ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-339

Detecting the Reputation Polarity of Microblog Posts

Cristina Gârbacea and **Manos Tsagkias** and **Maarten de Rijke**[1]

Abstract. We address the task of detecting the reputation polarity of social media updates, that is, deciding whether the content of an update has positive or negative implications for the reputation of a given entity. Typical approaches to this task include sentiment lexicons and linguistic features. However, they fall short in the social media domain because of its unedited and noisy nature, and, more importantly, because reputation polarity is not only encoded in sentiment-bearing words but it is also embedded in other word usage. To this end, automatic methods for extracting discriminative features for reputation polarity detection can play a role. We propose a data-driven, supervised approach for extracting textual features, which we use to train a reputation polarity classifier. Experiments on the RepLab 2013 collection show that our model outperforms the state-of-the-art method based on sentiment analysis by 20% accuracy.

1 INTRODUCTION

Reputation management has become a key component in designing the marketing strategy for businesses.[2] Reputation managers monitor and analyze social data related to an entity for alarming signals and take actions to prevent turn overs in the reputation of their customers. Reputation management has moved from offline to online, but algorithmic support for online reputation management is still limited.

The importance of reputation management is illustrated in a recent incident involving the Coca-Cola company.[3] In February 2014, the company released a series of adverts that were aimed at raising awareness about obesity. The ads quickly became a controversial topic of debate in the news because of the nutritional facts of the company's product line. To counter for the negative publicity the company retracted the ads' videos from YouTube until a consensus was reached. Reputation managers analyzed the news, broadcasts and polled public opinion to assess the impact on the company's reputation. They found that although the ads were negatively received by experts in commercial and scientific communities, end consumers regarded the ads with positive sentiment, and that the company's reputation had, overall, been strengthened.

The takeaway message of this incident is that despite the trend setting role of news and experts, it is important to poll public opinion in a direct and rapid manner. The utility of traditional data sources (e.g., news outlets, broadcasts, and surveys) here is limited because they neglect large parts of the end consumer population and they suffer from an inevitable time lag between when something happens and when it is reported. A natural way to overcome these limitations is to mine information from the online world, and from social media in particular. Social media have become the de facto outlet for self-reporting on "what's happening right now," and for people to share viewpoints on products, brands, and organizations. The real-time and personal nature of social media content makes it a proxy for public opinion and a source for tracking reputation management [25].

Online Reputation Management (ORM), the online flavor of traditional reputation management, tracks and analyzes online content using automatic methods for monitoring the reputation of real-world entities, e.g., organizations, people, or products. ORM consists of two main steps. In the first step, an input stream of documents is filtered to obtain documents that are relevant for a given entity. In the second step, each relevant document is automatically assessed for its reputation polarity. Reputation polarity refers to whether a document will have a positive, neutral, or negative impact on the entity's reputation. In this paper, we concentrate on the second step, namely, developing effective methods for detecting the reputation polarity of a social media update.

More precisely, the *reputation polarity detection* task is defined as follows. Given an entity and a social media update (tweet) relevant to this entity, we want to classify whether the tweet content has positive, neutral or negative implications for the entity's reputation. One of the many ways to do so is to use methods from information analysis and text mining for extracting features, and to use them for training a reputation polarity classifier at a later stage. Deciding on what features to extract (i.e., feature engineering) is currently a manual process. An early observation was that sentiment affects reputation; as a consequence, sentiment analysis methods were used as top-class feature extractors for detecting reputation polarity. Textual features and topic modeling have also been investigated as a means to capture more context; see Amigó et al. [1] for an overview.

We focus on improving textual feature selection to learn discriminative features for each reputation polarity class. This choice, rather than a focus on improving sentiment analysis, is motivated by three limitations we identify in lexicon-based sentiment-analysis approaches for reputation polarity detection. First, they require the development of language-specific sentiment lexicons, which are expensive to assemble and maintain as they depend on human labor. Second, the short, noisy, and unedited text of social media updates limits the coverage of the lexicon, and results in less effective sentiment analysis compared to edited texts [8, 20]. Third, and most importantly, reputation polarity is not only encoded in sentiment-bearing words but also embedded in other words, including, for instance, entities. For example, as we will see later in Table 2, mentions of financial organizations are generally correlated with negative reputation polarity, while mentions of entertainment-related celebrities are associated with positive reputation polarity. Our observations distance reputation polarity from traditional sentiment analysis and amplify the need for automatic methods that capture discriminative words for the task at hand, which are not necessarily constrained to sentiment-

[1] University of Amsterdam, The Netherlands. Email: c.garbacea@uva.nl, e.tsagkias@uva.nl, derijke@uva.nl
[2] http://bits.blogs.nytimes.com/2011/04/04/the-growing-business-of-online-reputation-management/
[3] http://opinionator.blogs.nytimes.com/2013/01/22/coke-blinks

bearing words.

