10.12871/clicit2014210

# UNIBA at EVALITA 2014-SENTIPOLC Task: Predicting tweet sentiment polarity combining micro-blogging, lexicon and semantic features

**Pierpaolo Basile** and **Nicole Novielli**
Department of Computer Science, University of Bari Aldo Moro
Via, E. Orabona, 4 - 70125 Bari (Italy)
{pierpaolo.basile,nicole.novielli}@uniba.it

## Abstract

**English**. This paper describes the UNIBA team participation in the SENTIPOLC task at EVALITA 2014. We propose a supervised approach relying on keyword, lexicon and micro-blogging features as well as representation of tweets in a word space. Our system ranked 1st in both the subjectivity and polarity detection subtasks. As a further contribution, we participated in the unconstrained run, investigating the use of co-training to automatically enrich the labelled training set.

**Italiano**. *Questo articolo riporta i risultati della partecipazione del team UNIBA al task SENTIPOLC di EVALITA 2014. L'approccio supervisionato che abbiamo proposto affianca alle keyword la rappresentazione semantica dei tweet in uno spazio geometrico, l'utilizzo di feature tipiche dei micro-blog e di dizionari per la definizione della polarità a priori del lessico dei tweet. Abbiamo sperimentato, inoltre, l'uso del co-training per l'arricchimento del dataset tramite annotazione automatica di nuovi tweet.*

## 1 Introduction

Sentiment analysis is the study of the subjectivity and polarity (positive vs. negative) of a text (Pang and Lee, 2008). With the worldwide diffusion of social media, a huge amount of textual data has been made available and sentiment analysis on micro-blogging is now regarded as a powerful tool for modelling socio-economic phenomena (O'Connor et al., 2010). Dealing with such informal text poses new challenges due to the presence of slang, misspelled words and micro-blogging features such as hashtags or links.

This paper describes our participation at EVALITA 2014 SENTIment POLarity Classification (SENTIPOLC) task (Basile et al., 2014). We discuss methods and results of our experimental studies for the subjectivity and polarity classification subtasks. SENTIPOLC focuses on Italian texts from Twitter. Data provided for training are annotated according to the subjectivity/objectivity of the content carried by the tweet. Moreover, each tweet is categorized as positive, negative, or neutral. Tweet expressing both positive and negative sentiment are also included.

We build a system based on supervised approaches. For training, we exploit three different kinds of feature based on keywords and micro-blogging properties of tweets, on their representation in a distributional semantic model (Vanzo et al., 2014) and on a sentiment lexicon. The purpose of this study is twofold: (i) we propose a method to represent both the tweets and the polarity classes in the word space; (ii) we automatically develop a sentiment lexicon for the Italian starting form SentiWordNet (Esuli and Sebastiani, 2006). Additionally, we propose an approach that exploits co-training to automatically create labelled tweets using the lexicon extracted from a small set of manually annotated data.

The paper is structured as follows: we introduce our system and report the details about features in Section 2. We describe the evaluation and the system setup in Section 3. We conclude by reporting and discussing results in Section 4.

## 2 System Description

In this section we provide details about the adopted supervised strategy according to the two kinds of run provided by the organizers. In the first one, the *constrained run*, only the provided training data can be used to build the system, but lexicons are allowed. In the second one, the *unconstrained run*, additional training data can be

included. We investigate several kinds of features, which are thoroughly described in Subsection 2.1. To follow the guidelines, we arrange two settings: constrained and unconstrained. In the constrained setting we extract the features from the training data and run the learning algorithm. In the unconstrained condition it is possible to exploit additional training data, (e.g., other corpora with sentiment annotation). Rather than using further manually annotated tweets, we decide to investigate a co-training approach to automatically add new examples to the training set. Figure 1 sketches how co-training is implemented in our system. *Training data* are represented by two different sets of features: *"Feature set 1"* and *"Feature set 2"*. For each feature set we built a separated training model: *"Model 1"* and *"Model 2"*. Unlabeled data, in our case tweets without polarity annotation, are classified using both models. The class selector chooses between predicted classes exploiting classifier confidence: the class with the highest confidence is chosen and the corresponding label is given to the new tweet. The obtained examples can be used as additional training data.

