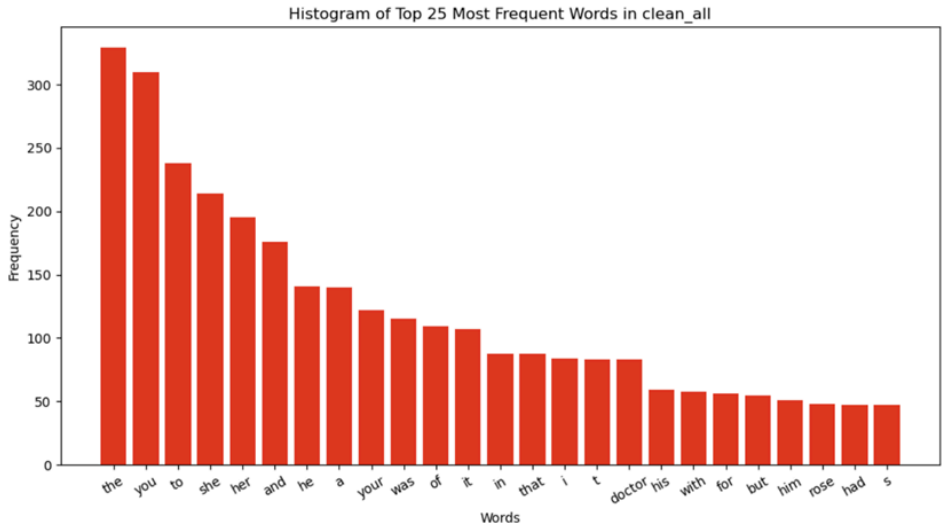
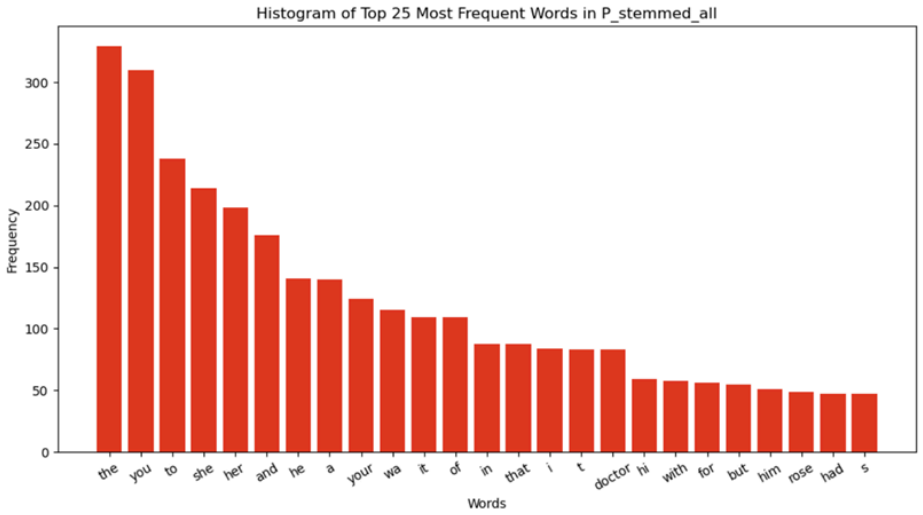
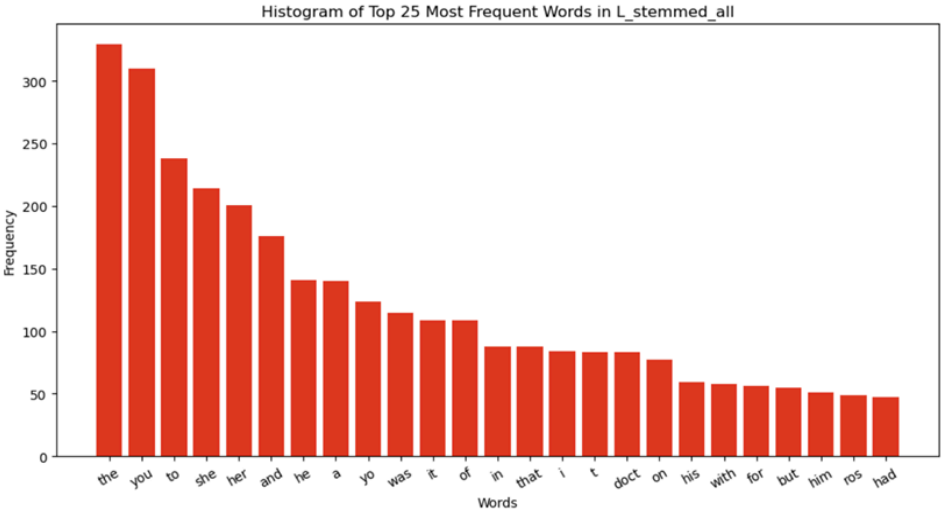
AD Assignment: Part 1







