FBM-Yahoo! at RepLab 2012

Jose M. Chenlo[1], Jordi Atserias[2], Carlos Rodriguez[2] and Roi Blanco[3]

[1] josemanuel.gonzalez@usc.es
Centro de Investigación en Tecnoloxías da Información (CITIUS)
Univ. de Santiago de Compostela
Spain
[2] {jordi.atserias,carlos.rodriguez}@barcelonamedia.org
Fundació Barcelona Media
Av. Diagonal 177
Barcelona, Spain
[3] roi@yahoo-inc.com
Yahoo! Research
Av. Diagonal 177
Barcelona, Spain

Abstract. This paper describes FBM-Yahoo!'s participation in the profiling task of RepLab 2012, which aims at determining whether a given tweet is related to a specific company and, in if this being the case, whether it contains a positive or negative statement related to the company's reputation or not. We addressed both problems (ambiguity and polarity reputation) using Support Vector Machines (SVM) classifiers and lexicon-based techniques, building automatically company profiles and bootstrapping background data. Concretely, for the ambiguity task we employed a linear SVM classifier with a token-based representation of relevant and irrelevant information extracted from the tweets and Freebase resources. With respect to polarity classification, we combined SVM lexicon-based approaches with bootstrapping in order to determine the final polarity label of a tweet.

1 Introduction

RepLab [1] addresses the problem of Reputation analysis, i.e. mining and understanding opinions about companies and individuals, a harder and still not well understood problem. FBM-yahoo! participates in the RepLab Profiling task [1] where systems are asked to annotate two kinds of information on tweets:

- **Ambiguity**: To determine if a tweet is related to the company using.
- **Polarity for Reputation**: To determine if the tweet have positive or negative implications for the company's reputation;

2 Ambiguity task

2.1 Company Representation

Twitter messages are short (up to 140 characters), hence, measures that account for the textual overlap between tweets and company names are in general not

enough to classify a given tweet as relevant or irrelevant [2], mostly due to data sparsity and lack of context [3]. In order to alleviate this problem, we turned into using the Freebase[4] graph and Wikipedia[5] as reliable sources of information for building expanded term-based representations of the different companies.

From the Freebase/Wikipedia pages of the companies we extracted automatically two sets of entities, namely *related concepts* and *non-related concepts*:

- *Related Concepts (RC)*: represents the set of entities that are connected with the company in Freebase through the incoming (outgoing) links connected to the company's Freebase page. For example, in the case of *Apple Inc.*, the related concepts set includes *iPhoto, ichat, ibook, iTunes Store*.
- *Non-Related Concepts (NRC)*: represents the set of common entities with which the current company could cause spurious term matches. This set is comprised of all Freebase entities with a name similar to that of the company's. This set is built automatically by querying Freebase with the query that identifies the company in the training data. From this set we remove the target company (if it was found), and all the entities that are already included in *RC*, and all entities that shared at least one non-common category with the target company. As an example of this process, in the case of *Apple Inc.* some of the non-related entities selected were *"big apple"* or *"pine apple"*.

Then for each entity obtained following the previous method we have crawled its Wikipedia page[6] and then we have used *Lucene*[7] software to compute the following lists of keywords for each set of entities (RC, NRC):

- *entity names*: Name of the entities related (non-related) to the company.
- *named entities in text*: All named entities extracted by the Stanford Named-Entity Recognizer [4].
- *ngrams*: Unigrams and bigrams (applying stemming and removing stopwords).

A weight w is associated to all of the obtained keywords (list of entities, named entities in text, ngrams). In the case of the entities, the weight is always 1. For named entities in text and ngrams, the weight is the ratio of documents that contain the concrete keyword.

These lists of keywords represent our profile for a given company as a bag of words model. We note that tweets could be written in English and Spanish and accordingly we have computed two different profiles for each company: one with the English version of Wikipedia an the other one with the Spanish version.

