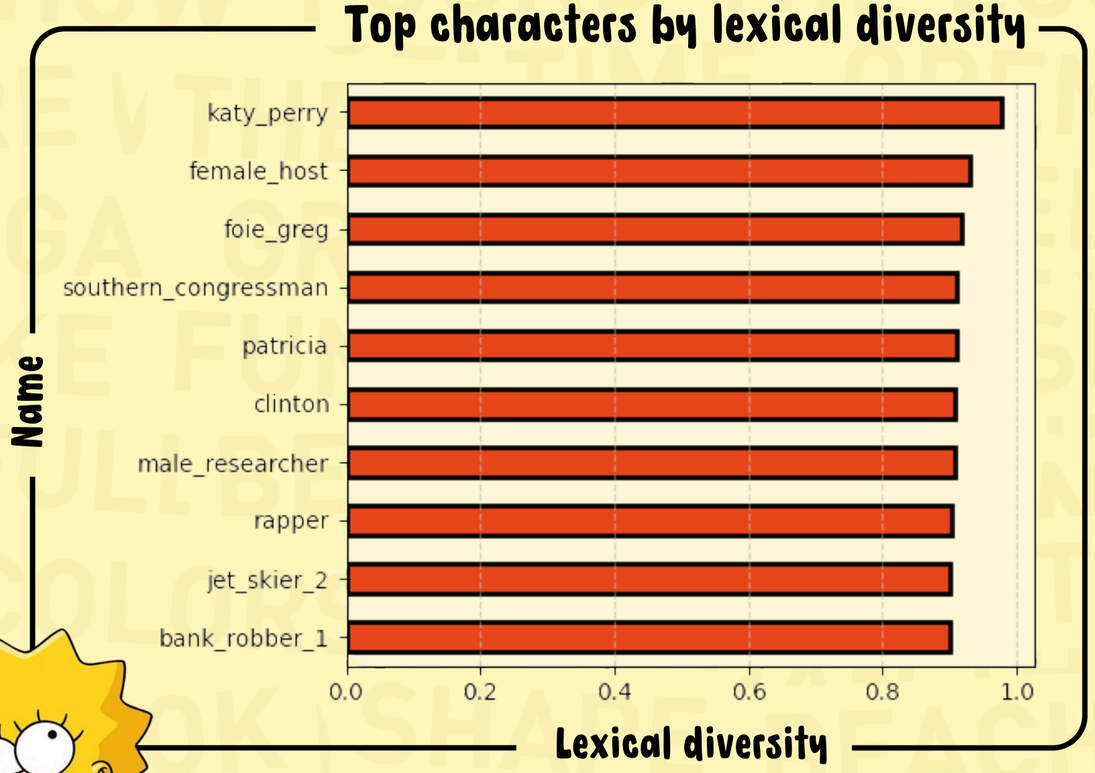
We standardized *characters* and *dialogue texts* converting them to lower case, replacing spaces with underscores, removing special characters ensuring uniformity and minimizing duplication errors. We then performed *tokenization* both with and without *stopwords*.

2.1. Exploratory data preparation

To facilitate the *EDA* we extended our dataframe by computing a variety of linguistic and stylistic features:

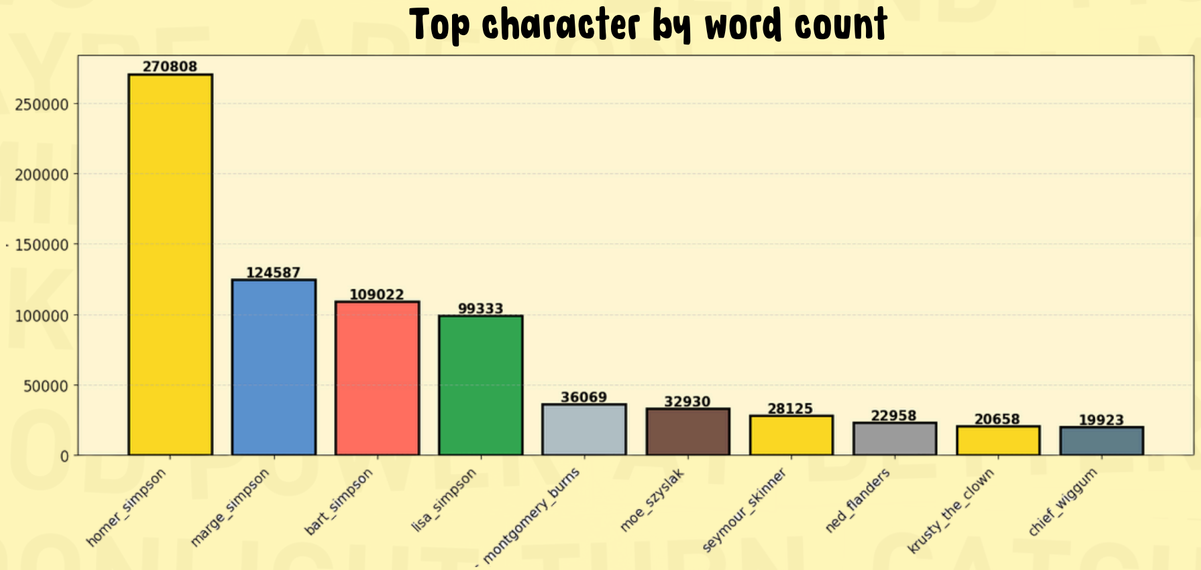
- *word count* and *letter count*
- *sentiment scores* (polarity and subjectivity)
- *unique word count, average word length*
- sentence-level metrics (*sentence length* and *sentence number)*
- punctuation-based features as *exclamation number* and *question number*
- number of *stopwords*, using a customized *stopwords* list

