CS 839 Data Science

Project Stage 1

＜Group 9＞

1. The names of all team members.
   - Abbinaya Kalyanaraman (akalyanaram2@wisc.edu)
   - Bob Effinger (bdeffinger@wisc.edu)
   - Rajan Dalal (rdalal@wisc.edu)
2. The entity type that you have decided to extract, give a few examples of mentions of this entity type.
   - We decide to extract person names from the CNN Dataset. For instance,“Secretary Hiliary Clinton commended the swift passage of the resolution”, we would want to extract Hiliary Clinton, and " Jamaica's Powell, who took 100m bronze at the last championships ", we want to extract Powell.
3. The total number of mentions that you have marked up.
   - 4864
4. The number of documents in set I, the number of mentions in set I.
   - 200 documents, number of mentions is 3154
5. The number of documents in set J, the number of mentions in set J.
   - 100 documents, number of mentions is 1710
6. The type of the classifier that you selected after performing cross validation on set I *the first time*, and the precision, recall, F1 of this classifier (on set I). This classifier is referred to as classifier M in the description above.
   - Type of the classifier: Random Forest
   - Precision: 0.562927
   - Recall: 0.415136
   - F1: 0.577
7. The type of the classifier that you have finally settled on *before* the rule-based postprocessing step, and the precision, recall, F1 of this classifier (on set J). This classifier is referred to as classifier X in the description above.
   - Type of the classifier: Random Forest
   - Precision: 0.520273
   - Recall: 0.391961
   - F1: 0.544
8. If you have done any rule-based post-processing, then give examples of rules that you have used, and describe where can we find all the rules (e.g., is it in the code directory somewhere?).

   - After quite a lot of discussion and experimentation, and considering the wide diversity in our source files, we decided on the following list of features:
   - capitalizations : count of number of words starting with capital letters, except for certain uncapitalized name sequences (such as 'van' or 'bin')
   - 'al' check : Given that a number of our articles were of Middle East coverage, this feature is True if a word in a sequence begins with 'al-' (as in 'Bashar al-Assad')
   - prefix check : check if a prefix from a given list precede the word(s). Examples

include 'dr', 'sen.' (short for senator), 'secretary' etc.

- suffix check : check if a suffix from a given list succeeds the word(s). After much testing on our dev set, this list contains only 'administration' (as in 'Obama administration')
- verb check : check if a verb from a given list precede the word(s). The list is "said", "told", "asked", "called". These usually indicate that the next word is a person.
- comma number after : A lot of our articles contained the age of the person after the mention, such as ('Belgian <n>D'Ambrosio<\n>, 26, tested for Lotus in preseason and drove for Virgin Racing last year' : dev file 160)
- parenthesis check : A lot of our articles were also about the movie industry, where character names are followed by the actors playing them
- hyphen check : To take into account names like 'Ban Ki-Moon'
- prefix article : Feature that tracks if a / an / the preceded the word. These articles are generally not followed by names.
- prefix preposition : Feature that tracks if in / on / at preceded the word. These prepositions are generally not followed by names.
- ‘atter checker : Almost every article contained an '@...' word such as '@highlight' or the twitter handle of a person. Since these words are placed in ungrammatical places, a feature to keep track of them was thought to be a good idea.
- comma middle : Checks if there's a comma in the middle of the words, indicating a non name n-gram
- possessive check : Checks if a word in the sequence is followed by the possessive "'s" which usually indicates a noun
- num words : number of words being considered, from 1 upto 4.
- check punctuation : similar to comma middle, but for terminator punctuations (. / ? / !).
- jr / sr check : checks if words like 'jr / sr / III' etc appear in the sequence
- common word checker : checks if the sequence contains a common stopword. Usually these are not names.
- len word : checks if any word in the sequence has length <= 2
- last capital : checks if the word(s) preceding were capitalized or not, and did not end with a possessive
- various other statistic features such as vowel percentage, phrase and average word length, whether it contains a digit and whether at least one lower case alphabet was present.
- The code can be found in vectorizer.py file in the code folder.

9. Report the precision, recall, F1 of classifier Y (see description above) on set J. This is the final classifier (plus rule-based post-processing if you have done any).
   - Precision: 0.577
   - Recall: 0.544
   - F1: 0.56

10. If you have not reached precision of at least 90% and recall of at least 60%, provide a discussion on why, and what else can you possibly do to improve the accuracy.
    - Our function did not meet the requirements. In order to improve our accuracy further there are a few things that we need to do. The First thing that we need to do to improve accuracy is to refine our Entity Type. Currently our domain is a bit too large. Our domain is currently all types of news that can be found at CNN. The size of this domain is definitely causing issues with our classification task. Due to some articles being about sports while other are focused on the war in the middle east our marked data lacks the focus required for this project.

    - In addition to the above, we focused a large amount of our efforts on creating features that focus on the spatial information of the data. These were things such as was there a prefix prior, did a comma shortly follow. However while these helped us to get good precision in our cross_validation step, these gave very low recall. Overall we would spend more time focusing on different features that help our models to reach higher recall and precision. In addition to generating new models we need to spend more time figuring out which of our features are giving meaningful input to our models, and which are functioning as noise.

    - The last thing that we would do to improve our accuracy further is by spending more time tinkering with the parameters of each of our models, we didn't have the necessary time to do as much tuning as we would have liked with the parameters that are used to generate the models.