UNIVERSITY OF WINDSOR, SCHOOL OF COMPUTER SCIENCE

Weekly Report

Xiutian Cui

June 28, 2015

1 JUNE 1ST - JUNE 7TH

1.1 DONE THIS WEEK

I have read 3 papers this week and written one part of related works. According to these papers, I can conclude some useful ideas that will be used during my thesis.

- Definition of spammers. There are some different definitions of spammers. Spammers can be the fake accounts who are generated to boost the Pagerank of some users (according to Yi Zhang's work), or can be the user accounts who send spam tweets, especially advertising tweets, to others (according to [1, 5, 2]). I think in my thesis, spammers should be defined as the latter one.
- How to collect spammers. In [2], they used same method as we used, which is to collected suspended users from old dataset. To use this method we not only need one large dataset (this is easy because there exists some), and the dataset should also contain tweets that users posted in order to indicate that the user is suspended because of spam activities. [5] provided one website - twitspam.org. But this website have already been changed and useless now.
- Methodology. [1] and [5] used Machine Learning algorithms to detect spammers, while [2] used Link Analysis method (A Pagerank-like algorithm) to do so. I think in my thesis we can combine these two methods together to improve the accuracy of classification.
- Feature Selection. All these 3 papers suggest that graph properties act very important role in classification. Other important features are the ones that are related to whether the tweet contains URLs. All of these features should be taken into consideration during my thesis.

- Weka and libsvm should be useful in this project.

1.2 Plan next week

I plan to read other papers next week and finish writing related works. I will also start writing data description and proposal slides.

2 June 8th - June 14th

2.1 Done this week

In this week I have finished the dataset parsing programming and started trying to detecting spammers by Machine Learning algorithms. The dataset I am using is combination datasets of [3] and [4]. [3] has been used as the graph structure of Twitter. And the suspended user list was depending on this dataset, by checking whether each user in this dataset is suspended now. [4] has provided the tweets and detail profile of users. Also I checked the changing from [3] to [4] during these two years.

I built one website (http://twitter.jekycui.net) to describe what I am doing by details. And the code of parsing these two dataset were provided on this website as well.

I have started to try some classification on the merged dataset. I have merged these two dataset and converted it to an ARFF file, which can be imported by Weka. I tried some classifiers in Weka, but the results are not good (see Figure 2.1). The accuracy of JRip, which are the best classifier in [5], is only 57%.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3471               57.0982 %
Incorrectly Classified Instances      2608               42.9018 %
Kappa statistic                          0.142
Mean absolute error                      0.4846
Root mean squared error                  0.495
Relative absolute error                 96.9177 %
Root relative squared error             99.0022 %
Total Number of Instances             6079

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.57      0.428      0.572     0.57      0.571      0.582    spammer
                 0.572     0.43       0.57      0.572     0.571      0.582    non-spammer
Weighted Avg.    0.571     0.429      0.571     0.571     0.571      0.582