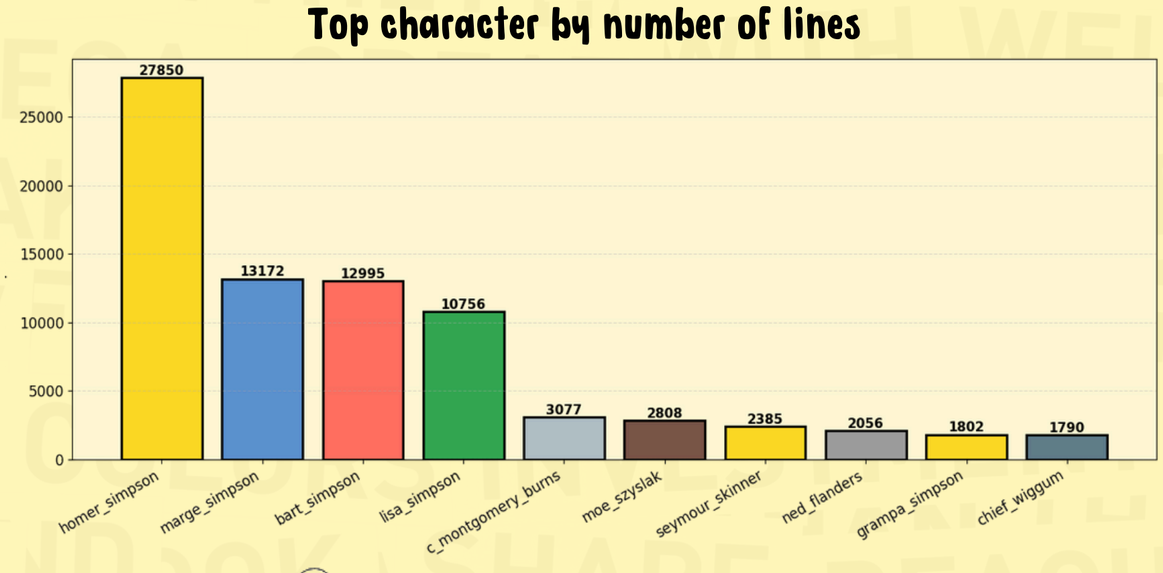
We then developed a function called ***lexical_diversity_analysis*** to compute the type-token ratio for each character, a value between 0 and 1 that indicates vocabulary richness.