[4] `http://www.freebase.com`
[5] `http://www.wikipedia.org`
[6] In the data-set tweets are written in either English or Spanish. For this reason we have downloaded and stored both versions when possible.
[7] `http://lucene.apache.org`

2.2 Training Process

In recent years, Machine Learning techniques have been deeply applied over Twitter data with relative success in many classification problems [5] [6,7]. Concretely, the best system in WePS-3 Evaluation Campaign [8], where the main task consisted in identifying if a tweet that contains a company name is related or not to the company, employed a linear SVM classifier. Following this approach, we have trained a SVM linear classifier using the LibLinear package [9]. Table 1 lists the features that are being used to represent the data, which are broken down into matches from terms in the tweet in the company's profile (profile), features related to the company name in the tweet (company) and company-independent (tweet-only) features.

Scope	description
Tweet	Size of tweet. Number of links. Number of hashtags. If the tweet could be spam (a simple word appears more than three times).
Company	Whether or not any hashtag contains the name of the company. If exists an URL that contains the name of the company. If the tweet mention the name of the company (first letter in uppercase).
Profile	Score in the related entities list (sum of weights over terms matched). Score in the related ngrams list (sum of weights over terms matched). Score in the related topics list (sum of weights over terms matched). Score in the non-related entities list (sum of weights over terms matched). Score in the non-related ngrams list (sum of weights over terms matched). Score in the non-related topics list (sum of weights over terms matched).

Table 1. List of disambiguation features.

Note that the last six features compare a given tweet with the profile computed for the company. The first six features are tweet-dependent, and they only need the text of the tweet and the query that represents the company. Using this representation we were able to learn a classifier over the trial set (six companies) that can be directly applied to the test data.

3 Polarity for Reputation task

The following sections explain three different approaches (lexicon-based and distant supervision using hashtags and lexicons) we explored in order to determine whether a tweet has positive or negative implications for the company's reputation.

3.1 Lexicon-Based Approaches

The most straightforward approaches employ an ensemble of several lexicons created with different methodologies in order to broaden coverage, especially across domains since some sentiment cues are used differently depending on the subject being commented.

In order to aggregate the lexicon scores into a final polarity measure, several formulas can be used, for instance:

$$polScore(t, lan, q_t) = \sum_{l_i \in lan} polLex(t, l_i, q_t) \tag{1}$$

where t is a tweet, lan is the language of the tweet, q_t is a query, l_i is one of the lexicons associated to lan and $polLex(t, l_i)$ is a matching function between the lexicon l_i and the tweet t. We have developed two different matching functions, $polLex_{raw}$ and $polLex_{smooth}$. $polLex_{raw}$ is a simple aggregation measure that takes into account just the matchings between tweets and lexicons to compute the final polarity:

$$polLex_{raw}(t, l, q_t) = \sum_{w_l \in l} tf_{w_l,t} \cdot priorPol(w_l) \tag{2}$$

where t represents a simple tweet, l is one of the lexicons associated to the language of the tweet, w_l is an opinionated word from lexicon l, $tf_{w_l,t}$ is the frequency of w_l in tweet t and $priorPol(w_l)$ is the polarity score of word w_l in lexicon l. [8]

On the other hand, $polLexSmooth$ is an aggregation measure that takes into account the matchings between tweets and lexicons and the distance of these matchings to the company name to smooth the score of polarity of each word:

$$polLex_{smooth}(t, l, q_t) = \frac{1}{|q_t|} \sum_{q_i \in q_t} \sum_{w_l \in l \cap t} \frac{1}{d_{w_l,q_i}} \cdot priorPol(w_l) \tag{3}$$

where d_{w_l,q_t} is the distance of the tweet term w_l to query term q_i.

Finally, we decide the final classification of each tweet using the following simple thresholding:

$$pol(t) = \begin{cases} positive & if\ polScore(t, l, q_t) > 0 \\ neutral & if\ polScore(t, l, q_t) = 0 \\ negative & if\ polScore(t, l, q_t) < 0 \end{cases} \tag{4}$$

[8] This score could be positive or negative depending on the orientation of w_l.

Note that it is possible to compute two different values for $polScore(t, l, q_t)$ by applying either Equation 2 or Equation 3 to the formula in Equation 1. Full details about which methods have been used in the runs submitted can be found in Section 4.

3.2 Distant Supervision

Traditional opinion mining methods proposed in the literature are often based on machine learning techniques, using as primary features a vocabulary of unigrams and bigrams collected from training data [10].

