Contributions

Name (Last, First)	Student ID	Contribution #1	Contribution #2	Other Contributions
Ranges, Lovely	301388497	Task 2	Code debugging	
Hawlader, Antanila	301332035	Task 3 & 4	Group coordination	
Way, Rachel	301435818	Task 1	Proofreading	

Task 1 (B Grade)

Comment on what you have found

For the first portion of the assignment we began by analysing the output that was delivered by Coreferee, looking to see how accurate the package was at properly identifying clusters. We ran five different texts through Coreferee and compared the results to the text file, looking to see if all members of a cluster were included, if there were clusters with wrong members and if there were clusters that should be merged.

Question 1: Are all members of a cluster included?

For the five different texts that were imputed into Coreferee, the first text was assigned 24 clusters, the second was also assigned 24, the third being a longer text was assigned 34 clusters, while the fourth was assigned 26 and the fifth just 15. To answer the question of whether all members of a cluster were included, we opened the actual text documents to compare what Coreferee identified as a cluster and its members compared to what we were able to identify by reading the text and using our knowledge of Linguistics.

While Coreferee was able to identify many of the clusters, we almost immediately noticed clusters that it failed to identify. For example, in the first text, the article talks about a couple who are attempting to adopt a child, Mr. and Mrs. Moran. The program is able to identify Kim and Clark as individuals, as well as a couple, but does not recognize that Kim and Clark are the couple in question, despite identifying the couple in the same sentence. While this may be a more complex identification than what the program can deliver, it is interesting to note that Coreferee does not identify a cluster of Kim and Clark with them, despite being in the same sentence. Coreferee was better able to identify all members of a cluster in other texts, however, failed to identify all members of a cluster in the third text. Where Coreferee was most successful was in the analysis of the fifth text. It was ultimately able to identify all clusters in the document. When attempting to understand why it was able to do so compared to the other texts, our group found that the text in question was rather short compared to the others and did not contain many different subjects which allowed for easier interpretation.

Question 2: Are there clusters with the wrong members?

For the second question, our group analysed the effectiveness of Coreferee by noting the clusters that Coreferee identified, then looking at the text to see whether the clusters that were identified contained any words or entities that didn't belong in the given cluster. We did so by finding the cluster that Coreferee was referring to and identifying the instances it listed within the text. We repeated this task for each of the five documents and did not find much variation in how well Coreferee was able to correctly identify clusters and their members. However, due to the variation in the length of the texts and the language that was used, for example a news article compared to a more passive story, there was small variation in how well Coreferee was able create clusters for each document.

Coreferee was not able to include all members of a cluster in the first document that we analysed and similarly there was an instance of a cluster with the wrong member. In a particular instance, Coreferee grouped the “Government” with the word “it”, however in the case it was referring to, the word “it” was not being used to describe Government. Besides this instance, Coreferee was quite accurate and did not put wrong members into clusters when analysing the first document. However, noticeable errors were found by our group in the third document in particular. For instance, in the third document Coreferee identified a hashtag and later put it in the same cluster as the word “it.” However, when reading the article it can be seen that the two were not meant to be clustered together. Our group felt this could be due to a variety of reasons, most likely because the use of a hashtag is not necessarily typical in text that would be imputed into Coreferee meaning that the text it had been trained on would likely not include many hashtags. Further, the third document also presented a couple of issues for the software which was likely due to the fact that it seemed to be extracted from a webpage. This presented challenges to Coreferee’s ability to accurately identify clusters as headlines for different articles and advertisements were extracted as if they were a part of the article when they were not meant to be included. This could likely be fixed by cleaning the text before inputting it into Coreferee but of course would be more labour intensive.

Question 3: Are there clusters that ought to be merged?

For the third question in this segment of the assignment, our group attempted to find individual clusters that should have been grouped together by Coreferee. To be more specific, we looked at each cluster that Coreferee returned and identified it in the text to see if the Coreferee had identified the same subject more than once. As was expected, there were clusters that should have been merged together. These instances were often in cases when the subject was referred to in different paragraphs of longer texts. Additionally, it occurred particularly in news articles, when how the person was originally referred to was changed part way throughout the text. For example, if a subject was referred to as Nigel Anderson and later Anderson in the text, Coreferee was not always able to distinguish that it was the same subject, consequently categorising them differently.

For the first document there are instances of clusters that should be merged, however they are not because they are separated by paragraphs, such as the ninth and twenty third clusters which both refer to the same person, Kim. The second document in particular seemed to provide many issues for Coreferee in regards to accurately merging clusters together. In just one instance a subject, Campbell, was identified by Coreferee three different times, or

once in each paragraph, despite the article discussing the same entity. When our group investigated this further in the remaining three documents it was found that this was a consistent issue across all texts. In each document that was analysed there were consistent clusters that were identified as separate entities however, the clusters should have been grouped together but were separated because of paragraph breaks.

Task 2 Bonus (for a B+):

Now think of the quotes extracted for the same text. How could you use the coreference chains to find the speaker of the quote? Use at most 1 more page.

Coreference chains can be used to find the speaker of the quote by using coreferee. To find the speaker of the quote, we would need the first pronoun or referent after the end of the quote. When we have the pronoun or referent, we need its index to look through the document, and ask coreferee to resolve it.

For example, in the text:

> Although he was very busy with his work, Peter had had enough of it. 'God, I'm frustrated!', he said. He and his wife decided they needed a holiday. They travelled to Spain because they loved the country very much.

When we extract the quote, ‘God, I’m frustrated!’, we can look for the speaker by searching for a pronoun or referent before or after the quote. We can find the index of he, in our code it’s ‘he(25)’ of the first coreference chain.

We then ask coreferee to resolve it through: “print(doc._.coref_chains.resolve(doc[25]))”, where it returns [Peter]. So we have our speaker, Peter.

Another method that is possible would be to go through ten words before the quote, or ten words after the quote, and find a named Person entity. Both of these methods are tried at the bottom of our code under A2_starter.ipynb.