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 results can be accessed through a website, which is implemented using Django.

II. Related work

In Suh et. al. [**?**] 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 [**?**] 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

IV. Results

A. Code checking

B. Testing

V. Discussion

VI. Conclusion

Appendix A
Code listings

Listings

APPENDIX B
AUTOMATIC GENERATION OF DOCUMENTATION

Demontration using epydoc:

```
epydoc --pdf -o /home/fnielsen/tmp/epydoc/ --name RBBase wikipedia/api.py
```

This example does not use `brede_str_nmf` but another more well-documented module called `api.py` that are used to download material from Wikipedia.