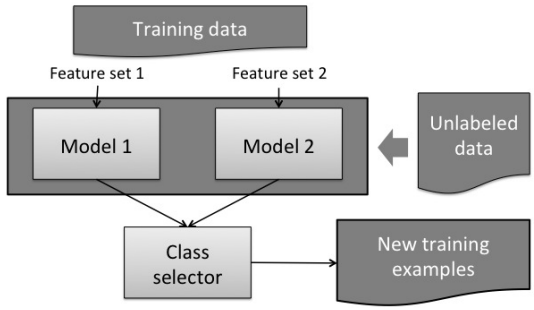


Figure 1: Co-training block diagram.

## 2.1 Features

We exploit the same features in both settings. In particular, we defined three groups of features based on: (i) keyword and micro-blogging characteristics, (ii) a sentiment lexicon, and (iii) a Distributional Semantic Model (DSM).

Keyword based features exploit tokens occurring in the tweets, only unigrams are considered. During the tokenization we replace the user mentions, URLs and hashtags with three metatokens: "_USER_", "_URL_" and "_TAG_". We create features able to capture several aspects of microblogging, such as the use of upper case and character repetitions[1], positive and negative emoticons, informal expressions of laughters[2], as well as the presence of exclamation and interrogative marks, adversative words[3], disjunctive words[4], conclusive words[5] and explicative words[6].

The second group of features concerns the DSM. Given a set of unlabelled downloaded tweets, we build a geometric space in which each word is represented as a mathematical point. The similarity between words is computed as their closeness in the space. To represent a tweet in the geometric space, we adopt the superposition operator (Smolensky, 1990), that is the vector sum of all the vectors of words occurring in the tweet. We use the tweet vector $\overrightarrow{t}$ as a semantic feature in training our classifiers. In the same fashion, we build also prototype vector for each class as the sum of all the tweet vectors belonging to the given class. We use two prototype vectors to represent, respectively, subjectivity $\overrightarrow{p_s}$ and objectivity $\overrightarrow{p_o}$. Analogously, we build four prototype vectors for positive $\overrightarrow{p_{pos}}$, negative $\overrightarrow{p_{neg}}$, positive and negative $\overrightarrow{p_{pn}}$, and neutral $\overrightarrow{p_n}$ polarity. To capture the subjectivity of a tweet $\overrightarrow{t}$, we add to the DSM features the cosine similarity between $\overrightarrow{t}$ and $\overrightarrow{p_s}$, and the similarity between $\overrightarrow{t}$ and $\overrightarrow{p_o}$. Thus, we compute all the similarity score with respect to the four prototype vectors for polarity.

Finally, the third block contains features extracted from the SentiWordNet (Esuli and Sebastiani, 2006) lexicon. We translate SentiWordNet in Italian through MultiWordNet (Pianta et al., 2002). It is important to underline that SentiWordNet is a synset-based lexicon while our Italian translation is a word based lexicon.

In order to automatically derive our Italian sentiment lexicon from SentiWordNet, we perform three steps. First, we translate the synset offset in SentiWordNet from version 3.0 to 1.6[7] using automatically generated mapping file. Then, we transfer the prior polarity of SentiWordNet to the Italian lemmata. Each synset in SentiWordNet has three polarity scores, negative, positive, and neutral, which are transferred to all the Italian lemmata belonging to the corresponding MultiWord-

[1]These features usually plays the same role of intensifiers in informal writing contexts.

[2]i.e., sequences of "ah".

[3]ma, bensì, però, tuttavia, peraltro, nondimeno, pure, eppure, sennonché, anzi, invece.

[4]o, oppure, ovvero, ossia.

[5]dunque, quindi, perciò, pertanto, onde, sicché.

[6]infatti, cioè, ossia.

