uc3m | Universidad Carlos III de Madrid

Machine learning applications 23/24-S2

Group 96

Report Final Project

Forum Enhancement

Made by:

Sara Piñas García (100472784)

Ángela Durán Pinto (100472766)

Marina Gómez Rey (100472836)

María Ángeles Magro Garrote (100472867)

Index

Introduction

Objective

In today's digital era, user interaction among different formats of platforms is paramount, and the enhancement of their experience plays a crucial role in order to obtain more engagement from them. This is where the objective of our project lays on; a diverse group of people converge to interact and exchange ideas in virtual communities, however, this volume and heterogeneity of forum data present significant challenges in organizing and extracting meaningful insights from these discussions. We look for an address of this problem by the use of different data analysis techniques to enhance the functionality and usability of a forum website, as well as to add a more visual aesthetic to the page, i.e., by the use of emojis, which convey emotions and sentiments that might be difficult to express with text alone. Our overarching objective is to leverage clustering, sentiment analysis, topic modeling, and recommendation systems to refine the forum dataset, thereby enriching the user experience and fostering deeper engagement within the online community.

Datasets

Two primary datasets were utilized in this project:

1. **Forum Dataset**: The forum dataset was obtained through web scraping from a popular online forum. The dataset exhibits characteristics typical of user-generated content, including variability in language, sentiment, and thematic content, however, no labels are given. Despite its richness and diversity, the dataset also poses inherent challenges, such as noise, ambiguity, and heterogeneity, which necessitate sophisticated analytical approaches for effective management and organization. The features in this dataset are the *post title* and the *post text*.

2. **Emojis Dataset**: This dataset was obtained from Kaggle (https://www.kaggle.com/datasets/subinium/emojiimage-dataset) and contains a comprehensive collection of *emojis* along with their *descriptions*.

Furthermore, other datasets were needed in order to complete the whole project:

1. **NRC Lexicon Dataset and Vader lexicon**: We'll utilize two lexicons for sentiment analysis: the NRC Lexicon Dataset and the Vader lexicon. These lexicons annotate words with associated emotions, in our case, in English. Each word is tagged with one or more emotions, providing insight into its emotional context or connotation. The emotions covered by NRC include anger, anticipation, disgust, fear, joy, sadness, surprise, trust, and disgust. Additionally, the Vader lexicon categorizes words into positive, negative, and neutral sentiments.

2. **Glove Dataset:** Global Vectors for Word Representation (GloVe).

Web Scraping for Data Collection

To collect data for this project, we used web scraping to extract content from an online forum. After evaluating several forums, we chose https://forums.somethingawful.com for its wide range of topics and compatibility with web scraping techniques. The scraping process used Python's BeautifulSoup library to parse HTML, with the results stored in a CSV file for further analysis.

We focused on extracting titles and texts from forum posts. Understanding the forum's HTML structure was key to this process. Titles were found within anchor (<a>) elements with specific classes, while the text content was extracted from table data (<td>) elements containing the body of each post. To scrape across multiple pages, we defined a base URL and dynamically generated the URLs for subsequent pages. This allowed us to collect data from various sections of the forum.

To avoid overloading the server, we implemented a loop with a delay between requests. This ensured that we could scrape without causing server issues or getting blocked.

With this approach, we scraped enough data to create a comprehensive dataset of 25,000 rows, containing post titles and texts. This dataset served as the foundation for further analysis, providing a rich source of content for our project. The data was saved in a CSV file for easy access and processing in later stages.

Task 1: Text Preprocessing and Vectorization

As previously noted, the **forum dataset** used for this project is notably **diverse** and **unorganized**, mirroring a perfect real-world scenario. Thus, effective text preprocessing and vectorization are essential for meaningful analysis and progress. These preprocessing steps are intended to convert raw text into a more organized and consistent format, providing a solid groundwork for subsequent vectorization and analysis endeavors.

Preprocessing pipeline used for text data

Firstly, in this preprocessing step, when performing the latter experiments, some **obscene words** were encountered; to maintain formality and suitability for all readers, entries containing obscene words were **eliminated** from the dataset before working with it so that the results were not influenced by those words.

