# Twitter Sentiment Polarity Classification using Barrier Features

**Anita Alicante, Anna Corazza, Antonio Pironti**
Department of Electrical Engineering and Information Technologies (DIETI)
Università di Napoli Federico II
via Claudio 21, 80125 Napoli, Italy
anita.alicante@unina.it, anna.corazza@unina.it,
antonio.pironti@gmail.com

## Abstract

**English.** A crucial point for the applicability of sentiment analysis over Twitter is represented by the degree of manual intervention necessary to adapt the approach to the considered domain. In this work we propose a new sentiment polarity classifier exploiting *barrier features*, originally introduced for the classification of textual data. Empirical tests on SemEval2014 competition data sets show that such approach overcomes performance of baseline systems in nearly all cases.

***Italiano.*** *Un punto cruciale per l'applicabilità della sentiment analysis su Twitter è rappresentato dal livello di intervento manuale necessario per adattare l'approccio al dominio considerato. In questo lavoro proponiamo un nuovo classificatore di sentiment polarity che sfrutta le* barrier features*, originariamente introdotte per la classificazione di relazioni estratte da testi. Test empirici sui data sets usati nella competizione SemEval2014 mostrano che l'approccio proposto supera le performance dei sistemi baseline nella maggioranza dei casi.*

## 1 Introduction

Sentiment analysis (SA) (Pang and Lee, 2008), or opinion mining, is mainly about finding out the feelings of people from data such as product reviews and news articles.

Most methods adopt a two-step strategy for SA (Pang and Lee, 2008): in the subjectivity classification step, the target is classified to be subjective or neutral (objective), while in the polarity classification step the subjective targets are further classified as positive or negative. Therefore, two classifiers are trained for the whole SA process: the subjectivity classifier and the polarity classifier. Polarity is an aspect of sentiment analysis which can be faced as a three-way classification problem, in that it aims to associate either a positive, negative or neutral polarity to each tweet.

Expressions in tweets are often ambiguous because they are very informal messages no longer than 140 characters, containing a lot of misspelled words, slang, modal particles and acronyms. The characteristics of the employed language are very different from more formal documents and we expect statistical methods trained on tweets to perform well thanks to an automatic adaptation to such specificities.

As evidenced by tasks included in competitions (Rosenthal et al., 2015) and (Nakov et al., 2016), twitter sentiment analysis is a relevant topic for scientific research. To the best of our knowledge (Ravi and Ravi, 2015; Kolchyna et al., 2015; Silva et al., 2016) present a comprehensive, State-of-the-Art (SoA) review on the research work done in various aspects of SA. Furthermore some approaches, as described in (Gonçalves et al., 2016), are based on the combination of several existing SoA "off-the-shelf" methods for sentence-level sentiment analysis[1].

(Saif et al., 2016) proposes an approach based on the notion that the sentiment of a term depends on its contextual semantics and some trigonometric properties on SentiCircles, that is a 2D geometric circle. These properties are applied to amend an initial sentiment score of terms, according to the context in which they are used. The sentiment identification at either entity or tweet-level is then performed by leveraging trigonometric identities

[1] A point of strength of this kind of systems is that combining several classification methods in an ensemble approach results to be very strong with respect to the input vocabulary size and to the amount of available training.

on SentiCircles.

The approach we are proposing has been experimentally assessed by comparing its performance with two baseline systems. In addition to that, the capability of adaptation of the approach to slightly different domains has been tested by comparing on a web-blog data set the performances of two systems in which the *Barrier Feature*(BF) dictionary has been respectively built on a collection of tweets and Wikipedia webpages. Eventually, the contribution of BF has been evaluated.

## 2 Proposed approach

Some automatic machine learning approaches recently applied to Twitter sentiment polarity classification try new ways to run the analysis, such as performing sentiment label propagation on Twitter follower graphs and employing social relations for user-level sentiment analysis (Speriosu et al., 2011). Others, not differently from the one we are proposing here, investigate new sets of features to train the model for sentiment identification, such as microblogging features including hashtags, emoticons etc. (Barbosa and Feng, 2010; Kouloumpis et al., 2011). Indeed, we are proposing to add *Barrier Features* (BFs) (Alicante and Corazza, 2011) to unigrams, bigrams and input parse tree and to provide them as input to a Support Vector Machine (SVM) classifier.