[7]Since MultiWordNet is based on WordNet 1.6.

Net synset. By using this approach, a lemma can receive multiple polarity scores if it occurs in more than one synset. In such cases, we assign to the lemma the average polarity score. In the lexicon we add also emoticons as taken from Wikipedia[8]: we assign a positive score equal to 1 to the positive emoticons, and a negative score equal to 1 to the negative ones. Finally, we expand the lexicon using Morph-it! (Zanchetta and Baroni, 2005), a lexicon of inflected forms with their lemma and morphological features. We extend the polarity scores of each lemma to its inflected forms. Our strategy for creating the Italian polarity lexicon is similar to the one adopted in (Basile and Nissim, 2013), which however deal differently with multiple polarity scores for an ambiguous lemma.

The obtained Italian translation of SentiWordNet is used to compute a set of features based on prior polarity of words in the tweets, as reported in Table 3. To deal with mixed polarity cases we defined two sentiment variation features so as to capture the simultaneous expression of positive and negative sentiment in the same tweet.

The complete list and description of microblogging, semantic and lexicon features are reported in Tables 1, 2 and 3, respectively. A boolean feature that indicates if a tweet concerns the politic topic or not is finally added. Since this feature is only present in the training data, we remove it in the unconstrained run.

## 3 Evaluation

The EVALITA-2014 SENTIPOLC Task is designed for evaluating systems on their ability in: Task 1) decide whether a given tweet is subjective or objective; Task 2) decide the tweet polarity with respect to four classes: positive, negative, neutral and mixed sentiment (both positive and negative).

Organizers provided 4,513 manually annotated tweets as training data. At the time of the evaluation, 495 tweets are not available for the download and are removed from the training. We use the annotated data to extract the features and independently train the classifiers for Tasks 1 and 2. Section 3.1 reports details on our system setup.

As test set, organizers provided a collection of 1,935 manually annotated tweets (1,748 available at the time of the evaluation). Systems are compared against the gold standard in terms of F measure. Results are reported in Section 4.

[8] http://it.wikipedia.org/wiki/Emoticon

### 3.1 System Setup

The system is completely developed in JAVA, and the Weka[9] library is adopted for the Support Vector Machine[10]. Tweets are tokenized using "Twitter NLP and Part-of-Speech Tagging"[11] API developed by the Carnegie Mellon University. We use only the tokenizer since previous research has shown that part-of-speech features are not crucial for sentiment analysis on tweets (Kouloumpis et al., 2011).

Regarding the DSM, we download 10 million tweets using the Twitter Streaming API. Tweets are downloaded by querying the API using four lexicons extracted from the training data for each class. In particular, tweets in training set are divided in two classes: subjective and objective. For each class we extract a lexicon. Analogously, tweets in training set are divided into positive and negative. We add mixed polarity tweets to both positive and negative classes. Thus, we extract a lexicon for the positive class and a lexicon for the negative one. To extract the lexicons we use a probabilistic approach. We compute the probability for each token as:

$$P(t|c_i) = \frac{\#t + 1}{\#tot_i + |V|} \quad (1)$$

where $c_i$ is the class, $\#t$ are the occurrences of $t$ in $c_i$, $\#tot_i$ are the total occurrences in $c_i$, and $V$ is the vocabulary.

For each lexicon, we rank tokens in descending order according to the Kullback-Leibler divergence (KLD). For example, in the case of subjectivity detection, we compute token probabilities for both subjective $c_s$ and objective $c_o$ classes. For each token $t$ in $V$ we calculate the KLD between $P(t|c_s)$ and $P(t|c_o)$ as:

$$KLD = P(t|c_s) * log\frac{P(t|c_s)}{P(t|c_o)} \quad (2)$$

The top terms in the rank are relevant for the $c_s$ class. We perform this computation for each lexicon to extract the most 50 relevant terms for subjective, objective, positive and negative classes. We use these terms as seeds for downloading the same number of tweets for each lexicon.