2.2. Character analysis

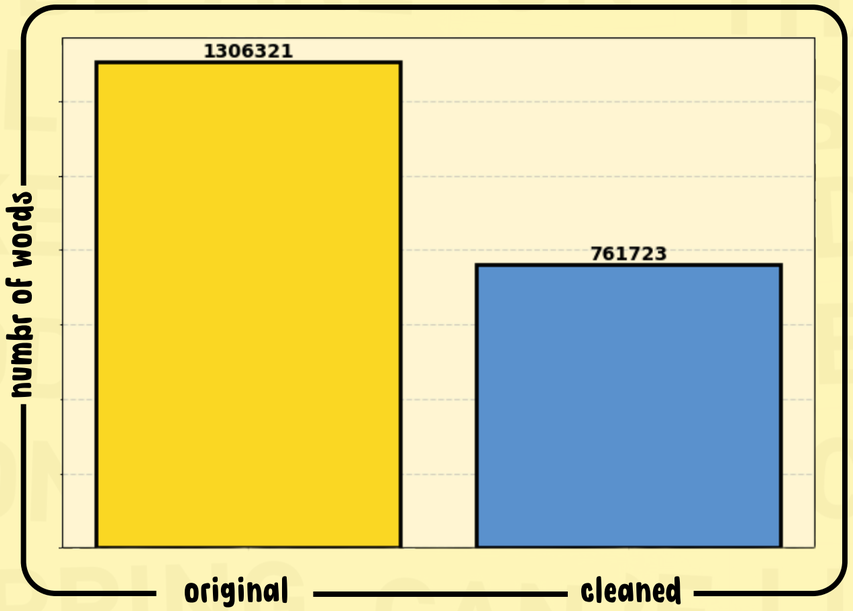
Unsurprisingly, Homer, Marge, Bart and Lisa are consistently ranked highest in these metrics:

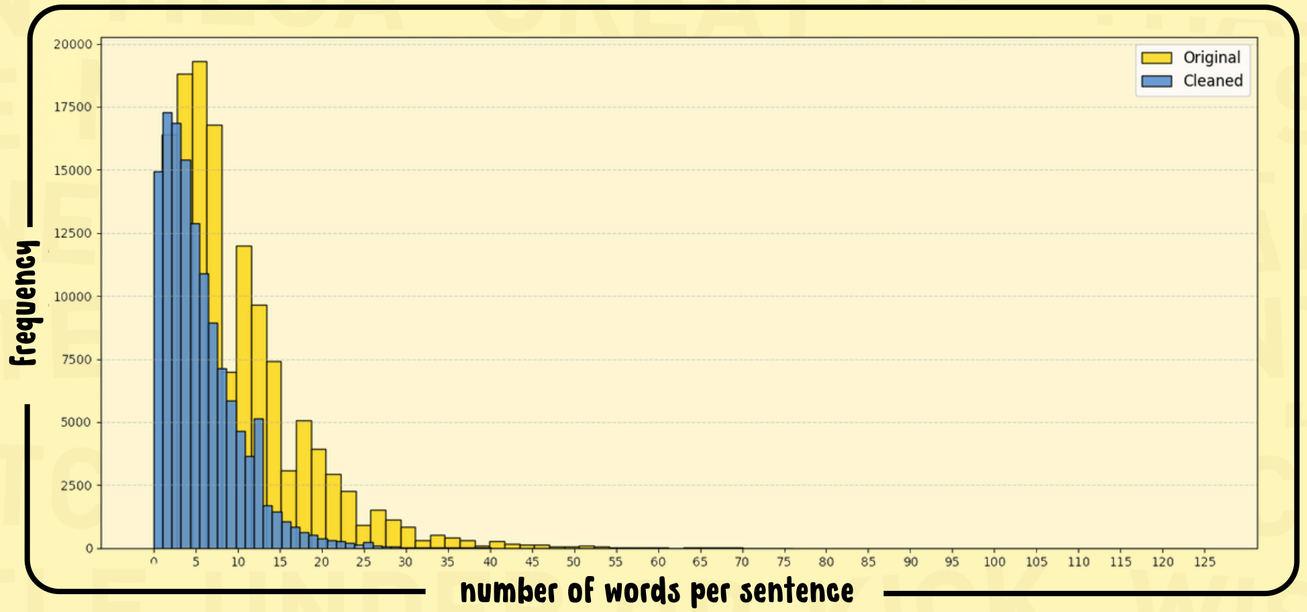




2.3. Dataset refinement and comparison

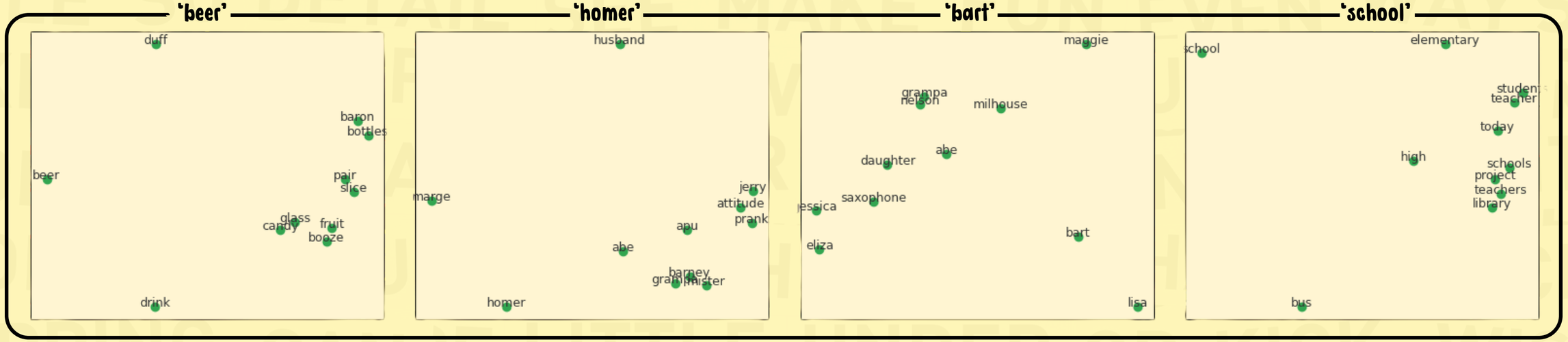
After completing the *preprocessing* part, we compared the original and cleaned *datasets*. As shown in this graph, **the total word count dropped from *1.3 million* in the original dataset to less than *800k* in the cleaned version**: a significant reduction that highlights the impact of *normalization* and *lemmatization*. We also observed a shift in the distribution of sentence lengths post-cleaning, as shown in the figure below:





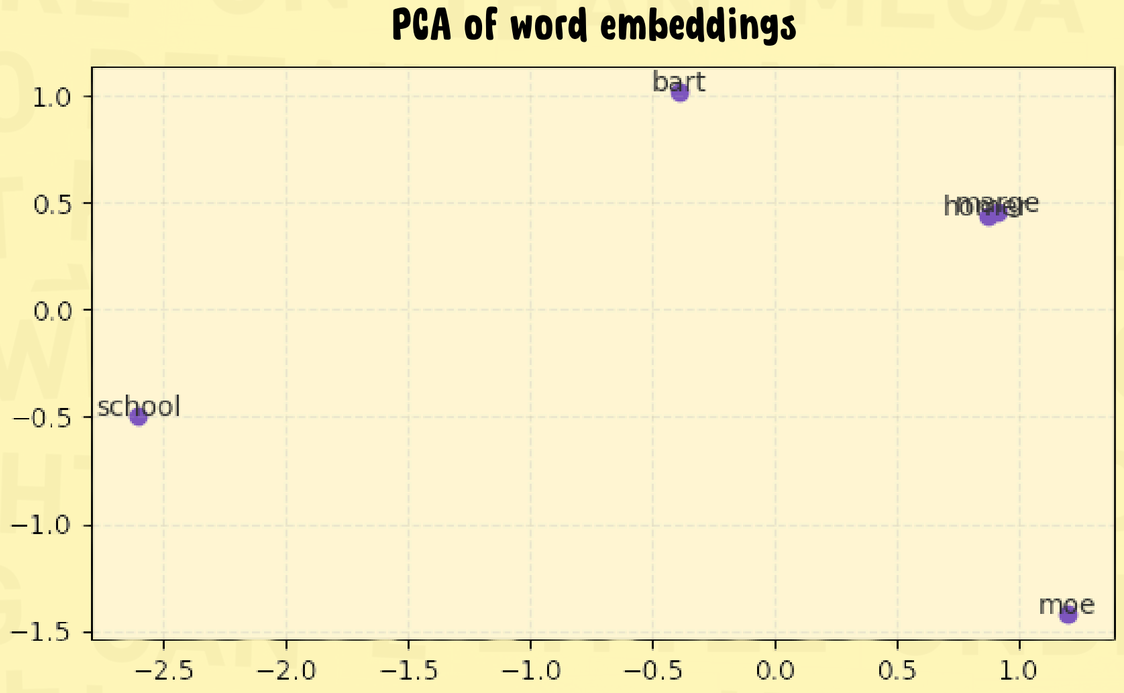
3. Text representation & Topic modeling

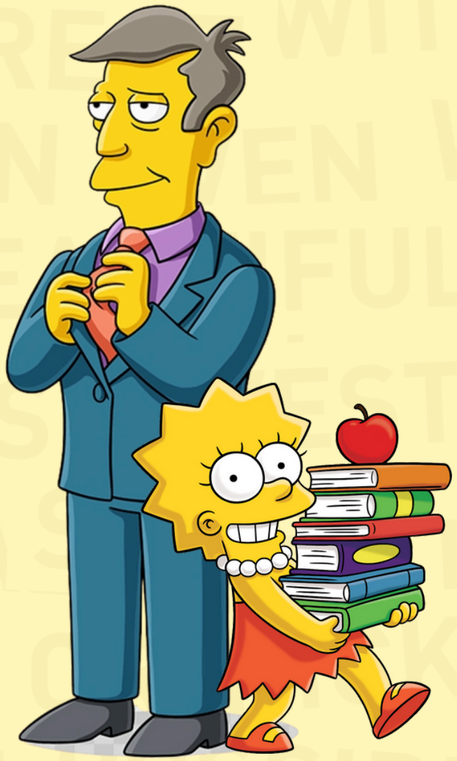
Our goal was to explore multiple methods of representing and modeling the dialogues in order to extract semantic patterns and latent structures. We adopted both *statistical* and *embedding-based techniques*, covering traditional *vectorization*, *word embeddings*, and *unsupervised topic modeling*. We trained a custom *Word2Vec* model using the *Skip-gram* architecture. This model was trained on the cleaned dialogues using a window size of 5, vector dimension of *100*, and including low frequency words to extract specific lexical signals. We computed word similarity among commonly used *tokens*.



3.1. Word Embedding with Word2Vec

To further explore the spatial relationships between selected tokens, **we applied *Principal Component Analysis (PCA)*** to reduce the *100* dimensional vectors to *2D* space for visualization of the four previously chosen words.





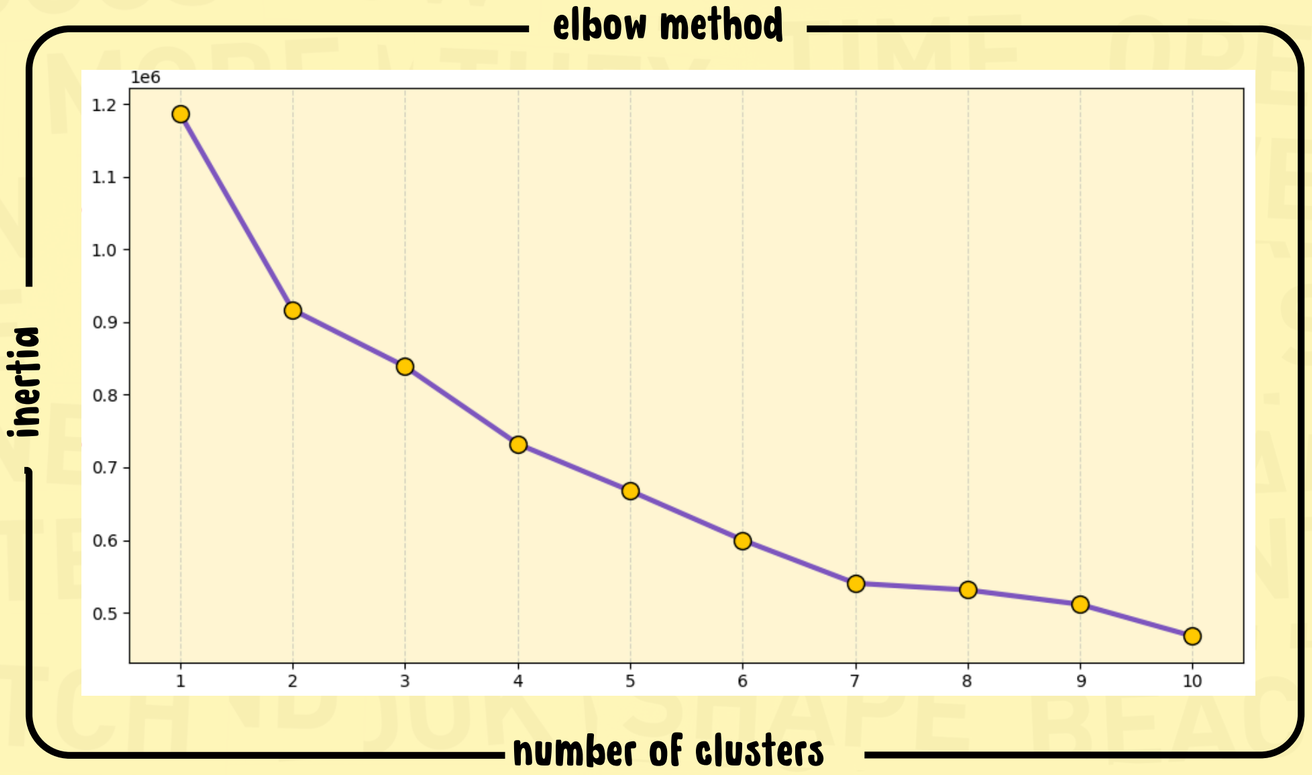
3.3 TF-IDF Vectorization and WordClouds

In addition to *word embeddings,* **we applied the *TF-IDF vectorizer* to obtain sparse textual representations that reflect word importance relative to the entire corpus**. This method was particularly useful for later tasks such as topic modeling and clustering. We also visualized the lexical distribution across main characters by generating ***word clouds***. This helped provide an intuitive understanding of each character's most frequently used words.