Introduced in the context of another application of text mining, namely relation classification, BFs are inspired by (Karlsson et al., 1995) for Part-of-Speech (PoS) tagging, but they have been completely redesigned as features rather than rules. BFs have also been exploited in (Alicante et al., 2016) for Italian Language in a unsupervised entity and relation extraction system, proving the language portability of these features. BFs describe a linguistic binding between the entities involved in each relation.

BFs require PoS tagging of the considered texts, which can be automatically performed with very high accuracy (Giménez and Màrquez, 2004). In fact, they consist of sets of PoS tags occurring between a predefined PoS pair, namely (endpoint, trigger). Similarly to unigrams and bigrams of words, these features are Boolean: for each tweet, their value is *true* if the feature occurs in the tweet, *false* otherwise.

Given a set of (endpoint, trigger) pairs *P* and a sentence (or tweet, in our case) *s*, the BFs extraction algorithm loops over the PoS tags in *s* and, for each trigger tag *t*, it looks backward in the sentence finding the closest occurrence of a PoS tag *e* such that $(e, t) \in P$ . If such endpoint is found, then the algorithm extracts the barrier feature *(e, t,* $PT_{e,t}$*)*, where $PT_{e,t}$ is the set of PoS tags occurring between *e* and *t*. Otherwise it extracts as many barrier features as the number of the elements in *P* having *t* as trigger tag and, for each of them, the related tag set is the set of POS tags of all the words in the sentence preceding the trigger.

While in the preceding work (Alicante and Corazza, 2011) (endpoint, trigger) pairs were predefined, in this work we apply an innovative approach: we choose such pairs in a completely automatic and unsupervised way, starting from an unannotated data set, not necessarily in the same domain as the final task. In fact, BFs are unlexicalized as they only depend on PoS tags: for any text collection, we can perform this analysis basing on a different one which has to be similar in the kind of language but not necessarily in the domain. For instance, we expect the pairs which are more effective for the language adopted in tweets to be generally different from the ones adopted for standard texts.

In choosing the (endpoint, trigger) pairs, our purpose is two-fold: we aim to obtain a high variability of the identified sets of tags while only considering statistically significant patterns, that is, patterns having a rather large number of occurrences. In addition to this, we do not want to penalize longer patterns, although they usually correspond to larger and then more infrequent sets.

Table 1: Endpoints and triggers of the BFs employed for the tweet and text messages task.

| **Endpoint** | **Trigger** |
|---|---:|
| DT | JJR or NNPS |
| NNP | NNP or VBZ |
| IN | NNS |
| NN | NN or VBG or VBN |
| RB | RBR |
| PRP | VBD or VBP |
| TO | VB |

For each possible trigger, we therefore choose the endpoint ep which maximizes the *expected information per tag* of the set corresponding to the

Table 2: BFs built considering the (endpoint, trigger) pairs listed in Table 1 and the following text: *Now/RB I/PRP can/MD see/VB why/WRB Dave/NNP Winer/NNP screams/NNS about/IN lack/NN of/IN Twitter/NNP API/NNP ,/, its/PRP limitations/NNS and/CC access/NN throttles/NNS !/.*

| Barrier Feature | Combined Text |
|---|---|
| (TO, VB, {MD, PRP, RB}) | Now I can see |
| (NNP, NNP, {MD, PRP, RB, VB, WRB}) | Now I can see why Dave |
| (NNP, NNP, {}) | Dave Winer |
| (IN, NNS, {MD, NNP, PRP, RB, VB, WRB}) | Now I can see why Dave Winner screams |
| (NN, NN, {IN, MD, NNP, NNS, PRP, RB, VB, WRB}) | Now I can see why Dave Winner screams about lack |
| (NNP,NNP, {IN, NN, NNS}) | Winer screams about lack of Twitter |
| (NNP, NNP, {}) | Twitter API |
| (IN, NNS {,, NNP, PRP}) | of Twitter API, its limitations |
| (NN, NN, {,, CC, IN, NNP, NNS, PRP}) | lack of Twitter API, its limitations and access |
| (IN,NNS, {,, CC, NN, NNP, NNS, PRP}) | of Twitter API, its limitations and access throttles |

