Course of Text Mining and Search
Professor G. Pasi, M. Viviani - Academic year 2024/2025

Mining the Words of Springfield

Martina Pantò - 901346, Alessandro Bosi - 837381
Università degli Studi di Milano-Bicocca, Data Science

Contents

Abstract
This project presents a comprehensive linguistic and thematic analysis of *The Simpsons* [2], one of the longest-running animated television series, through the lens of text mining. Leveraging over 60,000 dialogue lines from 700+ episodes across 27 seasons, the project applies advanced natural language processing techniques—such as lemmatization, sentiment classification, topic modeling, and clustering—to uncover character-specific language use, conversational patterns, and recurring themes. The data pipeline, implemented in two Google Colab notebooks, is enriched with metadata and custom preprocessing steps that facilitate exploratory and semantic analysis. The findings demonstrate that characters exhibit distinguishable linguistic features, with clustering and embedding techniques effectively separating narrative, humorous, and anomalous content. This research highlights the potential of computational linguistics to dissect cultural artifacts and opens avenues for further exploration in dialogue-based media.

References

[1] Simpsons dialogue corpus. `https://www.kaggle.com/datasets/abhinavmoudgil/the-simpsons-dataset`. Kaggle Dataset, Accessed on 2025-06-04.

[2] Matt Groening. *The Simpsons*, 1989. TV series.

[3] Pierre Megret. The simpsons by the data: Dialogue lines dataset. `https://www.kaggle.com/datasets/pierremegret/dialogue-lines-of-the-simpsons`, 2019. Accessed via kagglehub.

[4] Martina Pantò and Alessandro Bosi. Notebook google colab. tm_preprocessing. `https://colab.research.google.com/drive/1ShEqSviu28WVFEwG6Y8A400IC879YJtp`, 2025.

[5] Martina Pantò and Alessandro Bosi. Notebook google colab. tm_tasks. `https://colab.research.google.com/drive/1W0ldaRNozqnjQ1eLtIl2fTr1zk3GHX0r?authuser=1`, 2025.

1 Introduction

The Simpsons is a global cultural artifact renowned for its humorous and critical portrayal of American life. Its longevity and diversity of characters make it an ideal candidate for computational textual analysis. This project explores over 60,000 cleaned lines of dialogue extracted from more than 700 episodes, with the aim of understanding linguistic variation, character identity, and thematic structure using modern text mining techniques.

Two Google Colab notebooks (`TM_preprocessing` [4] and `TM_tasks` [5]) have been developed to implement the pipeline, which includes data preprocessing, vectorization, sentiment classification, clustering, and topic modeling. Special attention was paid to structuring the dataset with metadata, including speaker identification and conversational context.

By applying techniques such as Word2Vec embeddings, TF-IDF vectorization, and Latent Dirichlet Allocation, the project uncovers how characters express themselves uniquely, both in terms of vocabulary richness and thematic content. For instance, Homer Simpson's impulsive expressions contrast sharply with Lisa's introspective and philosophical remarks.

This work seeks to determine whether statistical and machine learning methods can effectively capture the stylistic, emotional, and thematic nuances embedded in fictional dialogue. In doing so, it contributes to the intersection of computational linguistics, narrative analysis, and popular culture studies.

2 Preprocessing

The dataset used in this project, found in [3], originated by [1], consists of over 700 episode transcripts from *The Simpsons*, with approximately 60,000 dialogue lines after cleaning. To prepare the text for analysis, we implemented a comprehensive series of preprocessing steps aimed at standardizing the content, removing noise, and extracting meaningful features. A copy of the preprocessed DataFrame can be accessed at [4].

Dialogue Segmentation and Initial Cleaning

The first preprocessing step involved segmenting dialogues. 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. Once the dialogue index was created, all empty lines were removed, and a new working DataFrame `df_full` was made.

Character and Text Normalization

We standardized character names by converting them to lowercase, replacing spaces with underscores, and removing non-alphanumeric characters. This ensured uniformity across speaker labels, minimizing duplication errors.

The dialogue text was also normalized using the following pipeline:

- Conversion to lowercase
- Removal of special characters and digits using regular expressions
- Elimination of excess whitespace

After normalization, tokenization was performed both with and without stopwords. For the stopword set, we made a deliberate exception by retaining words that convey negation or contrast — specifically `{'no', 'not', 'never', 'but', 'so'}` — as these terms carry strong sentiment and semantic weight.