For the preprocessing pipeline, several steps were employed to clean and prepare the both datasets for further analysis:

1. **HTML Tag Removal:** Since the forum is obtained from the internet, it is no surprise that some html tags will be encountered so with the use of BeautifulSoup library and the ‘lxml’ parser, these will be eliminated from the text data.
2. **URL Removal:** Also URLs will be removed as well since no important information is provided from them.
3. **Contraction Expansion:** Using the contractions library ‘contractions’, contraction words will be expanded and consistency will be kept.
4. **Number Removal:** We have decided that numbers did not give us any important insights when performing the experiment, so they were removed with the same library as the URL tags, i.e., the regular expression operations one.
5. **Tokenization:** Using ‘wordpunct_tokenize’, text is converted into tokens.
6. **Homogenization and Lemmatization:** Once the tokens are obtained, lemmatization is performed with the WordNetLemmatizer, once the words are converted to lowercase. Here an additional function called ‘get_wordnet_pos’ has been used to map each token’s part-of-speech tag to the format accepted by the lemmatizer, this function has been added because when using the lemmatizer by its own, some words were not fully lemmatized.
7. **Stopwords Removal:** Since stopwords do not carry significant meaning, they can be eliminated.
8. **Word Existence Check:** Since after all preprocessing, some words in other languages were kept, each token has been checked against the WordNet corpus so that tokens that are not valid English words are filtered out. On top of that, as it is a forum many non-existent words were present that could not be processed by the algorithms.

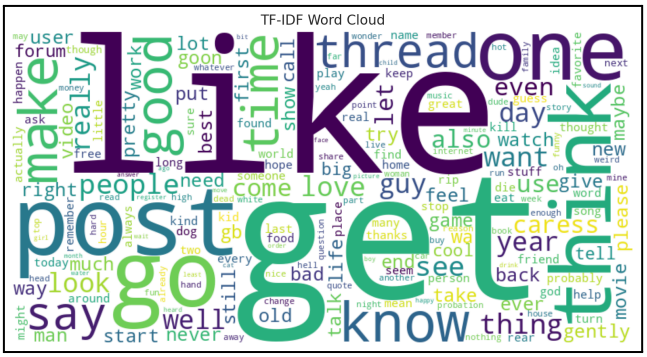
9. **Text Reassembly:** Finally, the already processed tokens are joined back into a single string, representing the cleaned and preprocessed version of the original text data, and a column with these is added to the forum dataset.

All these previous steps were integrated into a pipeline so that the cleaning is done directly. After the function is called for both datasets, in the post text column of the forum dataset and the emoji description of the emojis dataset, duplicates and empty rows of the output datasets have been also checked and eliminated.

Vectorization strategies explored: TF-IDF, GloVe, or LDA.

Regarding the vectorization techniques, some strategies have been explored and must be noted that here in this part is were we realized some popular words included bad connotation that and subsequently decided to eliminate those in the preprocessing step:

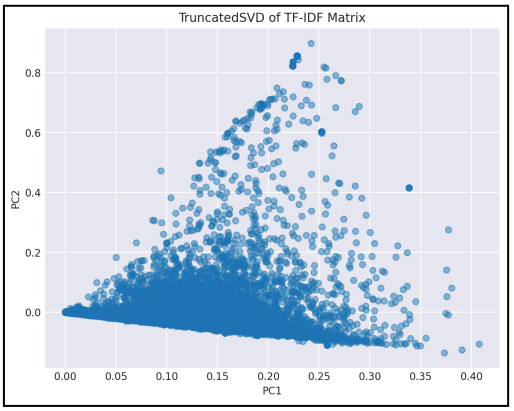
TF-IDF gives key insights calculating the importance of the words and their frequency across all documents. With the library TfidfVectorizer we have been able to obtain the words that have been repeated.



TF-IDF Word Cloud

This representation proved helpful for comprehending our TF-IDF scores. The biggest word in the vectorization word cloud corresponds to the highest score. It's worth noting that a high TF-IDF score indicates that the term is more distinctive for a particular document. This suggests that the terms are more unique to that document compared to others in the corpus.

Singular Value Decomposition (SVD) is applied to the TF-IDF representation to mitigate noise and facilitate a latent semantic representation of the data. This process is particularly beneficial when dealing with a large number of texts. By decomposing the TF-IDF matrix into its constituent parts, SVD helps uncover underlying patterns and relationships within the data, enabling more effective analysis and interpretation.