BF, that is:

$$\mathrm{sc}(\mathrm{ep}) = -\sum_{\mathrm{BF}} \Pr(\mathrm{BF}) \frac{1}{\mathrm{len}(\mathrm{BF})} \log \Pr(\mathrm{BF}) \quad (1)$$

where $\Pr(\mathrm{BF})$ has been estimated by the corresponding frequency. In order to cut off insignificant cases, a threshold has been put on the minimum number of occurrences of the considered BF candidates. The normalization on the set size len(BF) has been introduced to avoid penalizing larger sets.

Table 1 reports the pairs resulting from this new approach and adopted for the experiments described in Section 3, Table 2 shows an example of BF extraction based on those pairs.

While in the system presented in (Alicante and Corazza, 2011) BFs were collected by only using the training set, in this work we consider an additional feature reduction step: we only take into account the BFs contained in a BFs dictionary, which is built by only considering the BFs whose number of occurrences within an unannotated data set is greater or equal than a threshold value. The data set employed in the BFs dictionary construction step is not necessarily constrained to the training set [2].

Being unlexicalized, BFs lead us to improve the portability of our approach not only towards new languages but also towards new kinds of applications. In particular, this and the dictionary construction steps are decisive both for the automation of the process and for its performance.

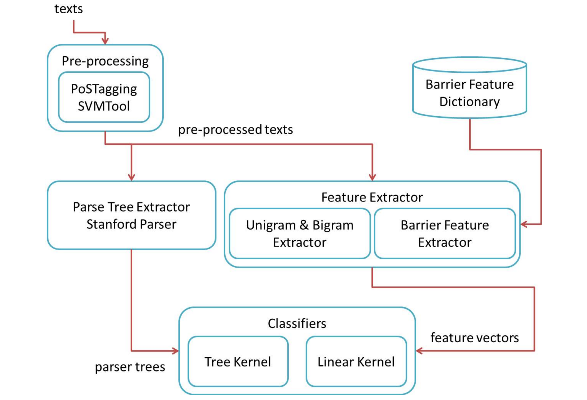

Figure 1: Architecture of the system for Twitter sentiment polarity classification.

## 3 Experimental Assessment

In order to evaluate our system performance, we implemented a solution for the *Message Polarity Classification* subtask of *SemEval-2014 Task 9 (Sentiment Analysis in Twitter)*[3] (Rosenthal et al., 2014). For each input tweet, our classification system decides whether it expresses a positive, negative, or neutral sentiment. According to the competition rules, the only training data we used are the ones that have been provided by the task organisers. We used a training set of about 8,000 tweets, a subset of the training and the development data released by the organisers[4].

After training the classifier on this training set, the performance of the obtained system have been evaluated against the test datasets provided for

[2] Specifically, we used the same data set employed for the identification of the PoS (endpoint, trigger) pairs. This aspect is not as trivial as it might seem: such strategy allows the application of the approach also to tasks where the size of the annotated training set is limited.

[3] The SemEval-2014 Task 9 competition website: `http://alt.qcri.org/semeval2014/task9/`

[4] The only way to collect the data is by using the downloader script available to the participants to the competition and some of the tweets were no longer available on Twitter at the time we ran the script.

the competition: Twitter2013 (T2013), tweets provided for testing the task in 2013; Twitter2014 (T2014), a new test set delivered in 2014; Twitter2014Sarcasm( T2014Sa), a dataset of sarcastic tweets; LiveJournal2014 (LJ2014), a set of sentences extracted from the *LiveJournal* blog; SMS13, text messages provided for testing the same task in 2013. The statistics for each test datasets are shown in Table 3.

