Filtering and Polarity Detection for Reputation Management on Tweets

Viktor Hangya and Richárd Farkas

University of Szeged,
Department of Informatics
hangyav@gmail.com, rfarkas@inf.u-szeged.hu

Abstract. In this paper we introduce our contribution to the RepLab 2013 – *An evaluation campaign for Online Reputation Management Systems* challenge. We participated in the filtering and polarity detection subtasks. The task of filtering is to determine whether a tweet is related to an entity. Then we classify tweets into positive, negative or neutral classes from the entity point of view. To solve these problems we employed supervised machine learning techniques. We applied several Twitter specific text preprocessing and features engineering methods. Besides supervised methods, we experimented with incorporating clustering information as well. Our system was ranked 2nd in the filtering task and 1st in the polarity detection task.

Keywords: Reputation management, Sentiment analysis, Twitter

1 Introduction

In the past few years, the popularity of social media has increased. People post messages on a variety of topics for example products, political issues, etc. Thus a big amount of user generated data is created day-by-day. The manual processing of this data is impossible, therefore automatic procedures are needed. Many studies have been made in the area [6, 13], for example predicting the results of elections [14], monitoring brands [9] or predicting stock price changes [15].

In this paper we introduce our contribution for the RepLab 2013 *An evaluation campaign for Online Reputation Management Systems* challenge [3]. Here, the end-user application is monitoring the reputation of several entities, like companies, organizations, celebrities, etc. from Twitter messages. The organizers defined four tasks, namely filtering, polarity classification, topic detection and assigning priority, from which we take part in the first two ones. In the case of the filtering task the goal was to determine which tweets are related to a given entity and which are not, for instance, distinguishing between tweets that contain the word “Stanford” referring to the University of Stanford or to Stanford as a place. This step is necessary as we can ignore the unrelated tweets, so this step could be considered as a preprocessing step. The task of polarity detection is to decide whether a given tweet carries positive, negative or neutral message towards the entity in question.

The data provided by the organizers of the challenge consists of English and Spanish messages. The data was crawled from 61 entities each from automotive, banking, universities or music/artists domains. The data was collected using the entities canonical names as queries. Besides the train and test databases a set of background tweets were also provided which could be used while preparing our system.

We created a similar system for the filtering and polarity detection tasks. It is an n-gram based supervised model as it has been shown that it works well on short messages like tweets [1, 5, 10, 11]. We reduced the size of the dictionary by normalizing the messages. Furthermore we examined novel methods which increased the precision of our classifiers for example specially weighting our features and using the results of topic modeling technologies. In case of the filtering task the official evaluation metric was an F measure calculated from the reliability and sensitivity values [2]. Besides the organizers provided the accuracy of the participated systems as well. Our system achieved 0.438 F measure in this task which is the second best result from all off the participated systems, and achieved 0.928 accuracy which is the best. In the polarity detection task, the official measure was the accuracy. We achieved an accuracy of 0.685, the highest value among the participants.

2 Approach

We employed a unigram model along with tweet-specific normalization techniques. We investigated novel features as well, which increased the accuracy of our classifier. For implementation we used the MALLET toolkit, which is a Java-based package for natural language processing [12].

2.1 Normalization

One reason for the unusually big dictionary size of the standard unigram model is that it contains one word in many forms, for example in upper and lower case, in a misspelled form, with character repetition, etc. On the other hand, it contains various special annotations which are typical for blogging, such as Twitter-specific annotations, URL's, smiles, etc. Keeping these in mind we made the following normalization steps:

- First, in order to get rid of the multiple forms of a single word we converted them into lower case form then we stemmed them. For this purpose we used the Porter Stemming Algorithm.
- We replaced the @ Twitter-specific tag and every URL with the *[USER]* and *[URL]* notations, respectively. Besides, in case of a hash tag we deleted the hash mark from it, for example we converted *#funny* to *funny*. This way we do not distinguish Twitter specific tag from other words.
- Smileys in messages play an important role in polarity classification. For this reason we grouped them into positive and negative smiley classes. We

considered *:), :-),:), :D, =), ;), ;), (:* and *:(, :-(, : (,):,) :* smileys as positive and negative, respectively.

