**Named Entity Recognition on the Wall Street Journal**

Firstly, the journal is read into an array, one file per item of the array (as this makes it easier to write back at the end), and this array is then sent to be tagged. The tagger that has been used is the Brill Tagger as it is much more efficient than most taggers, without having to sacrifice in time efficiency. Upon research, another tagger was suggested, a classifier based tagger, but although the accuracy could be slightly better than the brill tagger, it’s extremely slow (Jacob-StreamHacker April 2010) and so the decision was made that, since the corpus is so large, the marginal improvement in accuracy wouldn’t be worth the large loss in time efficiency.

The brill tagger uses a wrapper created by Harish Tayyar Madabushi, and the tagger is backed off onto the trigram tagger, which is backed off to the bigram tagger. The bigram is backed off to the unigram tagger, and the unigram tagger is backed off to the default tagger, which automatically tags the word as a named entity (NNP). The back-offs improve the accuracy of the brill tagger as there are then more statistics that can be used about the data, in order to tag it.

Another way the accuracy of the tagger was improved was by adjusting the cut-off points. Tests were run over twelve hours to test all the possible combinations of cut-offs with taggers and back-offs and the results showed that the tagger worked most accurately when the cut-off for the trigram tagger was four, and one for both the bigram and unigram taggers (see photo in references).

The brill tagger was trained on the Penn Treebank as it contains the Wall Street journal text, and would again greatly improve the accuracy of the tagger, when used on the test data. Once the data was tagged, the program searches through it, looking for words which have been tagged as named entities. It also checks that all these words that have been tagged in this way, begin with a capital letter (as all named entities do and this helps to get rid of anomalies). It collates these all into one list, and also combines adjacent words which have both been tagged as named entities, as these will most probably be one single entity. E.g. ‘Michael’ and ‘Jackson’ would both be tagged as named entities, so the program combines them into one entity.

Next the program uses lists of given surnames, male first names, female first names and titles in order to classify some names in a simple and time efficient way, so that they don’t need to be queried using Wikipedia.

The possible named entities are then sent to a function where Wordnet is used to try and define the words. This can give a very simple definition, where it is easy to work out the nature of the word, i.e. if it’s a location, person, organisation or anomaly. If a definition is found, then specific words are looked for, within the definition, which are indicators as to the type of word it is (these have been predefined as lists). The word is given a score for each category; person, organisation, location, based on how many of the indicator words were found in the text, for that category. The category with the highest score, is the category that the word is assigned to.

If Wordnet has no definition for the word, then Wikipedia is used to retrieve data on the word. The program collects url data from Wikipedia on the word, and finds all the possible page titles for this word. It then collects the information under that title and categorises the word, based on the data, in the same way as it did before.

**Evaluation:**

My tagger is good at tagging the named entities, but also tags non-named entities as named entities; so if the accuracy of the tagger improved, I feel the program would be that much more effective. Saying this, the named entities that are tagged as so, then appear to be categorised correctly as well, most of the time. Those which are tagged as people are tagged correctly most of the time, and this is the most accurate, and location is the least accurate, where roughly half of the entities which are tagged as locations, are actually locations. I collated adjacent words which had been tagged as named entities, and tried to use Wikipedia to search for them, but this was unsuccessful, so if I had more time, this would be something I would look into further.

**Bibliography:**

Jacob (2010) Part of speech tagging with NLTK part 4 – Brill Tagger vs Classifier Taggers. Available at: http://streamhacker.com/2010/04/12/pos-tag-nltk-brill-classifier/ (Accessed: 4 December 2016).