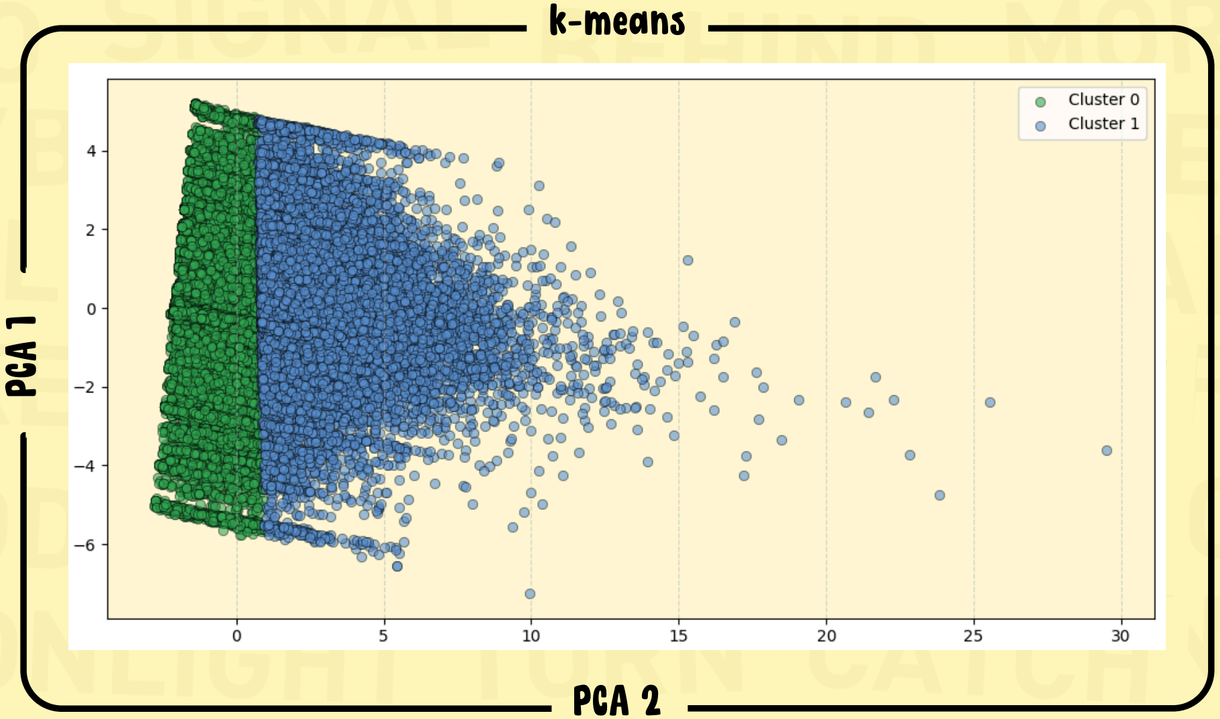
Homer **Marge** **Lisa** **Bart**

4.1. Text clustering and sentment classification

We conducted a comprehensive *clustering* and *sentiment analysis* on the dataset, enriched with various extracted features. We first standardized the features using *StandardScaler* and applied the *Elbow Method* to determine the optimal number of clusters.
K-Means clustering was then applied with *k = 2*. To visualize the results, *Principal Component Analysis (PCA)* was used to reduce the dimensionality to two components.