We exploit these unlabeled new tweets to build a DSM, using the "word2vec"[12] tool based on a re-

[9] http://www.cs.waikato.ac.nz/ml/weka/

[10] We also experimented with Random Forest with comparable performance.

[11] http://www.ark.cs.cmu.edu/TweetNLP/

[12] https://code.google.com/p/word2vec/

| Keyword and micro-blogging features | |
|---|---|
| $n-grams$ | only unigrams are considered. User mentions, URLs and hashtag are replaced with metatokens |
| $count_{USER}$ | total occurrences of user mentions |
| $count_{URL}$ | total occurrences of URLs |
| $count_{TAG}$ | total occurrences of hashtags |
| $upperCase_{ratio}$ | the ratio between the number of upper case characters and the total number of characters |
| $emo_{pos}$ | the number of positive emoticons |
| $emo_{neg}$ | the number of negative emoticons |
| $count_{Laugh}$ | the count of sequences of 'ah' as slang expression of laughters |
| $count_{Intensif}$ | the ratio between the number of tokens with repeated characters and the total number of tokens |
| $count_{QMark}$ | the total occurrences of question marks |
| $count_{ExMark}$ | the total occurrences of exclamation marks |
| $count_{advers}$ | the total occurrences of adversative words |
| $count_{disj}$ | the total occurrences of disjunctive words |
| $count_{concl}$ | the total occurrences of conclusive words |

Table 1: Description of keyword and micro-blogging features.

| Semantic features | |
|---|---|
| $\overrightarrow{t}$ | the representation of the tweet vector in the word space |
| $sim_{subj}$ | the similarity between $\overrightarrow{t}$ and the subjective prototype vector $\overrightarrow{p_s}$ |
| $sim_{obj}$ | the similarity between $\overrightarrow{t}$ and the objective prototype vector $\overrightarrow{p_o}$ |
| $sim_{pos}$ | the similarity between $\overrightarrow{t}$ and the positive prototype vector $\overrightarrow{p_{pos}}$ |
| $sim_{neg}$ | the similarity between $\overrightarrow{t}$ and the negative prototype vector $\overrightarrow{p_{neg}}$ |
| $sim_{posneg}$ | the similarity between $\overrightarrow{t}$ and the mixed polarity prototype vector $\overrightarrow{p_{pn}}$ |
| $sim_{neutral}$ | the similarity between $\overrightarrow{t}$ and the neutral prototype vector $\overrightarrow{p_n}$ |

Table 2: Description of semantic features.

vised implementation of the Recurrent Neural Net Language Model (Mikolov et al., 2013) using a log-linear approach. In particular, we use the Continuous Bag-of-Words Model (CBOW) with 200 vector dimensions. We remove the terms with less than ten occurrences, obtaining a total number of about 200,000 terms overall.

We trained our classifiers using a SVM with the RBF kernel, setting the C parameter to 4. We select these values after a 10-fold validation on training data to select the best combination. The total number of features is 12,117. In the constrained run, the entire set of features is used for both subjectivity and polarity classification tasks. Regarding the unconstrained run, we split the features in two subsets to implement the co-training approach. The first set (Feature set 1 in Figure 1) is composed by keyword and micro-blogging, and lexicon features used to learn Model 1; the second set (Feature set 2) exploits the semantic features to learn Model 2. In the co-training strategy we obtained about 40,000 new examples automatically tagged.

## 4 Results and Discussion

The overall system performance is assessed in terms of F measure, according to the measure adopted by the task organizers. Table 4 reports the system performance, its rank, and the percentage improvement over the baseline calculated assigning the most frequent class in the gold standard.

