From Sentiment to Reputation

ILPS at RepLab 2012

Maria-Hendrike Peetz, Maarten de Rijke and Anne Schuth
{M.H.Peetz, deRijke, A.G.Schuth}@uva.nl

ISLA, University of Amsterdam

Abstract. We report on our participation in the *profiling* task of the first edition of the CLEF RepLab evaluation initiative. We assume that a statement—such as a tweet—that caries negative sentiment can have a positive impact on the reputation of the entity it talks about (and vice versa). Our model directly captures this impact by observing the reactions—such as replies—the statement solicits. We present the assumptions behind our model and the model itself. We find that given the current setting, results on the test set are strongly entity-dependent and that the test data is very different from the trial data. We conclude with a proposal on how to create a task that avoids such dataset dependent problems.

1 Introduction

The Reputation of an entity is an opinion about that entity, typically in a social context. Twitter as a opinion-conveying medium provides this social context: here, reputation is what Twitterers think of the entity.
The reputation task in RepLab was to identify if a tweet's content has positive or negative implications on the reputation of a brand name. A tweet has mainly implications on opinions of people it reaches, directly or indirectly. Thus, an oracle or manual annotation would select all tweets uttered by Twitterers directly after they were read. One could then analyse manually if the original tweet had any implication on the opinion about the brands in those uttered tweets. Unfortunately, this is infeasible on many different levels. However, if we assume that tweets uttered in response to a tweet convey the implication on reputation of the entity mentioned in this tweet, we come to the first research question:

> Can we bootstrap the implication on reputation from sentiment-annotated tweets in the replies and retweets to a source tweet?

This research question implies that our understanding of reputation is something very different to sentiment.
Additionally, tweets have a lot of metadata that may contain information as to how far a tweet can be considered polarized with respect to reputation.

> Can we use machine learning to learn an appropriate combination of features to classify the polarity of a tweet?

Our work is organized as follows: We first describe our filtering methods in Section 2, we then continue with our polarity methods in Section 3. In Section 4, we describe how we use a machine learning approach to perform feature selection and classification. We then describe our experiments in Section 5. Results and and analysis are presented in Section 6. We finish with a conclusion in Section 7.

2 Filtering Methods

The filtering task is to classify a tweet as relevant to a source entity or not. We were provided with the Wikipedia pages of a source entity. For the disambiguation of the source entities we semanticised the tweets with Wikipedia pages and disambiguated on the grounds of these pages. For each entity, we automatically assemble sets of Wikipedia pages that, if they are linked to in a tweet, this indicates the relevance of the tweet for a source entity.
In the baseline we assume all tweets to be relevant for a source entity.
In the following, we lay out how the related Wikipedia pages for a tweet are found using semanticising (see Section 2.1) and how from this, sets of entities (as their Wikipedia page) that are related to the source entity are created (see Section 2.2). In Section 2.3, we shortly explain how relevant tweets are selected.

2.1 Semanticising

Each tweet can have possible semantic links to Wikipedia pages. Finding those links means disambiguating and finding concepts in a tweet. Following Meij et al. [2012], we use two features: the LINKPROBABILITY and the COMMONNESS feature. The earlier is the probability that an n-gram in a tweet is a link in Wikipedia: how many occurrences of this n-gram are actually within hyperlinks to a page? The second feature COMMONNESS, is the probability of an n-gram to link to a certain concept. The product of the two features is the number of links to a concept if this n-gram is a link to a Wikipedia page.

2.2 List Aggregation

Top Entities For each source entity, we aggregate the number of times Wikipedia pages are linked in tweets. The top N most linked pages are the set TOPPAGES.

Entities in Wikipedia Page Another group on entities is selected with the help of the provided Wikipedia pages of the source entities (SOURCEPAGES). Here, we select all outgoing links to internal Wikipedia pages. Those pages are called WIKIPAGES.

Combination of List For each source entity, TOPWIKIPAGES is the intersection of the sets TOPPAGES and WIKIPAGES. Additionally, every list contains the pages in SOURCEPAGES.

2.3 Disambiguation

Finally, for the disambiguation, we assume that a tweet is relevant to a source entity if

1. there are links to Wikipedia pages found by the semanticiser, and
2. those links are in a set of related Wikipedia pages for the source entity: either TOPWIKIPAGES, TOPPAGES, WIKIPAGES, or SOURCEPAGES.

Our runs for the filtering task differ in the use of the set of related Wikipedia pages.