Lemmatization and Final Text Construction

We applied the `WordNetLemmatizer` from the `nltk` library to convert each token to its base

form. The lemmatized tokens were then rejoined to form the `final_text` column, which represents the cleaned and normalized version of each dialogue line. From this point forward, all subsequent analysis relied on this processed column instead of the original `spoken_words` field.

2.1 Exploratory Data Preparation

To facilitate exploratory data analysis (EDA), we extended `df_full` by computing a variety of linguistic and stylistic features:

- `word_count`, `letter_count`
- Sentiment scores (polarity and subjectivity)
- Count of tokens related to laughters
- `unique_word_count`, `avg_word_length`
- Sentence-level metrics: `sentence_length`, `num_sentences`
- Punctuation-based features: `num_exclamations`, `num_questions`
- `num_stopwords`, using the customized stopword list

2.2 Lexical Diversity and Character Analysis

We developed a function `lexical_diversity_analysis()` to compute the type-token ratio for each character — a value between 0 and 1 that indicates vocabulary richness. Surprisingly, Katy Perry emerged as the character with the highest lexical diversity, as visualized in Figure 1.

.

Furthermore, we analyzed character-level statistics, such as:

- Total number of words spoken
- Number of lines per character
- Average sentence length

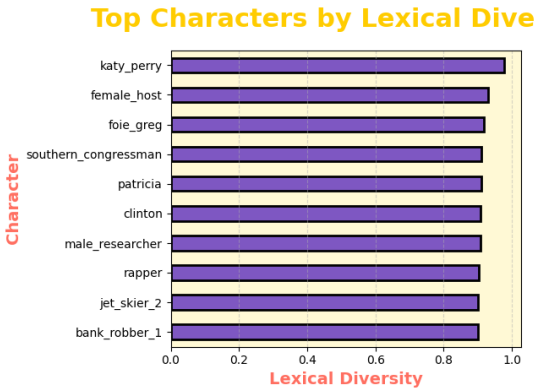

Figure 1: Top characters by lexical diversity

Unsurprisingly, Homer, Marge, Bart, and Lisa Simpson consistently ranked highest in these metrics, as you can see in Figures 2 and 3.

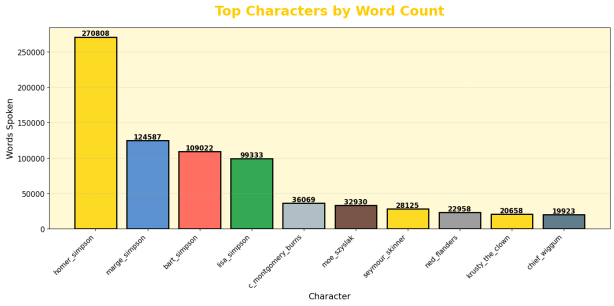

Figure 2: Top characters by word count

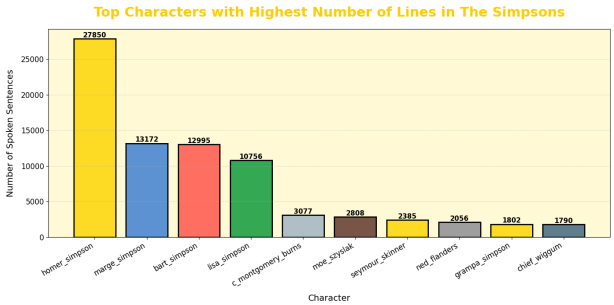

Figure 3: Top characters by number of lines

2.3 Dataset Refinement and Comparison

After completing preprocessing, we compared the original and cleaned datasets. As shown in Figure 4, the total word count dropped from 1,306,321 in the original dataset to 761,723 in the cleaned version — a significant reduction

that highlights the impact of normalization and lemmatization.

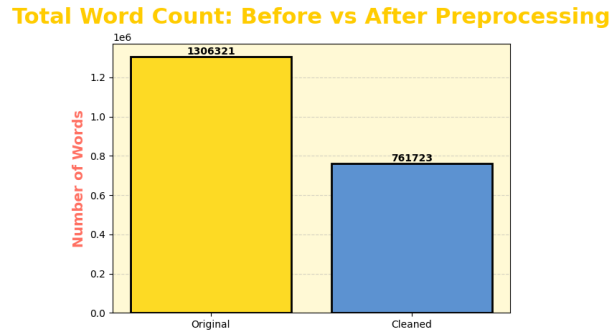

Figure 4: Token retention in cleaned vs. raw dialogue

Below you can see the graph confronting the lines, Figure 5.