Table 3: Dataset Statistics of SemEval2014-taskB, *Message Polarity Classification*

| Data Set | Positive | Negative | Neutral | Total |
|---|---|---|---|---|
| **LiveJournal2014** | 427 | 304 | 411 | 1142 |
| **SMS2013** | 492 | 394 | 1207 | 2093 |
| **Twitter2013** | 1281 | 542 | 1426 | 3249 |
| **Twitter2014** | 633 | 125 | 453 | 1211 |
| **Twitter2014Sarcasm** | 33 | 40 | 13 | 86 |

Table 4: Experimental results to compare the performance of our systems with the two baseline systems. Bold cases correspond to the best performance, while symbol ‡ indicates statistical significance of the comparison with the confidence interval.

| Data Set | LJ2014 | SMS2013 | T2013 | T2014 | T2014Sa |
|---|---|---|---|---|---|
| BLS1 | **74.84** | 70.28 | 70.75 | 69.85 | 58.16 |
| BLS2 | 69.44 | 57.36 | 72.12 | 70.96 | 56.50 |
| BFS | 68.91 | **72.01**‡ | **72.88**‡ | **72.10**‡ | **58.79**‡ |

Table 5: Experimental results to evaluate the BF contribution. Bold cases correspond to the best performance. Symbol † indicates the improvement of BFS is statistical significant, verified with approximate randomisation.

| Data set | System | P | R | F1 |
|---|---|---|---|---|
| LiveJournal 2014 | WOBFS | 68.94 | 62.34 | 65.32 |
| | BFS | **74.69**† | **63.97**† | **68.91**† |
| SMS 2013 | WOBFS | 64.25 | **72.92** | 67.14 |
| | BFS | **74.80**† | 69.43 | **72.01**† |
| Twitter 2013 | WOBFS | 74.30 | 67.32 | 70.62 |
| | BFS | **77.96**† | **68.45**† | **72.88**† |
| Twitter 2014 | WOBFS | 76.09 | 65.81 | 70.57 |
| | BFS | **78.19**† | **66.98**† | **72.10**† |
| Twitter2014 Sarcasm | WOBFS | 61.12 | 52.92 | 55.28 |
| | BFS | **64.75**† | **54.59**† | **58.79**† |

The Barrier Features System (BFS) implements the approach we are proposing and follows the schema depicted in Figure 1. Input is tagged by using SVMtool[5] (Giménez and Màrquez, 2004) an SVM-based tagger able to achieve a very competitive accuracy on English texts. Although accuracy is likely to be lower on tweets, classification performance does not appear to be affected; this is probably due to the robustness of the statistical learner against such kind of errors. In order to reduce syntactical irregularities, we remove hashtags from tweets before providing them to the PoS-tagger component.

In the BFS system, we use the STS data set[6] to build both the (endpoint, trigger) PoS pairs and the BFs dictionary. For the BFs dictionary construction step we considered a threshold value of $10$, chosen by $5$-fold cross validation on the SemEval2014 training set. This resulted in $44,536$ different BFs. In conclusion, once BFs are extracted from the SemEval2014 datasets, a vector of binary features which encodes all the related unigrams, bigrams and BFs is associated to every tweet.

We use the Stanford Parser[7] (Klein and Manning, 2003a; Klein and Manning, 2003b) to extract the parse trees for each of the sentences contained in the datasets. Since a tweet can be composed by several sentences, we use Tsurgeon[8] (Levy and Andrew, 2006) to build a single parse tree for each dataset's item (tweet, text messages, etc.).

