# Report on NLP Assignment 2: Distributional Semantics

**Q.1: Improve Pre-processing**

The **'pre_process()** function was enhances with these techniques as follows:

- **Tokenization:** To utilize **nltk.tokenize.word_tokenize** to segment the text into words.
- **Text Normalization:** To convert all tokens to lowercase to ensure uniformity.
- **Punctuation Removal:** To strip tokens of punctuation using the **string.punctuation** set.
- **Stopword Removal:** To filter out stopwords with **nltk.corpus.stopwords**
- **Lemmatization:** Reducing words to their base form via **nltk.stem.WordNetLemmatizer**.

These preprocessing steps were used to refine the text data so that it can contribute to better vector representations for each character.

**Q.2: Improve Linguistic Feature Extraction**

The **to_feature_vector_dictionary()** and **create_document_matrix_from_corpus()** functions were improved with the addition of:

- **N-grams of different lengths**
- **POS tags**
- **Matrix transformation techniques like TF-IDF**

These techniques were used to improve the feature set and provide a more nuanced representation of the character's dialogue

**Q.3: Add Dialogue Context and Scene Features**

Modifications to include lines spoken by other characters in the same scene were done, providing context to the target character's lines. This helps to embed a deeper level of context into the character vectors.

**Q.4: Parameter Search**

A grid search was done on various parameters including **ngram_range** and **use_idf**, to find the optimal settings. The best parameters found were:

- **Best mean rank: 1.4**
- **Best parameters: {'ngram_range': (1,2), 'use_idf': True}**

This approach aimed to identify the most effective configurations, leading to improved performance metrics.

**Q.5:** Analyse the Similarity Results

The **plot_heat_map_similarity()** function was utilized to heldout character vectors ranked closest and furthest from each character's training vector.

**Closest Heldout Character Vectors:**

**Chandler Bing:** The highest non-self similarity score is with **Joey Tribbiani (0.967)**, which could be due to their close friendship and frequent interactions, likely leading to shared vocabulary and conversational dynamics.

**Joey Tribbiani:** Similar to Chandler, Joey's closest match is with **Chandler (0.959)**, for the same reasons mentioned above.

**Monica Geller:** Her vector is closest to **Ross Geller (0.958)**, which could be attributed to them being siblings.

**Phoebe Buffay:** The closest match is with **Monica Geller (0.958)**, possibly reflecting shared social circles and interactions on the show.

**Rachel Green:** Her highest similarity is with **Ross Geller (0.947)**, which may be due to their romantic relationship within the series, influencing their dialogue with common themes and emotional content.

**Ross Geller:** Ross is closest to **Monica Geller (0.963)**, which is consistent with the sibling relationship they share, influencing their dialogue.

**Furthest Heldout Character Vectors:**

**Chandler Bing:** The furthest vector is with **'Other_None' (0.938)**, indicating a distinct linguistic style or less shared content.

**Joey Tribbiani:** Joey's furthest is also with **'Other_None' (0.939)**, suggesting unique speech patterns not captured in the generic "Other" category.

**Monica Geller:** The furthest away is the **'Other_Male' category (0.935)**, indicating Monica's linguistic features are quite distinct from those in this category.

**Phoebe Buffay:** Phoebe's furthest match is also with **'Other_Male' (0.933)**, supporting her unique and often quirky linguistic style.

**Rachel Green:** Similar to Monica and Phoebe, Rachel's vector is furthest from **'Other_Male' (0.935)**.

Ross Geller: The furthest from Ross is **'Other_Male' (0.934)**, again indicating a clear distinction from this category.

**Q.6: Run on Final Test Data**

Tested with the best configurations on the test data. The outcomes were:

- **Mean Rank: 1.1**
- **Mean Cosine Similarity: 0.9599084479096278**
- **Accuracy: 90% (9 correct out of 10)**

This final test aimed to validate the effectiveness of the methodologies and parameter tunings applied in previous questions.

| | **Training and Validation** | **Training and Testing** |
|---|---|---|
| **First run** | mean rank **4.2**<br>mean cosine similarity **0.8915725404768657**<br>1 correct out of 10 / accuracy: **0.1** | mean rank **4.0**<br>mean cosine similarity **0.8925164628242618**<br>3 correct out of 10 / accuracy: **0.3** |
| **After pre-processing steps** | mean rank **3.8**<br>mean cosine similarity **0.9118679204172677**<br>3 correct out of 10 / accuracy: **0.3** | mean rank **2.5**<br>mean cosine similarity **0.9160305887747778**<br>5 correct out of 10 / accuracy: **0.5** |
| **After feature extraction steps** | mean rank **2.2**<br>mean cosine similarity **0.9071222176738631**<br>5 correct out of 10 / accuracy: **0.5** | mean rank **2.3**<br>mean cosine similarity **0.9031576517144124**<br>5 correct out of 10 / accuracy: **0.5** |
| **After adding dialogue context** | mean rank **1.7**<br>mean cosine similarity **0.9607120828148268**<br>7 correct out of 10 / accuracy: **0.7** | mean rank **1.1**<br>mean cosine similarity **0.9599084479096278**<br>9 correct out of 10 / accuracy: **0.9** |
| **Final run (After grid search)** | Best mean rank: **1.4**<br>Best parameters:<br>{'ngram_range': **(1, 2)**,<br>'use_idf': **True**} | mean rank **1.1**<br>mean cosine similarity **0.9599084479096278**<br>9 correct out of 10 / accuracy: **0.9** |