The results are very encouraging: the system always obtains the best performance in all settings and in Task 1 of the un-constrained run it differs for only 0.0005 from the first ranked one. We observe that the co-training approach seems

| Sentiment lexicon based features | |
|---|---|
| $p_{subj}$ | the subjectivity polarity, it is the sum of the positive and negative scores |
| $p_{obj}$ | the objectivity polarity, it is the sum of the neutral scores |
| $o_{subj}$ | the number of tokens having the positive or negative score higher than zero |
| $o_{obj}$ | the number of tokens having the neutral score higher than zero |
| $r_{subj}$ | the ratio between $p_{subj}/o_{subj}$ |
| $r_{obj}$ | the ration between $p_{obj}/o_{obj}$ |
| $subjobj_{diff}$ | the difference between $r_{subj} - r_{obj}$ |
| $sum_{pos}$ | the sum of positive scores for the tokens in the tweet |
| $sum_{neg}$ | the sum of negative scores for the tokens in the tweet |
| $o_{pos}$ | the number of tokens that have the positive score higher than zero |
| $o_{neg}$ | the number of tokens that have the negative score higher than zero |
| $r_{pos}$ | the ratio between $sum_{pos}/o_{pos}$ |
| $r_{neg}$ | the ration between $sum_{neg}/o_{neg}$ |
| $posneg_{diff}$ | the difference between $r_{pos} - r_{neg}$ |
| $max_{pos}$ | the sum of the positive scores, where $positive\ score > negative\ score$ |
| $max_{neg}$ | the sum of the negative scores, where $negative\ score > positive\ score$ |
| $max_{subj}$ | the sum of $max_{pos}$ and $max_{neg}$ |
| $max_{obj}$ | the sum of the neutral scores, where the neutral score is higher than both the positive and negative ones |
| $subjobj_{max_{diff}}$ | the difference between $max_{subj} - max_{obj}$ |
| $posneg_{max_{diff}}$ | the difference between $max_{pos} - max_{neg}$ |
| $sentiment\ variation$ | for each token occurring in the tweet a tag is assigned, according to the highest polarity score of the token in the Italian lexicon. Tag values are in the set {OBJ, POS , NEG}. The sentiment variation counts how many switches from POS to NEG, or vice versa, occur in the tweet. |
| $sentiment\ variation\ pos/neg$ | it is similar to the previous feature, but the OBJ tag is assigned only if both positive and negative scores are zero. Otherwise, the POS tag is assigned if the positive score is higher than the negative one, vice versa the NEG tag is assigned. |

Table 3: Description of sentiment lexicon features.

| Setting | Task | F | Rank | Imp. |
|---|---|---|---|---|
| baseline | Task 1 | 0.4005 | - | - |
| | Task 2 | 0.3718 | - | - |
| constrained | Task 1 | 0.7140 | 1 | 78% |
| | Task 2 | 0.6771 | 1 | 82% |
| unconstrained | Task 1 | 0.6892 | 2 | 72% |
| | Task 2 | 0.6638 | 1 | 79% |

Table 4: System results for each task and setting.

to introduce noise and need to be tuned in future replication of our study. A deep analysis of the results shows that the co-training system slightly improves the performance in classifing positive tweets, while the performance in other classes decreases. Details about each class are reported in Table 5, improvements in the un-constrained task are underlined by the ↑ symbol. The evaluation criteria for the polarity task involve consideration of mixed cases as both negative and positive.

After an error analysis, we discover a bias in our classifier due to the domain-specific lexicon about political topics. This is the main cause of error in the classification of the objective tweets, which are labeled as subjective in 58% of misclassified cases due to the presence of lexicon related to topics for which people generally express a negative opinion[13]. For the same reason, the 37% and the 44% of misclassified neutral and positive cases, respectively, are classified as negative. Furthermore, we observe that the recall of our classifier could be improved for both positive and negative classes by enriching our lexicon with jargon and idiomatic expressions. Finally, in the 43% of misclassified negative cases common sense reasoning would be required to detect the negative opinion expressed

[13] e.g., Monti, governo, Grillo.

