FBM: Combining lexicon-based ML and heuristics for Social Media Polarities

Carlos Rodríguez-Penagos, Jordi Atserias, Joan Codina-Filbà,
David García-Narbona, Jens Grivolla, Patrik Lambert, Roser Saurí
Barcelona Media
Av. Diagonal 177, Barcelona 08018
Corresponding author: carlos.rodriguez@barcelonamedia.org

Abstract

This paper describes the system implemented by Fundació Barcelona Media (FBM) for classifying the polarity of opinion expressions in tweets and SMSs, and which is supported by a UIMA pipeline for rich linguistic and sentiment annotations. FBM participated in the SEMEVAL 2013 Task 2 on polarity classification. It ranked 5th in Task A (constrained track) using an ensemble system combining ML algorithms with dictionary-based heuristics, and 7th (Task B, constrained) using an SVM classifier with features derived from the linguistic annotations and some heuristics.

1 Introduction

We introduce the FBM system for classifying the polarity of short user-generated text (tweets and SMSs), which participated in the two subtasks of SEMEVAL 2013 Task 2 on *Sentiment Analysis in Twitter*. These are: *Task A*. Contextual Polarity Disambiguation, and *Task B*. Message Polarity Classification. The former aimed at classifying the polarity of already identified opinion expressions (or cues), whereas the latter consisted in classifying the polarity of the whole text (Wilson et al., 2013).

The literature agrees on two main approaches for classifying opinion expressions: using supervised learning methods and applying dictionary/rule-based knowledge (see (Liu, 2012) for an overview). Each of them on its own has been used in workable systems, and a principled combination of both of them can yield good results on noisy data, since generally one (dictionaries/rules) offers good precision while the other (ML) is able to discover unseen examples and thus enhances recall.

FBM combined both approaches in order to benefit from their respective strengths and compensating as much as possible their weaknesses. For Task A we used linguistic (lexical and syntactic) annotations to implement both types of approaches. On the one hand, we built machine learning classifiers based on Support Vector Machines (SVMs) and Conditional Random Fields (CRFs). On the other, we implemented a basic classification system mainly based on polarity dictionaries and negation information, as well as simple decision tree-like heuristics extracted from the training data. For task B we trained an SVM classifier using some of the annotations from Task A.

The paper first presents the process of data compilation and preprocessing (section 2), and then describes the systems for Tasks A (section 3) and B (section 4). Results and conclusions are discussed in the last section.

2 Data Compilation and Processing

2.1 Making data available

The **corpus of SMSs** was provided to the participants by the organizers of the task. As for the **corpus of tweets**, legal restrictions on twitter data distribution required the participants to download the textual contents of the corpus from a list of tweet ids. We retrieved the tweet text using the official twitter API instead of script provided by the organizers, but not all the tweets were available for download

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 483–489, Atlanta, Georgia, June 14-15, 2013.

due to restrictions of different types (e.g. geographical), or because the twitter account was temporarily suspended. In total, we managed to retrieve 10,764 tweets out of 11,777 ids provided by the organizers (91.4%). It is worth pointing out that the restrictions on tweets distribution can become an issue for future users of the dataset, as the amount of available tweets will diminish over time. By contrast, the twitter test corpus was distributed with the full text to avoid those problems.

2.2 Leveraging the data with rich linguistic information

We applied the same linguistic processing to both corpora (SMSs and tweets), even though the SMS test data presents very different characteristics from the twitter data, not only because of what can be appreciated as genre differences, but also due to the fact that is apparently written in Singaporean English, which differs significantly from American or British English. No efforts were made to adapt our linguistic processing modules and dictionaries to this data.

Tweets and SMSs were processed with a UIMA[1]-based pipeline consisting of a set of linguistic and opinion-oriented modules, which includes:

Basic linguistic processing: Sentence segmentation, tokenization, POS-tagging, lemmatization.

Syntax: Dependency parsing.

