Project Report Stage Two

Hongyi Wang, Hao Fu, Miao Yang

March 13, 2017

1 Name of All Team Members

- Hongyi Wang
- Hao Fu
- Miao Yang

2 Entity Extraction

In the stage two of this project, we implemented information extraction (IE) from the text files we collected in stage one, using an unsupervised learning approach. Specifically, the entity we are going to extract is the **name of university**.

2.1 About the Data Set (Mentions)

In our data set, we have entirely 1522 instances (i.e. mentions or data points). In addition, we have 1031 mentions, which were extracted from 206 documents from our text files in set I (we call it dev set in the following), and 491 mentions, which were extracted from 102 documents from the text files in set J (we call it test set in the following).

2.2 How we Implement Information Extraction (IE)

We manually labeled 1322 mentions, which contains 659 positive mentions and 623 negative mentions. We also generated 200 negative mentions randomly from all raw text files. The rule we used to generate the random negative mentions can be find at:

https://github.com/samfu1994/cs838webpage/blob/master/code/extra_neg.py

The mentions that labeled "True" (i.e. positive labeled mentions) were always the real university names, the mentions that labeled "False" (i.e. negative labeled mentions) were chose manually from each text files. We use the format "<p university name p>" to label positive mentions, and format "<n fake-university name n>" to label negative mentions. For further use, we also distinguished private university names and public university names. In specific, we labeled private university name using "<p1 private university p1>" while public university name using "<p2 public university p2>". Here we show a example of the labeled text files:

<p1 Harvard University p1> is a private institution that was founded in 1636. It has a total undergraduate enrollment of 6,699, its setting is urban, and the campus size is 5,076 acres. It utilizes a semester-based academic calendar.
<p1 Harvard p1>'s ranking in the 2017 edition of Best Colleges is National Universities, 2. Its tuition and fees are $47,074 (2016-17).
In addition to the College, Harvard is made up of 13 other schools and institutes, including the top-ranked Business

School and Medical School and the highly ranked Graduate Education School, School of Engineering and Applied Sciences, Law School and John F. Kennedy School of Government. Eight U.S. presidents graduated from Harvard College, including Franklin **<n Delano n>** Roosevelt and **<n John F. Kennedy n>.**

After labeling all text files, we wrote a script for IE to extract labeled mentions, and feature information contained. All our code for this project can be found at:

https://github.com/samfu1994/cs838webpage/tree/master/code

2.3 Entity Type

The features we extract in this stage from all mentions contain:

- length of mentions (i.e. the string)
- if the word "university" contains in the mention
- if dash (i.e. -) contains in the mention
- if the mention contain number

The labels in our extracted data instances are "True" (i.e. the mention is a university name) or "False" (i.e. the mention isn't a university name)

2.4 Entity Examples

We give several examples of mentions here:
MIT, Rochester Institute of Technology, University of California–Riverside, UCR, ASU-Temple, CUA, and etc. The those foregoing mentions were all labeled True in our data set.
Here are also several False labeled instances:
School's, Undergraduate, NCAA-division, Institution, and etc.

3 Performance on Different Models and Model Selection

3.1 How We Implement Machine Learning

For dev set, the training set is shuffled at the beginning, instead of implementing simple cross validation, we implement k**-fold cross validation**, i.e. split the whole training set into k subsets, and then use $k-1$ of k subsets of training set to train the model, and test the trained model on the remaining set. Under this case, we trained k models for each ML algorithm, and we select the one with highest cross validation accuracy. In particular, we implement 8-fold for cross validation.
We tried following ML models:

- **SVM** with RBF kernel
- **Decision Tree** using CART Algorithm
- **Random Forest**
- **Logistic Regression**
- **Linear Regression** (set 0.5 as threshold, with no $l1$ (LASSO) or $l2$ (Ridge) regularization)
- **Neural Network** with one hidden layer, which contains 15 hidden units with an extra bias unit. We use SGD method with batch size of 64 data points, and constant learning rate ($\gamma = 0.001$)

After training, we test the trained model on J set. The results is given as follow:

Evaluations Among Ml Models			
ML Models	Precision	Recall	F_1 Score
SVM	0.8277	0.8874	8565
Decision Tree	0.8270	0.8829	0.8540
Random Forest	0.8243	0.8874	0.8547
Logistic Regression	0.9810	0.6982	0.8158
Linear Regression	0.9910	0.6351	0.7768
Neural Network	0.9891	0.6847	0.8128

Clearly, we found that, using these features, we can get relatively good results using several models, especially linear regression and neural network. However, there are trade-off between precision, recall, and F_1 score. Thus, we determine to do some rule-based post-processing to obtain better results.

