Q1. Improve pre-processing

After applying the following preprocessing techniques, the accuracy increased from 0.1 to 0.5, the mean rank improved from 4.2 to 3.3, and the mean cosine similarity changed from 0.8915725404768657 to 0.5363425520099641.

(1) Tokenization: It can correctly handle contractions and punctuation and divide the text into individual words(tokens).

(2) Lowercasing: It converts all tokens to lowercase. And it is crucial for ensuring that the same words in different cases (e.g., "Hello" and "hello") are treated as identical, reducing the overall feature space.

(3) Removing Non-Alphabetic Words: Any tokens containing numbers or non-alphabetic characters (such as punctuation) are filtered out. The goal of this step is to keep only important words.

(4) Removing Stop Words: Stop words (e.g., "the", "is", "and") are generally filtered out in NLP tasks because they appear frequently and usually don't carry significant semantic meaning.

(5) Lemmatization: Based on a word's part-of-speech, it accurately reduces words to their base or dictionary form (lemma). It assists in simplifying word inflectional forms so that they can be examined as a single unit of analysis.

Q2. Improve linguistic feature extraction

After employing the following feature extraction methods, the metrics are as follows:

Mean Rank: 2.9, Mean Cosine Similarity: 0.23860835266689467**, Accuracy: 0.5**

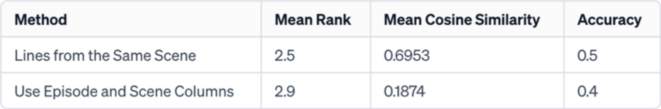
We can see that the mean rank was improved by 12.12%. Although the mean rank decreased, no change in the accuracy was recorded:

(1) N-Grams: Incorporation of n-grams, such as bigrams and trigrams, goes beyond single words to encompass more context within the text. This technique not only provides a richer representation of language patterns but also allows for the computation of a probability distribution that predicts the likelihood of a subsequent word in a sequence.

(2) POS Tags: Append POS tags to tokens to add syntactic information. it can be useful in understanding the context better.

(3) TF-IDF: The rationale for using TF-IDF instead of mere frequency counts of tokens within a document lies in its ability to reduce the impact of tokens that are common across the entire corpus. These frequent tokens are generally less informative than rarer tokens, which tend to appear in a limited segment of the training corpus and can offer more specific insights.

Q3. Add dialogue context and scene features



Method	Mean Rank	Mean Cosine Similarity	Accuracy
Lines from the Same Scene	2.5	0.6953	0.5
Use Episode and Scene Columns	2.9	0.1874	0.4

The data's context was integrated by including dialogue from other characters in the same scene. And I limited the number of context lines to five. Despite employing the character names to track the dialogue context, the addition of this feature, as indicated by the table, did little effect on the mean rank that **the mean rank improved from 2.9 to 2.5**. However, solely utilizing the Episode and Scene columns had no effect on the mean rank.

Q4. Parameter Search

(1) In Question 1 (Q1), by experimenting with permutations and combinations of five optimization methods in the pre_process function, I selected ten combinations. Through comparison, I found that the combination of word tokenize, lowercase, Remove stop words, and lemmatization yielded the best mean rank value, with a mean rank of **2.8**.

(2) Building upon the optimal mean rank from Q1, in Question 2 (Q2), I experimented with combinations of N-Grams, POS Tags, and TF-IDF. I discovered that using all three methods together produced the optimal mean rank value, resulting in a mean rank of **2.5**.

(3) Based on the optimal mean ranks from Q1 and Q2, in Question 3 (Q3), there are currently two methods under consideration. The first method involves including dialogue from other characters in the same scene, while the second method solely utilizes the Episode and Scene columns as the context. Through comparison, I found that using the first method yielded the best mean rank value. The mean rank achieved with this approach was **2.2**.

Q5. Analyse the similarity results

From the heat map generated by the plot_heat_map_similarity function, it is evident that the language use of Phoebe, Rachel, and Ross is more similar, whereas "ALL" generally shows lower similarity to other characters. This is primarily due to the individual language use and dialogue style of the characters, including similarities in word choice, phrasing, and the use of specific n-grams. Additionally, thematic elements, tone, and other linguistic features contribute to this observation.

(1) **High similarities in the characters Phoebe, Rachel, and Ross:** All three characters often use humor in their dialogue, with Phoebe's eccentricity, Rachel's sarcasm, and Ross's self-deprecation. They openly express emotions, sharing personal challenges and feelings. Common interjections like "Oh God" and "Alright" across their dialogues reflect their reactions and emphasize conversation points.

(2) **Low similarity between "ALL" and other characters:** The "ALL" character's lines typically represent a group speaking, which blends individual styles into a more neutral voice and may lead to less distinctive data for similarity analysis. If "ALL" contributes fewer lines, the data becomes sparser, further reducing similarity scores with individual characters.

Q6. Run on final test data

After running on the final test data, we obtained the following metrics:
Mean Rank: 2.8, Mean Cosine Similarity: 0.95724503449231**, Accuracy: 0.4**
The mean rank of 2.8 is higher than the 2.2 observed in the train set, which could be attributed to the following reasons:

(1) **Data Distribution:** Differences in the distribution of data between the training and test sets can impact performance. If the test set contains new, unseen patterns or variations not present in the training set, the model may struggle to accurately predict these cases.

(2) **Limited Training Data:** If the training data is not sufficiently diverse or is too small, the model may not learn enough to generalize well to new data.