Determining the function of political tweets

©IEEE Accepted version of the paper appearing in the Proceedings of the 13th IEEE International Conference on eScience. IEEE, Auckland, New Zealand, 2017, pages 438-439, ISBN 978-1-5386-2686-3, DOI: 10.1109/eScience.2017.60. See also: ieeexplore.ieee.org

Erik Tjong Kim Sang
Netherlands eScience Center

Herbert Kruitbosch
University Of Groningen

Marcel Broersma
University of Groningen

Marc Esteve del Valle
University of Groningen

***Abstract*—We study the discursive practices of politicians and journalists on social media. For this we need annotated data but the annotation process is time-consuming and costly. In this paper we examine machine learning methods for automatically annotating unseen tweets based on previously processed tweets. For improving the performance of the learner, we focus on methods related to training data expansion, like artificial training data, active learning and incorporating language models developed from unannotated text.**

I. Introduction

We are interested in what discursive practices politicians and journalists use on social media. For this purpose we have created data sets of tens of thousands of manually annotated tweets written in four different languages. We would like to expand these data sets so that we can perform large-scale data analysis. However manual annotation is costly and time-consuming. Therefore we are examining methods for automatically assessing interesting latent properties of social media messages.

II. Related work

Graham et al. [3] studied the behavior of politicians on Twitter by a quantitative analysis of their tweets. Banko and Brill [1] showed that for natural language tasks, performance does not depend primarily on the chosen machine learning method but on the size and quality of the training data. Halevy et al. [4] promote the use of unlabeled data for machine learning because there is so much of it available.

III. Data

Our data sets consist of tens of thousands manually annotated tweets of politicians and journalists, written in four different languages (Dutch, English (UK), Swedish and Italian). For this study, we concentrate on the Dutch data from the elections of 2012: 55,029 tweets written by 372 politicians. The data have nine annotated features of which the tweet function topic is the most interesting. This feature can take one of 12 values, for example Campaign Trail (1), Campaign Promotion (2) or Critique (7). Table I contains a complete overview of the class values.

Additionally, we created two sets with unannotated data. The first consisted of all the tweets of 326 current Dutch national politicians available on Twitter on 18 April 2017 (covers the years 2009-2017). We removed tweets written in other languages than Dutch and tweets already present in our annotated set. This resulted in a data set of 251,279 tweets, mostly about politics. The second data set contains all Dutch tweets from the month March 2017 as collected by the website twiqs.nl [6]. This data set contains 21 million tweets. Although there was a parliament election in this month, this data set should be regarded as a general data set since most of its tweets are not about politics.

TABLE I
Topics, frequencies and prediction scores

Topic	Frequency	Precision	Recall	F1
Campaign Trail	1189	58.3%	64.1%	61.1%
Own / Party Stance	1033	46.7%	53.5%	49.9%
Campaign Promotion	989	47.7%	60.4%	53.3%
Critique	856	56.1%	53.6%	54.8%
Personal	560	47.0%	27.5%	34.7%
Acknowledgement	387	53.9%	62.5%	57.9%
News/Report	232	47.4%	27.6%	34.9%
Advice/Helping	172	42.9%	7.0%	12.0%
Requesting Input	36	100.0%	2.8%	5.4%
Campaign Action	30	100.0%	6.7%	12.5%
Other	15	0.0%	0.0%	0.0%
Call to Vote	4	0.0%	0.0%	0.0%
All test data	5503	51.7%	51.7%	51.7%

Finally, we selected 1,000 tweets from small unannotated political data set, to be used in active learning experiments (see section IV). 500 tweets were selected randomly from the 10% most difficult tweets for our machine learner: the tweets in which the confidence scores of the two best alternative values were closest to each other. The other 500 tweets were selected randomly from the complete data set, as recommended by Banko and Brill [1] to avoid tuning towards difficult tweets.

IV. Methods

We will predict the values of tweet function topic for unseen tweets automatically. For this purpose we use the machine learning method fastText [2] [5], using lowercased tokens[1] with words represented by vectors of 300 numbers each. The oldest 10% of the data set was reserved for test data while the most recent 90% were employed as training data. We find that the different feature values did not occur equally frequent (Table I, the column Frequency concerns frequency in the test data set). Because the random weight initialization of fastText, the performance of the learner will vary. For this reason, we repeated each run 25 times and where one run depended on the results of another, we repeated both the first run and the depended run five times.

We have applied three methods for improving the performance of the machine learner, all based on using additional

[1] All urls, email adresses and user mentions were collapsed before training to respectively the unique tokens HTTP, USER and MAIL.