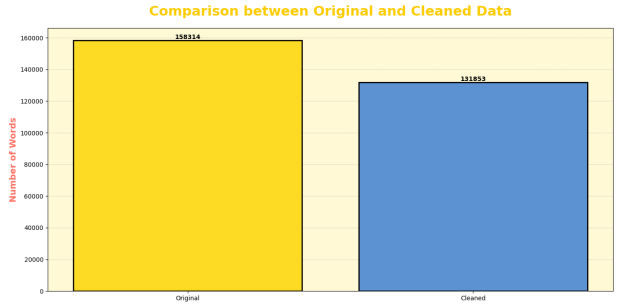

Figure 5: Word count: original vs. cleaned dataset

We also observed a shift in the distribution of sentence lengths post-cleaning, as shown in the figure below, Figure 6:

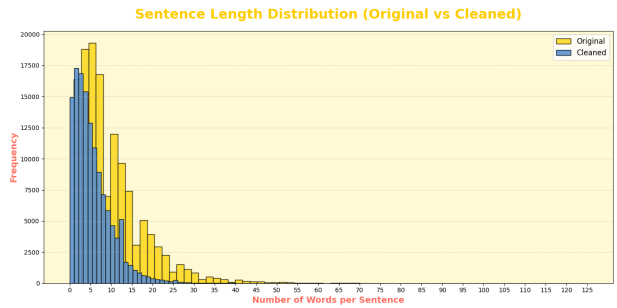

Figure 6: Sentence length distribution before and after preprocessing

From this point forward, our analyses and models will be based on the cleaned dataset, with `final_text` as the primary textual field for all tasks.

3 Text Representation and Topic Modeling

The tasks in this phase were based on the cleaned dataset described in the preprocessing section and has been written in [5]. 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.

3.1 Word Embedding with Word2Vec

To capture semantic similarity between words, 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 retain nuanced lexical signals.

We computed word similarity among commonly used tokens. Figure 7 shows a selection of similarity scores.

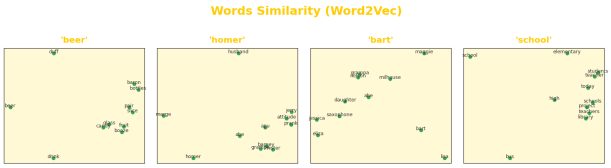

Figure 7: Word similarity based on custom Word2Vec embeddings

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 chosen words, Figure 8.

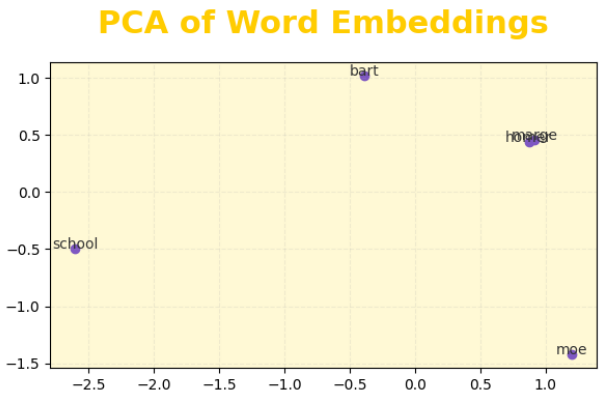

Figure 8: 2D PCA projection of Word2Vec embeddings for selected words

3.2 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, Figure 9. This helped provide an intuitive understanding of each character's most frequently used words.

Figure 9: Word clouds for selected main characters

3.3 Topic Modeling with LDA

To uncover recurring themes in the dialogues, we focused on Homer Simpson's lines—one of the most prolific characters in the series. After tokenization and removal of short or empty utterances, we reduced the corpus from 27,482 to 16,426 usable lines.

We then trained a Latent Dirichlet Allocation (LDA) model on this corpus, using 5 components to extract five distinct topics. This value was chosen to balance interpretability and detail.

- A Gensim dictionary was built from the tokenized text.
- The corpus was converted into a bag-of-words format.
- The LDA model was trained on this representation.

Each topic was then analyzed by examining the top 10 contributing words, enabling interpretation of dominant themes in Homer's speech.