4.2. K-Means clustering



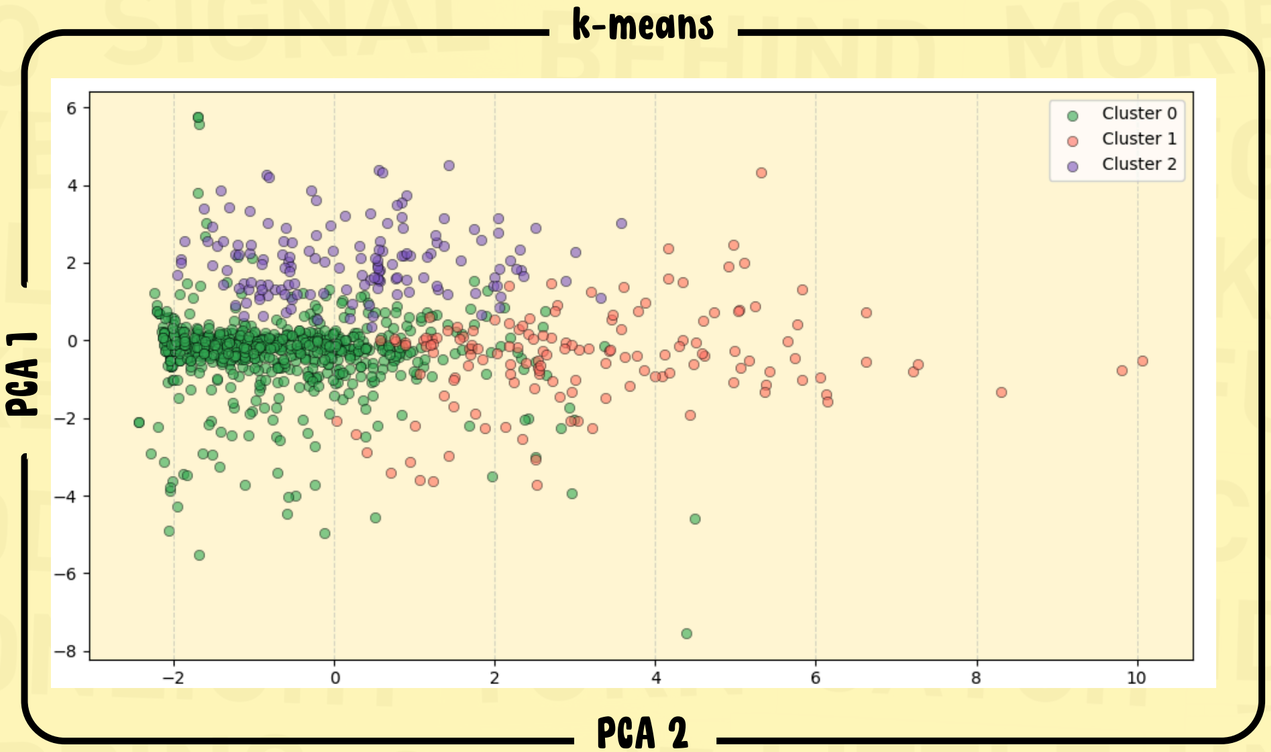
• ***Cluster 0***: Tightly packed, likely representing repetitive or stylistically **similar sentences.**
• ***Cluster 1***: More dispersed, suggesting greater **linguistic variability.**

For both clusters, we performed *topic modeling* using *Latent Dirichlet Allocation (LDA)* to understand the dominant themes.

Topics for Cluster 0:
• ***Topic 1***: *really, ll, mom, let, just, lisa, marge, like, right, bart*
• ***Topic 2***: *dad, like, little, homer, good, man, yes, okay, uh, oh*
• ***Topic 3***: *homer, did, just, got, ve, ll, yeah, know, hey, don*

Topics for Cluster 1:
• ***Topic 1***: *gonna, right, ve, oh, want, like, know, just, ll, don*
• ***Topic 2***: *guy, new, homer, oh, got, ll, know, like, ve, just*
• ***Topic 3***: *going, hey, good, okay, man, just, right, ve, like, oh*

4.3. Agglomerative clustering



• ***Cluster 0*** contains mostly **full sentences that are grammatically complete** and include narrative or descriptive content. These sentences are relatively long, with higher lexical diversity and more complex syntax.
• ***Cluster 1*** comprises **shorter utterances** that are often emotional, humorous, or reactive in nature.
• ***Cluster 2*** appears to isolate **anomalous or marginal sentences.** This cluster is significantly smaller and more homogeneous, suggesting the model identified it as outlier content.

These observations indicate that *Agglomerative Clustering* was effective in capturing structural and functional differences in sentence types
Unlike *K-Means*, which tended to mix short and long exclamations into broader groups, *Agglomerative Clustering* provided more complex separation, especially for extreme or outlier cases.

