Target-Oriented Opinion Mining from Tweets

Viktor Hangya, Richárd Farkas

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

***Abstract*—People express their opinion about many things like products, political parties, ideas using the facilities of social media. The analysis of these opinions is a gold mine for marketing experts and for humanities research as well. We introduce a system for opinion mining from the textual content of tweets and discuss the differences between tweet-level and target-oriented opinion mining.**

I. Introduction

In the past few years, the popularity of social media has incredibly increased. People post messages on a variety of topics for example products, political issues, etc. Thus a big amount of user generated content is created day-by-day. Several applications were developed exploiting the information present in user generated content [1], for example predicting the results of elections [2], monitoring brands [3] or disaster management [4].

Here, we introduce an approach for analyzing the textual content of tweets for assign sentiment labels to messages. More precisely, it classifies tweets into positive, negative or neutral polarity classes. We used a database which was created for the RepLab 2013 – *An evaluation campaign for Online Reputation Management Systems* challenge [5]. The chief novelty of this dataset is that it was collected for **target-oriented sentiment analysis**, i.e. instead of a message-level polarity classification – which was the objective of previous evaluation campaigns – the task is to decide the polarity of the message towards an entity in question. For example, the tweet

> *I get more compliments on my mazda then my old modded subaru #mazdalove*

bears positive polarity from Mazda's point of view while negative for Subaru and classifying the whole message as a whole does not make any sense.

In our system we developed text normalization steps which improves the accuracy of simple unigram and bigram based document classifiers by removing unnecessary elements from messages. Furthermore we experimented with novel features which can characterize the polarity of these messages. Our system achieved an outstanding 0.69 accuracy on the test database.

II. Approach

We employed a bag-of-words-based supervised classifier along with tweet-specific normalization techniques and experimented with novel features [6]. We followed the supervised classifier approach and employed a Logistic Regression classifier [7]. We used the implementaion of the MALLET toolkit, which is a Java-based package for machine learning [8].

A. Normalization

The size of the lexicon is usually huge in document classification problems in the social media domain. One reason for this 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 unified the twitter-specific user tags, URLs and numbers and we deleted the hash mark from hash tags, for example we converted *#funny* to *funny*.
- We grouped smileys into positive and negative smiley classes. We considered *:), :-),:), :D, =), ;), ;), (:* and *:(, :-(, : (,):,) :* smileys as positive and negative, respectively.
- We removed the unnecessary characters `'"#$%&()*+,./:;<=>\^{}~`.
- 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 tokens. After the above introduced steps we managed to reduce its size to 38,000 tokens.

B. Feature Space

Our baseline feature set is the the unigrams of the messages – using a whitespace tokenizer on the normalized texts –. 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 used **bigrams** besides **unigrams**.

We investigated novel 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** [9]. 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 II-A), we created a new feature as well, which represents the number of this kind of words in a tweet. Furthermore, we added a new feature which is the **number of negation words** in a message.

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.

Besides the standard message-level features, we developed several **target-oriented features** as well. We found it important to sign whether the message contains the **mention of the entity** or not. 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 inverse distance from the mention of given entities.

In addition, we used the **entity names** as feature. This way we incorporate apriori knowledge whether people usually like or dislike the given entity.

III. Experimental Setting

A. Dataset

The database which were provided by the RepLab organizers consists of $45,679$ training and $96,848$ test tweets, the rate of the English and Spanish messages was approximately $4:1$. Tweets were crawled by the name of a given entity. There were 61 entities in the database from the automotive, banking, universities and music/artists domains. These domains were chosen to create different scenarios. The automotive domain contains entities which reputation depends on their products only. In case of the banking and universities domain, the entities reputation depends on their economic activities and on a very broad and intangible set of products, respectively. The reputation of entities from the music/artists domain depends on their products and personality as well.

The data was labeled by thirteen annotators by the following way:

- RELATED/UNRELATED: is the tweet related to the given entity. For instance whether the String "Stanford" refers to the university or the town.
- POSITIVE/NEUTRAL/NEGATIVE: the polarity of the tweet with respect to the given entity. Note we used only these labels in our experiments.
- Identifier of the topic cluster the tweet has been assigned to.
- ALERT/MILDLY IMPORTANT/UNIMPORTANT: the priority of the topic

In case of the polarity labeling, the rate of the agreement between the annotators was 0.68 and the Kappa value was 0.41. The majority class in the database is positive.