3.2 Any Rule-based Post-processing?

In previous step, when we tried to debug our develop set I, we find that mentions like "MIT", "UWM", and other public university names were wrong predicted for most of times. Then, we choose to add following features:

- if the word "state" contains in the mention
- if a name of state (e.g. Wisconsin, California, and etc) contains in the mention
- if all letters in the mention are capital (e.g. MIT)

After this post-processing, we use the same method mentioned before to train the models on I set, and test on J set. And we give the final result here.
To give a result of trained models, it's relatively easy to plot the trained decision tree, while it's quite hard to visualize other trained models. Therefore, we give visualized our trained decision tree in Fig1. The results on different models are given below, also visualized in Fig2:

Evaluations Among Ml Models			
ML Models	Precision	Recall	F_1 Score
SVM	0.9648	0.9357	0.9500
Decision Tree	0.9649	0.9406	0.9526
Random Forest	0.9739	0.9455	0.9595
Logistic Regression	0.9546	0.9369	0.9457
Linear Regression	0.9647	0.9307	0.9474
Neural Network	0.9649	0.9405	0.9525

Due to the result, obviously the final result from **random forest** is the best. Thus we finally select the result from random forest.

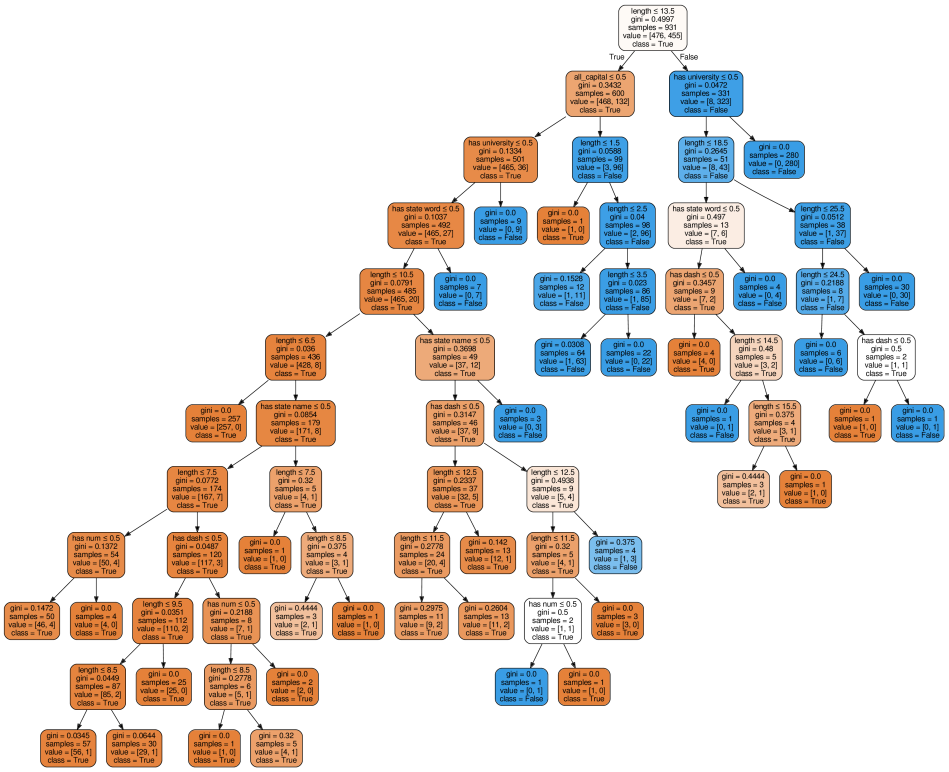

Figure 1: Trained Decision Tree

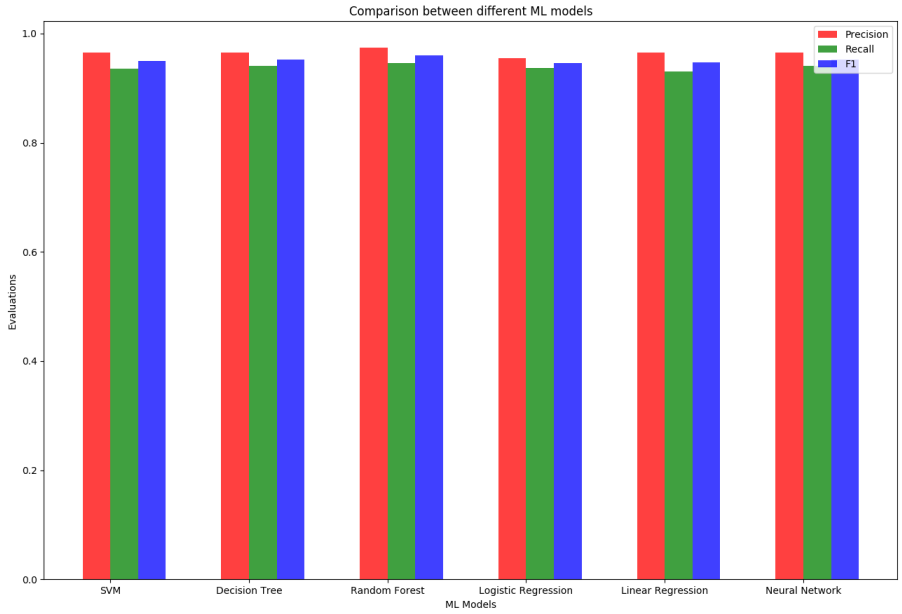

Figure 2: Comparison among ML Models