CS839 Stage 1 Report: Information extraction from natural text

Xiuyuan He [*1] **Chenlai Shi** [*1] **Mingren Shen** [*1]

1. Name of all team members

- Xiuyuan He
- Chrissie Watts
- Mingren Shen

2. Entity Type

We want to extract **people names** from moive review texts. The moview reviews are from Large Movie Review Dataset v1.0 (Maas et al., 2011) by Stanford University [1].

Examples are:

- Gina Yashere
- Chrissie Watts
- John's

Detailed rules of the entity type are:

1. Prefix and Titles like Mr., Mrs., Ms., Director, etc are **not included**
2. Suffix Names like Sr., Jr., IV, etc **are included**
3. Names form a possessive with the suffix -'s like John's, Mike's **are included**
4. Both Actor Names and Movie Character Names **are considered names**
5. People Names used in Movie Titles like "Mr. & Mrs. Smith" or Company Names like "Warner Bros. Entertainment Inc" **are considered names**

We use "<>" and "</>" to mark up all the occurrences of person names. So for the example above, we will mark them like this:

- <> Gina Yashere </>
- <> Chrissie Watts </>
- <> John's </>

3. Data Set

3.1. the total number of mentions that you have marked up

There are **1695** mentions of person names are marked up.

3.2. the number of documents in set I, the number of mentions in set I

There are **200** documents in set I and **1103** mentions of person names are marked up.

3.3. the number of documents in set J, the number of mentions in set J

There are **100** documents in set J and **592** mentions of person names are marked up.

4. Pre-processing

For the marked up text files, we do the following steps to clean the generated examples.

- delete all numbers
- delete all punctuation
- delete all stopping words
- delete all 4 words examples(we do not see names with this pattern in our data set)

5. Training and Model Selection

5.1. Feature Engineering

We propose the following binary and numeric features for the data:

Is the first letter of the string uppercase? Jim Green

the length of the words too short or too long words are not names

Whether the string contains common names if yes, the string can be a name

[*]Equal contribution [1]University of Wisconsin,Madison , USA. Correspondence to: Xiuyuan He <xhe75@wisc.edu>, Chenlai Shi <cshi29@wisc.edu>, Mingren Shen <mshen32@wisc.edu>.

CS 839 Class Project, UW-Madison, 2018.

[1] http://ai.stanford.edu/~amaas/data/sentiment/

Whether the string contains city names if yes, it may not be a person name

Whether the string contains county names if yes, it may not be a person name

Whether the string contains common words if yes, a string contains words like a, is, are may not be a person name

Is the word before the current example uppercase? "**like**" Jim Green

Is the word after the current example uppercase? Jim Green **says**

Is the word before the current example a common word? personal name should not contain too common word like a, is, too

Is the word after the current example a common word? personal name should not contain too common word like a, is, too

Is the word before the current example a country name? Country names like Afghanistan, Albania, Germany are not person names although they looks like person names.

Is the word after the current example a country name? Country names like Afghanistan, Albania, Germany are not person names although they looks like person names.

For 1000 common English words, we get from this website, https://gist.github.com/deekayen/4148741, slightly modified during training stage. And country names are from this website, https://gist.github.com/kalinchernev/486393efcca01623b18d. And our common names are based on data from U.S. Census Bureau at 2010 which is from this website, https://surnames.behindthename.com/top/lists/united-states/2010.

5.2. Performance of classifier M

We chose 5 different machine learning models:

- SVM with linear kernel
- Decision Tree
- Random Forest
- Logistic Regression
- Linear Regression

So based on the table 1, we choose Random Forest as the final method to improve on rule based methods which is referred as classifier X below.

Table 1. Performance of classifier M

classifier M	precision	recall	F1
Linear regression	0.023	0.975	0.045
Logistic regression	0.038	0.681	0.073
Random Forest	0.078	0.112	0.092
Decision tree	0.054	0.114	0.073
SVM	0.004	0.011	0.006

5.3. Any Rule-based Post-processing?

We are using two main rules for post-processing of the prediction results.

First, based on our data set, we think all words that can be a name should be capitalized. This rule will result in wrong prediction for 23 names (like Lars von Triers) and we put all those names into white name list.

Second, we mark all examples which contains words in the frequent words list as not a name. The frequent word list are downloaded from URL?

Table 2. Performance of classifier X

classifier	precision	recall	F1
Linear regression	0.408	0.620	0.492
Logistic regression	0.401	0.650	0.496
Random Forest	0.608	0.622	0.615
Decision tree	0.528	0.529	0.528
SVM	0.398	0.686	0.504

From table 2 we can find that Random Forest is also the classifier that has the best performance.

So if we apply classifier X and rules to form a new classifier Y on set J, the final result is: **precision : 0.519, recall : 0.607, F1 : 0.560.**

6. Discussion

Unfortunately, we have not achieved the required precision.

One of the biggest problems that we could not solve was the partial name in our data. For names of more than one word(e.g. Steve Jobs), our prepossessing steps will generate examples of partial names(e.g. Steve and Jobs), which are negative examples. We find it difficult to distinguish and get rid of these partial names, without the use of the real label. We tried to use a list of rules and features (e.g. preIsCap, postisCap, position, etc.) to filter out these partial names, but this will result in two hundred of positive examples being deleted, which is unacceptable. If we can find a way to filter out these nosy partial names in our data, we will finally reach a precision/recall of about 0.8/0.7.To

avoid over-fitting, we do not apply white-list and black-list for the result although they can greatly help the improvement of final results.

Software and Data

Our group website for this project is https://sites.google.com/view/cs839projectgroup7/home/stage-1 and there are links for all the required directories for the files. And we store all our data and program in Github https://github.com/iphyer/CS839ClassProject.

We use scikit-learn (Pedregosa et al., 2011) as our machine learning program library and Pandas (McKinney, 2015) for data processing.

References

Maas, Andrew L., Daly, Raymond E., Pham, Peter T., Huang, Dan, Ng, Andrew Y., and Potts, Christopher. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

McKinney, Wes. pandas: a python data analysis library. *see http://pandas. pydata. org*, 2015.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.