Truncated SVD of TF-IDF Matrix

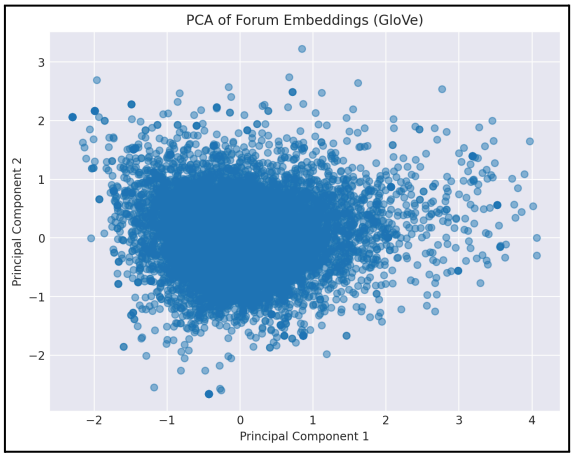
Glove representation has been also studied since it provides a dense vector representation of the words where the distance and direction between vectors encodes semantic relationships between words. Allowing to study co-occurrence. It also provides a lower dimensionality compared to the TF-IDF vector and encoding the similarity measures makes it suitable and a great candidate for performing clustering and the recommendation task.

For the GloVe-based representations, the conversion of a set of word vectors into a document vectorization involved the creation of a function '*topic_embeddings_glove*' which takes as inputs the tokens for each topic, and a pre-trained Glove word embedding mode 'glove-wiki-gigaword-300' , then the following steps were performed:

1. For each topic, it iterates through the tokens, extracts the corresponding word vectors from the GloVe model, and computes the mean of these vectors.
2. If a token is not found in the GloVe model, a random embedding is assigned.
3. The function returns a list of document embeddings (topic_embeddings), where each embedding represents a topic.

The model 'glove-wiki-gigaword-300' has been the one chosen among others due to its size and coverage because it is trained among a large corpus which will result in high-quality embeddings. In addition, its embeddings are likely to generalize well in a wide range of domains and topics; which in our heterogeneous data is most likely to happen.

For dimensionality reduction of the Glove vectorization, Principal Component Analysis (PCA) was applied to them. By projecting the high-dimensional GloVe embeddings onto a lower-dimensional space, PCA captures the most significant variations in the data. And with this step, we look to enhance the efficiency of the clustering by reducing computational complexity and noise in the data.

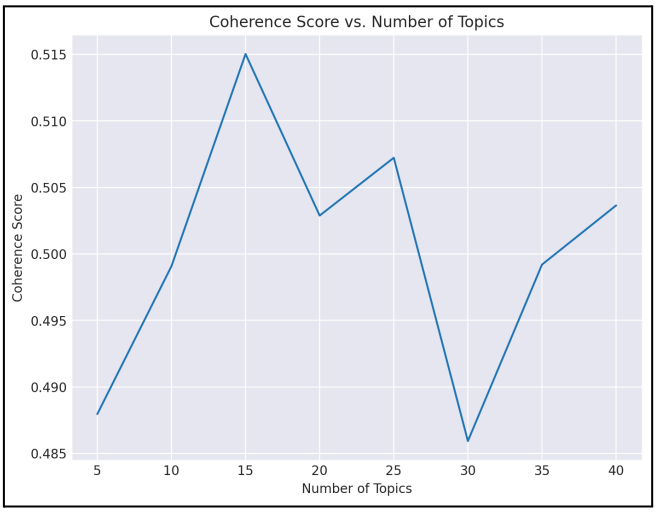


PCA on GloVe embeddings for Forum dataset

Topic Modeling with LDA (Mallet) has been finally used for the extraction of themes and the vector representation of documents using the LDA algorithm implemented by Mallet.

A list containing a different number of topics was used for testing and the best LDA Model has been chosen by their coherence scores. Finally a topic from the best model has been assigned to each forum using the *num_topics* generated by the model. The results that have been obtained are a model **15** topics with a coherence score of **0.51** approximately.

LDA is key in our project because it identifies underlying themes or topics present, without labels already given, and the impossibility of manually assigning some. Therefore, this step is key for the objective of the project.



LDA Coherence Score depending on the Number of Topics

Topic Interpretation: In the model creation, 10 words per topic were used. From them we have identified the topic subjects, and assign each forum post their corresponding label..

