E-034

Improving Tweet Classification Accuracy through Automatic Tweaking of Training Set

Muhammad Asif Hossain Khan[1], Masayuki Iwai[1], Kaoru Sezaki[1,2]

Abstract- **Twitter has become a valuable source of information for extracting early symptoms to predict changes in different economic and social indicators. However, misclassification of relevant tweets can easily lead to a 'cry wolf' situation. We have presented a framework to automatically identify noisy tweets in the training set that may confound the judgment of a classifier. We have also modified conventional likelihood based collocation feature selection method. Even with relatively small training set, our method could achieve better classification accuracy.**

I. INTRODUCTION

The spatial resolution of coverage by social media has made them a promising network for assessing the evolution and dynamics of social systems. Recent research shows that Twitter posts can be used for capturing the overall trend of a particular disease outbreak [1, 2]. The general assumption made in such research is that at the onset of an epidemic within a locality, public concern will take a hike, which can be captured and quantified through the 'disease relevant' tweets generated from that locality. A set of disease related keywords are chosen, assigned weights and are considered as distinguishable features for identifying such tweets. In the next phase, tweets generated from within the locality of interest are analyzed and classified based on the selected features. Decision about the intensity of disease propagation is then inferred from statistic and textual information content of the tweets classified as 'disease relevant'. However, our observation suggests that a significant proportion of tweets, which contains such important features and hence regarded as 'disease relevant', do not report individual illness within the locality. Our experiment shows that a bag-of-word classifier trained with conventional *n*-gram features fails to achieve acceptable accuracy level in classifying tweets with self-reported illness from the general collection of disease-relevant tweets. In this paper, we have proposed a method to deal with this problem.

II. MOTIVATING EXAMPLES

We are interested in tweets in which authors express their flu infection either directly or indirectly. We refer to this class of tweets as '***Self***'. Example tweets from this class are "*In bed all day with the flu. Not fun …*" or "*Drinking Theraflu to try to beat this .... wish someone were here taking care of me …*".

Our experiment data, tweets generated from New York (within 30 km from Manhattan) contained many tweets reporting epidemic outbreak in different parts of the world. Main source of such tweets are mainstream news media. This is the class we call '***News***'. Example tweet from this class is "*Chinese bus driver infected with H1N1 bird flu virus dies; Country's first reported human case in 18 months ….*". Such tweets, though rich in information, has nothing to contribute in assessing the disease intensity in New York.

There is another class of tweets that contain the disease related keywords, but do not fall into any of the above categories. We refer to this class as '***False***'. Example tweet belonging to this class is "*Yesss finally found the link so I can download slime flu 2…*". Here the author is referring to a popular music album, not reporting any disease.

We aim at isolating tweets of class '*Self*' from those of class '*News'* and '*False'*. It is evident that the inter-class margin is quite narrow. What confound the situation more are tweets like "*In bed with DJ Khaled's TheraFlu ….*". '*Theraflu*' is a prominent drug for influenza and hence referenced in many tweets belonging to class '*Self*'. However, there is a famous music track titled '*Theraflu*', which is actually referenced in the example tweet. Google page count for '*theraflu drug*' is 1,200,000 while that for '*theraflu music*' is 2,610,000. We have proposed a method that would automatically tweak such misleading tweets from the training set and thus improve classification accuracy.

III. METHODOLOGIES

In this experiment, we have used only unigram and bigram features and have trained a multinomial Naïve Bayes classifier with the extracted features.

We have used χ^2 feature selection method to select unigram features. We have selected bigram features in two steps. To capture bigrams with flexible structure, we have used a collocation window of 4. For example, when the collocation window is set to 2 or a higher number, the text string '*powerful personal computer*' would generate three bigrams: '*powerful personal*', '*powerful computer*' and '*personal computer*'.

Bigram Feature Selection – First Step

The simplest way of finding significant bigram features is to select most frequently occurring bigrams. However, when the training corpus is small, as in our case, two words might co-occur a lot just by chance. To determine whether the bigram has some real structural importance, we have adopted the `Likelihood Ratio' approach for hypothesis testing of independence, which takes into account the volume of data that has been considered for calculating the frequency of the bigram as well as the frequency of the individual words comprising the bigram. The bigrams, for which the hypothesis testing suggests that the collocation is just a chance event, are not considered as potential features. The detail of the algorithm can be found in Khan *et al.* [3]. We refer to the set of these selected features as '***rawBigrams***'.