3 Polarity Methods

The polarity task asks to classify a tweet for a given entity into having an impact on the reputation of that entity or not. There are three classes of polarity, positive, negative, and neutral.
This section proposes two groups of models: sentiment models (see Section 3.1) and reputation models (see Section 3.2). The two sentiment models build upon another, where sentiment model 1 is the first iteration of the iterative sentiment model 2. All reputation models are iterative and based on sentiment terms. They differ in the way they split positive and negative polarity vectors and in their initialization.

3.1 Sentiment Baselines

In the following we introduce two sentiment models. Sentiment model 1 (see Section 3.1) estimates sentiment based on the sentiment value of terms in a tweet, whereas sentiment model 2 (see Section 3.1) uses this as a initialization for an iterative approach.

Sentiment Model A simple way of estimating sentiment is to define sentiment as the sum of the sentiment of terms in a tweet.
Manually annotated sentiment lists can be found in Hu and Liu [2004], Liu et al. [2005], and Pérez-Rosas et al. [2012]. We say $S(w)$ is the sentiment for a term w. The sentiment for a tweet t and its terms $\mathit{terms}(t)$ is

$$\text{sent}(t) = \frac{1}{|\mathit{terms}(t)|} \sum_{w \in \mathit{terms}(t)} S(w). \tag{1}$$

We refer to this model as sentiment model 1.

Iterative Sentiment Model Language use in Twitter is very different from traditional texts. We use a more elaborate sentiment model where the sentiment terms are learnt on a Twitter corpus. For that, we use the sentiment vectors $S(w)$ from Section 3.1 and learn Twitter specific sentiments in an iterative approach.
We estimate the sentiment of a tweet t in iteration i as

$$\text{sent}_i(t) = \frac{1}{|\mathit{terms}(t)|} \sum_{w \in \text{terms}(t)} S_i(w), \tag{2}$$

and update the sentiment vector $S_i(w)$ based on all tweets T

$$S_i(w) = \sum_{t \in T} \text{sent}_{i-1}(t). \tag{3}$$

We refer to this model as sentiment model 2. The initial sentiment $\text{sent}_0(t)$ is equivalent to the sentiment in sentiment model 1.

3.2 Unsupervised Reaction Models Bootstrapped with Sentiment

The goal of this method is to learn the polarity of *words* with respect to an entity. In the sentiment models, we simply use the general sentiment that words carry, irrespective of the entity that is talked about. From examples, we know that the baseline approach is too simplistic. Depending on the context—the entity in question—the polarity of a word can be completely opposite from the general sentiment it caries. The obvious example is: *R.I.P. Michael Jackson, we miss you*. In this example, words carry negative sentiment (sadness) while the statement itself has a positive impact on the reputation of Michael Jackson. In the context of another entity, however, these words can carry a negative impact on the reputation. In this model, we intend to learn this in an unsupervised manner.
In the following we lay out the assumptions underlying the models in Section 3.2 and introduce the different reaction models, Model 1 (see Section 3.2), Model 2 (see Section 3.2), and Model 3 (see 3.2).

Assumptions Based on interviews with experts we hypothesize the following:

1. The message in a tweet is not necessarily about the entity we are concerned with. But, as tweets are rather short, we assume it is about some entity as soon as we find a reference to it.
2. A tweet with positive (negative) sentiment from a user who tweets mainly negative (positive) tweets has more impact on the reputation.
3. Positive sentiment can cancel negative sentiment and vice versa; positive reputation can cancel negative reputation and vice versa.
4. The impact on the *reputation* of an entity as represented in a tweet is based on the *sentiment* the tweet causes in other users.

Assumption 4 is the intuition that underlies our model. We hypothesize that the impact of a statement on reputation can be deduced from the sentiment of reactions.

Reaction Model 1 We propose an iterative approach to estimate an entity e specific term vector $W(e)$. The term vector is initialized with sentiment terms, similar to sentiment model 2 (see Section 3.1). We assume that the impact of reputation can be measured by the kind of replies and retweets it solicits. Thus, every iteration we estimate the polarity of a tweet based on the polarity contribution of the retweets and replies to a tweet. At the end of the iteration we update the term vectors W_i based on this estimated polarity of a tweet and the previous term vector W_{i-1}. We assume that after N iterations $W_N(e) \approx W(e)$ for all entities and we can estimate the polarity of every tweet, even if unseen.