-**Crime**: Conversations about law enforcement and criminal activity. Keywords: "police," "gun," "kill."
-**Wholesome**: Positive, uplifting topics. Words like "love," "friend," "gently."
-**American Politics**: U.S. politics and government. Keywords: "president," "government," "america."
-**Body**: Themes related to the human body. Keywords: "face," "head," "fat."
-**Family and Loss**: Discussions about family and loss. Words like "wife," "baby," "death."
-**Movie**: Focus on films. Keywords: "movie," "film," "episode."
-**Entertainment**: Broader entertainment topics, including video games. Keywords: "game," "music," "book."
-**Time**: Topics on time and daily activities. Keywords: "day," "minute," "time."
-**Memories**: Themes of nostalgia and personal experiences. Keywords: "picture," "remember," "walk."
-**Work and Money**: Conversations about work and financial matters. Keywords: "money," "job," "pay."
-**Life**: Broader themes about life and human experiences. Keywords: "people," "life.
-**Social Media**: Discussions about social media. Keywords: "twitter," "message," "social."
-**Forums**: Focus on forum behavior and user interactions. Keywords: "user," "register," "forum."
-**Food**: Conversations about food and dining. Keywords: "eat," "food," "pizza", "dinner", "meat"
-**Humanity**: Broader cultural and human-related topics. Keywords: "human," "space," "planet."

Rationale behind choosing each vectorization method.

In terms of rationale behind our selections, it must be highlighted the following points:

The sparse representation of text data in **TF-IDF**, leveraging the frequency of word appearance, is particularly advantageous for our subsequent clustering experiment. By highlighting unique words or terms within each document, TF-IDF enables the identification of distinct clusters effectively. This emphasis on unique terms enhances the clustering process, leading to more meaningful and interpretable results.

GloVe embeddings capture semantic similarities between words, making them suitable for tasks like topic modeling and mapping of emojis and topics. By representing documents as the average of their word vectors, GloVe embeddings can effectively capture the underlying topics present in the text. In addition, for the experiment of topic modeling, different numbers of **LDA** topics have been studied and used together with GloVe in order to classify the forum entrances in different topics.

Additional techniques used: dimensionality reduction techniques.

TF-IDF representation yields a sparse matrix, therefore, its sparsity makes them well-suited for Singular Value Decomposition (**SVD**), while PCA is better suited for dense data. As a result, SVD has been chosen over PCA for TF-IDF matrices due to its ability to handle sparsity and capture the latent structure of the data more effectively.

On the other hand, **GloVe** embeddings result in dense vectors with 300 dimensions due to the model that we have chosen, providing a compact representation of semantic relationships between words. For this scenario, although our data may not be strictly linear, **PCA** is well-suited for capturing the principal components of the data.

Attempts were made to apply **kernel** methods to handle non-linearity, but the results were found to be similar to those obtained with PCA and SVD. Therefore, considering the computational complexity and similarity of outcomes, we decided to keep SVD for TF-IDF and PCA for GloVE.

Task 2: Machine Learning Model: Clustering

Clustering is a fundamental unsupervised technique in data analysis used to group similar data points together, allowing for a more structured understanding of complex datasets. In the context of our project, clustering is employed to group different posts from a forum, aiming to identify trends and common themes within these discussions. By categorizing the forum content into clusters, we can uncover patterns that may not be immediately apparent, helping to inform further analysis and user experience improvements.

The objective of clustering in this scenario is to create meaningful groupings of forum posts based on their textual content. These groupings will allow us to explore the variety of discussions occurring within the forum and identify overarching topics that connect different posts. This process can ultimately lead to a better understanding of the forum's structure and user behavior.

To achieve this objective, we leverage the vectorized representations of the forum posts computed in Task 1. By converting text into numerical vectors, we can apply clustering algorithms to group similar posts together, leading to insights about the types of content that users are discussing.

Clustering Algorithm

After exploring various clustering algorithms such as DBSCAN, BIRCH, and hierarchical clustering, we found that **K-means** provided the best **balance between performance and efficiency** for our dataset. DBSCAN offers the advantage of identifying clusters with arbitrary shapes and handling noise effectively, but its computational complexity can be high, especially with large datasets. BIRCH, known for its scalability, did not deliver the same level of precision in defining clusters. Hierarchical clustering, while offering a detailed structure of cluster relationships, tended to be computationally intensive. K-means, on the other hand, strikes an optimal balance by efficiently grouping data into well-defined clusters, making it suitable for large-scale text data. Its simplicity, scalability, and ease of implementation were significant factors in our selection, ensuring that we could process our forum dataset with high efficiency while maintaining meaningful clusters.