Lexicon-based annotations:

- *Basic polarity*, distinguishing among: *positive*, *negative*, and *neutral*, as encoded in Wilson et al. (2010).
- *Polarity strength*, using the score for positive and negative polarity in SentiWordnet 3.0 (Baccianella et al., 2010). Each SentiWordNet synset has an associated triplet of numerical scores (`positive, negative`, and `objective`) expressing the intensity of positive, negative and objective polarity of the terms it contains. They range from 0.0 to 1.0, and their sum is 1.0 for each synset (Esuli and Sebastiani, 2007). We selected only the synset with positive or negative scores higher than 0.5, containing a total of 16,791 words.
- *Subjectiviy* clues, from Wilson et al. (2010), which are classified as *weak* or *strong* depending on their degree of subjectivity.
- *Sentiment* expressions, from the Linguistic Inquiry and Word Count (LIWC) 2001 Dictionary (Pennebaker et al., 2001).
- In-house compiled lexicons of *negation markers* (such as 'no', 'never', 'none') and *quantifiers* ('all', 'many', etc.), the latter further classified into *low*, *medium* and *high* according to their quantification degree.

The different classifiers employed by FBM constructed their vectors from this output to learn global and contextual polarities.

3 Task A: Ensemble System

Our system combined Machine Learning and rule-based approaches. The aim was to combine the strengths of each individual component while avoiding as much as possible their weaknesses. In what follows we describe each system component as well as the way the ensemble system worked out the collective decisions.

3.1 Conditional Random Fields

One of the classifiers uses the Conditional Random Fields implementation of a biomedical Named Entity Recognition system (JNET from JulieLab) [2], exploiting the classification capabilities of the system (rather than its span detection) by strongly associating already defined "marked instances" with a polarity, and exploring a 5-word window. It uses dependency labels, POS tags, polar words, sentiwordnet and LWIC sentiment annotations, as well as indications for quantifiers and negation markers.

3.2 Support Vector Machines

This classifier was implemented using an SVM algorithm with a linear kernel and the C parameter set to 0.2 (determined using a 5 fold cross-validation). The features set includes those that we used in RepLab

[1] http://uima.apache.org/uima-specification.html

[2] http://www.julielab.de

2012 (Chenlo et al., 2012) (including number of: characters, words, links, hashtags, positive and negative emoticons, question-exclamation marks, adjectives, nouns, verbs, adverbs, uppercased words, words with duplicated vowels), plus a set of new features at tweet level obtained from the linguistic annotations: number of high/medium/low polarity quantifiers, number of positive and negative polar words, sentiwordnet applied to both the cue and the whole tweet.

Moreover, the RepLab polarity calculation based on different dictionaries was modified to take into account negation (in a 3-word window) potentially inverting the polarity (negPol). This polarity measure was applied to the cue and to the whole tweet, thus generating two additional features.

3.3 Heuristic Approach

In task A, in parallel to the supervised learning system, we developed a method (named Heur) based on polarity dictionary lookup and simple heuristics (see Figure 1) taking into account opinion words as well as negation markers and quantifiers. These heuristics were implemented so as to maximize the number of correct positive and negative labels in the training data. To this end, we calculated the aggregate polarity of a cue segment as the sum of word polarities found in the polarity lexicon. The aggregate values in the training set ranged from -3 to +3, taking respectively 1, 0 and -1 as the polarity of positive, neutral and negative words. The label distribution of cue segments with an aggregate polarity value of -1 is shown in Table 1.

Aggregate polarity	-1	
Negation	no	yes
negative	1,032	30
neutral	37	4
positive	178	71

Table 1: Cue segment polarity statistics in training data for an aggregate polarity value of -1.

In this case, if no negation is present in the cue segment, a majority (1,032) of examples had the negative label. In case there was at least a negation, a majority (71) of examples had a positive label. This behaviour was observed with all negative aggregate

```
1:  if has_polar_word(CUE) then
2:      polarity= lex(P)-0.5*lex(QP)
3:                -lex(N)+0.5*lex(QN)
4:      if polarity>0 then
5:          if has_negation(CUE) then negative
6:          else positive
7:          end if
8:      else if polarity<0 then
9:          if has_negation(CUE) then positive
10:         else negative
11:         end if
12:     else
13:         if has_negation(CUE) then positive
14:         else negative
15:         end if
16:     end if
17: else if has_negation(CUE) then negative
18: else
19:     polarity= tlex(P)-0.5*tlex(QP)
20:               -tlex(N)+0.5*tlex(QN)
21:     if polarity<0 then negative
22:     else if tlex(NEU)>0 then neutral
23:     else if polarity>0 then positive
24:     else if has_negemo(CUE) then negative
25:     else if has_posemo(CUE) then positive
26:     else unknwn
27:     end if
28: end if
```

Figure 1: Heuristics used by the lexicon-based system to classify the polarity of a segment marked up as opinion cue (Task A).

polarity values in training data, yielding the rule in lines 8 to 11 of Figure 1. Similar rules were extracted for the other aggregate polarity values (lines 4 to 16 of Figure 1).