Following this approach and we have used a linear SVM to classify tweets as positive, neutral or negative. Table 2 lists the features employed to represent the data, which are broken down into tweet-based features, part of speech-based features and lexicon-based features.

Scope	Description
Voc.	vocabulary features: Unigrams and bigrams from training examples.
Tweet	Size of tweet. Number of links. Number of hashtags. If the tweet could be spam (a single word appears more than three times). Number of exclamations and interrogations. Number of uppercase letters. Number of lengthening phenomena.
POS	Number of verbs. Number of adjectives. Number of proper names. Number of pronouns.
Pol.	Number of positive emoticons. Number of negative emoticons. Lexicon polarity score using as matching function 2. Lexicon polarity score using as matching function 3.

Table 2. List of polarity features.

The lexicon-based approaches previously described do not require training and can be directly applied over test data. However, the proposed data representation requires some amount of training data to compute the vocabulary features for each tweet which was not available at training time. Moreover, due to the fact that the companies in the test set belong to different domains (e.g. banks vs technology), the terms (and even their senses) used for express opinions may change from one company to another.

For that reason, we learnt different a model for each company in which we automatically generated a set of labelled examples from their background model. Other recent work on this area has focused on distantly supervised methods which learn the polarity classifiers from data with noisy labels such as emoticons and hashtags [6] [11].

3.3 Distant Supervision using Hashtags

Similarly to [11], for each polarity class (i.e. positive, negative and neutral) we have performed the following process to automatically generate positive, neutral and negative labelled examples:

1. Selecting all hashtags that were used in more than 5 tweets in the background model of the company.
2. Removing the noisy content (spam, repeated tweets, retweets, etc.) for each hashtag.
3. Using the equation 1 in conjunction with equation 2 as matching function to select the top 5 positive/negative/neutral hashtags, according to the ratio of tweets of each hashtag that were classified as positive/negative/neutral.
4. Selecting the top 20 tweets of each polarity from top hashtags.

This bootstrapping process enables to obtain up to 100 positive, negative and neutral labelled examples (i.e. up to 300 examples in total) to train different classifiers.

Once we have generated our labelled examples, we have trained a positive classifier (positive examples against negative plus neutral examples), and a negative classifier (negative examples against positive plus neutral examples) for each company in the test set. We have also trained the best thresholds that separated the positive and the negative examples for each classifier. Finally, we combined the two classifiers and the thresholds learned to decide if a given tweet had to be tagged as positive, neutral or negative.

Learning the Best Threshold In the previously described approach, we selected the class decision threshold for a classifier using data which could potentially contain noisy labels and consequently could harm the performance of our system. To alleviate this problem, we randomly assessed 50 examples from the background data of each company and we selected the positive/negative thresholds for each classifier according to the the class distribution found in the data. Full details about which runs submitted were built with this kind of training can be found in Section 4.

3.4 Distant Supervision using lexicons

This distant supervision method is similar to the one explained in Section 3.3, with the difference that it makes use of the polarity lexicons instead of the tweet hashtags.

The following process is undertaken for each polarity class (i.e. positive, negative and neutral), in order to automatically generate positive, neutral and negative labelled examples for each company:

1. Select as positive examples tweets that only have positive matches sorted by the number of matches in the lexicon.

2. Select as neutral examples tweets that no matches ordered by the tweet length.
3. Select as negative examples tweets that only have negative matches sorted by the number of matches in the lexicon.

Similarly to the distant supervision method using hashtags doing this bootstrapping process we select up to 100 positive, negative and neutral labelled examples (i.e. up to 300 examples in total) in order to train different classifiers for each company. These examples are selected in order of their number of matches.

The final classifier is built using the thresholded ensemble described in Section 3.3.

4 Submitted Runs

FBM-Yahoo! participated in the profiling task of RepLab 2012 competition with 5 different runs[9]. The particular details on how the FBM-Yahoo! 5 runs runs were made can be found in Table 3. All runs use the method explained in section 2.1 to classify a tweet as *relevant* or *irrelevant*, but they differ on the polarity method used to compute the final label of a tweet (i.e. *positive*,*negative* or *neutral*).