B. Evaluation Metric

For the evaluation of the systems two measures were used, which were the accuracy and the reliability/sensitivity based f-measure [10]. The accuracy is a highly interpretable measure, because it measures the rate of correctly classified documents. But it can be misleading in those cases where the classes are not balanced. Consider a database where there are much more documents for one class than for the others. If we always predict this label we can get high accuracy, but our system is not good for practical use. In other case, when we want to get the relation between two document (negative: -1, neutral: 0, positive: 1), this measure is not appropriate. Consider two documents the first is negative and the second is neutral. If our system predicts that the first is neutral the second is positive, we get 0.00 accuracy but the relation (first $<$ second) between the to documents was predicted correctly. For this reason f-measure was used as well, which can measure the relations between documents correctly.

IV. Results

A. The Added Value of Features

In figure 1 the accuracy of our system can be seen in the function of incrementally expanding our features set. Our baseline system used simple unigram features without any normalization steps or extra features. It reached 0.6502 accuracy. It is very important to normalize the messages appropriately, because if so, the classifier can infer properly the polarity of unigram features from the training database. But in some cases for example when a words meaning is modified by another word unigrams are not sufficient. Bigrams can characterize those cases when the modifier precedes the given word. For this reason bigram features can increase accuracy significantly (yielding an error reduction of 2.5%).

Extra features (number of negation words, word and acronym polarity values, character repetitions and upper case

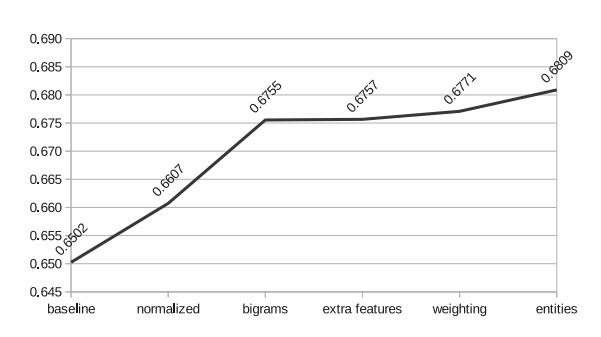

Fig. 1. Polarity accuracy on test data

team	accuracy
SZTE	0.69
LIA	0.65
POPSTAR	0.64
UAMCLYR	0.62
UNED ORM	0.62
BASELINE	0.58
NLP IR UNDED	0.58
IE	0.58
DIUE	0.55
VOLVAM	0.54
DAEDALUS	0.44
GAVKTH	0.37

TABLE I
OFFICIAL ACCURACY RESULTS ON THE TEST DATA FOR EACH PARTICIPANTS

words) represented peoples sentiments towards some topic. But our experiments showed that the error reduction by these features are not significant. The reason for this is that a bag-of-words classifier can learn sentiment values for unigram and bigram features from the training data and our additional features do not contain much extra information.

On the other hand, the target-oriented features, like using words in the classification process which are closer to the mention of the entity with bigger weight, we achieved a small but significant error reduction. Furthermore by using the knowledge that which entities do people like or dislike we managed to further increase our systems accuracy.

From these experiments we can conclude that regular methods can significantly increase classification accuracy. Furthermore it is important to gather information about the target entity of the sentiment analysis, for example the product or the persons name which the message refers to.

B. Comparison with Other Systems

In the next few tables the official RepLab results can be seen. The main evaluation metric was the accuracy measure. Our system achieved 0.69 accuracy. The results for each participated teams can be seen in table I. The BASELINE which was provided by the organizers is a simple Jaccard distance based system. For a given tweet it predicts the label of the most similar tweet in the train database. In table II the reliability, sensitivity and f-measures can be seen. Our system achieved a high score in this case too. It can be seen that there is a difference in the order of the teams by the two measures. There are teams with high accuracy but lower f-measure, for example team LIA [11]. On the other hand there are teams with high f-measure but lower accuracy, like VOLVAM [12]. The reason for this is that we do not need to predict the correct labels to achieve high f-measure, just the relation between tweets.