We develop methods that cope with all three of these challenges. Viewed abstractly, sentiment words are a subset of the words in the entire vocabulary, and we aim at extracting this subset (and more) automatically via short-circuiting the sentiment analysis step. This insight has important repercussions in other tasks where sentiment analysis plays a role. It means that for a given task we can automatically extract a discriminative set of words using text mining methods that are independent of domain or language (lexicon-based sentiment analysis is costly to acquire because it needs human annotations per language). This property makes such methods easier to apply because they require less human annotations, and also are significantly more effective when compared to sentiment analysis methods.

We approach the problem of detecting reputation polarity using a supervised method. We start with annotated documents based on their reputation polarity: positive, neutral, or negative. We use these annotations to build three corpora, each one corresponding to a reputation polarity class. Then, we contrast the effectiveness of five state-of-the-art methods for extracting textual features, and test the approaches on an end-to-end reputation polarity detection scenario. The main research question we aim to answer is: What is the effectiveness for reputation polarity detection of using (i) sentiment analysis methods, and (ii) their combination with words as features. Our main contribution is a method that can automatically select discriminative features for detecting the reputation polarity of a social media post without the need of lexicon-based sentiment analysis.

§2 covers related work. In §3 we describe the sets of features we consider and zoom in on methods for extracting discriminative textual features for identifying reputation polarity, in §4 we describe our experimental setup, in §5 we report on our results, in §6 we provide an in-depth analysis of our findings, and finally we conclude in §7.

2 RELATED WORK

We present work in reputation polarity detection that is related to ours. The reputation polarity task is one of the tasks in the evaluation campaign RepLab [27], which aims to provide an evaluation testbed for developing methods for ORM. As we describe in §4, RepLab provides an annotated corpus of tweets in English and Spanish for a large set of entities. Each tweet is associated with an entity, and human annotators have assessed the reputation polarity of a tweet for the entity as positive, neutral, or negative. This set consists of the training data; a similar set has been provided as test data.

Most participants approached the problem as a classification problem, and focused on extracting discriminative features for reputation polarity detection. All systems use some sort of sentiment lexicons either to directly probe the sentiment polarity of a tweet, or to extract features based on sentiment. Most participants used textual features on top of sentiment features [5, 9, 11, 16, 28, 29], and a range of textual selection methods have been explored, e.g., frequency [28], tf.idf [5] with Gini purity score [6], information entropy [9]. Enhancing the representation of tweets via clustering methods [5, 9], and incorporating external content from the links found within tweets has also been tried [5]. We describe two systems of particular interest.

Spina et al. [29] use a semantic graph approach for extending sentiment lexicons: nodes represent WordNet concepts and the edges represent semantic relations between concepts. They extract features based on this improved sentiment analysis method for training a classifier. They find that errors in linguistic analysis propagated and amplified in the final output. Also, they find that because the vocabulary contains more positive labeled words, it biases the entire graph towards positive reputation polarity. Our approach is independent of linguistic analysis and domain characteristics.

Hangya and Farkas [11] follow a very elaborate data cleansing procedure before extracting surface, sentiment, and textual features. Their system was the best performing in RepLab and we consider it our baseline. We follow their preprocessing steps, and their use of surface, sentiment, and textual features so our approaches are comparable to theirs. Our approaches differ in how we perform the selection of textual features and we provide a more in-depth analysis of the importance of each set of features.

Finally, our work is close to that of [21], who investigate the usefulness of information retrieval weighting schemes for sentiment analysis, but we concentrate on reputation polarity detection, and the combination of surface, sentiment, and textual features.

3 FEATURE ENGINEERING

We consider three sets of features: (i) surface, (ii) sentiment, and (iii) textual. We motivate and describe each set in turn, below.

3.1 Overview of features used

Surface features. We identify six surface features that can encode reputation polarity: (i) number of words with overly repeating letters as character repetition can be indicative of emotions; (ii) the number of words in all capitals as indication of shouting, the number of positive (iii) and negative (iv) emoticons, (v) whether the entity name is mentioned in the tweet, (vi) the number of negation words as they can change the polarity of the tweet.