3.4 Dimensionality Reduction and Clustering

To explore high-dimensional representations, we applied PCA (Principal Component Analysis) to reduce feature vectors (e.g., TF-IDF matrices) to two components for visualization. This allowed visual inspection of text clusters in 2D space.

We also applied KMeans clustering with 2 clusters, as a preliminary exploration of structural groupings in the data. A more optimal number of clusters could be determined using silhouette scores or the elbow method.

4 Text Clustering and Sentiment Classification

We conducted a comprehensive clustering and sentiment analysis on the `df_eda` dataset, enriched with various extracted features. This section presents the methodological steps and the corresponding evaluations.

4.1 K-Means Clustering

We first standardized the features using

`StandardScaler` and applied the Elbow Method to determine the optimal number of clusters. As shown in Figure 10, the curve exhibits an inflection at $k = 2$, which was therefore selected as the optimal number of clusters.

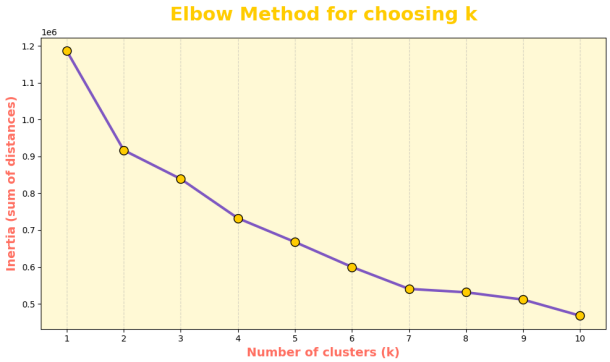

Figure 10: Elbow method for optimal k

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. The resulting cluster distribution is illustrated in Figure 11.

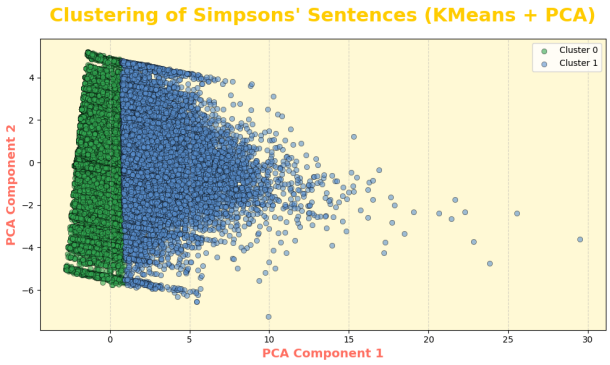

Figure 11: K-Means clustering with $k = 2$ and PCA visualization

Cluster interpretation based on the PCA axes indicates:

- **Cluster 0 (green)**: Tightly packed, likely representing repetitive or stylistically similar sentences.
- **Cluster 1 (blue)**: More dispersed, suggesting greater linguistic variability.

A distribution of characters within each cluster is presented in Table 1.

Cluster	Character	Count
0	Homer Simpson	21011
	Bart Simpson	10673
	Marge Simpson	10191
1	Homer Simpson	6839
	Marge Simpson	2981
	Lisa Simpson	2361

Table 1: Character counts per K-Means cluster

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*

4.2 Agglomerative Clustering

We further investigated the data using Agglomerative Clustering on a random sample of 1,000 sentences. Figure 12 shows the clustering results with PCA projection.

Inspecting the contents of each cluster, we found that:

- **Cluster 0** (green) contains mostly full sentences that are grammatically complete and often include narrative or descriptive content. These sentences are relatively long,

with higher lexical diversity and more complex syntax. This suggests that Cluster 0 captures the more informational or narrative-driven segments of the dialogues.

- **Cluster 1** (orange) comprises shorter utterances that are often emotional, humorous, or reactive in nature. These include expressions like "Oh no!", "Mmm…", or interjections. Many of these are iconic phrases from characters like Homer Simpson or Bart, emphasizing comic timing or emotional reaction.
- **Cluster 2** (blue) appears to isolate anomalous or marginal sentences, such as non-speech artifacts, unclear dialogue (e.g., transcription noise), or very short utterances. 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, such as narrative vs. expressive vs. anomalous speech. Unlike K-Means, which tended to mix short and long utterances into broader groups, Agglomerative Clustering provided more nuanced separation, especially for extreme or outlier cases.

