# Friends-Character-Similarity-Analysis

# Improve pre-processing ( used 6 )

In this in used **tokenization to** splits the text into words, and converting to lowercase to normalize them, **Removing stop words** and **punctuation** clears out common or non-essential elements, streamlining analysis Through **Lemmatization,** reduces words to their base form, aiding in recognizing different forms of the same word also I systematically **expand contractions** ensures consistency by converting shortened words to their full forms. Finally, start **discarding non-alphabetic characters and single letters** removes noise, focusing on meaningful words and sentences required to analyse the FRIENDS Characters.

# Improve linguistic feature extraction using and Matrix transformation techniques

In my process, I first iterate through extra features, adding a count of 1 for each to the counts dictionary. By **Extracting n-grams** (sequences of n words) of different lengths (example bigrams ) to capture word order and context. **Adding POS Tags:** Uses a Part- of-Speech (POS) tagger to identify the grammatical role of each word (e.g., noun, verb, adjective). Further **Applying Sentiment Analysis:** Uses TextBlob to extract sentiment polarity (positive, negative, neutral) from the text. (Commented for best results). Also, **Includes Other Features:** Calls extract\_other\_features() and predict\_gender() for additional features. (Did not include for best results) Used **Matrix transformation techniques** like **TfidfVectorizer** is initialized to converts the text documents into a matrix of TF-IDF features with a **minimum document frequency** as passed in min\_df=2 parameter and **SelectKBest** with **chi2** and k='all' is used for feature selection based on the chi-squared statistic.

# Add dialogue context and scene features

Two dictionaries, character\_docs and character\_line\_count, are created to store the concatenated lines and count of lines for each character, respectively. DataFrame is **grouped by 'Episode' and 'Scene** to process dialogue scene by scene as asked in question. The given function prints the line counts for each character and returns the character\_docs dictionary, which contains the processed text for each character

# Parameter Search (15 marks)

**param\_grid** defines a dictionary of parameters for a **`TfidfVectorizer`** and a feature selector **`SelectKBest`.** Parameters include

**`tfidf sublinear\_tf`** (`True`), **`tfidf min\_df`** (minimum document frequency with values 2), and **`select\_k k`** (1000, 2000, and 'all') creates a list of all possible combinations of these parameters.The code iterates through each parameter combination. For each combination, it extracts the values of `k`, `mindf`, and `sublinear\_tf` using `params.values()`. It attempts to create feature matrices for both training and validation datasets using the`create\_document\_matrix\_from\_corpus` function, with the current parameters. If an error occurs during this process, it prints "Error" and continues to the next iteration.The code compares the mean rank from each parameter combination to the current best score. If it finds a lower mean rank. It updates `best\_score` and

`best\_params` with the new values. print best parameters and score.

# Analyse the Similarity matrix

The given similarity matrix shows how similar each document (or character vector in this context) is to every other document, the values range from 0 to 1, where 1 indicates a very high similarity, and values closer to 0 indicate lower similarity. **Closest Characters: Chandler Bing**: The character vector closest to Chandler that is not Chandler himself appears to be Joey Tribbiani, given the high similarity score of around 0.49. This suggests that Chandler and Joey might share similar language patterns, topics, or have interactions that make their dialogue representation similar. **Phoebe Buffay**: Phoebe's dialogue is most similar to Joey's, excluding her own, with a similarity score close to 0.49.**Furthest Characters: Chandler Bing**: The furthest character from Chandler seems to be the "Other\_Female" category, with a similarity score around 0.214. **Phoebe Buffay**: Phoebe's dialogue shows the least similarity with "Other\_Male", with a score around 0.175. Several linguistic and contextual factor can consider to check highest match between the target character in the training set and that character's vector in the held-out set:Such as **Dialogue Context, Vocabulary and Topics**, **Interactions with Other Characters, Amount of Dialogue**, **Speech Style and Idiosyncrasies**, **Model Limitations**(The choice of features (e.g., word embeddings, n-grams) and the vectorization technique (e.g., TF-IDF, count vectorizer) might not capture the nuances necessary for distinguishing characters effectively)

# Final Test Data

1. **Mean Rank**: The average position where the correct document is ranked by the model is 4.1 for testing and 3.3 for validation after pre-processing, linguistic feature extraction , group by dialogue and scene while applying grid search. The

lower mean rank in validation indicates that the correct documents tend to be ranked closer to the top of the list compared to the testing dataset.

1. **Mean Cosine Similarity**: The average cosine similarity score is a measure of how similar the document vectors are to each other. Higher scores indicate more similarity between documents. The validation set has a slightly higher mean cosine similarity (0.3973) than the testing set (0.3915), suggesting that the documents in the validation set are more similar to each other than those in the testing set.
2. **Accuracy**: The accuracy indicates the proportion of documents that were correctly identified as the most relevant. The model has an accuracy of 0.1 (10%) in the testing set and 0.3 (30%) in the validation set, meaning it correctly identified the most relevant document 1 out of 10 times in testing and 3 out of 10 times in validation.

|  |  |  |
| --- | --- | --- |
| **Training/Testing** | | **Validation/Results** |
| **First run, when code is given as it is** | mean rank 4.2  mean cosine similarity 0.8915725404768657  1 correct out of 10 / accuracy: 0.1 | mean rank 4.0  mean cosine similarity 0.8925164628242618 3 correct out of 10 / accuracy: 0.3 |
| **Ques 1 - Pre- processing steps:** | mean rank 3.3  mean cosine similarity 0.9170514279446987  3 correct out of 10 / accuracy: 0.3 | mean rank 2.8  mean cosine similarity 0.9173610869536564 4 correct out of 10 / accuracy: 0.4 |
| **Ques 2 :**  **TF-IDF Transformer :** | mean rank 3.3  mean cosine similarity 0.9170514279446987  3 correct out of 10 / accuracy: 0.3 | mean rank 2.8  mean cosine similarity 0.9173610869536564 4 correct out of 10 / accuracy: 0.4 |
| **Ques 2 :**   1. **TF-IDF Vectorizer** 2. **Select K-Best** 3. **Chi- square** | mean rank 3.1  mean cosine similarity 0.1385298237845539  3 correct out of 10 / accuracy: 0.3 | mean rank 2.8  mean cosine similarity 0.13201139424789687 4 correct out of 10 / accuracy: 0.4 |
| **Ques 2:**  **min\_df =2 :** | mean rank 3.1  mean cosine similarity 0.21360415902356164  4 correct out of 10 / accuracy: 0.4 | mean rank 2.7  mean cosine similarity 0.20046919188497178 5 correct out of 10 / accuracy: 0.5 |
| **Ques 3 :**  **grouped by 'Episode' and 'Scene' :** | mean rank 5.3  mean cosine similarity 0.22622310377786387  1 correct out of 10 / accuracy: 0.1 | mean rank 4.5  mean cosine similarity 0.23561828317726774 2 correct out of 10 / accuracy: 0.2 |
| **Ques 4: Grid Search** | mean rank 4.3  mean cosine similarity 0.38667813647485805  0 correct out of 10 / accuracy: 0.0 | mean rank 3.8  mean cosine similarity 0.3820474113672166 0 correct out of 10 / accuracy: 0.0 |
| **Ques 6 :**  **With Best combination** | mean rank 3.9  mean cosine similarity 0.4034230024554968  1 correct out of 10 / accuracy: 0.1 | mean rank 3.4  mean cosine similarity 0.41060125708980894 3 correct out of 10 / accuracy: 0.3 |