Reaction Model 2 The number of tweets with positive and negative sentiment is skewed in the dataset. This influences the estimation of the term vector W — positive and negative influences do not cancel another out.
We propose separate reputation vectors W^+ and W^-. The difference to Model 1 is the estimation of the polarity contribution and the iterative updating W^+, W^-: here we have different vectors for positive and negative polarities. As the reputation vectors are normalized at the end of each iteration and the influence of positive tweets is not overwhelming the negative tweets.

Reaction Model 3 The third reaction model differs from Model 1 with respect to the initialization. In Model 1 (see Section 3.2), the initial vector $W_0(e)$ is the sentiment vector S, so $W_0(e, w) = S(w)$. In this model, Model 3, the vector of the original sentiment does not interpolate into W_1. That way, we have a stronger focus on the the actual reputation and have only terms in the sentiment lexicon that feature a strong polarity within Twitter.

4 Classification

We use a machine learning approach for classification the classification of polarity. We view all our models, described in Section 3.1 through Section 3.1, as features and train a classifier based on the trial data.

4.1 Feature descriptions

For each tweet we collect 25 features. Those 25 features include the sentiment and reputation models, as well as metadata features.

reactionmodel1 as described in Section 3.2
reactionmodel2 as described in Section 3.2
reactionmodel3 as described in Section 3.2
sentimentmodel1 as described in Section 3.1
sentimentmodel2 as described in Section 3.1
scaledreactionmodel3 scaled—or, centered—version of the *reactionmodel3* feature
scaledsentimentmodel1 scaled—or, centered—version of the *sentimentmodel1* feature
entity a reference to the entity as provided by the track organizers. Note that this feature can not be used in classifier that generalizes to the test collection.
lang detected language
knownlang whether *lang* is either *english* or *spanish*, the languages for which we have sentiment lexicons
nrrt the number of retweets
nrrp the number of replies
nrreact the total number of reactions (*nrrt* + *nrrp*)
nrpos the number of reactions with positive sentiment
nrneg the number of reactions with negative sentiment
nrreactionfriends the sum of the number of friends of the authors of all reaction tweets

fractionpos the fraction of positive reactions (*nrpos* / *nrreact*)
fractionneg the fraction of negative reactions (*nrneg* / *nrreact*)
reactpossum the sum of sentiment of negative reactions
reactnegsum the sum of sentiment of positive reactions
friends the number of friends
favorite whether a tweet was favorited
userreactionmodel3 the sum of the reactionmodel3 for this user
usercount the number of thwarts from this user
useravgreactionmodel3 the average value of *reactionmodel3* for this user

4.2 Classifier

We train a simple tree classifier[1] using the above features and subsets of these features on the trial data. We select the subsets of features based on the information gain of individual features, as illustrated in Table 5.

5 Experimental setup

In this section we describe the experiments to answer the research questions mentioned in Section 1. We describe the official and external datasets as well as their preprocessing in Section 5.1. The runs and their evaluation are described in Section 5.2.

5.1 Data

Twitter Dataset We used the dataset provided by the organizers of RepLab@CLEF. The dataset was split in labeled (unlabeled of the test set) and background datasets. In particular, the background dataset contains 238,000 and 1.2 million tweets for trial and test set, respectively. This means 40,000 and 38,000 tweets per entity, respectively. The set of labeled tweets in the trial dataset contains 1649 tweets, of which we managed to download 1553 tweets (94.1%). The set of unlabeled tweets for the test data contains 12400 tweets, of which we managed to download 11432 tweets (92%).

Replies and Retweets to Tweets The reputation models are based on the reactions to the tweets. For us, a reaction is a tweet that is either a reply or a retweet. We extracted $\sim$ 434,000 (17,000 per entity) reactions from the test background dataset and $\sim$ 50,000 (8,000 per entity) from the trial background dataset. These are supplemented with all ($\sim$ 228,000,000) reactions from an (external) Twitter spritzer stream after the earliest date of a tweet in either trial or test data (25 October 2011). Those reactions were not necessarily reaction to tweets in the background and (un)-labeled corpora. Consider Table 1 for the number of reactions to tweets in the background dataset.

Sentiment Lexicons We use publicly available sentiment word lexicons in English [Hu and Liu, 2004, Liu et al., 2005] and Spanish [Pérez-Rosas et al., 2012] as the vast majority of tweets are in either of these languages.