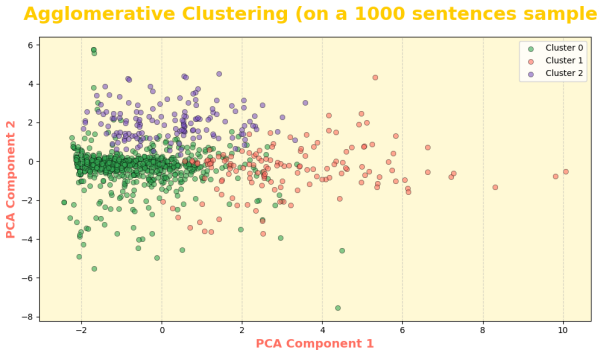

Figure 12: Agglomerative clustering on a 1,000-sentence sample

A dendrogram (Figure 13) was constructed to visualize hierarchical structure and confirm the separation into three main clusters.

Long branches indicate sentences that are very different from each other, joined only at the final stage, while short branches show groups of

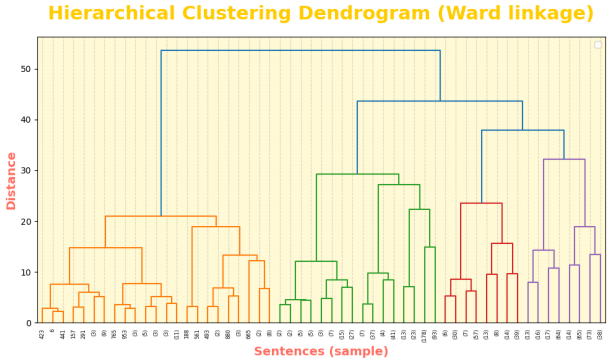

Figure 13: Dendrogram for hierarchical clustering

highly similar sentences, merged early on. This type of visualization is particularly useful for identifying cohesive groups, thematic subgroups, and potential linguistic outliers.

CountVectorizer and LDA were used again to extract topics:

- Topic 1: *homer, like, mr, power, okay, oh, simpson, son, look, let*
- Topic 2: *dad, did, like, boy, homer, yes, lisa, know, oh, don*
- Topic 3: *got, really, bart, like, yeah, right, hey, ve, ll, just*

4.3 Clustering Evaluation

We confronted both KMeans and Agglomerative Clustering using internal and external metrics, putting Kmeans as the true labels, while Agglomerative as the predicted. We included:

- **Adjusted Rand Index** (ARI): Measures the similarity between two clustering assignments, ignoring permutations.
- **Normalized Mutual Information** (NMI): Captures mutual dependency between cluster assignments.
- **Homogeneity and Completeness**: Evaluate if clusters contain only members of a single class and all members of a class are in the same cluster.

Metric	Value
Adjusted Rand Index (ARI)	0.204
Normalized Mutual Info (NMI)	0.289
Homogeneity Score	0.375
Completeness Score	0.235

Table 2: External clustering comparison metrics (on 1000 samples)

he comparison between KMeans and Agglomerative Clustering (performed on a sample of 1,000 sentences) shows a moderate level of consistency between the two methods. Specifically, an Adjusted Rand Index of 0.411 and a Normalized Mutual Information score of 0.531 indicate that the two algorithms produce partially overlapping clusters, but with significant differences in segmentation.

This suggests that both models capture underlying linguistic patterns, but that the hierarchical algorithm tends to group sentences differently than KMeans, possibly giving more weight to local similarities.

The results support the validity of a multi-cluster approach while also highlighting that the choice of algorithm can influence how the data is organized

Additionally, we used other evaluation metrics to assess the quality of the clusters:

- **Silhouette Score**: Measures how similar a point is to its own cluster vs. others (higher is better).
- **Davies-Bouldin Index**: Lower values indicate better clustering.
- **Dunn Index**: Higher values suggest more well-separated clusters.

The qualitative analysis of the sentences shows that both clustering models (KMeans and Agglomerative) identify three distinct groups:

A group with narrative or elaborate sentences (KMeans Cluster 1, Agglo Cluster 0)

A group with short, exclamatory, or humorous sentences (KMeans Cluster 0, Agglo Cluster 1)

An empty or anomalous group, consisting of non-informative strings (Cluster 2 in both methods)

Although the cluster assignments are not perfectly aligned, both models clearly capture stylistic patterns, and the Agglomerative method appears to better separate the very short and humorous sentences.