Sentiment features. Sentiment plays a role in the polarity of a tweet [25]. We use SentiWordNet [7] to gauge the sentiment of a tweet. We encode sentiment as the sum of SentiWordNet scores for all positive, neutral, and negative terms individually normalized against the tweet length, which results in three features respectively. Abstract linguistic features have been found to improve accuracy of sentiment analysis methods [10] because they are likely to capture the variance in a word's meaning, e.g., "love" in "I love this movie" (indicating sentiment orientation) vs. "This is a love story" (neutral with respect to sentiment). We consider the part-of-speech-tag (POS) of the term and retrieve the SentiWordNet score that corresponds to the appropriate category only. This results in three additional features, similar as before but using the POS tag of terms. Additionally, we expand acronyms with their mappings from an online resource,[4] e.g., *h82sit* is expanded to *hate to say it* and *gr8* is replaced by *great*, resulting in two features encoded as the sum of (i) positive and (ii) negative acronyms found in the post.

Textual features. Pang et al. [22] found that linguistic features are less useful than unigrams for sentiment analysis. We expect to find additional evidence for this finding in our setting because of the noisy character of microblog posts, where sentiment is hard to capture using only predefined lexicons such as SentiWordNet. For this reason we consider unigrams and bigrams as textual features. The higher order n-grams aim to better capture the "context" of a term [23]. In the next section we describe methods for selecting textual features that are discriminative of the polarity of a microblog post.

3.2 Textual feature selection

Our main methodological contribution is a comparative study of methods for extracting textual features that aid in detecting the rep-

[4] http://www.noslang.com

Table 1. Features we consider for the reputation polarity detection task.

Type	Gloss
Surface	Shouting
Surface	Positive emoticons
Surface	Negative emoticons
Surface	Character repetition
Surface	Entity mention
Surface	Negation words
Sentiment	Sum of positive terms
Sentiment	Sum of neutral terms
Sentiment	Sum of negative terms
Sentiment	Sum of positive terms (POS)
Sentiment	Sum of neutral terms (POS)
Sentiment	Sum of negative terms (POS)
Sentiment	Positive acronyms
Sentiment	Negative acronyms
Textual	Unigrams and/or bigrams

utation polarity of a microblog post for given entity. The main idea is to leverage the training data not only for training a classifier, but also for learning what are discriminative textual features for each class. In particular, we regard each annotation set as a corpus and extract features from each corpus with well established, state-of-the-art methods from text mining [30] and information retrieval [14].

We consider five methods: (i) frequency, (ii) tf.idf, (iii) χ^2, (iv) Log-Likelihood Ratio (LLR), and (v) Latent Dirichlet Allocation (LDA). All methods draw candidates from looking at only one corpus at a time except χ^2 and LLR. For the frequency method for example, we consider the top-N scoring terms for the positive corpus, for the neutral corpus, and for the negative corpus. χ^2 and LLR compare the statistics of a term in one corpus with its statistics in the other two corpora before scoring the term. We hypothesize that χ^2 and LLR will generate more discriminative textual features than the rest of our methods. Below, we describe each method we consider.

Frequency. Frequency distribution is one modality of automatically detecting the most informative words in a text. For each polarity class we rank terms by their term frequency and use the top-N most frequent terms as features.

tf.idf. A more elaborate method for weighing discriminative terms is tf.idf. Given a corpus C of d microblog posts and a term t, the tf.idf score is computed as $tf \cdot \log(\frac{|C|}{|\{d \in C : t \in d\}|})$, where tf is the term's frequency, and $|C|$ the number of posts in C. For each polarity class we rank terms by their tf.idf score and select the top-N as features.

χ^2. For discovering keywords that differentiate one corpus from another, frequency profiling methods can be used. A good statistical goodness of fit measure is the χ^2 test, a supervised learning method that determines the correlation of two words in two corpora by comparing the observed co-occurrence frequencies of these words with their expected frequencies, when they are assumed to be independent [15]. The greater the difference between the observed and expected values, the less likely it is that this difference is caused by chance and that the two words are random samples of the same population. One known shortcoming is that when a relatively small corpus is compared with a much larger one, χ^2 becomes unreliable and sensitive to small expected frequencies, presenting a tendency to overestimate with high frequency words which makes it very dependent on the sample size [13].

Log-Likelihood Ratio (LLR). LLR is another approach for identifying discriminative terms between corpora and has proven useful in classification and regression tasks [19, 26]. To extract the most discriminative words for a polarity class, we construct two corpora as follows. We consider all microblog posts in the target class as our target corpus, and a second corpus is made of the posts in the other two classes. Then, the LLR score is computed for each term in the target corpus. In practice, we generate three pairs of corpora: positive vs negative and neutral, negative vs positive and neutral, and neutral vs positive and negative. For each target corpus we rank terms by their LLR score and consider the top-N as features.

