Information extraction before and after neural coref

Introduction

Coreference resolution is the study of what people meant but didn't explicitly say. Finding references that are related to one another within text documents is a problem that is found when applying coreference resolution. In this report we will look at the effectiveness of applying coreference resolution to text documents before applying an information extraction algorithm.

To help us further research this we used 2 tools: NeuralCoref and OpenIE. NeuralCoref is a type of coreference resolution module that is based on the super fast spaCy parser and uses the neural net scoring model described in Deep Reinforcement Learning for Mention-Ranking Coreference Models by Kevin Clark and Christopher D. Manning, EMNLP 2016 [1] [2]. As for the open information extraction (OpenIE) software developed by Stanford, it refers to the extraction of tuples, typically binary relations, such as (*Mark Zuckerberg; founded; Facebook*)[3]. Within our research we used OpenIE to extract these tuples in a form of triples such as {subject, object, and verb}.

Development was done within a Jupyter lab notebook within an anaconda environment. In this anaconda environment we used both the '*pip*' & '*conda'* to install the necessary python packages. To view more in-depth details regarding what libraries were used as well as instructions on duplicating the used environment, can be found within a github gist that was published for this project specifically [4].

As mentioned, our goal is to see the effectiveness in applying coreference resolution to text documents before applying an information extraction algorithm. In terms of development, we first created a global coreference resolution function that would pass in a string as a parameter and use NeuralCoref to apply the coreferencing to the string so that it could output additional strings that could be passed into the information extraction algorithm. Having done this, we proceeded to develop another global function that would apply an information extraction algorithm, OpenIE, to the passed in parameter, a list of strings. Having these two global functions complete, the necessary data to help evaluate the problem we are trying to research, can be created. A brief overview on how we developed the datasets is that we began by obtaining articles found within WikiData and inserting them into a text file. We began by obtaining the basis of our data by passing in the text file into each of the global functions and writing their individual outputs to different text files. After having our base cases we could create the primary data that would be evaluated, which is passing in the output of the coreference resolution function to the information extraction function.

Something that should be mentioned is that when my peer and I began testing the NeuralCoref library in python using different operating systems, Windows and Ubuntu Linux, different outputs were being given when running the same exact code. This was further tested on other computers, and the provided the same result of having different outputs.

A more in-depth look into our data development can be seen in a write-up on github created for this project [5].

Methods

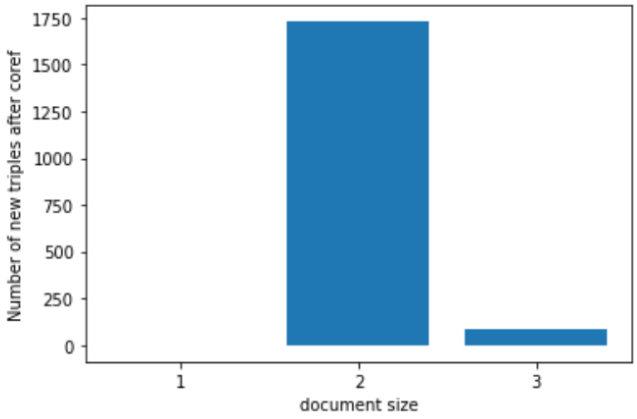
We use 3 different measures to see if NeuralCoref makes a difference in information extraction. Those measures are

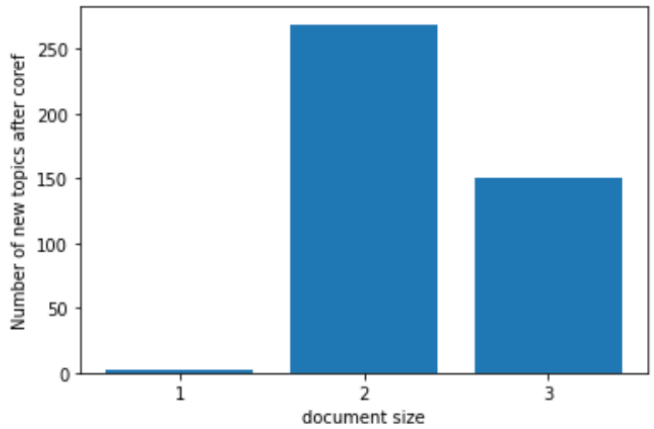
1. Number of new triples generated by OpenIE (not necessarily unique) after using NeuralCoref
2. Number of new topics generated by OpenIE after using NeuralCoref
3. Number of new triples(unique) by OpenIE after using NeuralCoref

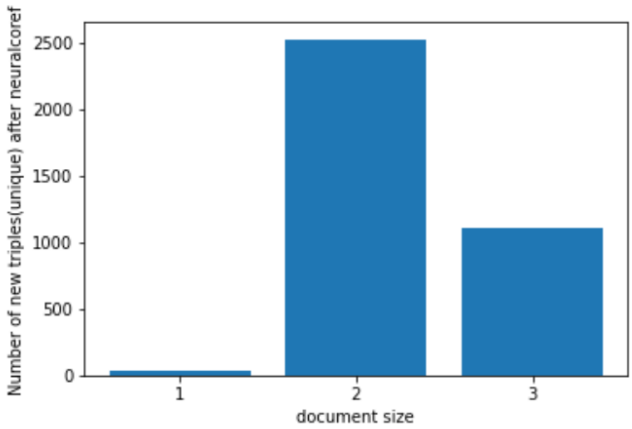
We run these measures on 3 sets of documents. 1 small, 1 large, and 1 extra large.

Results

The results for the 3 measures are provided below. The first is the number of new triples(not necessarily unique, the second is the number of new topics generated after OpenIE, and the third is the number of new unique triples generated by OpenIE after NeuralCoref.







These graphs show that all 3 measures went up from a small document to a large document but dropped when going to an extra large document.

Conclusion

NeuralCoref does help extract more information with OpenIE, but not as much as we expected as going to a very large document did not result in the highest measures for any of the 3 categories that we tested. This could have been due to varying themes of documents, something that would be worth keeping constant in a future study,

References

[1] "neuralcoref · spaCy Universe," *neuralcoref*, 2016. Available: https://spacy.io/universe/project/neuralcoref [Accessed: May 11, 2021]

[2] K. Clark and C. Manning, "Deep Reinforcement Learning for Mention-Ranking Coreference Models,." [Online]. Available: https://cs.stanford.edu/people/kevclark/resources/clark-manning-emnlp2016-deep.pdf [Accessed: May 12, 2021].

[3] "The Stanford Natural Language Processing Group," *Stanford.edu*, 2015. Available: https://nlp.stanford.edu/software/openie.html [Accessed May 12, 2021].

[4] R. B. Cohen, "The Economic Impact of Information Technology," *Business Economics*, vol. 30, no. 4, pp. 21–25, 1995, Available: https://www.jstor.org/stable/23487730?seq=1. [Accessed: May 12, 2021]

[5] 262588213843476, "NeuralCoref + Jupyter Notebook / Lab in Anaconda Environment," *Gist*, Apr. 08, 2021. Available: https://gist.github.com/luisegarduno/a8d9fb39bb2749b316442a0b34d24357 [Accessed May 12, 2021]

[6] luisegarduno, "luisegarduno/NLP-Projects," *GitHub*, 2021. https://github.com/luisegarduno/NLP-Projects/blob/main/Co-reference_Resolution/NeurlCoref%20%26%20OpenIE.pdf [Accessed May 12, 2021].