[1] We use the WEKA [Hall et al., 2009] implementation of C4.5 by Quinlan [1993]

	trial data				test data			
	mean	min	max	std	mean	min	max	std
#retweets	4767	2620	8982	2131	5282	2059	14831	2925
#replies	72	28	151	39	554	57	1806	464
#reactions	4839	2648	9066	2153	5836	2203	15119	2930

Table 1. Mean reactions per entity, statistics per dataset. The min, max and standard deviation are shown as well. Note that the number of replies is very different for the test data.

Preprocessing We preprocess all tweets using the following procedure:

1. separate punctuation characters from word characters but:
2. keep mentions, hashtags and smilies intact;
3. casefolding;
4. tokenize by splitting on whitespace.

Additionally, we perform language identification on tweets using the method described in Carter et al. [2013].

5.2 Evaluation

We participated with 5 runs, see Table 2 for a description of these runs. The sentiment models were trained on the entire background corpora, entity unrelated. The reputation models were trained on the reactions as explained in Section 5.1. We estimated the performance of each run on the trial data for polarity and filtering separately and paired the best polarity run with the best filtering run, the second best polarity run with the second best filtering run, etc.
The evaluation measures we use are accuracy for the polarity and F-score for the relevance filtering.

6 Results and Analysis

In this section we answer the research questions mentioned in Section 1. We first analyze the official results of the runs in Section 6.1. Section 6.2 analyses how a different approach to set up the experiments is likely to be more realistic and successful in estimating polarity and relevance for tweets given an entity.

6.1 Results of the Runs

Table 3 shows the results of our runs on the trial data and the test data. We can see that the performance with respect to the evaluation measures of the test runs are roughly inversely proportional to the performance with respect to the evaluation measures of the trial runs for the polarity task as well as the filtering task.

run	polarity name	description	filtering name	description
ilps_1	best	J4.8, feature selection with: reactionmodel2, reaction-model3, sentimentmodel2, reactionmodel1, lang, senti-mentmodel1, knownlang	allrel	all tweets are relevant
ilps_2	model1	*best* excluding reactionmodel2 and reactionmodel3	top30-0.2-outlinks-origin	tweets where one linked Wikipedia page is in TOP-WIKIPAGES, with $N = 30$
ilps_3	model2	*best* excluding reactionmodel1 and reactionmodel3	top30-0.2-origin	tweets where one linked Wikipedia page is in TOP-WIKIPAGES, with $N = 30$
ilps_4	model3	*best* excluding reactionmodel1 and reactionmodel2	outlinks-origin	tweets where one linked Wikipedia page is in WIKIPAGES
ilps_5	non-sent	all features excluding sentiment or reactions models and the en-tity feature	onlyWikipedia	tweets where one linked Wikipedia page is SOUR-CEPAGES
base_6	random	assigns classes randomly		
base_7	zero	picks the majority class		
base_8	all	J4.8 using all features		
base_9	all+ent	J4.8 on *all* features plus entity information		
base_10	best+ent	J4.8 on the *best* features plus en-tity information		

Table 2. Run descriptions, sorted in descending performance on the trial set (see Table 3) for polarity and filtering independently. The first 5 runs were submitted, the others serve as baselines for comparison in our analysis.

In particular, for the polarity task our best runs on the trial data are using all reputation and sentiment models and the language feature, while on the test data, this performs worst with respect to accuracy: the run with the highest accuracy uses no reputation models at all.

Table 1 shows the number of reactions and replies for the trial and test data. We can see that for the test data we used significantly more replies than the trial data, while the number of retweets remains about the same. We suspect that with a higher number of replies comes more noise that misguides the bootstrapping approach. In this respect, the trial and test data are very different and it is only natural that this is reflected in the quantitative evaluation.

For filtering, the highest F-score on the trial set was using all tweets. All more informed attempts to disambiguate could never reach the F-score of 96%: We found that the bigger

run	polarity accuracy trial	test	filtering F-score trial	test
ilps_1	0.77	0.36	0.96	0.28
ilps_2	0.69	0.38	0.84	0.36
ilps_3	0.71	0.41	0.85	0.35
ilps_4	0.74	0.40	0.87	0.34
ilps_5	0.61	0.43	0.79	0.29
base_6	0.33	0.33	-	-
base_7	0.55	0.44	-	-
base_8	0.74	-	-	-
base_9	0.81	-	-	-
base_10	0.82	-	-	-

Table 3. Results on the trial and test data for both the polarity and filtering. For most baseline runs, no test data is available.