=== Confusion Matrix ===

    a    b   <-- classified as
 1735 1310 |    a = spammer
 1298 1736 |    b = non-spammer
```

Figure 2.1: Result of JRip in Weka

2.2 PLAN NEXT WEEK

I will try to figure out why the classification result is so low and continue writing proposal slides.

3 JUNE 15TH - JUNE 21ST

3.1 DONE THIS WEEK

I have tried to integrate the tweets user posted into feature set. Algorithm 1 shows the basis idea of this step.

Data: User Tweets set T
Result: Feature set F
for $t \in T$ **do**
  $tokens \leftarrow tokenized(t)$
  $words \leftarrow normalized(tokens)$
  $words \leftarrow removeStopWords(tokens)$
  $grams \leftarrow makeGrams(words, n)$
  for $g \in grams$ **do**
    add g into feature set F
  end
end

Algorithm 1: Process Tweet Feature

In normalize step, I tried two algorithms, Porter[7] and Lancaster[6] stemming algorithms. Porter is the most common used stemming algorithm and Lancaster is more aggressive one. In my test part, in some cases Porter is better while Lancaster is better in some other cases. So In my experiments I used Porter because the result of Porter is more readable.

In makeGrams step, I tried splitting the tweets into unigram, bigram and trigram. And then I used Naive Bayes classifier to classify them, the formula of which is

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B}$$

10-fold validation has been used to test this classifier. The results are shown in Table 3.1 and Table 3.2.

Unigram		Bigram		Trigram	
2422	623	1775	1270	864	2181
2185	849	1411	1623	648	2386

Table 3.1: Confusion Matrix of unigram, bigram and trigram

	Accuracy	F1	Recall	Precision
Unigram	53.81%	63.30%	79.54%	52.57%
Bigram	55.90%	56.97%	58.29%	55.71%
Trigram	53.46%	37.91%	28.37%	57.14%

Table 3.2: Results of unigram, bigram and trigram

In order to remove the noises from the feature set, I tried two different feature selection methods, Mutual Information and χ^2. The formulas of these two algorithms are:

$$I(U;C) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} P(U = e_t, C = e_c) log_2 \frac{P(U = e_t, C = e_c)}{P(U = e_t)P(C = e_C)}$$

$$X^2(\mathbb{D}, t, c) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}}$$

(This part is still running. The size of feature set of bigram is more than 4,500,000. I will update this part once the experiment is finished)

3.2 Plan next week

