Project Report Stage Two

Hongyi Wang, Hao Fu, Miao Yang

March 12, 2017

1 Name of All Team Members

- Hongyi Wang
- Hao Fu
- Miao Yang

2 Entity Extraction

2.1 Entity Type

In the stage two of this project, we planed to extract the **name of university** from the text files we collected in stage one. The features we extract from all mentions contain:

- length of mentions (i.e. the string)
- if the word "university" contains in the mention
- if the word "state" contains in the mention
- if a name of state (e.g. Wisconsin, California, and etc) contains in the mention
- if dash (i.e. -) contains in the mention
- if the mention contain number
- if all letters in the mention are capital (e.g. MIT)

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)

2.2 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.

2.3 How we Implement Information Extraction (IE)

All mentions are label manually, the mentions that labeled "True" were always the real university names, the mentions that labeled "False" were chose randomly 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 Educa-
tion 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 that implement the hand-craft rules, which using regexes and dictionary based method 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`

3 About the Data Set (Mentions)

In our data set, we have entirely 1322 instances (i.e. mentions or data points). In addition, we have 931 mentions, which were extracted from 206 documents from our text files in set I (we call it dev set in the following), and 391 mentions, , which were extracted from 102 documents from the text files in set J (we call it test set in the following).

4 Performance on Different Models and Model Selection

4.1 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.
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)

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:

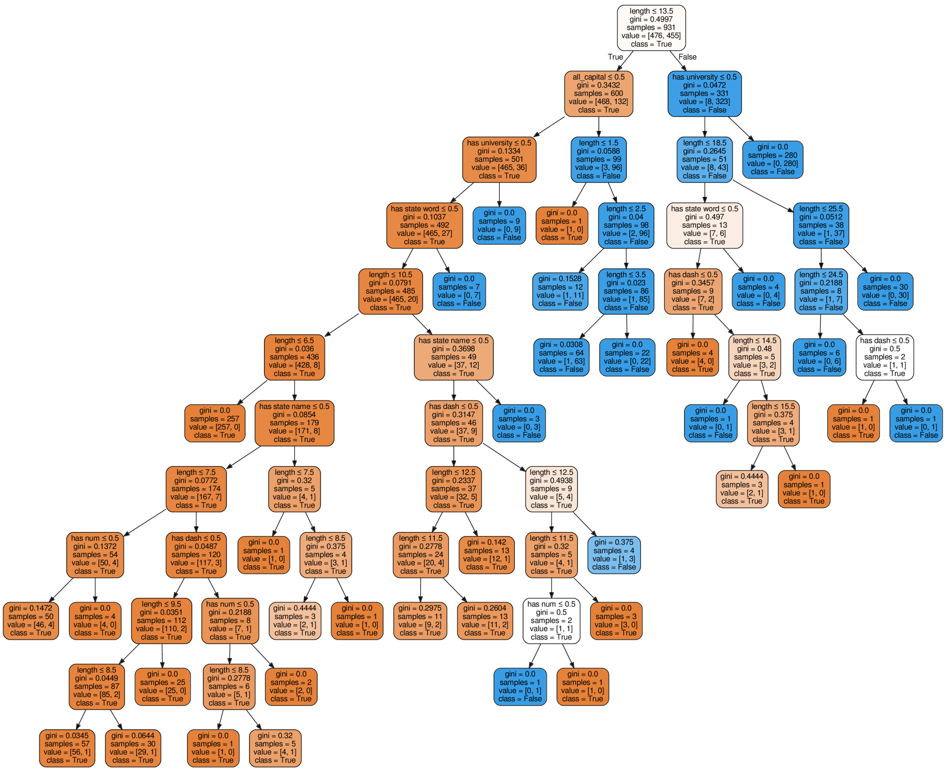
length ≤ 13.5 gini = 0.4997 samples = 931 value = [476, 455] class = True
True
False
all_capital ≤ 0.5 gini = 0.3432 samples = 600 value = [468, 132] class = True
has university ≤ 0.5 gini = 0.0472 samples = 331 value = [8, 323] class = False
has university ≤ 0.5 gini = 0.1334 samples = 501 value = [465, 36] class = True
length ≤ 1.5 gini = 0.0588 samples = 99 value = [3, 96] class = False
length ≤ 18.5 gini = 0.2645 samples = 51 value = [8, 43] class = False
gini = 0.0 samples = 280 value = [0, 280] class = False
has state word ≤ 0.5 gini = 0.1037 samples = 492 value = [465, 27] class = True
gini = 0.0 samples = 9 value = [0, 9] class = False
gini = 0.0 samples = 1 value = [1, 0] class = True
length ≤ 2.5 gini = 0.04 samples = 98 value = [2, 96] class = False
has state word ≤ 0.5 gini = 0.497 samples = 13 value = [7, 6] class = True
length ≤ 25.5 gini = 0.0512 samples = 38 value = [1, 37] class = False
length ≤ 10.5 gini = 0.0791 samples = 485 value = [465, 20] class = True
gini = 0.0 samples = 7 value = [0, 7] class = False
gini = 0.1528 samples = 12 value = [1, 11] class = False
length ≤ 3.5 gini = 0.023 samples = 86 value = [1, 85] class = False
has dash ≤ 0.5 gini = 0.3457 samples = 9 value = [7, 2] class = True
gini = 0.0 samples = 4 value = [0, 4] class = False
length ≤ 24.5 gini = 0.2188 samples = 8 value = [1, 7] class = False
gini = 0.0 samples = 30 value = [0, 30] class = False
length ≤ 6.5 gini = 0.036 samples = 436 value = [428, 8] class = True
has state name ≤ 0.5 gini = 0.3698 samples = 49 value = [37, 12] class = True
gini = 0.0308 samples = 64 value = [1, 63] class = False
gini = 0.0 samples = 22 value = [0, 22] class = False
gini = 0.0 samples = 4 value = [4, 0] class = True
length ≤ 14.5 gini = 0.48 samples = 5 value = [3, 2] class = True
gini = 0.0 samples = 6 value = [0, 6] class = False
has dash ≤ 0.5 gini = 0.5 samples = 2 value = [1, 1] class = True
gini = 0.0 samples = 257 value = [257, 0] class = True
has state name ≤ 0.5 gini = 0.0854 samples = 179 value = [171, 8] class = True
has dash ≤ 0.5 gini = 0.3147 samples = 46 value = [37, 9] class = True
gini = 0.0 samples = 3 value = [0, 3] class = False
gini = 0.0 samples = 1 value = [0, 1] class = False
length ≤ 15.5 gini = 0.375 samples = 4 value = [3, 1] class = True
gini = 0.0 samples = 1 value = [1, 0] class = True
gini = 0.0 samples = 1 value = [0, 1] class = False
length ≤ 7.5 gini = 0.0772 samples = 174 value = [167, 7] class = True
length ≤ 7.5 gini = 0.32 samples = 5 value = [4, 1] class = True
length ≤ 12.5 gini = 0.2337 samples = 37 value = [32, 5] class = True
length ≤ 12.5 gini = 0.4938 samples = 9 value = [5, 4] class = True
gini = 0.4444 samples = 3 value = [2, 1] class = True
gini = 0.0 samples = 1 value = [1, 0] class = True
has num ≤ 0.5 gini = 0.1372 samples = 54 value = [50, 4] class = True
has dash ≤ 0.5 gini = 0.0487 samples = 120 value = [117, 3] class = True
gini = 0.0 samples = 1 value = [1, 0] class = True
length ≤ 8.5 gini = 0.375 samples = 4 value = [3, 1] class = True
length ≤ 11.5 gini = 0.2778 samples = 24 value = [20, 4] class = True
gini = 0.142 samples = 13 value = [12, 1] class = True
length ≤ 11.5 gini = 0.32 samples = 5 value = [4, 1] class = True
gini = 0.375 samples = 4 value = [1, 3] class = False
gini = 0.1472 samples = 50 value = [46, 4] class = True
gini = 0.0 samples = 4 value = [4, 0] class = True
length ≤ 9.5 gini = 0.0351 samples = 112 value = [110, 2] class = True
has num ≤ 0.5 gini = 0.2188 samples = 8 value = [7, 1] class = True
gini = 0.4444 samples = 3 value = [2, 1] class = True
gini = 0.0 samples = 1 value = [1, 0] class = True
gini = 0.2975 samples = 11 value = [9, 2] class = True
gini = 0.2604 samples = 13 value = [11, 2] class = True
has num ≤ 0.5 gini = 0.5 samples = 2 value = [1, 1] class = True
gini = 0.0 samples = 3 value = [3, 0] class = True
length ≤ 8.5 gini = 0.0449 samples = 87 value = [85, 2] class = True
gini = 0.0 samples = 25 value = [25, 0] class = True
length ≤ 8.5 gini = 0.2778 samples = 6 value = [5, 1] class = True
gini = 0.0 samples = 2 value = [2, 0] class = True
gini = 0.0 samples = 1 value = [0, 1] class = False
gini = 0.0 samples = 1 value = [1, 0] class = True
gini = 0.0345 samples = 57 value = [56, 1] class = True
gini = 0.0644 samples = 30 value = [29, 1] class = True
gini = 0.0 samples = 1 value = [1, 0] class = True
gini = 0.32 samples = 5 value = [4, 1] class = True

