CS839 - Project Stage 1

Chakshu Ahuja, cahuja2@wisc.edu
Geetika, geetika@wisc.edu
Swati Anand, sanand@cs.wisc.edu

Dataset

Data Source	News articles are taken from http://mlg.ucd.ie/datasets/bbc.html
Description	These articles are taken from the Entertainment category in BBC News and were found to have many person name mentions
Total articles	386
Labeled articles	300
Entity Type	Person Names

Examples

Dame Muriel Spark is among three British authors who have made the shortlist for the inaugural international Booker Prize.

Doris Lessing and Ian McEwan have also been nominated. McEwan and Margaret Atwood are the only nominees to have previously won the main Booker Prize.

Gabriel Garcia Marquez, Saul Bellow, Milan Kundera and John Updike also feature on the 18-strong list of world literary figures.

But other past winners of the regular Booker Prize, such as Salman Rushdie, JM Coetzee and Kazuo Ishiguro have failed to make the shortlist.

Document Labelling

Data Set	Number of Documents	Number of Person Names
I (Training Data)	200	2118
J (Test Data)	100	1322
B (All Documents)	300	3440

Candidate Token Generation

We generated all tokens of word length 1 - 4 in the labeled files. If a token appears between the and tag, they are marked as positive labels and rest all tokens are marked as negative labels.

Total number of generated tokens: 231,696 (Training Data)
113,064 (Test Data)
344,760 (Total)

Since the number of generated tokens were very large, we decided to block some candidates to consider only the cases where the probability of finding names is large (around 99.99%).

Candidate Blocking

We have applied the following blocking strategies to eliminate extra tokens with negative labels:

- Remove tokens where no words are capitalized
  Example: "Elon Musk owns Tesla". The list of tokens generated are as follows:

Elon	***Elon Musk***	*Elon Musk owns*	*Elon Musk owns Tesla*
Musk	*Musk owns*	*Musk owns Tesla*	
owns	*owns Tesla*		
Tesla			

Our candidate tokens used for training and testing only include Elon, Musk, Tesla and Elon Musk

- Remove tokens with special characters (**, . ! 's s'**) in the first or middle word (eg, "*Musk's Tesla*"; this will not be considered as a candidate token as the first-word *Musk's* includes a special character 's.

- If the frequency of the token in a document is greater than the threshold value **10**, that will not be considered as the candidate token. (Eg, tokens such as *The*, *She* will not be considered as candidate token)

After the candidate blocking, the number of tokens generated is:

Data Set	Candidate Tokens	Positive Tokens	Negative Tokens
I	17524	4978	12546
J	9674	3417	6257
B	27198	8395	18803

Similar distribution of positive and negative candidates was observed across all set of randomly generated I and J sets.

Feature Generation

We identified features according to the patterns observed while marking the labels. Some of the features include:

1. Whether the token contains name prefix (eg, Mr, Ms, Mrs, Sir, Professor)
2. Whether the token contains suffix (eg, Jr, Sr, XI, I)
3. Token ends with ‘s (eg, Elon Musk’s Tesla)
4. Whether the token has a partial name occurrence in the document (eg, Jennifer Lopez is referred to as Lopez in the document after the first full name occurrence)
5. Has full name occurrence (eg, Lopez was earlier referred to like Jennifer Lopez)
6. Preceded by occupation word (eg. Actress Emma Stone)
7. Succeeded by occupation word (eg. **La La Land**’s actress Emma Stone; here La La Land should not be considered as a name)
8. Preceded / Followed by family relation (eg. Sylvester’s mother Katherine Jackson)
9. Has preposition (eg, Emma Stone performed in **Los Angeles;** here Los Angeles should not be considered as a name)
10. Whether the token is a common word (eg, words such as Meanwhile, But, There etc)
11. Whether the token is a common name - we have used a dictionary of 2000 popular first names and last names used in the United States

Apart from these, we have also used a few other features such as the total length of the token, number of words in the token, whether the token is a popular location (country/city) etc.