the entity lists (thus recall) the higher the F-score. The relevance assignment by the retrieval method in the dataset creation seemed to have been very powerful.
The picture is different for the test data. Here, the F-score for the filter that considers all tweets to be relevant is 0.28, the lowest F-score of all. The best performing approaches are using the TOPPAGES set, either intersecting with WIKIPAGES or on its own. Again, in the trial set the observation is reversed: using the WIKIPAGES set lead to a higher performance with respect to F-score than using the TOPPAGES set.
On an entity-level we can see that for 13 out of the 31 tweets the baseline assigning all of the relevance performs best. However, it does hurt the filtering performance with respect to F-score for other entities so much that F-score drops to be the worst. The run ilps_2, even though it performs best with respect to the overall F-score, only has higher F-scores than ilps_2 for 8 entities, but the difference in F-score is on average 0.55, with the F-score for ilps_1 being zero or near zero in 5 out of the eight cases.

6.2 Entity-specific annotation

Table 5 shows the ranks of the features used for polarity of trial data when sorted by information gain. The feature **entity** encodes for which entity the datapoint (tweet) is supposed to be classified. Of course, this feature can not be used in a classifier trained for the test set. We can see that knowing the entity in beforehand has the greatest information gain. The accuracy of base_9 and base_10 on the trial set feature is 0.82, 25% better than the runs without this prior information. The trial set is too small for elaborate analysis, but we conclude that for the entities used in the trial set, a manual entity-specific seed annotation is more useful than an entity-ignorant annotation. As the number of entities is limited, we propose to manually annotate tweets for every entity and train classifiers on those tweets for future incoming tweets. To ensure that changes in the use of language in the tweets over time are captured, an adaptive interactive interface for the reputation manager seems most convenient.

	polarity			relevance	
run	neutral	positive	negative	yes	no
ilps_1	7433	3562	437	11432	0
ilps_2	7032	4191	209	5593	5839
ilps_3	5841	5036	555	7597	3835
ilps_4	6589	4638	205	6504	4928
ilps_5	5643	5700	89	5083	6349

Table 4. Runs on the test data with distribution of the 11432 instances over classes. If a value for the relevance was missing it was assumed to be relevant.

7 Conclusions

In general, we found that the trial and test set were very different. For the polarity task we are able to say that reputation models works well for all trial entities, but not for the test entities. Additionally, we also found that for the filtering task the best performing run strongly varied per entity.
Therefore, for future reputation management tasks we propose a more natural setting, where training entities and evaluation entities are the same. Entities are very different, and given the manpower of reputation management companies, it seems feasible to annotate a batch of tweets for each new entity that needs to be monitored. Results are likely to be more reliable and useful.

Acknowledgments. This research was supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 258191 (PROMISE) and 288024 (LiMoSINe), the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.061.814, 612.061.815, 640.004.802, 380-70-011, 727.011.005, 612.001.116, the Center for Creation, Content and Technology (CCCT), the Hyperlocal Service Platform project funded by the Service Innovation & ICT program, the WAHSP and BILAND projects funded by the CLARIN-nl program, the Dutch national program COMMIT, and by the ESF Research Network Program ELIAS.

8 References

S. Carter, W. Weerkamp, and E. Tsagkias. Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text. *Language Resources and Evaluation Journal*, 2013.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The WEKA data mining software: an update. *SIGKDD*, 11(1):10–18, 2009.

M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

information gain	feature
0.29387	entity
0.193489	reactionmodel3
0.107189	reactionmodel2
0.10614	sentimentmodel2
0.099352	useravgreactionmodel3
0.077697	userreactionmodel3
0.063932	lang
0.035735	reactionmodel1
0.012383	sentimentmodel1
0.000788	knownlang
0	nrrt
0	friends
0	favorite
0	scaledreactionmodel3
0	fractionneg
0	usercount
0	reactnegsum
0	scaledsentimentmodel1
0	reactpossum
0	nrpos
0	nrrp
0	nrreact
0	fractionpos
0	nrneg
0	nrreactionfriends

Table 5. Attribute ranker that uses information gain, produced with WEKA.

B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web*, pages 342–351. ACM, 2005.

E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *WSDM '12*, pages 563–572, 2012.

V. Pérez-Rosas, C. Banea, and R. Mihalcea. Learning sentiment lexicons in spanish. In *Proceedings of the International Conference on Language Resources and Evaluations*, 2012.

J. Quinlan. *C4. 5: Programs for machine learning*. 1993.