Latent Dirichlet Allocation (LDA). Topic models can increase precision in text classification tasks [12]. Latent Dirichlet Allocation (LDA) is an unsupervised machine learning technique that identifies topic information inside documents [3, 4]. A topic is defined as a discrete distribution over words from a finite lexicon. In LDA, each document can be represented as a mixture of topics where each topic has a certain probability of generating a particular word. We consider the topic distribution over each microblog post as features.

Table 2 lists examples of the top-10 features we extract for each polarity class using frequency and LLR on unigrams, and χ^2 on bigrams. Entity names are found in the top-10 extracted features for all three methods. Interestingly, there is a correlation of brand names with reputation polarity. Bands and entertainers are usually correlated with positive reputation polarity, car manufacturers with neutral reputation polarity, and financial institutions with negative reputation polarity. In positive lists, we find sentiment-bearing words, e.g., love, want, like, however, their amount decreases for the neutral and the negative classes. Two interesting examples are the words "fine" and "like." The former appears in the negative class for LLR, which seems to be "understood" as noun rather than adverb without linguistic analysis. The latter appears in the neutral class for frequency and seems to be "understood" as proposition or conjunction rather than verb.[5] These findings are encouraging evidence that discriminative corpus-based approaches can be used for complementing or replacing sentiment analysis methods for reputation polarity detection.

4 EXPERIMENTAL SETUP

In addressing the reputation polarity detection problem, we concentrate on developing features and combinations of features that can be used for reputation polarity detection and not on developing or optimizing machine learning techniques. In particular, we want to know the effectiveness of surface features, sentiment features, and their combination with textual features extracted using five term selection methods, i.e., frequency, tf.idf, χ^2, LLR, and LDA for classifying the impact of a microblog post on an entity's reputation as *positive*, *neutral*, and *negative*. To answer these research questions we conduct classification experiments.

Dataset. We use the Replab 2013 dataset [27]. The dataset spans the period June 2012–December 2012 and consists of tweets in English and Spanish for 61 entities drawn from four domains: automotive, banking, university and music. The average tweet length is 10 words/tweet. Each tweet in the corpus is manually annotated in either of three polarity classes: positive, negative, neutral. The dataset provides a training set (45,671 tweets) and a test set (105,099 tweets). We ignore tweets for which the content is not available because it has been previously deleted or user profiles went private. Table 3 summarizes our actual training and testing data.

Preprocessing. We ignore tweets that are annotated as non-relevant to an entity. We treat tweets in English and in Spanish equally mainly because of the missing lexical resources for sentiment analysis in Spanish. We apply five preprocessing steps. First, text is case

[5] "But I feel like I accomplished a little bit by destroying his BMW!", "I want a job were I wear a suit and have a nice company car like Audi.", "Looks like a lot of fun. Could the Volvo be in something like that one day?"

Table 2. Top-10 extracted textual features for the positive (P.), neutral (Nt.), and negative (Ng.) polarity classes using Frequency, χ^2, LLR.

Frequency			χ^2			LLR		
P.	Nt.	Ng.	P.	Nt.	Ng.	P.	Nt.	Ng.
one	Mazda	Bankia	Wisin, Yandel	Northbound, Reuter	Hong, Kong	Aerosmith	Mazda	Bankia
new	Honda	bank	celda, basada	Hilfiger, Savor	Charlie, Sheen	Zeppelin	manual	HSBC
love	car	HSBC	inventando, Nils	Brasstech, Drain	Congressman, Darrell	Led	Volkswagen	Barclays
this	Whitney	Barclay	Nils, Bohlin	carta, Hogwarts	Darrell, Issa	ACDC	Subaru	dinero
Led	Houston	Justin	Richie, Sambora	crucial, indicator	Fendi, purse	Jovi	Yamaha	fine
Zeppelin	like	Bieber	greatesthit, studiohit	Mylo, Xyloto	Laioflautes, ocupen	love	Honda	rato
want	Subaru	America	Greyson, Chance	international, urgencia	ocupen, oficina	Bon	PDF	libor
like	Volkswagen	Goldman	magnification, sorprendente	AndyLau, Iwish	artista, grabando	Maroon	Liga	lavado
Bon	Jennifer	Lady	Meis, Ancol	auxilio, secuestraron	atacado, phishing	PSI	Whitney	money
Jovi	Lopez	Gaga	Cristiano, Ronaldo	Bin, Laden	Bayern, Pharmaceuticals	MIT	download	Senat

Table 3. Summary of the training and test sets for the positive (P.), neutral (Nt.), and negative (Ng.) class in the RepLab 2013 dataset. We report on the total number of examples per class (Total), and the average (Avg.), maximum (Max.), and minimum (Min.) examples per entity in the respective set.