Regarding the polarity lexicon based method described in section 3.1 we employed a total of six different polarity lexicons for English (including OpinionFinder[13], AFINN [14], Qwordnet[15], dictionaries from the Linguistic Inquiry and Word Count (LIWC) text analysis system [16][10] and five polarity lexicons for Spanish. Following [17] we also combine these lexicons with a lexicon based on emoticons.

Since the resources available for Spanish are scarce, we translated some of the resources available for English, for instance, some baseline lexicons like the one used by OpinionFinder (the MPQA Subjectivity Lexicon), or AFINN [14]. In order to resolve ambiguities in this bilingual dictionary and to adapt it to micro-blogging usage, we selected the translation alternative that occurred most frequently on an alternative large (100,000) Spanish Twitter corpus (different from the one provided by RepLab).

As an additional approach we used author-assessed datasets to create polar lexicons from customer reviews, in this case, from 100,000 good vs. bad comments sent to Hotels.com and other such sites, like movie comments from volunteer reviewers and professionals. A Naive Bayes classifier was trained, from which a list of class-discriminative unigrams and bigram was extracted. Only adjectives and adverbs from those list were filtered to create a data-driven polar lexicon, similar to the method of Banea and Mihalcea [18] that employs an automatically translated corpus. Finally, starting from a small, manually crafted dictionary, we expanded its polar entries via WordNet synsets.

[9] Another run (UNED_5) was submitted in collaboration with UNED which combines all FBM-Yahoo! and UNED runs. The details on the combination are described at see section 3 of [12]

[10] Mapping positive and negative sentiments to numeric polarities, expanding the lexicon to possible morphological variants.

Table 3. List of submittedd runs to Profiling Task.

Run Id	Description
BMedia1	Distant Supervision using Hashtags (see section 3.2)
BMedia2	Lexicon-Based using $polLex_{raw}(t, l)$
BMedia3	Lexicon-Based using $polLex_{smooth}(t, l, q_t)$
BMedia4	Distant Supervision using Hashtags with threshold from dist. (see section 3.3)
BMedia5	Distant Supervision using Lexicons (see section 3.4)

5 Results and Conclusions

Table 4 shows the official evaluation results for Ambiguity and Polarity for Reputation tasks for the 5 runs submitted by FBM-Yahoo!.

Ambiguity task. On one hand, results show that our ambiguity method has a poor reliability (R) and sensibility (S) performance. On the other hand, the accuracy of the classifier is very high.

R and S are macro-measures that are equivalent to the product of precisions (reliability) and the product of recalls (sensitivity) over positive and negative classes. To put things in perspective, Table 5 reports the precision and recall values for each class. These results show that our model is classifying most of examples as positive (i.e related to the company), due to the fact that there is a lack of negative examples in training companies (more than 95% of examples are positive).

Table 4. Submitted runs for Profiling task at RepLab 2012.

	Ambiguity[11]				Polarity				Profiling
	Acc.	R	S	F(R,S)	Acc.	R	S	F(R,S)	Acc.
BMedia1	.736	.166	.123	.103	**.429**	.283	.270	.269	.333
BMedia2	.736	.166	.123	.103	.409	**.332**	**.365**	**.335**	.335
BMedia3	.736	.166	.123	.103	.375	.288	.347	.308	.326
BMedia4	.736	.166	.123	.103	.390	.265	.258	.252	**.358**
BMedia5	.736	.166	.123	.103	.409	.287	.321	.290	.335

Table 5. Ambiguity performance per class

	Precision	Recall	F1
positive	0.38	0.66	0.47
negative	0.13	0.04	0.06

Polarity for reputation task. According to the official measures (R and S), the runs that take into account just the overlapping between tweets and lexicons (i.e. *BMedia2* and *BMedia3*) performed the best for polarity classification. Nonetheles, bootstrapping approaches were very competitive in terms of accuracy. In fact, the performance they achieved is very close to that of the lexicon-based approaches, and therefore the first conclusion we can extract from this evaluation is that distant supervised approaches take a limited advantage of training data in this benchmark. This could be due to the fact that lexicons contribute for most of the model signal and might make difficult to learn anything from other sources of features. Moreover, the noise introduced by misclassification data in the training process could harm the performance of the learning process more than improve it.

