Twitter Sentiment Diffusion

Matthias Baetens & Karol Dzitkowski

***Abstract*—We build software to evaluate the most important features of a Tweet and their influence on the number of retweets. We used both the standard Tweet-features as well as a calculated value for the sentiment of a Tweet. We used a number of different Machine Learning models and algorithms including neural networks to compare the performance of these methods. Our system can be dynamically accesed using a webpage which is able to download new Tweets, run the different Machine Learning algorithms, perform analysis and generate relevant charts.**

I. Introduction

One of the most important features to measure the popularity of a Tweet is the number of retweets. Next to the number of favorites, which counts how much people like a post, the number of retweets counts the number of times another user reshared the post, and thus wants to identify himself with the post and wants to share it with other. This means it is very interesting to research the possibility to optimize Tweets in order to get more retweets and to spread your message.

Tweets not only consists of the message itself: they have a huge amount of metadata:

- The time of creation of the Tweet and the user profile.
- The location.
- Whether there is a URL, an image, ...
- The hashtags (and the amount of hashtags)
- The number of followers and friends
- ...

Using the text, it is possible to calculate a certain sentiment for each Tweet; which can also be seen as a feature of the Tweet. For example: a Tweet with “awesome day” will have a more positive sentiment value than a tweet with “bad day”.

We concentrated on building a basic system that downloads Tweets relevant to a certain query, calculate a sentiment and save them to a database. We implemented 6 different Machine Learning algorithms: 4 for classification and 2 for regression. The classification is used to classify tweets in a certain sentiment class using the favorite count of the Tweet, followers count of the user, retweet count of the Tweet and age of the Tweet as an input. The regression algorithms are used to predict the number of retweets based on the favorite count of the Tweet, followers count of the user, calculated word sentiment and the age of the Tweet. The results can be accessed through a website, which is implemented using Django.

II. Related work

In Suh et. al. [1] the authors tried to quantitatively identify factors that are associated with retweeting. They split up the factors in 2 classes of features: content features and contextual features and found that for the content features URLs and hashtags seemed to have an influence on the retweet rate and for the contextual features, the number of followers and followees and the age of the account seemed to have an influence.

Dan Zarrella [2] found that users with more followers indeed get more retweets, but there are certain users without a lot of followers who get a lot of retweets, so the content of the tweets must be of some importance too. He also found that there were significantly more links in the retweets than in the tweets (56.69 % versus 18.96 %). Novelty (“newness” of the ideas and information presented) also turns out to be an important feature. The late afternoon until night (3 PM until midnight) is the most popular time to retweet.

III. Software

In Figure 1 you can see a coarse overview of our software. In this section we will describe the different parts of our software and what their respective functionalities are.

A. The main logic/API (TwitterSentimentAnalysis)

The *TwitterSentimentAnalysis*-package contains all the logic and processing of data of our application.

1) Data: This folder contains the data used in the application:

- corpus.csv: the downloaded file containing records with the topics, sentiment rating and Tweet-id specified.
- words.txt: the AFINN wordlist containing different words and their sentiment value. This file is used to calculate the sentiment of our Tweets.
- ai-folder: contains the saved (and trained) Artificial Intelligences used on the website.

2) test: This folder contains all the tests written to test the logical part of our software.

3) core.py: core.py takes care of all initializing and sets up connections to Twitter and our database. It reads the necessary parameters from the *configuration.cfg*-file.

4) ai.py: This file contains all the Machine Learning algorithms used to predict the sentiment and the retweet count. We have implemented 6 different Machine Learning algorithms, 4 to classify the Tweets in a certain sentiment class and 2 to predict the retweet count. Classification:

- MultiClassClassificationNeuralNetwork: uses the Pybrain library [3] to construct an Artifical Neural Network with 2 hidden layers and 4 and 9 neurons in the first and second layer respectively.
- SimpleClassificationNeuralNetwork: uses the native Pybrain NNclassifier class from Pybrain tools to set up an Artificial Neural Network for classification.
- NaiveBayesClassifier: this class implements Machine Learning using the Naive Bayes Classifier from the NLTK package [4].

- MaxEntropyClassifier: this class implements Machine Learning using the Maximum Entropy classifier from the NLTK package.

Regression:

- SimpleRegressionNeuralNetwork: uses the native Pybrain NNregression class from Pybrain tools to set up an Artificial Neural Network for regression
- LinearRegression: this class implements regression using the LinearRegression class from the scikit-package [5].

All the models can be trained, evaluated (with or without using crossvalidation), saved and loaded.

5) datasets.py:

6) downloaders.py:

7) wordSentiment.py: We included Django blabla en nie zelf geschreven dus niet met pep enzo maar wel nodig en aangepast enal

IV. Results

A. Code checking

B. Testing

V. Discussion

VI. Conclusion

References

[1] B. Suh, L. Hong, P. Pirolli, and E. H. Chi, "Want to be retweeted? large scale analytics on factors impacting retweet in twitter network," pp. 177–184, Augustus 2010.

[2] D. Zarrella, "Science of retweets," 2009. [Online]. Available: http://danzarrella.com/science-of-retweets.pdf

[3] "Pybrain," 2010. [Online]. Available: http://pybrain.org/

[4] E. Loper, "Natural language toolkit." [Online]. Available: http://www.nltk.org/_modules/nltk/classify/naivebayes.html

[5] started by David Cournapeaul, "Scikit-learn: Machine learning in python." [Online]. Available: http://scikit-learn.org/stable/modules/linear_model.html

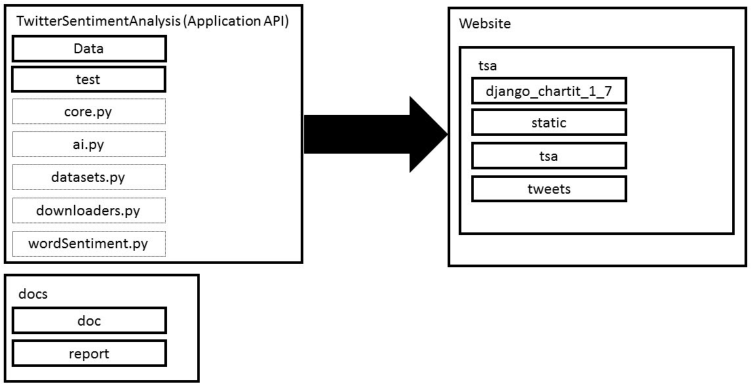

Fig. 1. Overview of the built software.

APPENDIX A
CODE LISTINGS

LISTINGS