Another important aspect of polarity detection, is the ability to predict the average polarity of an entity with respect to other entities. In table III the Pearson correlation between the average estimated and real polarity levels across entities can be seen. The values are relatively high, including the BASELINE system as well.

team	relativity	sensitivity	F-measure
SZTE	0.48	0.34	0.38
POPSTAR	0.43	0.34	0.37
DAEDALUS	0.31	0.40	0.34
VOLVAM	0.31	0.39	0.34
NLP IR UNDED	0.33	0.31	0.32
UAMCLYR	0.33	0.29	0.30
BASELINE	0.32	0.29	0.30
UNED ORM	0.32	0.29	0.30
LIA	0.37	0.27	0.29
GAVKTH	0.37	0.21	0.27
DIUE	0.33	0.22	0.25
IE	0.29	0.22	0.25

TABLE II
OFFICIAL RELATIVITY, SENSITIVITY AND F-MEASURE RESULTS ON THE TEST DATA FOR EACH PARTICIPANTS

C. Normalisation on Different Domains

Below the effects of normalization steps (II-A) on different domains will be presented. Our hipothesys is that the normalization have different effects on different domains. For example consider the music/artists domain. We expected that it is noisier than the other three domain as it uses more emoticons, picture or video urls, and that its text is less formal, it contains more slang words, acronyms and misspelled words. On the other hand in case of the banking domain we can expect

team	correlation level
POPSTAR	0.89
SZTE	0.88
BASELINE	0.87
LIA	0.82
UAMCLYR	0.82
NLP IR UNDED	0.79
UNED ORM	0.70
DAEDALUS	0.52
GAVKTH	0.49
VOLVAM	0.38
IE	0.22
DIUE	0.21

