Question 1: Pre-processing

There are several applications of different pre-processing techniques applied, namely – separation of punctuation at the front and back, conversion of word text numbers to digits, removal of punctations, replacing money digits to the word 'money', replacing digits to hashtags, normalising texts to all lower case, applying lemmatisation, removal of stop words and converting number digits to words. To separate the text into smaller units, tokenisation is applied. This is done by splitting the text via its white spaces.

After applying numerous combinations of the different techniques, it can be observed that the retrieval of the lowest mean rank with high mean cosine similarity, conversion of word text numbers to digits do not contribute with decreasing the mean rank. Whilst the most effective pre-processing techniques that help reduce mean-rank are separation of punctuation at the front and back, removal of punctations, normalising texts to all lower case, applying lemmatisation, and removal of stop words.

Therefore, after such observations, it is concluded that the best techniques to apply to improve pre-processing are separation of punctuation at the front and back, removal of punctations, replacing money digits to the word 'money', replacing digits to hashtags, normalising texts to all lower case, applying lemmatisation and removal of stop words.

After application, Figure 1 shows the 52.8% improvement in mean rank with the improved pre-processing technique. The mean ranking has improved from 4.5 to 2.125. Also, percentage of accuracy shows that it classifies 10 characters correctly out of 16 characters.

	Initial	After	%Δ
Mean rank	4.5	2.125	52.8
Mean cosine similarity	0.912814	0.942123	-
Accuracy	0.25	0.625	-

Figure 1: Results before pre-processing technique application

Question 2: Linguistic Feature Extraction

For linguistic extraction, 2 techniques are applied. One of which is the application of the n-gram model. To assist the n-gram modelling, '<s>' is added to in the beginning of the phrase and '<\s>' is added at the end of the phrase, indicating a new line. The values tested for n is from 1 to 5.

The second technique is the adding of weights to the frequency of the token or character phrase. There are 2 applications of the weights. Whereby, the weight is the frequency of occurrence of each token in the character script named as the 'counts' technique. And the other is the frequency of each token divided by the number of tokens for each character, named as the 'weighted' technique.

After applying the numerous combinations of the different linguistic feature extraction techniques, it can be observed that the counts and weighted technique with a unigram keeps the mean rank at the same position as the improved pre-processing which is 2.125.

The higher n-gram models such as bigram and trigram models had increased the mean rank. Hence, unigram is chosen with its initial weighing technique which is counts. Here, there is no improvement in mean rank for this case. It remains that the percentage of accuracy still classifies 10 characters correctly out of the 16 characters.

Question 3: Similarity Results

Heather, Jane, Max, Minty, Other, Phil, Ronnie, Stacey, and Tanya are identified correctly with the similarity results the highest on the correct character.

Christian and Clare are also identified correctly, however Christian is also ranked closely to Jane, and Clare to Roxy. This shows that there is a tendency of Christian and Clare being identified as Jane and Roxy respectively, if the character lines are pruned even more so, decreasing word count for each character. Hence, there is a need to add additional context such as gender context in the corpus, this could help to prevent this tendency.

Ian, Jack, Roxy, Sean, and Shirley are ranked furthest from its character themselves. Ian is identified as Christian, Jack is identified as Phil, Roxy as Christian, Sean as Ian, Shirley as Other.

From the characters that are furthest to its correct ranking, they seem to have more words than its identified character in the training set. With higher number of words, it can relate to higher word similarity to the other characters with lesser number of words. Thus, causing characters to not be ranked correctly.

Question 4: Adding additional context

Investigation of adding episode-scene values and/or gender of character into the dialogue context occurs here. For this to work, the characters with no lines in the episode-scenes will have an added line of '_N<episode-scene>N_' which would indicate 'N' for no character part for the episode-scene. Whilst those characters with character part in the episode scene will be labelled as '_<episode-scene>_' before the start of the dialogue lines. Character gender is also included as '_FEMALE_' and '_MALE_'.

Pre-processing is then adjusted, whereby, during pre-processing these added dialogue contexts are skipped and not processed.

For adding of weights in feature vector method, the n-gram model is only applied to the start of the dialogue lines and not on the added dialogue contexts. After, testing four different combinations of dialogue context, this resulted to a 26.47% improvement on the mean rank from 2.125 (from question 1 and question 2) to 1.5625. Also, percentage of accuracy shows that it classifies 12 characters correctly out of 16 characters.

Figure 4 shows the 65.28% improvement of the mean rank before any improvements were created.

	Initial	After	%Δ
Mean rank	4.5	1.5625	65.28
Mean cosine similarity	0.912814	0.996464	-
Accuracy	0.25	0.75	-

Figure 4: Results comparison before improvements created and after adding of additional context

Question 5: Improvement of vectorisation

The TF-IDF transformer is added after the Dictionary Vectorizer is applied. The TF_IDF uses the frequency of the words to see how relevant the words are to a given document. TF is the word frequency in the document and IDF is the inverse document frequency. Thus, this further refines the vectorisation method. With this, the mean rank improved from 1.5625 (from question 4) to 1.375, resulting in a 12% improvement. Also, percentage of accuracy shows that it classifies 12 characters correctly out of 16 characters. From the initial mean rank of 4.5, this is a 69.44% improvement.

	Initial	After	%Δ
Mean rank	4.5	1.375	69.44
Mean cosine similarity	0.912814	0.984415	-
Accuracy	0.25	0.75	-

Figure 6: Results comparison before improvements created and after adding TF-IDF vectorizer

Question 6: Final Test Data

After running on the final test data, the table below shows that the mean rank had improved from 4.5 to 1 creating a percentage improvement of 77.78%. Hence, reaching the lowest mean rank number.

Also, percentage of accuracy shows that it classifies 16 characters correctly out of 16 characters.

	Initial	After	%Δ
Mean rank	4.5	1	77.78
Mean cosine similarity	0.912814	0.985903	-
Accuracy	0.25	1	-

Figure 6: Results comparison before improvements created and after improvements created

In conclusion, the most effective methods that reduced the mean rank is the application of pre-processing techniques, adding of additional contexts such as gender and episode scene and the improvement of the vectorisation method.