	Training			Testing		
	P.	Nt.	Ng.	P.	Nt.	Ng.
Total	19,221	9,492	5,293	41,933	20,015	10,580
Avg.	315	155	86	687	328	173
Max.	766	522	716	1,483	1,033	1,570
Min.	7	6	2	12	8	11

folded, English and Spanish stop words are removed, and tokens are stemmed using the Porter stemmer. Numbers, punctuation marks except *?, !*, URLs, and user mentions (@user) are substituted with a place holder tag. Hashtags are kept after removing the hash. Repetition of the same character inside a word is reduced to at most three characters, e.g., *cooooooooool* is normalized to *coool*). Emoticons are grouped into positive (*:), :-), :), :D, =), ;), ;), (:, :], =], :-D, :-], ;D, ;-D, ;], ;-], ;-)*) and negative (*:(, :-(, : (,):,) :, D:, =(, :[, :-[, =[, :'(, :'[*). Finally, our set of negation words consists of: *not*, *no*, *never*, *cannot* and words ending in *n't*.

Feature selection. For selecting textual features, we run our textual feature selection methods on unigrams and on bigrams. We set $N = 500$ and select the top-N as features. Each textual feature is represented as boolean based on whether it appears in the example. When we refer to combining unigrams and bigrams, we use both the top-N from unigrams and the top-N from bigrams as features. For LDA, we follow [11] and set the number of topics to 50.

Training. We choose to train an entity-independent classifier, i.e., we want to detect a tweet's reputation polarity regardless of its association with an entity. To this end, we assemble our positive, neutral, and negative classes by combining examples from the respective class from each entity. Before combining, we balance the classes at entity-level. For each entity, we consider the top-N examples for each class ranked by their annotation order in the training set, where N is set to the number of examples found in the entity's least popular class. We perform our experiments using the natural language toolkit (NLTK) [2] and the scikit-learn framework [24].

Evaluation. We set our baseline to [11], the best performing system at RepLab 2013. The features used in that system consist of the sentiment polarity of a tweet, the number of character repetitions, the presence of the entity name inside a message, the distance between a token and the mention of the target entity, and textual unigram and bigram features. We report on the overall accuracy and F1-score for each class of the tested classifiers, i.e., Naive Bayes (NB), Maximum Entropy (ME), Random Forest (RF) and Support Vector Machines (SVM),[6] averaged over all entities. The choice of our classifiers is motivated by their good performance in many classification tasks and in previous research on reputation polarity detection [11, 17, 18, 22, 31].

Experiments are conducted on an Intel Core i5 processor (2.6 GHz) with 8 GB RAM. Training and testing all the classification algorithms takes, on average, 2.5 hours for each method.

5 RESULTS

Figure 1 lists the performance of the approaches we consider. We test the accuracy of our surface and sentiment features individually and in combination with textual features. Surface features (Su) achieve better performance than sentiment features (Se), and combining them (SuSe) does not necessarily provide better results. When we add unigram features to the SuSe approach we achieve scores comparable to, or better than, the baseline (third group). The use of bigram features on top of the SuSe system tends to hurt the accuracy scores, possibly due to data sparsity issues. Using unigram and bigram features on top of SuSe, we obtain improvements in accuracy over using only SuSe or using unigrams or bigrams. Although bigrams are not useful in isolation, they do help when used in combination with unigrams, likely because they manage to better capture the context of a term.

Next, we turn to our five methods for selecting textual features. Focusing on the performance of the SVM classifier, we observe that accuracy follows a stable pattern: LDA $<$ frequency $<$ χ^2 $<$ tf.idf $<$ LLR. The poor performance of LDA is likely due to the sparse language use within a tweet and its limited content, in line with findings on other microblog-related tasks [32]. Simply using the most frequent words inside a corpus as features does not provide enough relevant information to guide classifiers. The χ^2 statistics results in a considerable boost in accuracy, capturing terms with more meaning. The tf.idf weighting scheme is more reliable in identifying distinctive terms for each class. Finally, LLR outperforms all other previous textual feature selection approaches, suggesting that comparing corpora using frequency profiling is a good approach to automatically identify the key items that define each of our classes.