4.4. Clustering evaluation

KMeans Clustering	Agglomerative Clustering
Adjusted Rand Index (ARI): 0.204	— (ARI calculated against KMeans)
Normalized Mutual Information (NMI): 0.289	— (NMI calculated against KMeans)
Silhouette Score: 0.313	Silhouette Score: 0.153
Davies-Bouldin Index: 1.507	Davies-Bouldin Index: 1.886
Dunn Index: 0.019	Dunn Index: 0.022

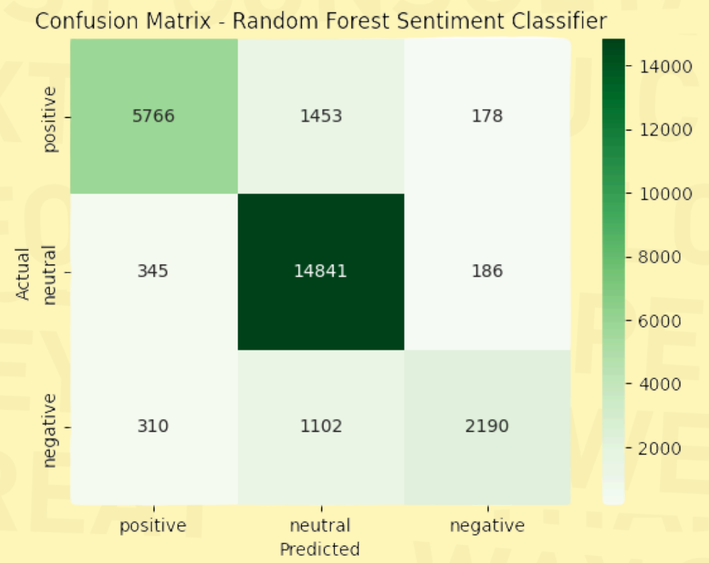
The comparison between *KMeans* and *Agglomerative Clustering* reveals distinct strengths for each method. ***KMeans* achieves higher *Silhouette* and *Davies-Bouldin* scores**, indicating more cohesive and well-separated clusters overall. Conversely, ***Agglomerative Clustering* scores slightly better on the *Dunn Index***, suggesting it is better at identifying extreme or distant clusters.
Both methods consistently identify three main types of sentence groups:

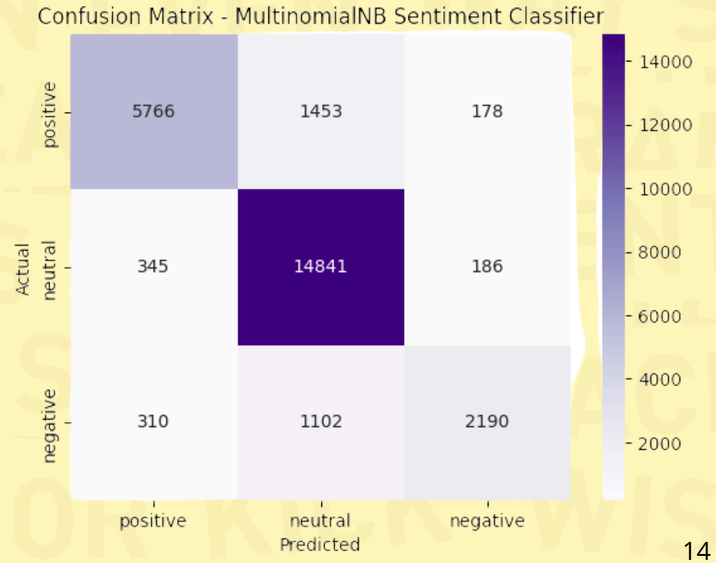
- narrative or elaborate sentences
- short and exclamatory or humorous sentences
- anomalous or non-informative sentences

While *KMeans* provides a more effective global segmentation, *Agglomerative* excels at isolating stylistically cohesive edge cases.
These complementary strengths highlight the value of applying multiple clustering techniques to uncover different linguistic patterns in the *dataset*.

4.4. Sentiment classification

The dataset's sentences were labeled with three sentiment classes: *neutral, positive, and negative*. *Neutral* sentences dominate the corpus, followed by *positive* and then *negative*. **We trained a *Multinomial Naive Bayes classifier* using *TF-IDF* features**, achieving an overall *accuracy* of *81%* on a test set of over *26,000 samples*. The model performs well on neutral and positive sentiments, with *precision* scores of *0.78* and *0.91* respectively, but struggles with the negative class, showing lower *recall (0.39)* and *F1-score (0.55)*. **A *Random Forest classifier* was also tested, confirming the general trend**. These results highlight the challenge of accurately detecting negative sentiment, yet demonstrate the robustness of the classification approach through multiple models and evaluation metrics.





Alessandro Bosi
Martina Pantò

THANK YOU