Optimal number of Clusters

Determining the optimal number of clusters is crucial for effective clustering. In our project, we conducted **hyperparameter tuning** to find the best number of clusters for K-means. This involved looping through a range of possible cluster numbers and evaluating their performance using two metrics: inertia and silhouette score.

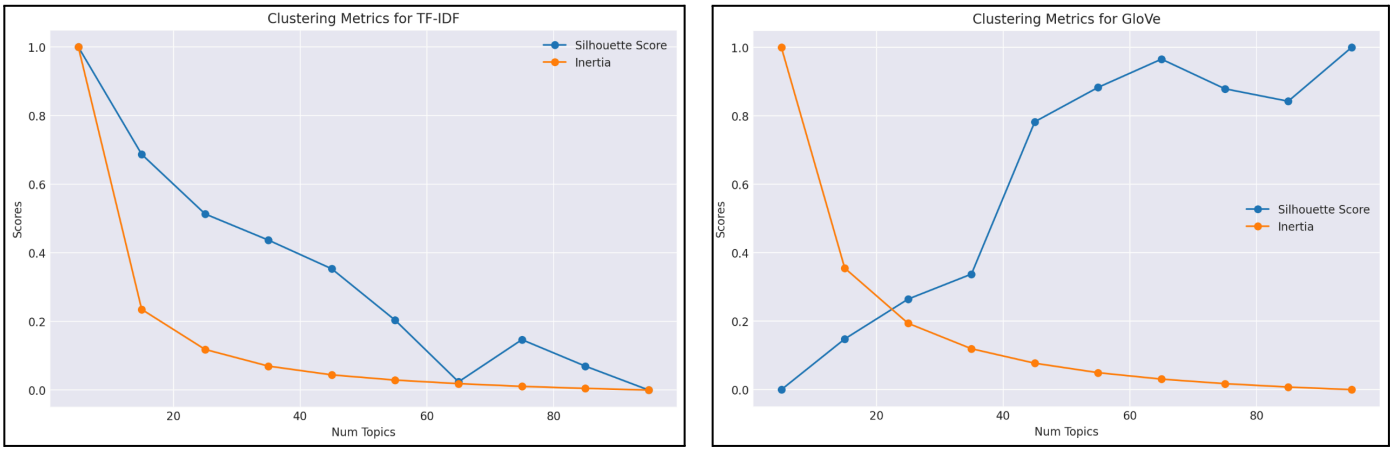
Inertia, commonly used in the elbow method, measures how tightly data points in a cluster are grouped around the cluster center. Lower inertia indicates more compact clusters. The elbow method involves plotting inertia values against the number of clusters and identifying the "elbow point" where the rate of decrease significantly slows, suggesting an optimal number of clusters.

The **silhouette score** assesses the separation between clusters by comparing intra-cluster distance to the nearest cluster's distance. Scores range from -1 to 1, with higher scores indicating that clusters are well-separated and data points are close to their own cluster center.

To find the optimal number of clusters, we normalized these metrics and assigned them equal weights, creating a **combined score** to **balance both inertia and silhouette score**. Normalization ensures that the metrics are on a comparable scale (from 0 to 1 in our case), while the equal weighting signifies that both metrics are equally important in determining the optimal number of clusters. The combined score for each possible cluster number is calculated by weighting and summing the normalized inertia and silhouette scores, with inertia contributing negatively to the combined score since lower values are preferred.

The cluster number with the highest combined score is selected as the optimal choice. This approach allows us to take into account both compactness and separation when determining the best number of clusters for our K-means clustering, leading to a more reliable and effective clustering outcome.

In this project, as mentioned in Task 1, we have experimented with different methods for vectorizing text data, specifically TF-IDF and GloVe, to perform clustering on our forum dataset. After tuning the hyperparameters for K-means clustering and evaluating the optimal number of clusters, we determined the following results:



Clustering metrics for Optimal Number of Clusters on TF-IDF and Glove vectorizations

For the **TF-IDF** representation, the optimal number of clusters was found to be **15**, with a normalized silhouette score of 0.687101 and a normalized inertia of 0.235071 (remember that the scores are normalized for comparison among both metrics).