Figure 1 details the complete classification algorithm. Note (lines 1 to 17) that we first rely on the basic polarity lexicon annotations (described in section 2). The final aggregate polarity formula (lines 2-3) was refined to distinguish sentiment words which act as quantifiers, such as *pretty* in *pretty mad*. The word *pretty* is both a positive polar word and a quantifier. We want its polarity to be positive in case it occurs in isolation, but less than one so that the sum with a following negative polar word (such as *mad*) be negative. We thus give this kind of words a polarity of 0.5 by substracting 0.5 for each polar word which is also a quantifier. In the polarity formula of lines 2-3, `lex(X)` refers to the number of words annotated as X, P and N refer respectively to positive and negative polar words, and QP and

QN refer to positive and negative polar words which are also quantifiers. Quantifiers which are not polar words are not taken into account because they are not likely to change the opinion polarity.

In case that no annotations from the basic polarity, quantifiers, and negative markers lexicons are found (lines 18 to 28), we look up in dictionaries built from the training data (`tlex` in lines 19-20). To build these dictionaries, we counted how many times each word was labeled positive, negative and neutral. We considered that a word has a given polarity if the number of times it was assigned to this class is greater than the number of times it was assigned to any other class by a given threshold. We calculated the polarity in the same way as before, but now with the counts from the lexicon automatically compiled from the training data. To improve the recall of the dictionary lookup, we performed some text normalization: lowercasing, deletion of repeated characters (such as *gooood*) and deletion of the hashtag "#" character. Finally, if no polar word is found in the automatically compiled lexicon, we look at the sentiment annotations (extracted from the LIWC dictionary).

3.4 Ensemble Voting Algorithm

As already mentioned, we combined the results from the described polarity methods to build a collective decision. Table 2 shows the performance (in terms of F1 measure) of the different single methods over the tweet test data.

	SVM	**Heur**	**Heur+**	**CRF**
Test	80.74	83.47	84.62	62.85

Table 2: Twitter Task A results for different methods

Although the heuristic method outperforms the ML methods, they are not only different in nature (ML vs. heuristic) but also use different information (see Table 5). This suggests that the ensemble solution will be complementary and capable of obtaining better results than any of the individual methods by itself.

The development set was used to calculate the ensemble response given the individual votes of the different systems in a way similar to the behavior knowledge space method (Huang and Suen, 1993). Table 3 shows an example of how the assemble voting is built. For each method vote combination (SVM-Heuristics-CRF) the number of positives / negatives / neutral is calculated in the development data. The ensemble (EV) selects the vote that maximizes the number of correct votes in the development data (in bold).

SVM	**Heur**	**CRF**	**EV**	**# Instances**		
				pos	**neg**	**neu**
−	+	−	−	0	**6**	0
−	−	+	−	1	**23**	2
−	−	−	−	3	**125**	2
−	u	+	+	**1**	0	0
+	u	n	−	0	**1**	0
+	−	+	+	**17**	13	2
+	+	+	+	**314**	18	17
+	−	n	+	**3**	1	0

Table 3: Oracle building example (EV: Ensemble Vote, +:positive, −:negative, n:neutral, u:unknown)

The test data contains some combination of votes that were not seen in the development data. Thus, in order to deal with these unseen combinations of votes in the test set we use the following backup heuristics based on the preformance figures of the individual methods: Use the vote of the heuristic method. If this method does not vote (*u*), then select the SVM vote.

