CS 839 Project-1

by Vishal Agarwal, Roshan Lal and Pratham Desai

We decided to extract "person names" from wikinews articles.
For example:

- <name>Alan Joyce</name>, CEO of Qantas, said
- United States President <name>Barack Obama</name> said he "would have fired him."
- Wikinews reporter <name>Iain Macdonald</name> has performed an interview with Dr <name>Isabella Margara</name>
- Governor <name>Rick Snyder's</name> plan for the schools

We marked up a total of 320 documents containing 1668 person names.

- Set I : num_docs = 214, num_mentions = 1086
- Set J : num_docs = 106, num_mentions = 582

The classifier M was ***decision tree*** and it gave us the following metrics:

- Precision: 0.65
- Recall: 0.43
- F1: 0.52

We ran several iterations of debugging and cross-validation after this.

Our classifier X was ***SVM***. Decision Tree, SVM and Random Forest were kind of close, but SVM had a slightly higher precision and recall.

- Precision: 0.92
- Recall: 0.73
- F1: 0.81

We did not do rule based post processing. So classifier Y = classifier X = SVM. Thus, the metrics on Y are same as what we obtained previously:

- Precision: 0.92
- Recall: 0.73

- F1: 0.81

Some insights we had during the assignment:

- Some document contained non-english letters and unicode characters. We had to do **data cleaning** to remove them before passing it to our algorithm.

- For generating examples, we considered **n-grams** with length upto 4 words.

- To handle apostrophe marks signifying possession, we removed them while generating examples. For example, if the tagged document had "... <name>Obama's<name> policies …", the positive example generated will be the string **Obama.**

- Initially, we had very low precision and recall on the ML algorithms. On debugging, we realised that it was because of the low ratio (~1:10) of positive:negative examples. Then we tried to increase this ratio by reducing the amount of negative examples, which resulted in a significant improvement of precision and recall.

- <u>Features explored:</u>
    - Length of n-gram (int)
    - Are all letter capital? (boolean)
    - Does the n-gram have a "title" before it? [Mr, Dr, Ms etc] (boolean)
    - proximity to a term indicating occupation [President, CEO etc] (float)
    - proximity to a term indicating specific verbs [said, told etc] (float)
    - is this n-gram preceded by an Article ['a', 'an', 'the' etc] (boolean)
    - frequency of this n-gram in the document (int)
    - Does this n-gram occur at the beginning of a sentence? (boolean)

  We did experiments to select the most useful features from this list.

- ML algorithms used: Decision tree, Random Forest, Support Vector Machine, Linear Regression, Logistic Regression