We briefly turn to a comparison of the different classifiers. In most settings, they perform at similar levels, with RF's accuracy being constantly surpassed by our three other choices. The only cases

[6] Parameters: NB–in the default Python NLTK implementation if a feature has never been seen with any label it will be ignored instead of being assigned a probability of 0; ME–Generalized Iterative Scaling methods are used to train the classifier with a maximum number of 10 iterations; RF–the number of trees in the forest is set to 100; SVM–kernel set to linear.

where it is as competitive as a NB or a ME classifier is when using surface features only or when combining them with sentiment features and tf.idf unigrams, or tf.idf unigrams and bigrams. NB and SVM classifiers generally present very competitive accuracy scores as the number of features is increased. NB yields the best overall accuracy when we use all unigram and bigram features extracted by LLR:[7] 0.8546, nearly 20% higher than the state-of-the-art (0.685).

6 ANALYSIS

The best performing approach identified in the previous section (NB, using unigrams and bigrams on top of our surface and sentiment features) uses 6,257 unigrams and 4,981 bigrams identified by the loglikelihood-ratio method.

Fig. 2 illustrates how the accuracy of our system is dependent upon the number of features we select and how it increases proportionally with the number of features we add: steady increases in performance (accuracy) when we increase the number of unigrams and bigrams considered as textual features. Discriminatively selecting textual features that characterize a corpus with the loglikelihood-ratio approach and combining these with sentiment and surface features results in an accuracy score of 0.8546.

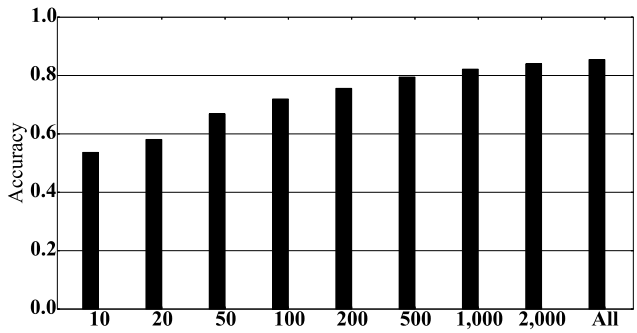

Figure 2. Performance of the best performing approach (NB using textual, surface and sentiment features), when the number of unigrams and bigrams identified by the loglikelihood-ratio is varied successively to 10, 20, 50, 100, 200, 500, 1,000, 2,000 and when all the extracted features are used.

We also test the performance of an NB classifier that combines features extracted from all methods that we consider: frequency, tf.idf, χ^2, LLR, and LDA. When using 100 features per class per method, accuracy drops to 0.6904. Increasing the number of features has a negative effect on accuracy, which reaches a minimum at 0.5045 when we consider all features from each method. These findings provide further evidence that more elaborate feature selection methods, such as χ^2 and LLR, are better in extracting discriminative features that characterize the polarity of social media posts.

Looking back at the distinctive features we use, our unigram and bigram-based approaches correctly guess the polarity of a tweet by exploiting specific language usage associated with each positive, neutral or negative class, such as names of famous artists, companies or institutions. Our understanding of reputation polarity (as expressed in the annotations provided by expert annotators) is not only limited to opinions and sentiment bearing words, but rather embedded inside entities. While previous work on detecting the reputation polarity of tweets mostly tackled the task as a sentiment analysis task, our findings show that sentiment analysis does not suffice. This has important implications for today's approaches to the task.

The performance of our best performing approach shows very good precision for the positive class (0.8855) and good recall for the negative class (0.6807), which may indicate a stronger and more explicit choice of terms for positive tweets and that users may be inclined to use terms from other classes to express remarks of opposite polarity. Below we list some examples of miss-classifications of our system that confirm this hypothesis:

- *Positive class:* "Priceless! Iran's nuclear computers hacked & forced to play AC/DC's Thunderstruck @ full volume in middle of the night!", "I'm surprised they've made it this far without U2 to be honest. Or Bono, at least.", "Not sure which AC/DC song is better - Hells bells, Back in black, Highway to hell or Night prowler."
- *Neutral class:* "New Course 2013 Oxford University introduction to managing crime & antisocial behaviour within the historic environment", "I thought about buying those Jay-Z/Coldplay tickets, but I'm feeling over saturated by Barclays", "Sometimes I feel bad for the members of Maroon 5 that aren't Adam Levine."
- *Negative class:* "Got a letter from BMW trying to sell me their new 3 series. It's very nice but I can't afford the insurance!", "Michelle Obama is a Princeton and Harvard Law Graduate but this is what every article I've seen today looks like", "Of course Harvard and other elite schools want to be associated with creative genius. It burnishes their brand and gives future lawyers."

