Named Entity Recognition

Abstract

There has been conscious effort in the NLP community to identify named entities – in other words entities which are supposedly names like characters in a novel or play or locations or anything that we assign a named existence. This task has been thoroughly studied and we have different techniques with varying accuracies to which these are solved. The relationship of these entities in a drama or play and how to draw them out has also been thoroughly contested. Our aim of this project has been twofold – first to identify these named entities. And to identify the relationships between them and the possible causes of the relationships.

Data

The challenge of this project was finding annotated text confirming these named entities and the relationships among them. This was a huge challenge. But with our Professor's guidance we consulted a reference librarian who pointed us out to Folger Digital Library - an online repository of Shakespeare's plays. We worked on five plays namely Comedy of Errors, Titus Andronicus, Julius Caesar, Macbeth, and Merchant of Venice. It has annotated character references for us to evaluate against. However it still did not have relationship annotations.

Techniques

Our first challenge was to find these named entities and test them against the ones annotated. We were first trying out a reference window slice of 10 words. However as we narrowed our domain to plays where our unit of discourse was dialogue – I dynamically arranged for the reference window slice to be of the dialogue. Then used NLTK's pos tagging which tagged the text accordingly with reference to the Penn Treebank. We then used the NNP reference tag i.e. the proper noun tag to find the proper nouns and also used capitalization to look for possible named entities. Also since it was plays – it was easy to find entities who were speaking – as they would typically be at the start of dialogue. The challenge was to find the entities hidden in dialogue. We still managed to find them using wordnet as a lexicon and trying to map words which were absent as named entities tend to not have meanings in English. Also capitalization was used to detect named entities. Another challenge was to include things like first murderer which is different from murderer. We detected possible entities and accounted for words before them as well to make things more distinct. We achieved a hundred percent accuracy on that account. We however had some stray entries because of old English not getting recognized like doust.

We then tried to place these entities as a network graph under the measure of co-occurrence metric. The nodes were the entities and the edges were the co-occurrence measure i.e. how many times these entities occur in the same dialogue. This gives us kind of a relation metric as to how these might be related and also a sense of relationship between these entities. It also gives us a sense of importance of a named entitiy i.e. if they were a major character as the size of each node was weighed according to the degree of the node. The bigger the node the more importance it had. This was verified by the annotations of the text. The annotations were in the increasing order of importance. Also we ran modularity algorithm for the clustering of nodes which clusters nodes based on their edge weights. The clusters gave us cluster of nodes. We wanted to assign what these clusters could be possibly based on and we had assigned the top 10 words according to the tf-idf of these clusters. So basically we used the text pertaining to the certain cluster as a document and for a particular cluster we would have a number of documents and we would have tf-idf for the words of each document and select top ten words of each document. These would help us to give a clue what could be the theme of each cluster - i.e. the basis of relationships for each cluster.

Design

The code was written using python. The visualization was done using sigma.js. We generated the json format for each data visualization. The dashboard has each button for each play and when clicked goes to the individual visualizations for each play. We used color to denote the classes. The visualization has a search bar to search for node names. On hover the names of the nodes are displayed. Size pertains to the importance of the nodes. On hover over a node it also dims the nodes that are not connected to it. Also clicking on a node also shows measures pertaining to the node – like betweenness centrality, the class it belongs to, the weighted degree and also the top tf-idf words of the node.

Result

We have a visualization which gives us all the named entities and all the named entities that are connected to it in a lucid manner. Also we verified the tf-idf scores for the words , if they give a particular meaning to the clusters. For a group in Macbeth where there is first murderer, second murderer, and third murderer we have light and affair which were the causes for the murders. The same can be said for the witches cluster which has witches and cauldrons as the top words, all words pertaining to witches. There are all such relationships which are meaningful which says we have been successful in our clustering. However the only problem is the granularity of the cluster which gives us a more general cluster which really does not give us anything if the cluster size is big.