It seems like Naive Bayes is not good enough to classify users. Last week I used JRip in weka and the result of it can beat all of the results of this week. So the next step I think I need to implement a Ripper or find a Ripper open source implementation to test if Ripper is much better than Naive Bayes. Weka cannot be used here because the data size is too big to load.

And I plan to introduce Pagerank, (maybe other rank algorithm), into feature set to test whether the result can be improved.

4 June 22nd - June 28th

4.1 Done this week

This week I was focusing on feature selection and data visualization. In the feature selection part, I run two feature selection algorithms on this dataset and get the top 100 features in Unigram, Bigram and Trigram. I then tried to resize the feature set to get the best result of classification. Figure 4.1 and 4.2 show the result of classification of different size of feature set.

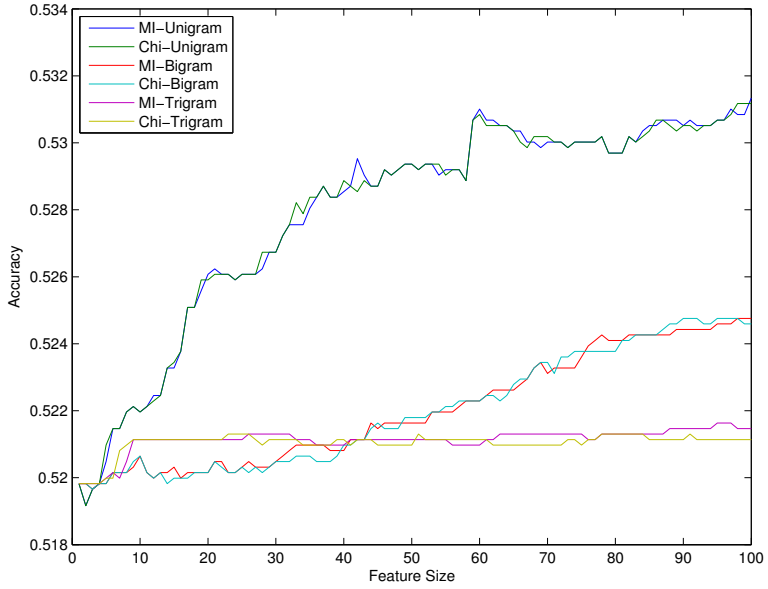

Figure 4.1: feature set size vs Accuracy

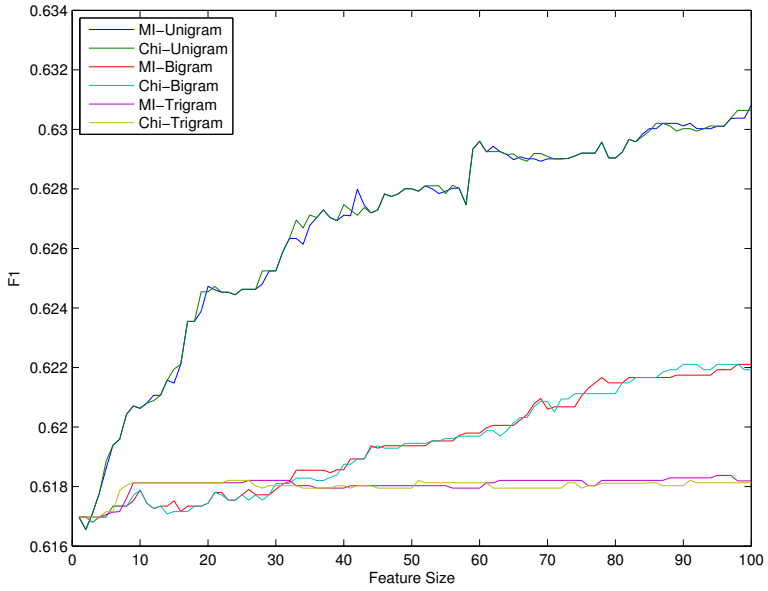

Figure 4.2: feature set size vs F1

I tried to get all the classification result of the size that is from 1 to 1000 on Unigram and Bigram. By now the experiment on Unigram has been finished and the results are shown in following figures. The experiment of Bigram is still running.

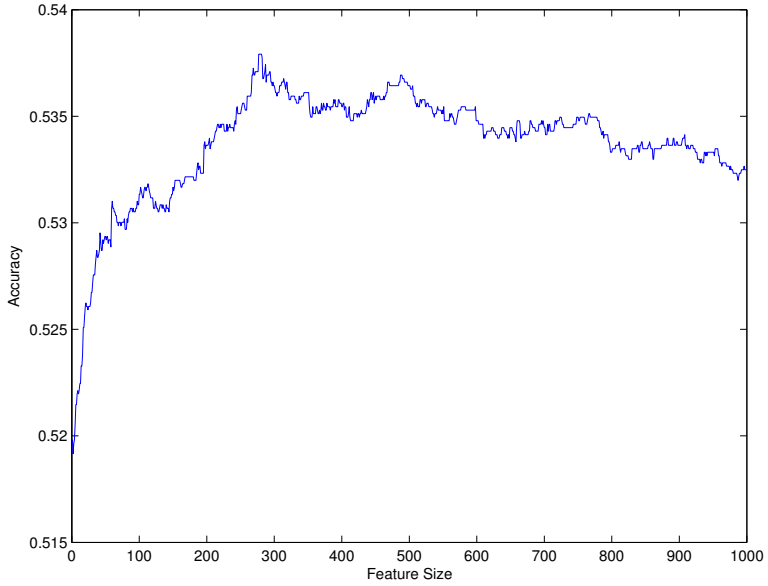

Figure 4.3: feature set size vs Accuracy

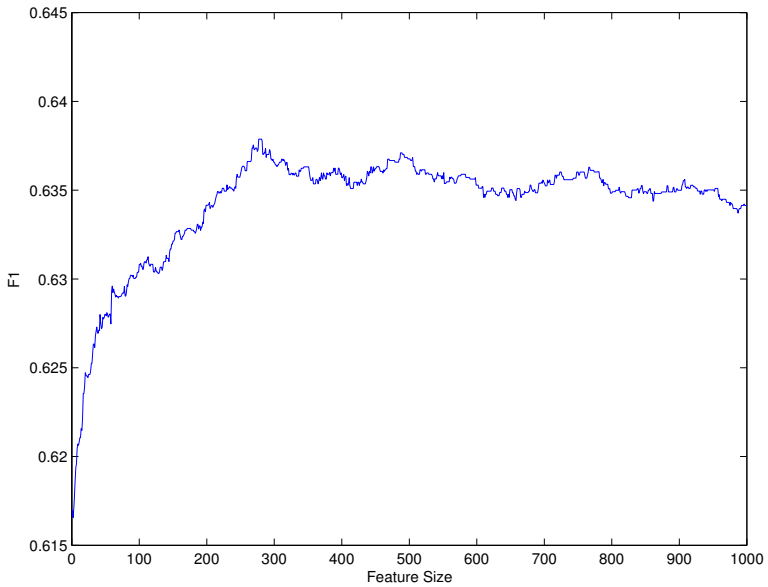

Figure 4.4: feature set size vs F1

The result of classification still can satisfy me so I will try some other feature selection algorithms later on.
I also tried to visualize the dataset to see if there is any cluster of words and the distances between words. The dataset I used here is the cleared tweets, which have been normalized,

stemmed and the stopwords are removed from them. The tweets have been merged together into 2 file, spam-tweets.txt and nonspam-tweets.txt. I then used **word2vec** to convert the merged tweets into word matrix. It first constructs a vocabulary from the training dataset and then learns vector representation of words. The result of word2vec is a $F \times N$ matrix, in which each row represents a word in vocabulary. To use a 2D figure to plot the distances between words, the dimension of result matrix should be reduced. Here I used t-Distributed Stochastic Neighbor Embedding (**t-SNE**) to do so. After all words have been done, the result is shown in Figure 4.5. The red ones are words that in spam tweets and the blue ones are the ordinary words. I didn't plot which word each node represents because it will be too hard to read. But I built one **web page** (http://twitter.jekycui.net/?category=2&page=tsne.html) to show every word interactively.

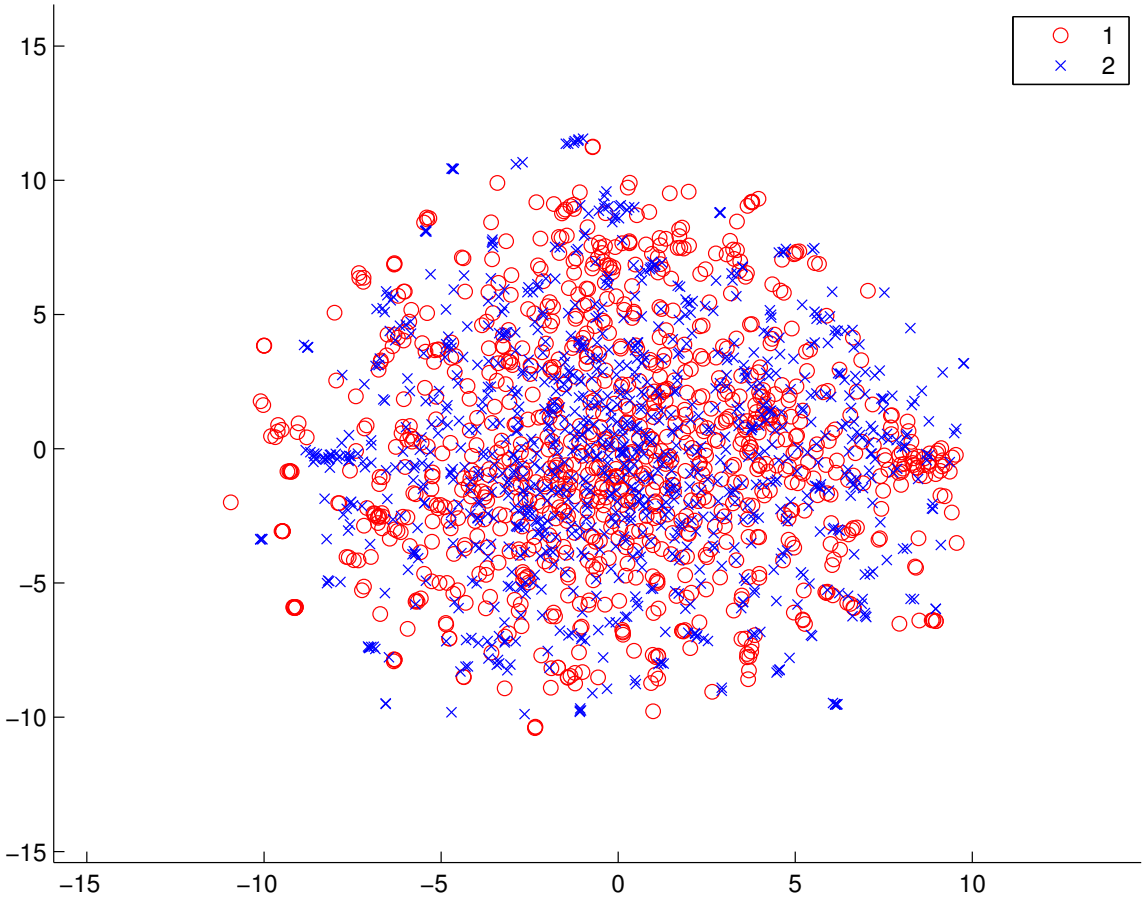

Figure 4.5: Word Visualization

We can see that there are some clusters of words. I listed some of them:

- The red cluster laying on the middle left contains the words about Food and Health.
- The red cluster laying on the bottom left contains the words about Game and Contest.

- The red cluster laying on the top left contains the words about Tax and Budget.
- The red cluster laying on the middle right contains some dirty words.

The first 3 are all about advertisement and the last one is another kind of spam activity. I think this result can be really help full to the classification.

REFERENCES

[1] BENEVENUTO, F., MAGNO, G., RODRIGUES, T., AND ALMEIDA, V. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)* (2010), vol. 6, p. 12.

[2] GHOSH, S., VISWANATH, B., KOOTI, F., SHARMA, N. K., KORLAM, G., BENEVENUTO, F., GANGULY, N., AND GUMMADI, K. P. Understanding and combating link farming in the twitter social network. In *Proceedings of the 21st international conference on World Wide Web* (2012), ACM, pp. 61–70.

[3] KWAK, H., LEE, C., PARK, H., AND MOON, S. What is Twitter, a social network or a news media? In *WWW '10: Proceedings of the 19th international conference on World wide web* (New York, NY, USA, 2010), ACM, pp. 591–600.

[4] LI, R., WANG, S., DENG, H., WANG, R., AND CHANG, K. C.-C. Towards social user profiling: unified and discriminative influence model for inferring home locations. In *KDD* (2012), pp. 1023–1031.

[5] MOH, T.-S., AND MURMANN, A. J. Can you judge a man by his friends?-enhancing spammer detection on the twitter microblogging platform using friends and followers. In *Information Systems, Technology and Management*. Springer, 2010, pp. 210–220.

[6] PAICE, C. D. Another stemmer. *SIGIR Forum 24*, 3 (Nov. 1990), 56–61.

[7] VAN RIJSBERGEN, C. J., ROBERTSON, S. E., AND PORTER, M. F. *New models in probabilistic information retrieval*. Computer Laboratory, University of Cambridge, 1980.