CS838 - Project Stage-1

Person Name Extraction from BBC News Article Text Files

Manjunath Nagaraj Shettar (shettar@wisc.edu)
Jayashankar Tekkedatha (tekkedatha@wisc.edu)
Samhith Venkatesh (svenkatesh5@wisc.edu)

Introduction

Dataset Source

Politics news of BBC Dataset is taken as the dataset for this project.

Entity Type:

We chose **person name** as the entity to be extracted from the given dataset. We marked the entities with **<person>** and **</person>** tags in 300 text files.

Description

1. A total of **1732** person names were tagged among the above-mentioned documents.

2. Number of documents in set I: **200**

3. Number of documents in set J: **100**

4. **Preprocessing:** We noticed that the names present in the headlines of the document had lot of inconsistencies, e.g. "*Blair government under attack*" instead of "*Blair's government under attack*". Hence, as a part of pre-processing step we **remove** the first line from all the 300 documents during the runtime.

5. **Generation of Negative Samples**:

Since the person names in our dataset was consistent throughout the document i.e all the person names were capitalized, instead of creating the n-gram of all possible phrases in our dataset, we created that n-gram of only capitalized phrases. **This helped us in pruning away most of the negative samples.**

6. **Features Chosen**

1. **Words ending with Apostrophe:** If a capitalized phrase is ending with apostrophe then it is chosen as feature for a name.

2. **Words followed by punctuations:** If a capitalized phrase is followed up by a comma or full stop then it can be a potential feature for the name.

3. **Words present at the beginning of the line:** If a capitalized phrase is starting at the beginning of the line then it is chosen a feature for name.

4. **Words with possible prefixes:** If a capitalized phrase is followed by prefix word belonging to the list *['and', 'said', 'leader', 'justice', 'chairman', 'secretary', 'chancellor', 'reporter', 'journalist', 'pm', 'mp', 'candidate']* then it is chosen as weak feature.

5. **Words with possible prefixes:** If a capitalized phrase is followed by the prefix word belonging to the list [*'mr', 'ms', 'mrs', 'dr', 'lord', 'sir', 'lady', 'prince', 'minister', 'director', 'president', 'spokesman', 'spokeswoman', 'spokesperson', 'prime minister'*] then it is chosen as a strong feature. We noticed that in most of the cases a person name will be followed words present in the list, hence it is provided as definite feature.

6. **Words followed by punctuation:** If a name is followed by comma, full stop punctuation marks then it can be a potential feature for name. If the person name is followed by comma or full stop, couple of other features are directly **set to zero** to avoid the corner cases or unnecessary work.

7. **Words followed by list of suffixes:** If a name is followed by words present in the following list *['said','told', 'claimed', 'mp', 'spokesman', 'spokesperson', 'spokeswoman', 'sr', 'jr', 'and', '-', 'has', 'had']* then it could be a potential feature for the person name.

8. **Words followed by strong suffixes:** We noticed that the words present in the list *['is', 'and']* occurred next to most of the capitalized phrases which were names, hence this feature was provided separately to the model.

9. **Words with exhaustive list:** An exhaustive list was made combining the words present in the lists presented in feature 4,5 and 7. If any of capitalized phrases contain the words present in these lists, then it is a feature to identify the negative samples. **Note**: **This helped to us prune away most of the negative samples.**

10. **Words with suffix and prefix:** If a capitalized phrase has a one of the salutations *['mr','ms','mrs','dr']* as prefix and is followed by a capitalized word then it cannot be a name. This feature will help us identify the negative samples which are partial names.

11. **Words with suffix and prefix:** If a capitalized phrase is present between words starting with small letters, then it can be potential feature for identifying a name without any standard prefix or suffix list. **Note: This vastly helped us in increasing the precision as the number of positive samples were identified properly.**

12. **HASH code of the prefix and suffix:** This feature was included to make sure that if word was repetitively present in suffix or prefix and yet not present in our predefined list, then the model would learn its occurrence based on the hash code value calculated from the MD5 hash function.

7. Cross Validation was performed for following classifiers:
    1. Decision Tree
    2. Random Forest
    3. Support Vector Machines
    4. Logistic Regression
    5. Linear Regression

8. **Stop-Words –** To further prune the negative samples, we identified the list of standard English language stop-words and included it in our code. This helped is in bringing down the negatives samples by higher value.

9. **Random File Generator Code**: To split the 300 samples into I and J we wrote a random file generator code which randomly picks files and separates the 300 samples to test and train.

10. Performance of above classifiers on the **set I**:

Classifier	Precision	Recall	F1
Decision Tree	0.82	0.68	0.74
Random Forest	0.48	0.79	0.60
SVM	0.67	0.80	0.73
Logistic Regression	0.71	0.72	0.72
Linear Regression	0.72	0.59	0.64

Performance of classifiers after adding features 11,12 and cleaning the data:

Classifier	Precision	Recall	F1
Decision Tree	0.87	0.71	0.78
Random Forest	0.55	0.84	0.66
SVM	0.71	0.85	0.77
Logistic Regression	0.78	0.78	0.78
Linear Regression	0.79	0.64	0.71

Classifier Chosen: **Decision Tree**

11. Performance of Decision Tree Classifier on Test Set J:

Classifier	Decision Tree
Precision	0.91
Recall	0.73
F1	0.79

12. Reason for less precision in the Train phase:

    1. The uneven distribution of the positive and negative samples might be the reason for drop in the accuracy during the train phase. If the classifier is trained on folds containing only negative samples, and if our fold contains only + sample, then there is high probability that that sample might be classified as negative. Thus, we “believe” that this might be the reason for not reaching 90+ accuracy in the train phase.
        a. **Solution:** As the number of negative candidates in our datasets is more that the positive ones, we observed increasing the **number folds** in cross validation calculation increased the overall average precision and recall.

Conclusion:

“Know you model, know your data”, We noticed that the longer we spent on the looking at the data, the better our features were. Features 11 and 12 mentioned above helped us in boosting our accuracy rates. We believe that addition of few more features for the names appearing in the middle of the sentences could still improve our accuracy levels. But for now, we are good to go and test our model on the real-world politics data.