Figure 1: Trained Decision Tree

- **SVM:** Test set accuracy: 0.9693 Precision: 0.9848 Recall: 0.9557 F_1: 0.97
- **Decision Tree:** Test set accuracy:0.9719 Precision: 0.9849 Recall: 0.9606 F_1: 0.9726
- **Random Forest:** Test set accuracy: 0.9770 Precision: 0.9899 Recall: 0.9655 F_1: 0.9776
- **Logistic Regression:** Test set accuracy: 0.9719 Precision: 0.9746 Recall: 0.9569 F_1: 0.9713
- **Linear Regression:** Test set accuracy: 0.9668 Precision: 0.9847 Recall: 0.9507 F_1: 0.9674

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

4.2 Any Rule-based Post-processing?

Clearly, since the foregoing results are nearly "perfect" we do not need any rule-based post-processing for our information extraction work.

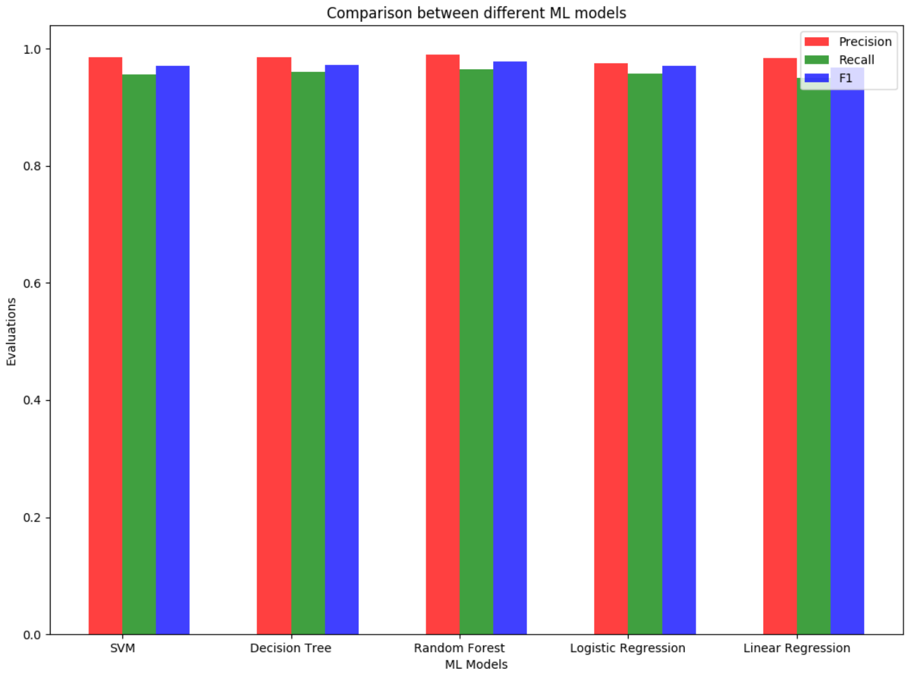

Figure 2: Comparison among ML Models