- Since numbers do not contain information regarding a message polarity, we converted them as well to the *[NUMBER]* form. In addition, we replaced the question and exclamation marks with the *[QUESTION_MARK]* and *[EXCLAMATION_MARK]* notations. After this we removed the unnecessary characters `'"#$%&()*+,./:;<=>\^{}~`, with the exception that we removed the `'` character only if a word started or ended with it.
- In the case of words which contained character repetitions – more precisely those which contained the same character at least three times in a row –, we reduced the length of this sequence to three. For instance, in the case of the word *yeeeeahhhhhhh* we got the form *yeeeahhh*. This way we unified these character repetitions, but we did not loose this extra information.

Before the normalization step, the dictionary contained approximately 113, 000 words. After the above introduced steps we managed to reduce the size of the dictionary to 38, 000 words. It is important to mention that we handle English and Spanish tweets the same way.

2.2 Features

After normalizing Twitter messages, we investigated feature space for a supervised classifier. In many cases, phrases are important because they can catch aspects of messages that simple unigrams can't. For example *"don't like"* if we handle the two words separately we lose the knowledge that the negation word refers to the word *"like"*. From this reason we examined the effects of bigrams and trigrams and we realized that bigrams can improve the accuracy of our classifier. However trigrams did not improve our result significantly so we used only **bigrams** besides unigrams. Furthermore, we added a new feature as well which is the **number of negation words** in a message.

We searched for special features which characterize the polarity of the tweets. One such feature is the polarity of each word in a message. To determine the polarity of a word, we used the **SentiWordNet sentiment lexicon** [4]. In this lexicon, a positive, negative and an objective real value belong to each word, which describes the polarity of the given word. We created three new features for each tweet which are the sum of the positive, negative and objective values divided by the number of words in a message.

For handling **acronyms**, we used an acronym lexicon which can be found on the `www.internetslang.com` website. For each acronym we separately summed up the positive and negative values of each word in the description of the acronym and we normalized them by the number of words in the description. Then for each tweet we added two new features which are the sums of the positive and negative values of the acronyms in the message divided by the number of acronyms.

Our intuition was that people like to use **character repetitions** in their words for expressing their happiness or sadness. Besides normalizing these tokens (see Section 2.1), we created a new feature as well, which represents the number of this kind of words in a tweet.

Beyond character repetitions people like to write words or a part of the text in upper case in order to call the reader's attention. Because of this we created another feature which is the **number of upper case words** in the given text.

Since the task is to filter tweets related to given entity and to classify them by their sentiments, we found it important to sign whether the message contains the **mention of the entity** or not (e.g. the username can also contains the name of the entity). For this purpose we created a binary feature which indicates this aspect.

Furthermore it could be helpful to take into consideration the **distance between the token in question and the mention of the target entity**. The closer a token is to an entity the more the possibility that the given token is related to the entity. For example consider following message where the first sentence does not refer to *BMW* at all:

> I do agree that money can't buy happiness. But somehow, it's more comfortable to sit and cry in a BMW than on a bicycle!

For this reason we weighted each word in the message by its distance from the mention of given entities:

$$\frac{1}{e^{\frac{1}{n}|i-j|}} \tag{1}$$

where n is the length of the message, i and j are the position of the actual word and the mention of the entity in the message. Besides this, because different properties could be positive or negative to different entities we created a new feature which is the name of the entity, this way the classifier could handle entities differently.

Beyond the above supervised steps we experimented with leveraging unlabeled data as well. We used Latent Dirichlet Allocation (LDA) [7] for detecting topics on the train, test and background tweets provided by the RepLab 2013 organizers. The goal of **topic modeling** is to discover abstract topics that occur in a collection of documents. As a result of LDA, we get the topic distribution for each tweet and the topic for each word. We used the topic distributions over each tweet as features. Each word belongs to a topic, so for a given message we calculated the number of each topic by its content and used it as a feature as well.

3 Results

In this section we report the results of the several systems which we experimented with and their parametrization and the official results of the RepLab 2013 challenge. The training data which was provided by the organizers consists of 36,940 English and 8,731 Spanish tweets. From this 15,123 tweets are from automotive, 7, 774 from banking, 6, 964 from universities and 15, 814 from music/artists domains. The test data consists of 79, 981 English and 25, 118 Spanish tweets which are distributed over the four domains similarly like the train data. In our system we used Maximum Entropy classifier because we earlier showed that it works well in document classification [8].