TABLE III
OFFICIAL PEARSON CORRELATION RESULTS ON THE TEST DATA FOR EACH PARTICIPANTS

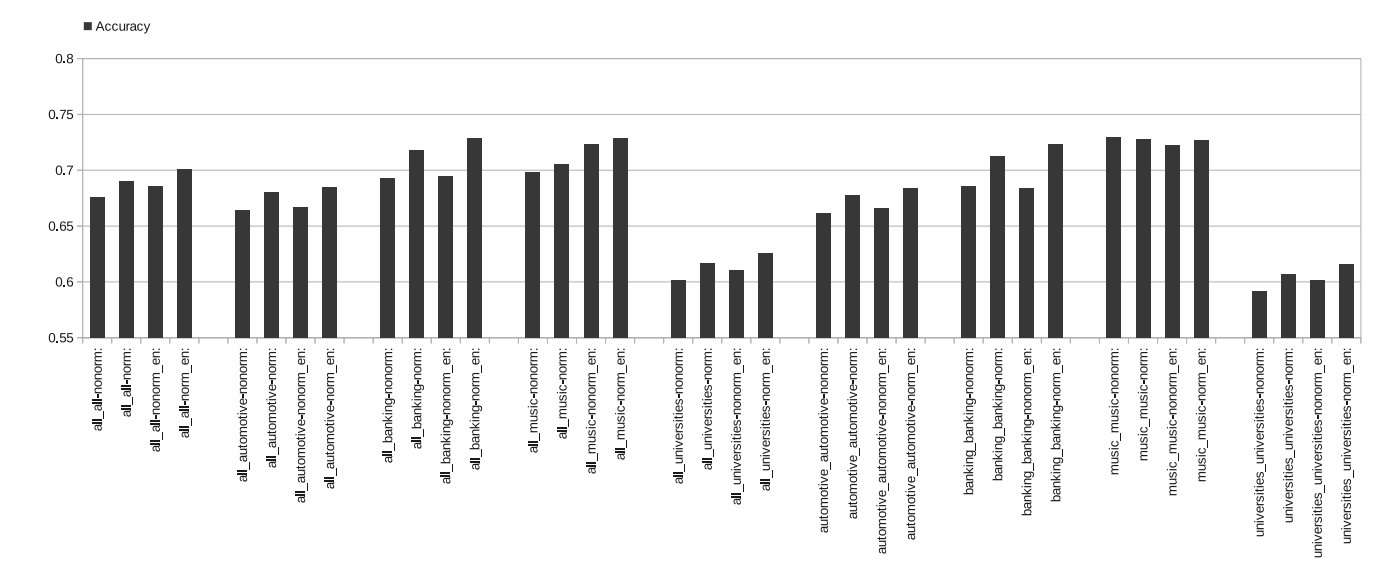

Fig. 2. Effects of normalization on system accuracy

more formal tweets. The content of these tweet are financial related, so we can expect less emoticons and urls.

In figure 2 our results can be seen. The naming convention of the systems are the following. The first word indicates the domain on which we trained our model, the second word is the test domain. The words *norm* and *nonorm* indicates whether we used normalization. Lastly the *en* means that we run our tests only on English tweets. There are no test on only Spanish tweets because our normalization method is optimized for English. During the tests we used all our features which were introduced already.

The results showed that the normalization improved our accuracy in all cases, except when we trained and tested on the music/artist domain. In this case the accuracy is slightly lower when we normalized, but the difference is not significant. Furthermore it can be seen that the lowest improvement was achieved in the music/artists domain and the highest in the banking domain which is fully contradictory to our assumption.

V. Conclusions

In conclusion, our task was to detect the polarity of a tweet toward a given target. We introduced a simple unigram and bigram based system. We concluded the in case of tweet classification it is important to normalize text appropriately, because it can significantly improve the performance of the classifier. Furthermore we proposed novel target oriented features and our system achieved high accuracy and F measures. We think that this research area is very important because more and more people use some kind of social media application, which is a gold mine for data miners.

Acknowledgment

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] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "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.

[2] E. T. K. Sang and J. Bos, "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, pp. 53–60.

[3] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury, "Twitter Power: Tweets as Electronic Word of Mouth," in *Journal of the American society for information science and technology*, 2009, pp. 2169–2188.

[4] I. Varga, M. Sano, K. Torisawa, C. Hashimoto, K. Ohtake, T. Kawai, J.-H. Oh, and S. De Saeger, "Aid is out there: Looking for help from tweets during a large scale disaster," in *Proceedings of the 51st Annual Meeting of the ACL*, 2013, pp. 1619–1629.

[5] E. Amig, J. Carrillo de Albornoz, I. Chugur, A. Corujo, J. Gonzalo, T. Martn, E. Meij, M. de Rijke, and D. Spina, "Overview of RepLab 2013: Evaluating Online Reputation Monitoring Systems," in *Fourth International Conference of the CLEF initiative*, 2013.

[6] V. Hangya and R. Farkas, "Filtering and polarity detection for reputation management on tweets," in *Working Notes of CLEF 2013 Evaluation Labs and Workshop*, 2013.

[7] A. L. Berger, S. D. Pietra, and V. J. D. Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996. [Online]. Available: citeseer.ist.psu.edu/berger96maximum.html

[8] A. K. McCallum, "Mallet: A machine learning for language toolkit," 2002, http://mallet.cs.umass.edu.

[9] S. Baccianella, A. Esuli, and F. Sebastiani, "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, 2010.

[10] E. Amigó, J. Gonzalo, and F. Verdejo, "Reliability and sensitivity: Generic evaluation measures for document organization tasks," Tech. rep., UNED, Tech. Rep., 2012.

[11] J.-V. Cossu, B. Bigot, L. Bonnefoy, M. Morchid, X. Bost, G. Senay, R. Dufour, V. Bouvier, J.-M. Torres-Moreno, and M. El-Beze, "Lia@replab 2013," in *Working Notes of CLEF 2013 Evaluation Labs and Workshop*, 2013.

[12] A. Mosquera, J. Fernandez, J. M. Gomez, P. Martinez-Barco, and P. Moreda, "Dlsi-volvam at replab 2013: Polarity classification on twitter data," in *Working Notes of CLEF 2013 Evaluation Labs and Workshop*, 2013.