2. Difference between stemmed and unstemmed words

When you stem a text, we remove all the extra parts of a word, to just have their root form. This results in sometimes incomplete words. The unstemmed words are the base words that have been cleaned of punctuation and lowered but remain the same. The histogram of the most frequent words from the unstemmed text shows that words like ‘the’, ‘you’ and ‘to’ are very common. This is the same for the stemmed texts, this is because they are already in their root forms. However, when we continue comparing the stemmed and unstemmed texts we see that some words have been changed, such as ‘was’ to ‘wa’ and ‘doctor’ to ‘doct’. Stemming has reduced these words. In these cases, it has overly reduced them and caused them to lose their meaning. The unstemmed words can retain their meaning and context and so can be useful when considering this. Stemmed words have lost their word meaning, although it may result in more words being counted, such as ‘goes’ becoming ‘go’ which increases the frequency of ‘go’ and makes the dataset more reliable. This comes with the risk of also damaging the rest of the data that may be inadvertently affected by these modules. Stemmed texts have fewer word variations and can give more accurate results. Thus, when you are analysing general topics or themes, the stemmed texts will present you with more reliable data.

Difference between PorterStemmer() and LancasterStemmer()

There seems to be a difference in approach between the Porter and Lancaster stemmers. The Lancaster stemmer has cut off the ends of two-character names, ‘doctor’ to ‘doct’ and ‘rose’ to ‘ros’. This is not the case in the Porter stemmer, this stemmer seems to be more conservative with what it chooses to stem, likely due to how they were both made, they both tend to stem different kinds of words. Based on these findings alone, I would say that the Lancaster stemmer is likely more aggressive with how it performs its stemming in comparison to the Porter. Thus, for our purposes, the Porter stemmer would be more useful as it does not affect other words that do not need to be stemmed, in comparison to the Lancaster stemmer. However, when performing the stemming, I did notice that the Lancaster stemmer was faster, so in some cases where you have extremely large datasets, the Lancaster stemmer may be more efficient in time-sensitive situations.

3. Assumptions on POS frequencies between Tom Sawyer translations

```
eng_pos_counts = Counter(token.pos_ for token in doc_eng)
print('English words pos', eng_pos_counts)
```

```
English words pos Counter({'NOUN': 12669, 'VERB': 11634, 'PRON': 10141, 'ADP': 7242, 'DET': 6900, 'ADV': 4852, 'ADJ': 4427, 'AUX': 3938, 'CCONJ': 3843,
'PROPN': 2445, 'SCONJ': 1883, 'PART': 1716, 'PUNCT': 873, 'INTJ': 528, 'NUM': 495, 'X': 52})
```

```
ger_pos_counts = Counter(token.pos_ for token in doc_ger)
print('German words pos', ger_pos_counts)
```

```
German words pos Counter({'NOUN': 10352, 'ADV': 10022, 'PRON': 9005, 'VERB': 8828, 'DET': 7537, 'ADP': 5775, 'AUX': 4128, 'ADJ': 3388, 'CCONJ': 2864, 'P
ROPN': 2087, 'PART': 1646, 'SCONJ': 1084, 'X': 387, 'NUM': 301, 'PUNCT': 9, 'INTJ': 3})
```

```
dut_pos_counts = Counter(token.pos_ for token in doc_dut)
print('Dutch words pos', dut_pos_counts)
```

```
Dutch words pos Counter({'NOUN': 12405, 'VERB': 11574, 'PRON': 9994, 'ADP': 8851, 'DET': 7051, 'ADV': 6230, 'ADJ': 5276, 'AUX': 4177, 'CCONJ': 3463, 'PR
OPN': 2539, 'SCONJ': 2055, 'NUM': 397, 'SYM': 210, 'INTJ': 173, 'X': 12, 'PUNCT': 2})
```

Above are all the translations of Tom Sawyer, and their frequencies of different word types. Immediately what we notice is that the German text has significantly fewer nouns than the other two translations. This would make me assume that the German language uses fewer nouns than English or Dutch to tell the same story. They have a comparatively very high rate of adverbs being used instead, with 10022. While the English text only uses 4852 and the Dutch text uses 5276. I would assume that this is where the key lexical difference lies between the 3 languages, with German using a much higher number of adverbs compared to, for example, the English noun-based sentence structure. Based on these findings I would also argue that when it comes to sentence structure, Dutch and English might be surprisingly similar, especially when compared to German. They have similar levels of word types, for example, adjectives being 4852 in English and 5276 in Dutch. Based on this similarity in number, I would assume that structurally Dutch and English are more similar than when compared to German.

AD Assignment 1: Part 2

File_01: Another plot?

Automated:

Missy smiles and leans back, placing a kiss on River **PRODUCT**'s jaw, a smile on her face, River **LOC** reaches onto the table for a cup of coffee, gracefully sitting herself on the chair next to Missy **PERSON**, she raises the cups to her lips, wondering whether Missy **PERSON** had noticed yet.

Manual:

Missy **PERSON** smiles and leans back, placing a kiss on River **PERSON**'s jaw, a smile on her face, River **PERSON** reaches onto the table for a cup of coffee, gracefully sitting herself on the chair next to Missy **PERSON**, she raises the cups to her lips, wondering whether Missy **PERSON** had noticed yet.