Below we will show the effects of the methods which was introduced in section 2. On figures 1 and 2 the accuracy of our system for both filtering and polarity detection tasks can be seen while adding more and more features to it. Our baseline system is named *naive* which uses simple unigram features without any normalization steps or extra features. In the *norm* version of our system we used all of the normalization steps which we introduced earlier. It can be seen that this step increased the accuracy for both tasks with a relatively large value. The next step was to use the bigrams as features besides unigrams and normalization in the *N-gram* system. This step increased the accuracy as well. The next system which we named *features* uses several abstract features too, which are the polarity of words and acronyms, the presence of character repetitions and upper case words and the number of negation words. These features increased our results marginally. In the so called *weighting* system we applied the distance based weighting for each word in the message and in the *entities* system we used the presence or absence of the given entity's name in the messages. These steps further increased our accuracy. In the last two systems we used the results of LDA topic modeling with 50 topics. In the *topic50* we used the topic distributions over the messages as features and in the *topic50-num* we used the number of topics feature as well. These methods slightly improved our classifier only for the polarity task. In total, the introduced methods and features improved our results by 3-4 percents compared to our baseline.

In table 1 we show the effects of LDA with different number of topics. Here we used a system that uses all of the above introduced methods and features. It can be seen that LDA increased our results in both tasks, mainly the F measures. The best topic number was 50 and 100 for the filtering and polarity detection tasks, respectively.

Table 1. Results with different number of topics

topic number	filtering acc.	filtering F	polarity acc.	polarity F
0	0.919	0.380	0.680	0.368
20	0.920	0.387	0.680	0.370
50	0.919	0.391	0.682	0.375
100	0.920	0.388	0.683	0.379

The RepLab 2013 participants were allowed to send up to ten different runs per subtask. In case of the filtering task we achieved our best result with the following system. We used all of the above mentioned methods and features, we detected 50 topics with LDA on the train and test data. Furthermore we run

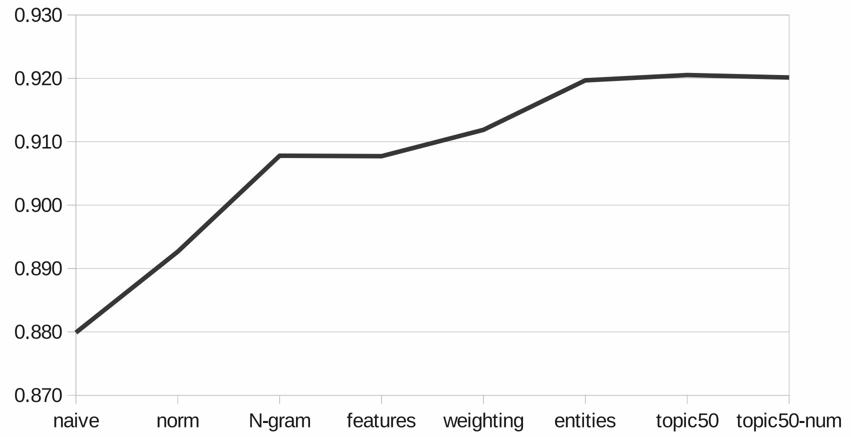

Fig. 1. Filtering accuracy on test data

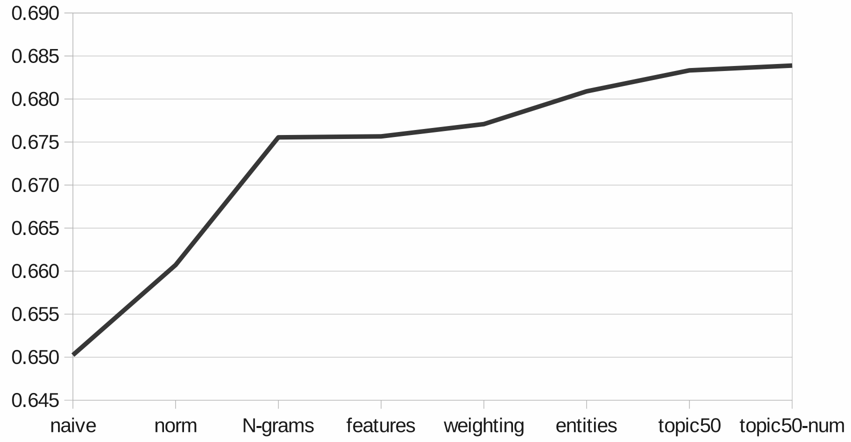

Fig. 2. Polarity accuracy on test data

our classifier separately on the four domains (automotive, banking, university, music/artists). This way we reached 0.438 F measure comparing to the best system which reached 0.488 and the baseline provided by the organizers was 0.325. With this system we reached 0.928 accuracy which turns out to be the highest value. In case of the polarity detection task our best system was parametrized as follows. Like before, we used all the technologies which we developed, we detected only 20 topics on the train and test data. Unlike the filtering task, here we run our classifier on the four domains at the same time. We achieved 0.685 accuracy which is the highest among all of the participated systems, the given baseline was 0.584. We achieved our highest F measure with a different system which similar to the previous but it did not use the polarity of the words and the acronyms. This way we reached 0.381 F measure, whilst the given baseline was 0.297.