7 CONCLUSIONS

We have presented a new approach for tackling the task of identifying the reputation polarity of microblog posts by using discriminative textual features inferred from labeled data using corpus-based methods. We show that they turn out to be very effective in addressing the reputation polarity detection task. We conclude that sentiment analysis on its own does not suffice, but rather that reputation polarity is encoded in broader language usage than is typically captured by a sentiment lexicon. In future work we plan to examine entity-specific methods, which mine the relevant language usage conditioned on an individual entity whose reputation is being monitored.

Acknowledgments. This research was supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nrs 288024 and 312827, the Netherlands Organisation for Scientific Research (NWO) under project nrs 727.011.005, 612.001.116, HOR-11-10, 640.006.013, the Center for Creation, Content and Technology (CCCT), the QuaMerdes project funded by the CLARIN-nl program, the TROVe project funded by the CLARIAH program, the Dutch national program COMMIT, the ESF Research Network Program ELIAS, the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences (KNAW), the Netherlands eScience Center under project nr 027.012.105, the Yahoo! Faculty Research and Engagement Program, the Microsoft Research PhD program, and the HPC Fund.

REFERENCES

[1] E. Amigó et al. Overview of RepLab 2013: Evaluating online reputation monitoring systems. In *CLEF '13*, pages 333–352. 2013.
[2] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009.
[3] D. M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.
[4] D. M. Blei, A. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Machine Learning Research*, 3:993–1022, 2003.
[5] J.-V. Cossu et al. LIA at RepLab 2013. In *CLEF*, 2013.

[7] LLR unigram features/class: 4,071 positive, 731 negative, 1,455 neutral
LLR bigram features/class: 3,065 positive, 542 negative, 1,374 neutral