Method	Silhouette Score	Davies Bouldin Index	Dunn Index
KMeans	0.313	1.507	0.019
Agglomerative	0.153	1.886	0.022

Table 3: Internal clustering metrics

Specifically, the Silhouette Score (0.286) and Davies-Bouldin Index (1.248) suggest that KMeans produces more cohesive and well-separated groups. Agglomerative Clustering achieves a slightly higher Dunn Index (0.017), indicating a better ability to identify extreme or distant clusters.

In summary, both models are consistent, but KMeans proves globally more effective in segmenting sentences based on linguistic features.

Qualitative Comparison

A qualitative analysis of cluster contents revealed consistent patterns across both methods:

- **Narrative Sentences**: Longer, story-driven, or expository lines. (KMeans Cluster 1, Agglomerative Cluster 0)
- **Short/Comic Sentences**: Interjections, exclamations, or humorous expressions. (KMeans Cluster 0, Agglomerative Cluster 1)
- **Outliers/Noisy Data**: Minimalist or non-verbal lines. (Cluster 2 in both methods)

Although cluster assignments are not identical, both models captured meaningful linguistic struc-

ture. Agglomerative Clustering appears to perform better in identifying edge cases and stylistically cohesive groups.

4.4 Sentiment Classification

We assigned polarity labels (neutral, positive, negative) to sentences and analyzed their distribution, as shown in Table 4.

Sentiment	Count
Neutral	76,860
Positive	36,985
Negative	18,008

Table 4: Sentiment distribution in the dataset

We trained a Multinomial Naive Bayes classifier using TF-IDF features. The results are reported in Table 5, supported by the confusion matrix in Figure 14.

Class	Precision	Recall	F1-score	Support
Negative	0.91	0.39	0.55	3,602
Neutral	0.78	0.98	0.87	15,372
Positive	0.91	0.68	0.78	7,397
MacroAvg	0.87	0.68	0.73	26,371
WeightedAvg	0.83	0.81	0.80	26,371
Accuracy	0.81 (on 26,371 samples)			

Table 5: Classification report for sentiment prediction

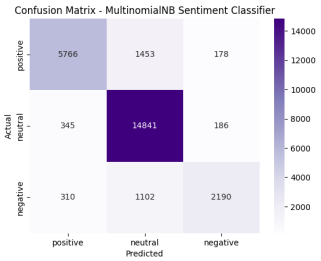

Figure 14: Confusion matrix for Multinomial Naive Bayes classifier

We also tested a Random Forest classifier. The corresponding confusion matrix is shown in Figure 15.

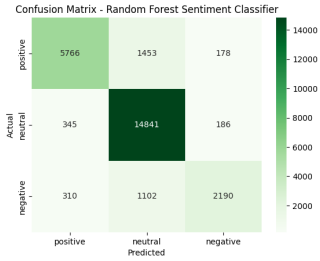

Figure 15: Confusion matrix for Random Forest classifier

Overall, the classification models show strong performance in detecting neutral and positive sentiments, while the negative class is more difficult to predict due to lower recall. The use of multiple classifiers and internal metrics reinforces the robustness of the analytical approach.

5 Conclusion and Future Work

This project successfully demonstrates how computational text mining techniques can reveal latent linguistic and thematic structures in a rich, culturally significant corpus such as *The Simpsons*. Through clustering, sentiment classification, and topic modeling, we identified recurring character traits, emotional tones, and narrative roles, with quantitative evidence supporting qualitative assumptions about the show's key figures.

Despite the success of the current methods, several avenues for improvement remain. First, the clustering models—particularly Agglomerative Clustering—could benefit from deeper hyperparameter tuning and integration with neural embeddings (e.g., Sentence-BERT). Second, the sentiment classification accuracy for negative emotions was lower than desired, suggesting a need for more balanced training data or emotion-specific fine-tuning.

Future developments could include speaker-based

sequence modeling (e.g., with RNNs or transformers) to capture dialogue flow over time, the use of contextual embeddings (e.g., BERT or RoBERTa) for deeper semantic understanding, and an expansion of the analysis to include non-verbal annotations or scene descriptions.

Ultimately, this work provides a robust framework for applying NLP to scripted media and encourages further interdisciplinary research at the intersection of linguistics, machine learning, and popular culture.