| Setting | Class | False (F) | | | True (T) | | | Comb. |
|---|---|---|---|---|---|---|---|---|
| | | $P_F$ | $R_F$ | $F_F$ | $P_T$ | $R_T$ | $F_T$ | $F$ |
| Constrained | sub | 0.6976 | 0.5271 | 0.6005 | 0.8498 | 0.8064 | 0.8275 | 0.7140 |
| | pos | 0.8102 | 0.8364 | 0.8231 | 0.7195 | 0.4162 | 0.5274 | 0.6752 |
| | neg | 0.7474 | 0.6869 | 0.7170 | 0.6882 | 0.5995 | 0.6408 | 0.6789 |
| Un-constrained | sub | 0.6937 | 0.4629 | 0.5553 | 0.8317 | 0.8148 | 0.8232 | 0.6892 |
| | pos | 0.8189 | 0.7696 | 0.7935 | 0.5969 | 0.4780 | 0.5309 ↑ | 0.6622 |
| | neg | 0.7400 | 0.6654 | 0.7007 | 0.6658 | 0.5984 | 0.6303 | 0.6655 |

Table 5: System results for each class.

by the author[14], including ironic tweets.

As a further investigation of the predictive power of the features in our model, we perform an ablation test for both tasks. We removed each group of features to assess the decrease of F measure on test data with respect to the setting including all features. Results are reported in Figures 2 and demonstrate the importance of all feature groups. Particularly, semantic features plays a key role, as we observe how removing them causes the highest decrease in performance in both tasks.

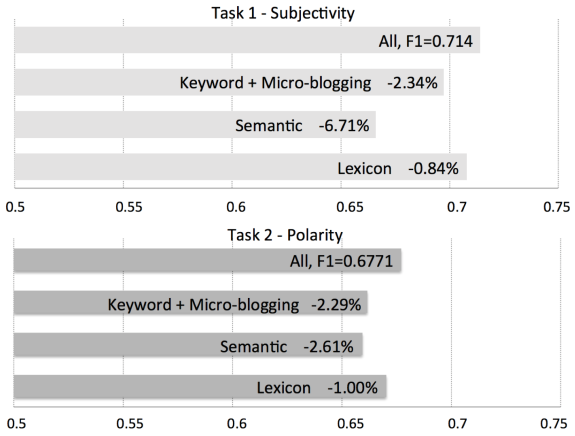


Figure 2: Decrease of F by removing each feature group, compared to the complete feature setting.

Future replication of this study will involve further data, to validate and generalize our findings.

## Acknowledgements

This work is partially funded by the ATS Romantic Living Lab under the Apulian ICT Living Labs program and the project PON 01 00850 ASK-Health (Advanced System for the interpretation and sharing of knowledge in health care).

[14]"Governo Monti: ipotesi #Passera allo Sviluppo. Candidatura spontanea della Minetti."

## References

Valerio Basile and Malvina Nissim. 2013. Sentiment analysis on italian tweets. In *Proc. of WASSA 2013*, pages 100–107.

Valerio Basile, Andrea Bolioli, Malvina Nissim, Viviana Patti, and Paolo Rosso. 2014. Overview of the Evalita 2014 SENTIment POLarity Classification Task. In *Proc. of EVALITA 2014*, Pisa, Italy.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proc. of LREC*, pages 417–422.

Efthymios Kouloumpis, Theresa Wilson, and Johanna D. Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! In *Proc. of ICWSM 2011*, pages 538–541.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proc. of ICLR Work.*

Brendan O'Connor, Ramnath Balasubramanyan, Bryan Routledge, and Noah Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *Intl AAAI Conf. on Weblogs and Social Media (ICWSM)*, volume 11, pages 122–129.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January.

Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. 2002. Multiwordnet: developing an aligned multilingual database. In *Proc. 1st Intl Conf. on Global WordNet*, pages 293–302.

Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1-2):159–216, November.

Andrea Vanzo, Danilo Croce, and Roberto Basili. 2014. A context-based model for sentiment analysis in twitter. In *Proc. of COLING 2014*.

Eros Zanchetta and Marco Baroni. 2005. Morph-it!: a free corpus-based morphological resource for the italian language. *Proc. of the Corpus Linguistics Conf. 2005*.