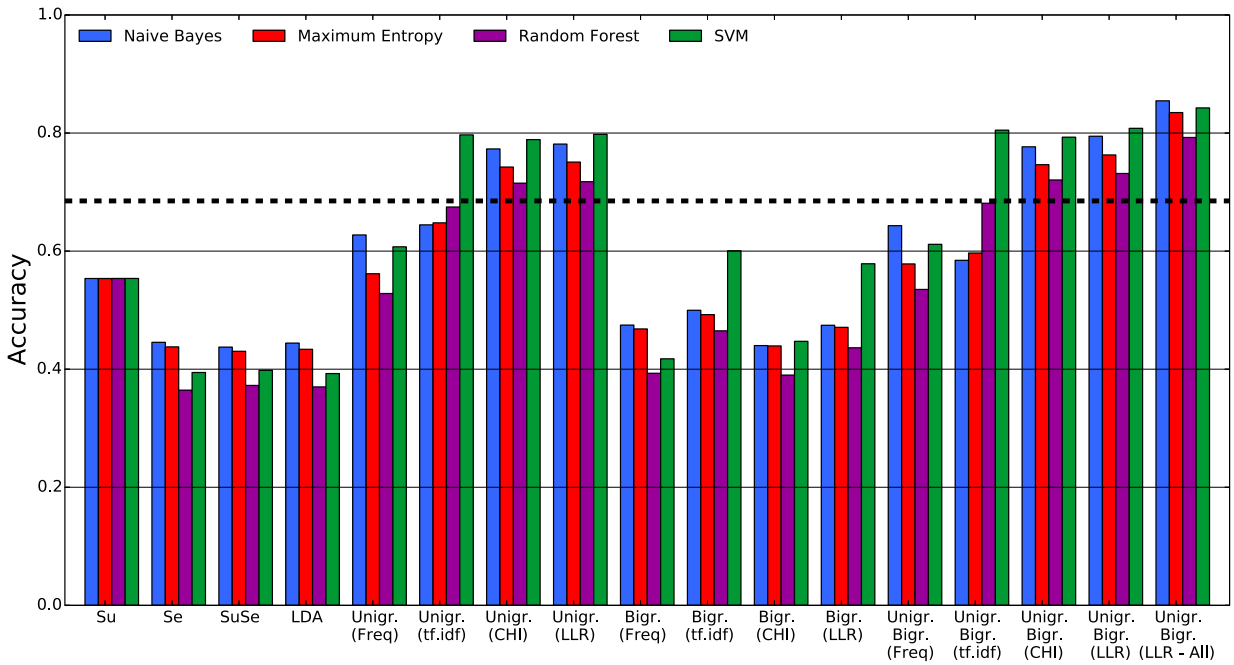

Figure 1. System performance for the reputation polarity detection task using only surface features (Su), only sentiment features (Se), surface and sentiment features without textual features (SuSe) and with top-500 unigram textual features selected using five methods: frequency (freq.), tf.idf, χ^2 (CHI), LLR, and LDA for a Naive Bayes classifier (NB), a Maximum Entropy classifier (ME), a Random Forest classifier (RF) and a Support Vector Machines (SVM) classifier. The baseline is set to [11], the best performing system in RepLab 2013 (black dashed line). We report on average accuracy over positive (P.), neutral (Nt.) and negative (Ng.) class and over all entities. NB using all unigram and bigram features from LLR performs the best.

[6] T. Dong, W. Shang, and H. Zhu. An improved algorithm of Bayesian text categorization. *J. Software*, 6(9), 2011.

[7] A. Esuli and F. Sebastiani. SentiWordNet: A publicly available lexical resource for opinion mining. In *LREC '06*, 2006.

[8] M. Feckzo, M. Schaye, A. Marcus, and A. Nenkova. SentiSummary: Sentiment summarization for user product reviews. Technical report, Univ. of Pennsylvania, 2010.

[9] J. Filgueiras and S. Amir. POPSTAR at RepLab 2013: Polarity for reputation classification. In *CLEF '13*, 2013.

[10] M. Gamon. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *COLING '04*, 2004.

[11] V. Hangya and R. Farkas. Filtering and polarity detection for reputation management on tweets. In *CLEF '13*, 2013.

[12] L. Hong and B. D. Davidson. Empirical study of topic modelling in Twitter. In *1st Worksh. Social Media Analytics*, 2010.

[13] A. Kilgarriff. Comparing word frequencies across corpora: Why chi-square doesn't work, and an improved LOB-Brown comparison. In *ALLC-ACH Conference*, 1996.

[14] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[15] J. M. Meena, K. R. Chandran, J. Brinda, and P. Sindhu. Enhancing feature selection using statistical data with unigrams and bigrams. *Int. J. Comp. Appl.*, 1(11), 2010.

[16] A. Mosquera, J. Fernández, J. M. Gómez, P. Martínez-Barco, and P. Moreda. Dlsi-Volvam at Replab 2013: Polarity classification on Twitter data. In *CLEF '13*, 2013.

[17] V. Narayanan, I. Arora, and A. Bhatia. Fast and accurate sentiment classification using an enhanced naive Bayes model. In *IDEAL '13*, volume 8206 of *LNCS*, 2013.

[18] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI '99 Workshop on Machine Learning for Information Filtering*, 1999.

[19] A. Oghina, M. Breuss, M. Tsagkias, and M. de Rijke. Predicting IMDB movie ratings using social media. In *ECIR '12*, 2012.

[20] B. Ohana and B. Tierney. Sentiment classification of reviews using SentiWordNet. In *Proceedings of IT&T*, 2009.

[21] G. Paltoglou and M. Thelwall. A study of information retrieval weighting schemes for sentiment analysis. In *ACL '10*, pages 1386–1395. ACL, 2010.

[22] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *EMNLP '02*, pages 79–86, 2002.

[23] T. Pedersen. A decision tree of bigrams is an accurate predictor of word sense. In *NAACL '01*, pages 1–8, 2001.

[24] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *J. Machine Learning Research*, 12:2825–2830, 2011.

[25] M. H. Peetz, M. de Rijke, and A. Schuth. From sentiment to reputation. In *CLEF '12*, 2012.

[26] P. Rayson and R. Garside. Comparing corpora using frequency profiling. In *Workshop on Comparing corpora*, pages 1–6, 2000.

[27] RepLab. An evaluation campaign for online reputation management systems, Sept. 2013. URL http://www.limosine-project.eu/events/replab2013.

[28] J. Saias. In search of reputation assessment: experiences with polarity classification in Replab 2013. In *CLEF '13*, 2013.

[29] D. Spina, J. Carrillo-de Albornoz, T. Martin, E. Amigo, J. Gonzalo, and F. Giner. UNED online reputation monitoring team at RepLab 2013. In *CLEF '13*, 2013.

[30] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, 2nd edition, 2005.

[31] B. Xu, X. Guo, Y. Ye, and J. Cheng. An improved random forest classifer for text categorization. *J. Computers*, 7: 2913–2920, 2012.

[32] W. X. Zhao et al. Comparing Twitter and traditional media using topic models. In *ECIR '11*, pages 338–349, 2011.