Classifiers

We used the following classifiers:

Neural Network	Hidden layers - *2*, Nodes - *30*, Solver - *adam*, Activation - *relu*
Random Forest	Number of estimators - *1000*, Criteria - *entropy*
Decision Tree	Criteria - *entropy*
Support Vector Machine	Gamma - *scale*, Tolerance - *0.0001*
Logistic Regression	Multi class - *ovr,* Solver - *lbfgs*
Linear Regression	Max iterations - *1000*

Cross-Validation

We have used a 10-fold stratified sampling on our training data set for cross-validation. Cross-Validation results from all classifiers on Training Set I are as follows:

		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Avg
NN	P	86.2	83.6	85.3	85.5	82.6	83.1	81.1	82.4	78.8	80.5	**82.9**
	R	79.5	79.1	83.1	82.1	91.7	84.9	79.7	78.1	79.6	79.8	**81.7**
RF	P	81.0	80.0	82.9	81.8	80.2	80.5	79.2	78.4	76.8	77.6	**79.8**
	R	80.9	79.5	80.9	83.5	87.3	83.1	75.9	79.1	79.2	80.8	**81.0**
DT	P	77.7	81.1	78.8	75.9	82.9	79.0	78.7	76.0	82.3	77.5	**79.0**
	R	81.1	80.5	74.5	81.3	81.8	77.8	84.7	80.5	85.4	83.6	**81.1**
SVM	P	80.7	83.2	83.8	85.8	88.9	84.7	85.8	86.3	86.2	79.7	**84.5**
	R	75.1	76.5	72.4	77.1	80.5	76.5	83.4	80.0	77.3	70.7	**77.0**
LOR	P	81.5	85.0	84.6	85.5	89.2	82.2	85.8	86.4	84.3	82.3	**84.7**
	R	77.6	78.5	74.0	78.2	79.6	73.3	85.1	81.1	80.2	76.1	**78.4**
LR	P	78.5	83.9	84.3	83.7	90.3	83.1	85.6	85.7	83.4	79.6	**83.8**
	R	76.2	76.7	70.2	74.5	79.4	73.6	83.3	80.0	78.0	72.0	**76.4**

Our precision and recall were around **83%** and **82%** for the best performing model (Neural Network classifier). We then tried adding a few more features to our model but realized that our precision did not increase by more than a percent. Then we decided to compromise recall to gain a higher precision and thus we changed the prediction probability threshold of our model from default 0.5 to 0.78. This increased our precision to **91%** but dropped the recall to **64%**. ((+/-) 5%)

		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Avg
NN	P	95.8	91.1	92.4	90.7	87.5	88.7	88.1	91.0	86.0	88.8	**90.0**
	R	60.4	66.0	64.0	67.2	78.9	69.8	62.6	59.0	59.7	62.5	**65.0**
RF	P	87.9	87.2	90.0	87.1	86.0	91.4	85.6	87.1	84.0	85.9	**87.2**
	R	66.0	64.8	65.0	62.6	70.6	68.6	62.2	63.8	62.7	64.1	**65.1**
DT	P	78.1	79.3	82.6	85.3	87.9	83.7	83.0	82.3	81.4	81.7	**82.5**
	R	70.1	61.7	67.1	71.9	74.5	66.2	74.2	74.0	67.6	70.4	**69.8**
SVM	P	86.5	85.9	91.4	93.4	94.6	88.6	87.3	87.9	87.5	88.3	**89.1**
	R	66.8	59.4	62.0	70.3	68.9	64.8	69.6	70.6	64.6	62.8	**66.0**
LOR	P	89.2	87.9	91.2	94.8	95.1	88.8	88.3	89.7	90.2	88.3	**90.3**
	R	64.7	55.3	58.9	65.1	65.5	63.0	65.5	66.6	58.8	58.5	**62.6**
LR	P	96.6	92.6	96.0	97.8	99.4	93.3	92.6	94.7	93.5	95.7	**95.2**
	R	30.8	22.2	29.8	31.6	32.8	29.6	33.2	35.1	25.7	27.2	**29.8**

Test Results

In the Cross Validation phase, we found that the **Neural Network** was the best performing model. So we ran it on our test data set J and our results are as follows.

Precision **92.731 %**
Recall **61.604 %**
F-score **74.015 %**