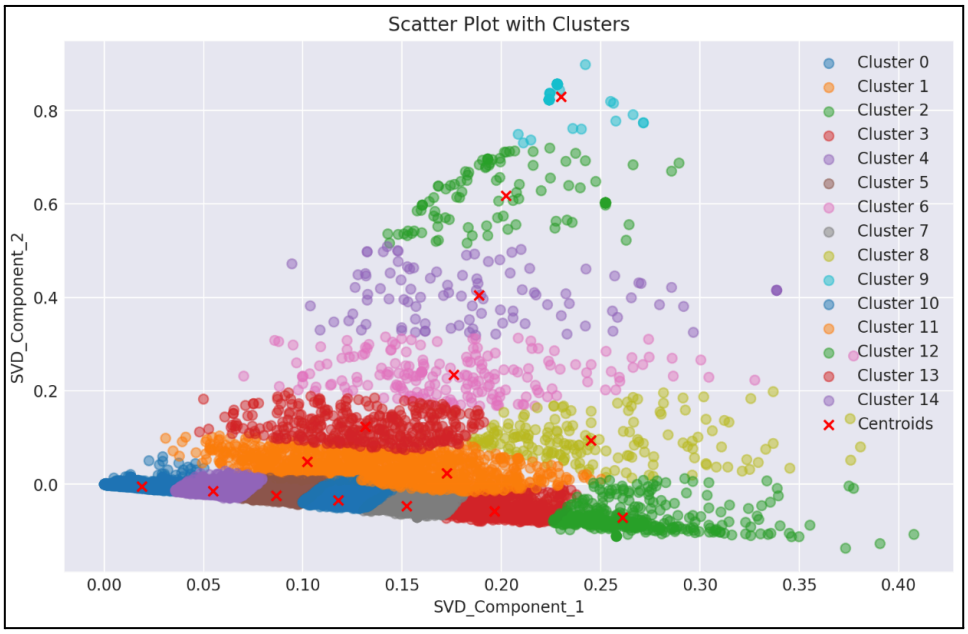
For the **GloVe** representation, the optimal combined score was identified at 95 clusters. However, 95 clusters could be overly complex and lead to challenges in interpretation and analysis. Therefore, we opted for a reduced number of

65 clusters, the second highest combined score, with a normalized silhouette score of 0.965802 and a normalized inertia of 0.965802.

Although the Glove combined score was better, the number of clusters did not align with the elbow method of Glove inertia, which corresponded to 15 and which combined score was worse. Then, the TF-IDF of 15 clusters, which aligned with its elbow method, was chosen, assuring a balance between performance and interpretability. This decision allows us to maintain a reasonable level of detail while minimizing the complexity of further analysis and visualization.

Dimensionality Reduction: We could not have achieved these results without using feature extraction techniques for dimensionality reduction. **SVD for TF-IDF** and **PCA for GloVe** played crucial roles in reducing the dimensionality of our text data and, therefore, in the clustering. This not only improved the performance and efficiency of the clustering algorithms but also allowed us to obtain meaningful results within a reasonable timeframe. Without these dimensionality reduction techniques, executing the clustering algorithm would have taken an **impractically long time** and required considerable computational resources, leading to a **less efficient analysis process**.

Clusters Interpretation



K-Means Clustering using TF-IDF with SVD

In the plot it can be seen that the cluster choice aligns with the topics extracted, a total of 15 clusters. This alignment provides an opportunity for comparison and interpretation to determine whether both representations capture the same information or diverge. In order to extract the interpretation, the closest texts to the centroids were analyzed. The following insights have been obtained from each cluster:

- **Cluster 0** consists of shorter, more cryptic messages.
- **Cluster 1** revolves around a mix of content, with discussions mainly focused on forum content.
- **Cluster 2** also groups together short messages that were banned with the message: **`(USER WAS AUTOBANNED FOR THIS POST)`**
- **Cluster 3** focuses on food experiences and informal conversation sharing personal experiences and seeking advice.
- **Cluster 4** has short texts in common as well as those that were banned with the message: **`(USER WAS AUTOBANNED FOR THIS POST)`**