TABLE II
EFFECTS OF ADDING TRAINING DATA

Method	Train size	Accuracy
Baseline: fastText, vector size 300	49,526	51.7±0.2%
+ extra data: 0.25M political tweets	300,805	51.1±0.2%
+ extra data: 21M general tweets	21,878,310	51.1±0.4%
+ language model, 0.25M political tweets	49,526	53.5±0.2%
+ language model, 21M general tweets	49,526	54.8±0.2%
+ language model, Wikipedia	49,526	54.2±0.3%
+ active learning data: 500 tweets	50,026	51.8±0.3%
+ active learning data: 1,000 tweets	50,526	51.6±0.3%
Ceiling: + test data as extra data	55,029	61.4±0.4%

unannotated data. The first method involves applying our machine learner to unseen tweets and using the tweets with their presumed classes as extra training data. With this approach we hope to increase the vocabulary associated with the different class values. It is called weakly supervised learning [1].

The second method involved using external skipgram language models. In our machine learner experiments, we represent words as vectors of numbers. These word vectors can also be obtained by training on large external unannotated text corpora [2]. Representation vectors of similar words will be similar. Using external corpora for word vector training will enable the machine learner to discover similarities involving words that are missing from its training data, thus again increasing its vocabulary.

The third method we applied, was active learning. The idea here is to create extra training data for the classifier, in particular for the items it found difficult to classify. Banko and Brill [1] found that by having the classifier select appropriate extra training data, less than 1% of their extra data needed to be annotated to get the same performance as with the complete extra data set. They recommend including extra randomly selected data in this process to avoid tuning the classifier towards difficult examples.

V. RESULTS

We applied fastText [2] [5] to the annotated data. For each experiment, 25 runs of fastText were performed and we report average accuracy[2] scores of all runs. The initial experiment achieved an accuracy score on the test data of 51.7% (Table I), which is not very high. However, the task is quite difficult. Graham et al [3] report a kappa score of 0.66 for a similar annotation class in Dutch data of a previous election. This corresponds with an interannotator agreement of 71%. We estimated the ceiling performance of the machine learner by analyzing the test data with a model learned from the training data *and* the test data. The accuracy of this model (61.4%) again shows that this is a difficult task (Table II).

In our first effort to improve the performance of the machine learner, we classified the two unannotated data sets with fastText and then added the classified tweets to the training data. Unexpectedly, the performance decreased both with the small topical data set and with the large general data set (51.1%, see Table II). Like Banko and Brill [1], we tried using only the most reliable classifications as extra training data but this also failed to produce better models.

Next, we build five skipgram language models for each of the two unannotated data sets. For each of the language models, we repeated the baseline experiment five times but this time using the word vectors from the models. In the 25 runs for the small topical data set we obtained a performance increase of about 2% (to 53.5%) over the baseline score while for the large general data set the performance increase was 3% (to 54.8%, see Table II). We also evaluated an external language model trained on Dutch Wikipedia[3] This obtained a performance increase of about 2.5% (to 54.2%, see Table II).

In our active learning experiments we used subsets of 500 and 1,000 tweets of the small political unannotated data set of 251,279 tweets. In each data set, half of the tweets were selected from the most difficult tweets while the other half were selected at random. Unfortunately these tweets as additional training data did not enable the machine learner to improve its performance (Table II).

VI. CONCLUDING REMARKS

We have an annotated data set which we want to expand with new data by applying a machine learning model to the unseen data. However, the task proved to be hard. Human interannotater agreement is only 71% and the machine learning method fastText obtains no more than 51% on this task. We have tried to improve these scores by applying three data-based techniques: using additional artificial training data, language models and active learning. Of the three methods, only the language models boosted performance. Of the three language models we tested, the one learned from the largest available set of tweets, worked best (+3%, see Table II).

We have shown that additional unannotated data can help our classification task, at least when they are used for building improved language models. Our next focus will be testing the effect of language models trained on even larger sets of tweets.

REFERENCES

[1] Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th annual meeting on Association for Computational Linguistics*, pages 26–33. Association for Computational Linguistics, 2001.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[3] Todd Graham, Dan Jackson, and Marcel Broersma. New platform, old habits? Candidates use of Twitter during the 2010 British and Dutch general election campaigns. *New Media & Society*, 18:765–783, 2016. DOI: 10.1177/1461444814546728.

[4] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24:2:8–12, 2009.

[5] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[6] Erik Tjong Kim Sang and Antal van den Bosch. Dealing with Big Data: the Case of Twitter. *Computational Linguistics in the Netherlands Journal*, 3:121–134, 2013. ISSN: 2211-4009.

[2] The accuracy scores reported for the test set are always the same as the precision, recall and F1 scores. Only for individual class values, precision, recall and F1 scores can have different values.

[3] github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md