Profiling task. In this task, all methods behave similarly in terms of performance, being *BMedia4* the best run. This method combines the hashtag bootstrapping approach with the selection of a threshold for each classifier learnt from hand-classified tweets from background models. It is worth to remark that we have selected the best threshold for a classifier using data which contains noisy labels and consequently could harm the overall performance of the system. In order to overcome this problem, we set a different threshold for each classifier using background data. Results indicate that setting this threshold alleviates the score noise coming from lexicon bootstrapped examples.

Finally, as future work, we would like to explore how sentiment in Twitter streams are affected by real-world events, which affect severely Twitter topic trends. For example, if a football team loses a match, probably the next day the overall opinion about this team will be to negative. We would also like to study how to detect the polarity changes across the time and how to adapt our classification models to this new scenarios. More concretely, we would like to apply propensity scoring techniques [19,20] to deal with the fact that training instances are governed by a distribution that differs greatly from the test distribution.

Acknowledgements

This work is partially funded by the Holopedia Project (TIN2010-21128-C02-02), Ministerio de Ciencia e Innovación.

References

1. Amigó, E., Corujo, A., Gonzalo, J., Meij, E., Rijke, M.d.: Overview of replab 2012: Evaluating online reputation management systems. In: CLEF 2012 Labs and Workshop Notebook Papers. (2012)
2. Surender Reddy Yerva, Zoltán Miklós, K.A.: It was easy, when apples and blackberries were only fruits. In: CLEF (Notebook Papers). (2010)
3. Blanco, R., Zaragoza, H.: Finding support sentences for entities. In Crestani, F., Marchand-Maillet, S., Chen, H.H., Efthimiadis, E.N., Savoy, J., eds.: SIGIR, ACM (2010) 339–346

4. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: In Proceedings of HLT-NAACL 2003. (2003) 252–259
5. Bermingham, A., Smeaton, A.F.: Classifying sentiment in microblogs: is brevity an advantage? In: CIKM. (2010) 1833–1836
6. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. Processing (2009) 1–6
7. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of twitter data. In: Proceedings of the Workshop on Language in Social Media (LSM 2011), Portland, Oregon, Association for Computational Linguistics (June 2011) 30–38
8. Artiles, J., Borthwick, A., Gonzalo, J., Sekine, S., Amigó, E.: Weps-3 evaluation campaign: Overview of the web people search clustering and attribute extraction tasks. In: CLEF (Notebook Papers/LABs/Workshops). (2010)
9. Fan, R.E., Chang, K.W., Wang, C.J.H.X.R., Lin., C.J.: LIBLINEAR: A library for large linear classification. In: Journal of Machine Learning Research 9(2008). (2008) 1871–1874
10. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of EMNLP. (2002) 79–86
11. Kouloumpis, E., Wilson, T., Moore, J.: Twitter sentiment analysis: The good the bad and the omg! In: ICWSM. (2011)
12. Jorge Carrillo de Albornoz, I.C.y.E.A.: Using an emotion-based model and sentiment analysis techniques to classify polarity for reputation. In: CLEF 2012 Labs and Workshop Notebook Papers. (2012)
13. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: HLT/EMNLP. (2005)
14. Nielsen, F.Å.: A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. CoRR (2011)
15. Agerri, R., García-Serrano, A.: Q-wordnet: Extracting polarity from wordnet senses. In Chair), N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D., eds.: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, European Language Resources Association (ELRA) (may 2010)
16. Pennebaker, J., Francis, M.E., Booth, R.J.: Linguistic inquiry and word count: Liwc 2001. Mahway: Lawrence Erlbaum Associates (2001)
17. Kun-Lin Liu, Wu-Jun Li, M.G.: Emoticon smoothed language models for twitter sentiment analysis. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI). (2012)
18. Mihalcea, R., Banea, C.: Learning multilingual subjective language via cross-lingual projections. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics. (2007)
19. Bickel, S., Brückner, M., Scheffer, T.: Discriminative Learning Under Covariate Shift. Journal of Machine Learning Research **10** (September 2009) 2137–2155
20. Agarwal, D., Li, L., Smola, A.J.: Linear-time estimators for propensity scores. Journal of Machine Learning Research - Proceedings Track **15** (2011) 93–100