- **Cluster 5** seems to keep entrances related to groceries and food, but overall a mix of topics.
- **Cluster 6** focuses on post contents and links.
- **Cluster 7** reports many texts that involve life and loss.
- **Cluster 8** does not fully involve forum entrances related to the same topic, it can be seen that it has entrances of people that were banned and others that focus on movies.
- **Cluster 9** keeps posts in which users were put on probiotics with links , possibly indicating violations of forum rules or guidelines.
- **Cluster 10** revolves around diverse topics. Despite this variety, there's a common idea related to different aspects of life, food and internet culture.
- **Cluster 11** talks about audiovisual content.
- **Cluster 12** has long texts in which it is mostly about the Internet and TV shows.
- **Cluster 13** revolves around social media, posting and online communities.
- **Cluster 14** talks mainly about news and video games but nothing very specific, these are very informal and short texts.

In conclusion, the clusters generated by TF-IDF/SVD and the topics obtained by LDA using Mallet give different outcomes derived from these two different vectorization techniques. The use of these multiple approaches for text analysis provides a more comprehensive understanding of the underlying content that can be found in this heterogeneous dataset. In the context of forum management, this analysis is key; it not only facilitates the identification of thematic trends for content cleaning but also enables the aggregation of posts with similar characteristics, such as those that have been banned, thereby enhancing the overall user experience. Using different text analysis methods helps us better understand forum content, which leads to smarter decisions and improved forum management strategies.

Sentiment analysis for emoji recommendation

For sentiment analysis in this project, we employed a combination of datasets and tools, including the **NRC Lexicon Dataset**, **Vader lexicon**, and **NLTK's SentimentIntensityAnalyzer**.

Data Classification and Emotional Scoring (NRC): The forum and emoji datasets are classified into various emotional categories, including 'Anger', 'Anticipation', 'Disgust', 'Fear', 'Joy', 'Sadness', 'Surprise', and 'Trust'. Each entry was analyzed based on its textual description, and a normalized score was assigned to reflect its emotional intensity across these categories.

Sentiment Labeling (Vader): After the initial classification, each forum post and emoji was further analyzed to determine its overall sentiment: positive, negative, or neutral. This was achieved using NLTK's SentimentIntensityAnalyzer, which provided polarity scores based on the sentiment expressed in the text.

Emoji Recommendation: Once the feelings retrieval was completed, we compared that score of NRC feelings among each forum post and the emojis labeled with the same Vader sentiment (positive with positive, etc…). This comparison allowed us to identify emojis that closely matched the emotional tone of the forum post. In order to do that, cosine_similarity() was used.

Emoji Selection: As a result, we selected the three emojis with the highest scores and the most similarity to each forum text, obtained by cosine similarity.

Topic modeling for emoji recommendation

Mapping Topics with subsets of Emojis

-Step 1 Tokenize Topics: Break down the topics into individual words or tokens, ensuring each topic is represented by its key words.
-Step 2 Clean Data: Clean topics and exclude emojis of faces and flags
-Step 3 Calculate Embeddings: Use GloVe to represent each token as a numerical vector. This step involves converting both the topic tokens and the emoji descriptions into dense vectors.
-Step 4 Calculate Cosine Similarity: To determine the similarity between topic embeddings and emoji embeddings, we use cosine similarity, which measures the cosine of the angle between two vectors, indicating how closely they align in a vector space. A similarity score close to 1 indicates high similarity, while a score close to 0 indicates low similarity. We create a similarity matrix, where each row represents a topic embedding and each column represents an emoji embedding.
-Step 5 Determine Relevant Emojis for Each Topic: Start by identifying the emoji with the highest cosine similarity score to each topic, as it represents the closest match to the topic's semantic meaning. To broaden the mapping, we include emojis with a similarity score that is at least 70% of the highest score for each topic. This threshold is a tuned hyperparameter, designed to prevent too few or too many emojis from being associated with each topic.
By allowing a set of emojis for each topic, we ensure a richer interpretation and avoid redundancy. This approach captures the diversity within topics and provides a more nuanced perspective.