From this analysis we can conclude that the normalization of the messages yielded a considerable increase in the accuracy of our classifier. We discussed above that this step also significantly reduced the size of the dictionary. The features and other methods increased the precision as well. During our experiments we realized that in some cases LDA did not detect topics well which can cause the low improvements in accuracy by LDA features. For example consider the following topic which contains these words *"the to for a in of and year a"*. In future it would be worth trying to improve the performance of topic modeling.

4 Conclusions and Future Work

Recently, sentiment analysis on Twitter messages has gained a lot of attention due to the huge amount of Twitter users and their tweets. A commercial extension of classical binary sentiment analysis is reputation management systems. In this paper, we examined several methods for filtering relevant tweet by a given entity and for classifying them by their sentiments for reputation management. We proposed special features which characterize the polarity and other aspects of tweets and we concluded that due to the informality (slang, spelling mistakes, etc.) of the messages it is crucial to normalize them properly. Our system achieved outstanding ranks in both filtering and polarity tasks of the RepLab 2013 challenge.

In the future, we plan to investigate the utility of relations between Twitter users and between their tweets. Furthermore we would like to examine several domain adaption methods in such way that we use messages for training from other sources than Twitter.

Acknowledgments

This work was supported in part by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TÁMOP-4.2.2.C-11/1/KONV-2012-0013).

References

1. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment Analysis of Twitter Data. In: Proceedings of the Workshop on Language in Social Media (LSM 2011). (June 2011) 30–38
2. Amigó, E., Corujo, A., Gonzalo, J., Meij, E., Rijke, M.: Overview of replab 2012: Evaluating online reputation management systems. In: CLEF 2012 Labs and Workshop Notebook Papers. (2012)
3. AmigÃş, E., Carrillo de Albornoz, J., Chugur, I., Corujo, A., Gonzalo, J., MartÃŋn, T., Meij, E., de Rijke, M., Spina, D. In: Fourth International Conference of the CLEF initiative, CLEF 2013, Valencia, Spain. Proceedings. Springer LNCS, location =
4. Baccianella, S., Esuli, A., Sebastiani, F.: SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In Chair), N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D., eds.: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, European Language Resources Association (ELRA) (May 2010)
5. Barbosa, L., Feng, J.: Robust Sentiment Detection on Twitter from Biased and Noisy Data. In: Poster volume. Coling 2010 (August 2010) 36–44
6. Bifet, A., Frank, E.: Sentiment Knowledge Discovery in Twitter Streaming Data. (2010)
7. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. the Journal of machine Learning research **3** (2003) 993–1022
8. Hangya, V., Berend, G., Farkas, R.: Szte-nlp: Sentiment detection on twitter messages. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Atlanta, Georgia, USA, Association for Computational Linguistics (June 2013) 549–553
9. Jansen, B.J., Zhang, M., Sobel, K., Chowdury, A.: Twitter Power: Tweets as Electronic Word of Mouth. In: Journal of the American society for information science and technology. (2009) 2169–2188
10. Jiang, L., Yu, M., Zhou, M., Liu, X., Zhao, T.: Target-dependent Twitter Sentiment Classification. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics. (June 2011) 151–160
11. Liu, B.: Sentiment Analysis and Subjectivity. In Indurkhya, N., Damerau, F.J., eds.: Handbook of Natural Language Processing. (2010)
12. McCallum, A.K.: Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu (2002)
13. O'Connor, B., Balasubramanyan, R., Routledge, B.R., Smith, N.A.: From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. In: Proceedings of the International AAAI Conference on Weblogs and Social Media. (May 2010)
14. Sang, E.T.K., Bos, J.: Predicting the 2011 Dutch Senate Election Results with Twitter. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. (April 2012) 53–60
15. Vu, T.T., Chang, S., Ha, Q.T., Collier, N.: An experiment in integrating sentiment features for tech stock prediction in twitter. In: 24th International Conference on Computational Linguistics. (2012) 23