TP - 2

FP - 2

FN - 1

Precision: TP/(TP+FP) = 2/(2+2) = 0.5

Recall: TP/(TP+FN) = 2/(2+1) = 0.6667

F1-Score: 2*((Precision*Recall)/(Precision+Recall)) = à

2*((0.5*0.6667)/(0.5+0.6667)) = **0.5714**

File_02: You’d looked me in my eyes and told me

Automated:

She wished it could have been him after he knew who she was, because what he was mostly talking about was 1969 **DATE**, she wanted to know how her parents were in New York **GPE**, she wanted the more mature version of him, but she lent against him, sitting in the garden, gently pulling him up to show him the three **CARDINAL** children she had been left with, it wasn’t night yet and they were playing together.

Manual:

She wished it could have been him after he knew who she was, because what he was mostly talking about was 1969 **DATE**, she wanted to know how her parents were in New York **GPE**, she wanted the more mature version of him, but she lent against him, sitting in the garden, gently pulling him up to show him the three **CARDINAL** children she had been left with, it wasn’t night yet and they were playing together.

TP - 3

FP -0

FN -0

Precision: TP/(TP+FP) = 3/(3+0) = 1

Recall: TP/(TP+FN) = 3/(3+0) = 1

F1-Score: 2*((Precision*Recall)/(Precision+Recall)) =2*((1*1)/(1+1)) = **1**

File_03: I Want You Safe

Automated:

Rose blinked back at him, confused. He was laughing, but something wasn't right and Rose **PERSON** could feel it.

Manual:

Rose **PERSON** blinked back at him, confused. He was laughing, but something wasn't right and Rose **PERSON** could feel it.

TP - 1

FP - 0

FN -1

Precision: TP/(TP+FP) =1/(1+0) = 1

Recall: TP/(TP+FN) = 1/(1+1) = 0.5

F1-Score: 2*((Precision*Recall)/(Precision+Recall)) = 2*((1*0.5)/(1+0.5)) = **0.6667**

File_04: Even If the Language of Flowers is Dead, Roses Always Mean Love

Automated:

The Doctor only gave the TARDIS **ORG** permission to translate certain things. He didn't want some people to snoop around the library and read on Gallifrey **PERSON**, so the TARDIS **ORG** only had permission to translate that in emergencies.

Manual:

The Doctor **PERSON** only gave the TARDIS **PRODUCT** permission to translate certain things. He didn't want some people to snoop around the library and read on Gallifrey **GPE**, so the TARDIS **PRODUCT** only had permission to translate that in emergencies.

TP - 0

FP - 3

FN - 1

Precision: TP/(TP+FP) = 0/(0+3) = 0

Recall: TP/(TP+FN) = 0/(0+1) = 0

F1-Score: 2*((Precision*Recall)/(Precision+Recall)) = **0**

File_05: WWTDD: What would the Doctor do?

Automated:

You didn't mean to be away for long, however, spending some time alone allowed you to really focus on the Doctor's words. And you realized all this time she had been right.

Manual:

You didn't mean to be away for long, however, spending some time alone allowed you to really focus on the Doctor's **PERSON** words. And you realized all this time she had been right.

TP - 0

FP - 0

FN -1

Precision: TP/(TP+FP) = 0/(0+0) = 0

Recall: TP/(TP+FN) = 0/(0+1) = 0

F1-Score: 2*((Precision*Recall)/(Precision+Recall)) = **0**

Results:

Each of these texts gave very different scores, partly due to the low sample size of only 1-2 sentences per text. However, with an average score of 0.4478, we can deduce that the named entity recognition (NER) performed very poorly for this text. Anything below 0.5 is usually considered poor and unusable for the research we want to undertake. However, from the results, we know that two of the F1 scores were 0s, which would have heavily skewed the results.

All of the manually annotated results are accurate and have no errors, which would give them a score of 1, or an excellent to perfect rating. However, this is using texts that would inevitably perform poorly using the automated annotation method. This is because in each of these texts that come from the television show 'Doctor Who', for example, Gallifrey is mislabelled as a PERSON rather than as a GPE. You would only know this information from watching the show to understand that Gallifrey was a planet and so is considered a geopolitical entity. Another example of this is the Tardis. The automated label labelled it as an organisation and not a product. It is a product because the Tardis is a vehicle in which you can travel. These are just some examples of why the automated method

performed poorly, and if it was on a piece of text which did not have these foreign terms it would have likely performed much better.

Lastly, there was an issue with labelling two characters from the show. It was unable to label ‘Missy’ in some cases and mislabelled the character ‘River’ as a location. Thus, the model completely missed some entities or mislabelled others that are more dependent on the context of the story, to understand that River is a character for example.

In conclusion, the model is very precise as it has broadly higher precision scores across the texts. This means that it does not over-predict too often on random potential entities. Its biggest drawback was missing entities which it likely would not have known without understanding the context behind the show, such as terms like Tardis. To improve this, some manual rules may need to be implemented, for example, to always label ‘Doctor’ as a person, to improve the model’s accuracy.