Recommend emojis for a Post

Now, to recommend emojis for individual forum posts, the following process is used:
-Step 1 Assign Topics to Posts: Using the document-topic distribution from previous LDA analysis, assign a topic to each post based on the highest probability from "doc_topics.txt."
-Step 2 Identify Relevant Emojis: For each topic, identify the subset of emojis associated with it. This subset will be used to recommend emojis to the post.
-Step 3: Assign Recommended Emojis: Calculate cosine similarity between the embedding of the forum post and the embeddings of the emoji descriptions for its assigned topic.
Assign the top 3 emojis with the highest similarity to each forum post. These recommended emojis align with the post's assigned topic and its underlying content, offering a contextually relevant emoji recommendation.

Task 3: Dashboard

An interactive dashboard has been created to enhance the users' engagement with the posts and forum. The dashboard provides users with insights into the emotional content of forum posts, facilitates exploration of post clusters, and offers emoji recommendations based on text and sentiment analysis.

Cluster Visualization and details: The dashboard includes an interactive scatter plot visualizing post clusters generated by the KMeans algorithm. Users can click on data points to view details of the selected post, including post title, text, cluster number, recommended emojis, and formatted emojis. This feature helps users understand the content and emotional context of posts. In addition, users have the option to explore distribution graphs of emotions, topics, emojis, or word clouds for the selected clusters.

Post Selector: Users can filter posts based on sentiment, topic, and maximum emotion to view example posts matching specific criteria. This feature allows users to explore posts relevant to their interests or emotional preferences.

Emoji Cloud: The dashboard includes an emoji cloud feature that displays emojis scaled by their emotional scores. Users can select an emotion from a dropdown menu to generate an emoji cloud highlighting the emotional content associated with that specific emotion.

Emoji recommendation based on sentiment analysis: Given a new text, we created a pipeline that reproduce all the work that has been done previously:

1) Preprocessing of the new text.
2) Using our best clustering (Kmeans of SVD of TF-IDF) to cluster the new text.
3) Sentiment analysis using NRC and Vader lexicon. Furthermore, we needed to scale again the sentiments of NRC using the same denominator (total emotions list) as in forum and compute the cosine similarities among the emojis with the same Vader sentiment.

The pipeline is called *recommending_emojis* and it only needs the new text as a string in the input.

Conclusion

During this project, we encountered the important role that data cleaning plays in the success of machine learning tasks. As we were advancing with the project, it became evident that the quality and cleanliness of our dataset were crucial. Having a real scenario dataset, we confronted challenges such as noise, ambiguity, and heterogeneity inherent in user-generated content. However, by employing meticulous data cleaning techniques, we ensured the integrity and reliability of our dataset. Furthermore, although we chose one of the most diverse forums in content, any dataset extracted from one forum tends to be biased to the topics that these forums reiterate.

On top of that, the volume of the data made it impossible to use some algorithms because of their cost, that is why dimensionality reduction techniques were crucial in order to solve this issue.

Regardless of that, another encountered issue was how to relate the different datasets, because of their nature. For example, in order to relate the emojis with the sentiments firstly we tried a clustering of the emojis based on the emotions that did not turn out as accurate as first expected so finally we decided to use cosine similarity whose results turned out much better.

Apart from the analysis of the forum, another main goal was to search for techniques that could improve the quality and engagement of it.

Uses of our code

With the clustering segmentation and the topic modeling performed in our real-case scenario dataset, many applications can be used of our code and future implementations in the enhancement of the forum:

Distinguishing Between Useful or Spam or Harassing Posts (Moderation): It enhances user experience by ensuring that the forum is contained in safe space with no triggering content, as well as improving the community engagement by filtering out harmful or irrelevant content. This can be done with the clustering performed, taking out clusters that were centered on that.

Adding Emojis Automatically: By the addition of visual elements such as the emojis to posts, it enhances the visualization of the site, helps convey emotions and tone more effectively, fostering better communication, and finally, improves user satisfaction.

Dividing the Posts into Sections Depending on the Topic and Recommending Posts with Similar Topics: Which organizes content for easier navigation and discovery, facilitating discussions on specific topics that can be obtained all through the topic modeling and clustering.

Applying an Age Filter: Ensuring compliance with legal regulations regarding age-restricted content by checking clusters that do not protect minors from exposure to inappropriate material.

Statistics for the Forum in Order to Engage the Clients: Provides useful insights into user behavior, preferences, and trends, which enables targeted marketing and content strategies.