**Student ID = 230049961**

**Full Name = Gondabal Ramakrishna Pavan Kumar**

**Q1. Improve pre-processing:**

To clean the text, we are using general text cleaning methods such as Tokenization, Lowercasing, Character name removal (Removes specific character names), Punctuation removal, Number removal, stop word removal, single character removal, lemmatization. After Using all these techniques, we drastically reduce the Mean Rank to 2.4 for Val and Test data set.

pre\_process(character\_text) Takes String as input and cleans it and returns tokens.

At the later stage, grid only best pre-processing methods will be implemented to optimize the Mean Rank

**Q2. Improve linguistic feature extraction:**

For each character Doc features such as Sentiments, POS tags, N-Grams, TF-IDF and K best selectors for top features are implemented as vector feature vectors to create Training Feature Matrix. Since the tokens which are generated from pro\_process() are not enough to draw meaningful relationships between different characters. Our Goal is to extract as much as features from the tokens and compare cosine similarities and differences between the character Features in next steps

We will be using tfidf\_vectorizer = TfidfVectorizer(ngram\_range=(1, 2), max\_df=0.85, min\_df=2, sublinear\_tf=True)

As a standard input for the tf-idf features after trying several combinations of the hyperparameters, these are the best combinations

And k\_best\_selector = SelectKBest(chi2, k=100) is the optimum parameter for K best estimator after trail and error

Bi grams and SentimentIntensityAnalyzer() function are used to create feature vectors within the function

to\_feature\_vector\_dictionary()

Tf-IDF and K Best Selectors are being used in create\_document\_matrix\_from\_corpus(corpus, fitting=False)

After this step, A document matrix vector is being created.

**Q3. Add dialogue context and scene features**

The create\_character\_document\_from\_dataframe function iterates through the DataFrame and is grouped by episodes and scenes. For each scene, each character's lines are concatenated into a single string in order to map the scenes and episodes for each character and get the context behind the character lines.

Empty lines and lines exceeding a specified maximum per character are ignored. An "EOL" token is inserted between lines to mark their boundaries. This results in a dictionary where each key is a character's name and the value is their compiled dialogue document for the entire dataset. After running these steps we get mean rank =1.6

mean cosine similarity = 0.9772362869701616 , 8 correct out of 10 / accuracy: 0.8

**Q4. Parameter Search**

Pre processing plays a pivotal role in this analysis as the features derived from the tokens are highly crucial for the model, All the preprocessing steps are again grouped into three Args, the doc text is being cleaned by using all the possible combinations of these groups for validation dataset to maximize the mean rank, after running the grid search we found that the best combination of args is arg1 = True, arg2 = False and arg3 = False with a Mean rank 1 which is the highest possible rank

A colorful squares with numbers

Description automatically generated

**Q5. Analyse the similarity results :**

From the Heatmap, we were able to extract the closest and Farthest characters from Test and Train dataset , we found that

Pair with the highest similarity: **Phoebe Buffay and Rachel Green**

**similarity - 0.964441**

Pair with the lowest similarity: **Rachel Green and Monica Geller**

**similarity - 0.910794**

The similarity is due to the usage of similar N grams and tokens

A close up of words

Description automatically generatedBy these characters

Top 50 words Train doc - Phoebe Buffay:

['\_eol\_', 'i', 'you', 'the', 'oh', 'to', 'a', 'it', 'and', 'that', 'okay', 'this', 'so', 'is', 'no', 'in', 'my', 'im', 'but', 'do', 'what', 'me', 'its', 'have', 'like', 'just', 'of', 'know', 'really', 'on', 'thats', 'go', 'all', 'right', 'na', 'was', 'your', 'well', 'get', 'dont', 'love', 'youre', 'gon', 'be', 'one', 'not', 'who', 'he', 'her', 'now']

A close up of words

Description automatically generated

Top 50 words Test doc - Rachel Green:

['i', '\_eol\_', 'and', 'that', 'im', 'you', 'it', 'me', 'oh', 'its', 'right', 'all', 'to', 'what', 'but', 'my', 'okay', 'know', 'yeah', 'better', 'about', 'the', 'hi', 'have', 'just', 'a', 'no', 'was', 'barry', 'sorry', 'am', 'so', 'think', 'this', 'is', 'with', 'your', 'on', 'isnt', 'look', 'not', 'of', 'um', 'are', 'really', 'do', 'guys', 'alright', 'see', 'doing']

**Q6. Run on final test data:**

After running the grid search and optimizing pre\_processing function, optimizing the context of each characters through mapping episodes and scenes, using feature extraction and selection, we run the model on test data set and were able to from initial Mean Rank of **4.0** and accuracy of **0.3, we have** achieved the Mean Rank of 1.0 , Cosine similarity of 0.9735 and accuracy of 10/10 = 1.0 by making all the above changes