1 Institute of Industrial Science, The University of Tokyo

2 Center for Spatial Information Science, The University of Tokyo

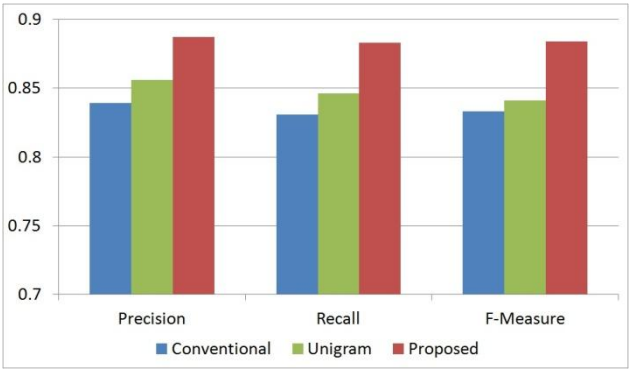

Fig. 1. Comparison of average performance among the three models

Determining Associated Class for Bigram Features

For each selected bigram, we determine the class for which the bigram contributes most as a feature. Let, c_1 and c_2 are frequencies of bigram b in the training tweets of class X and Y respectively. Let, N_1 and N_2 be the total number of identified bigrams from X and Y. We calculate the relative frequency ratio $r = \frac{c_1/N_1}{c_2/N_2}$. If $r \geq 1$, b's associated class is X, otherwise it is Y.

Tweaking the Noisy Tweets

To determine whether a tweet would be misleading for the classifier, we calculate the number of bigrams (n) from the set *rawBigrams* that it contains and the number of bigrams (l) among these n bigrams that are associated with the class it belongs to. If majority of the bigram features that it contains are not associated with its own class (i.e. $\frac{l}{n} < 0.5$), we tweak the tweet from the training set. The set of remaining tweets is referred as '***tweakSet***'.

Bigram Feature Selection – Second Step

After the tweaking phase, the bigrams among *rawBigrams* that are still present in at least one of the training tweets form the final bigram feature set. We refer to this set as '***fidels***'.

IV. Experiment and Results

Experiment Data

We have used tweets generated within 30 km around Manhattan, NY from December 06, 2011 to April 30, 2012. We have only considered those tweets that contain at least one of the keywords '*flu'* or '*influenza'*. A total of 3,955 tweets had these keywords and we randomly selected 1,000 tweets from this corpus. We have then manually labeled them into three classes: self (290), news (268) and false (329). As the annotator could not concord for 113 tweets about their appropriate class, we did not include them in training set.

Models Considered for Performance Evaluation

We have compared the performance of our proposed model with two other models. The difference among the models are in the set of features they consider during the learning phase. Moreover, the unigram and conventional models considers all annotated tweets while the proposed model considers only *tweakSet*.

Unigram model: Considers only unigram features.

Conventional model: Considers unigram and *rawBigrams*.

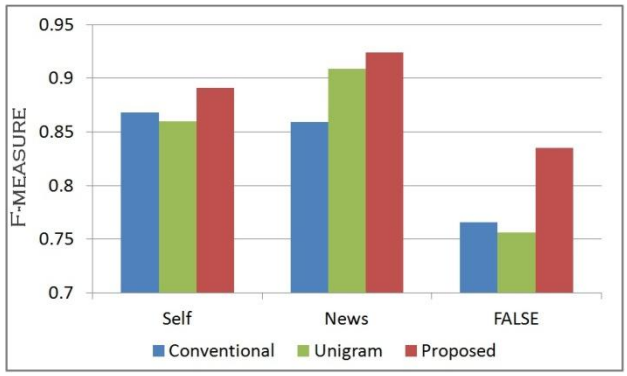

Fig. 2. Per class performance comparison among the three models

Proposed model: Considers unigram and *fidels*.

Selected Classifier

We have used a multinomial Naïve Bayes classifier for classifying the tweets into the three classes. We have used *Laplace smoothing* for accommodating unseen feature space. 10-fold cross validation method has been adopted for assessing the classification performance.

Results and Discussion

The proposed model shows better precision and recall compared to both the conventional and unigram models (figure 1). This improvement in accuracy can be attributed to the 'tweaking' of tweets and selection of '*fidels*'. As can be seen from figure 2, the conventional model performs slightly better than the unigram model for classes '*Self*' and '*False'*. However, for the class '*News'* unigram model outperforms the conventional model with significant margin due to which the overall performance of the conventional model suffers (figure 1). This is because, many of the '*News'* tweets shared the bigram feature of class '*False'*. In our proposed method, we could identify and remove these tweets to prevent them from convoluting the judgment of the classifier.

V. Conclusion

We have proposed a method for improving the performance of a bag-of-word classifier through refinement of training set. Our experiment results show that the proposed model outperforms two baseline models and achieve an accuracy of 88.7% on a real world twitter dataset. In future, we intend to incorporate higher order *n*-gram features in the model.

References

[1] V. Lampos, T.D. Bie and N. Cristianini, "Flu detector -- tracking epidemics on Twitter", *Machine Learning and Knowledge Discovery in Databases*, pp. 599-602, 2010.

[2] M. Krieck, J. Dreesman, L. Otrusina and K. Denecke, "A new age of public health: Identifying disease outbreaks by analyzing tweets", In *Proc. of HWS Workshop, ACM Web Science Conf.*, 2011.

[3] M.A.H. Khan, M. Iwai, K. Sezaki, "Towards urban phenomenon sensing by automatic tagging of tweets", In *proc. of International Conf. on Networked Sensing Systems*, 2012.