Table 4 shows the results of the proposed ensemble method, the well-known majority voting and the upper bound of this ensemble method (calculated with the same strategy over the test data), over the development and test tweet data

	Ensemble Voting	**Majority Voting**	**Upper Bound**
Dev	85.48	81.31	85.48
Test	85.50	82.70	89.37

Table 4: Results for different ensemble strategies

In the development corpus, the upper bound and ensemble results are the same, given that they apply the same knowledge. The difference is in the test dataset, where the ensemble voting is calculated based on the knowledge obtained from the development corpus, while the upper bound uses the knowledge that can be derived from the test corpus.

Table 5 illustrates the features used by each component.

	SVM (task A)	SVM (task B)	CRF	Heur
word	•		•	•
lemma				
pos	•		•	
deps			•	
pol	•	•	•	•
polW		•		
sent	•		•	•
sentiwn	•	•	•	
quant	•	•	•	•
neg	•	•	•	•
links	•			
hashTags	•			

Table 5: Information used (pos: part-of-speech; deps: dependencies; pol: basic polarity classification; polW: basic polarity word; sent: LIWC sentiments; sentwn: SentiWordnet; quant/neg: quantifiers and negation markers.)

4 Task B: A Support Vector Machine-based System

The system presented for task B is based on ML using a SVM model. The feature vector used as input for the SVM component is composed of the annotations provided by the linguistic annotation pipeline, extended with a feature obtained by applying negation to the next polar words (window of size 3).

The features used do not include the words (or their lemmas) because the number of tweets available for training is small (10^4) compared to the number of different words ($4 \cdot 10^4$). A model based on bag-of-words would suffer from overfitting and thus be very domain and time-dependent. If the train and test sets were randomly selected from a bigger set, the use of words could increase the model's accuracy, but the model would also be too narrowly applied to this specific dataset.

From the annotation pipeline we extracted as features: the polar words (PolW) and their basic polarity (Pol); the sentiment annotations from LIWC (Sent); the negation markers (Neg) and quantifiers (Quant). The model was trained using Weka (Hall et al., 2009).

The model used is SVM with the *C* parameter set to 1.0 and applying a 10 fold cross-validation. The option of doing first a model to discriminate polar and neutral tweets was discarded because Weka already does that when training classifiers for more than two training classes, and the combination of the two classifiers (a first one between polar and opinionated and a second one between positive and negative) would produce the same results.

5 Results and Discussion

The results of our system in each subcorpus and task are presented in Table 5 (average of the F1-measure over the classes positive and negative, constrained track), with the ranking achieved in the competition in parentheses.

	Tweet Corpus	SMS Corpus
Task A	0.86 (5th)	0.73 (11th)
Task B	0.61 (7th)	0.47 (28th)

Table 6: FBM system performance (F1 average over positive and negative classes, constrained track) and rankings

Given the differences in style and vocabularies between the SMS and tweet corpora, and the fact that we made not effort whatsoever to adapt our system or models to them, the drop in performance from one to the other is considerable, but to be expected since domain customization is an important aspect of opinion mining.

Task A: The confusion matrix in Table 7 shows an acceptable performance for the most frequent classes in the corpus (with an error of 7.75% and 19.5% for postive and negative cues, respectively) and a very poor job for neutral cues (98.1% of error), clearly a minority class in the training corpus (5% of the data).

	GOLD:	**Pos**	**Neg**	**Neu**
SYSTEM:	**Pos**	2,522	296	126
	Neg	206	1,240	31
	Neu	6	5	3

Table 7: Task A confusion matrix

Given the skewed distribution of polarity categories in the test corpus, however, neutral mistakes amount to only 23% of our system error, and so we

focus our analysis on the problems in positive and negative cues, respectively amounting to 31.7% and 44.8% of the total error. There are 2 main sources of error:

- *Limitations of the dictionaries* employed, which were short in covering somewhat frequent slang words (e.g., *wacky, baddest, shitloads*), expressions (e.g., *ouch, yukk, C'MON*), or phrases (e.g., *over the top*), some of which express a particular polarity but contain a word expressing just the opposite (*have a blast*, *to want something bad/ly*).

- *Problems in UGC processing*, mainly related to normalization (e.g., *fooooool*) and tokenization (*Perfect...not sure*), which put at risk the correct identification of lexical elements that are crucial for polarity classification.

Task B: The average F-score of positive and negative classes was 0.62 in the development set (that was included in the training set) and the averaged F-score for the test set was 0.61 (so they are very similar). If focusing on precision and recall, the positive and negative classes have higher precision but lower recall in the test set. We think that this low degradation of perfomance indicates the model's potential for generalization.

6 Conclusions

From our results, we can conclude that the use of ensemble combination of orthogonal methods provides good performance for Task A. Similar results could be expected for Task B (judging from mixing dictionaries and ML in similar tasks at RepLab 2012 (Chenlo et al., 2012)). The ML methods that we applied for Task B are essentially additive, and hence have difficulties in applying features such as polarity shifters. To overcome this, one of the features includes negation of polar words when a polarity shifter is near.

Overall, the SemEval Tasks have make evident the usual challenges when mining opinions from Social Media channels: noisy text, irregular grammar and orthography, highly specific lingo, etc. Moreover, temporal dependencies can affect the performance if the training and test data have been gathered at different times, as is the case with text of such a volatile nature as tweets and SMSs.

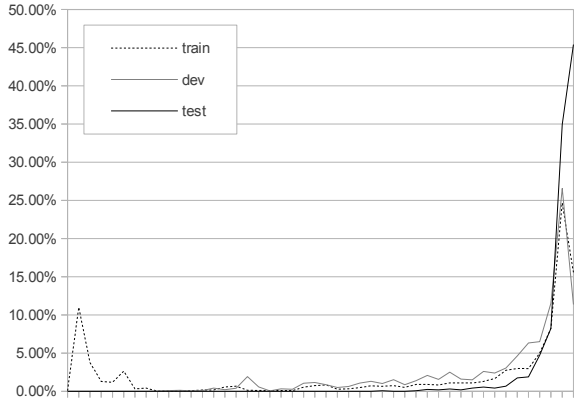

Figure 2: Distribution of tweets over time

The histogram in Figure 2 shows that this also applies to the Semeval tweets dataset. It illustrates the distribution of tweets over time (extrapolated from the sequential ids) in the 3 subcorpora (train, development and test), showing some divergence between the test corpus on the one hand, and the development and training corpora on the other. Nevertheless, our system shows little performance degradation between development and testing results, as attested in Table 4 (ensemble voting column).

Our work here and at other competitions already cited validate a system that combines stochastic and symbolic methodologies in a principled, data-driven approach. Time and domain dependencies of Social Media data make system and model generalization highly desirable, and our system hybrid nature also contribute to this objective.

Acknowledgments

This work has been partially funded by the Spanish Government project *Holopedia*, TIN2010-21128-C02-02, the CENIT program project *Social Media*, CEN-20101037, and the Marie Curie Reintegration Grant PIRG04-GA-2008-239414.

References

Baccianella, Stefano, Andrea Esuli and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the 7th conference on International Language Resources and Evaluation*, Valletta, Malta.

Chenlo, Jose M., Jordi Atserias, Carlos Rodríguez-Penagos and Roi Blanco. 2012. FBM-Yahoo! at RepLab 2012. In: P. Forner, J. Karlgren, C. Womser-Hacker (eds.) *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes*. http://clef2012.org/index.php?page=Pages/proceedings.php.

Esuli, Andrea and Fabrizio Sebastiani. 2007. SENTIWORDNET: a high-coverage lexical resource for opinion mining. Technical Report ISTI-PP-002/2007, Institute of Information Science and Technologies (ISTI) of the Italian National Research Council (CNR).

Hall, Mark, Frank Eibe, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann and Ian H. Witten. 2009. The WEKA data mining software: an update. In: *ACM* SIGKDD Explorations Newsletter, 1: 10–18.

Huang, Y. S. and C. Y. Suen. 1993. Behavior-knowledge space method for combination of multiple classifiers. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, 347–352.

Liu, Bing. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, (5-1), 1–167.

Pennebaker, James W., Martha E. Francis and Roger J. Booth. 2001. Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*.

Wilson, Theresa, Zornitsa Kozareva, Preslav Nakov, Sara Rosenthal, Veselin Stoyanov and Alan. Ritter. 2013. SemEval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '13.

Wilson, Theresa, Janyce Wiebe and Paul Hoffmann. 2010. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, *35*(3), 399–433.