The classification module based on Support Vector Machines has been implemented using the SVMLight-TK[9] (Moschitti, 2006) package. This module takes as input both the feature vectors and the parse trees. Moreover, by applying SVMs with a combination of two different kernel functions, we can handle at the same time both structured and non-structured information. Indeed, as in (Alicante et al., 2014; Alicante and Corazza, 2011), we applied tree kernels to the parse trees and a linear kernel to the vector of binary fea-

[5] The software can be freely downloaded from `http://www.lsi.upc.edu/~nlp/SVMTool/`

[6] The Stanford Twitter Sentiment (STS) data set can be freely downloaded from `http://help.sentiment140.com/for-students/`

[7] The parser can be freely downloaded from `http://nlp.stanford.edu/software/lex-parser.shtml`

[8] Tsurgeon can be freely downloaded from `http://nlp.stanford.edu/software/tregex.shtml`

[9] The software package can be freely downloaded from `http://disi.unitn.it/moschitti/Tree-Kernel.htm`

tures described above. We build three binary classifiers, one for each sentiment/class (positive, negative, neutral). Moreover, for each classifier, the training phase has been performed by considering gold positive examples for the considered class, while negative examples are represented by all the other messages. In this way, the number of negative examples is much larger than the positive ones. SVMLight allows to balance the number of positive and negative examples by using a cost factor given by the rate between the number of negative and positive training examples. In order to assess our classification system performance, we consider two baseline systems (BLS), namely the two systems that won the SemEval2014 competition (Rosenthal et al., 2014). The former, BLS1 (Zhu et al., 2014), is based on an SVM classifier and a feature set composed by some lexical and syntactical features, while the latter, BLS2 (Miura et al., 2014), exploits a Logistic Regression trained with features based on lexical knowledge.

## 4 Results

Performance is assessed by adopting the same evaluation metrics as in the SemEval2014 competition (Rosenthal et al., 2014). As usual, they are based on $F_1$*-measure*, which is separately computed for each class (positive, negative and neutral). Table 4 compares the classification performance of our tweet system, namely BFS, and the baseline systems, namely BLS1 (Zhu et al., 2014) and BLS2 (Miura et al., 2014) by adopting the same evaluation protocol used in the SemEval2014 competition (Rosenthal et al., 2014). Our system performs significantly better on all data sets except LiveJournal2014. However, additional experiments, whose results are here omitted due to space constraints, showed that our approach performs better than BLS2 on this data set when the BF dictionary is built on Wikipedia.

We think that the explanation for this behaviour depends on the capability of the approach to adapt to the employed data set. In fact, our strategy is based on the use of unsupervised mining of text to maximize the adaptation to the specificity of the type of the language. This also explains why BFS performs worse than the others on LiveJournal2014: the syntactical structure of the structured sentences contained in a weblog is quite different from the tweets' one. It is worth highlighting that the difference in performance is not statistically significant though. The main innovation of our system is the introduction of BFs and the way in which it learns them from data. We assess the BFs contribution to the overall classification performance by comparing the performance between the Barrier Features System (BFS) and Barrier Features System (WOBFS) systems we described in Section 3 and report results in Table 5. Note that this table is more detailed than Table 4 because in this case we can run both systems and collect all the different parameters. Barrier features almost always improve performance both in terms of precision and recall, and thus also in terms of $F_1$. In a few cases, the introduction of BFs improves precision while decreasing recall: however, in all these cases $F_1$ improves in BFS with respect to WOBFS.

In conclusion, the introduction of BFs always comes with an improvement in terms of $F_1$ and such improvement is nearly always statistically significant. We can therefore conclude that BFs provide a crucial contribution to sentiment polarity classification.

## 5 Conclusions and future work

We explored the effectiveness of BFs for sentiment polarity classification in Twitter posts and we showed on SemEval2014 data sets that they can be very effective. In our approach, the need of a manual intervention is really minimum. Indeed, the BFs dictionary can be built from any collection of tweets, even one that do not belong to the same domain of the considered task. This is quite interesting because it suggests that BFs are able to capture hints about the polarity of the expressions in a domain independent way.

## References

Anita Alicante and Anna Corazza. 2011. Barrier features for classification of semantic relations. In Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, and Nicolas Nicolov, editors, *RANLP*, pages 509–514. RANLP 2011 Organising Committee.

Anita Alicante, Massimo Benerecetti, Anna Corazza, and Stefano Silvestri. 2014. A distributed information extraction system integrating ontological knowledge and probabilistic classifiers. In *Proceedings of the 9th International 3PGCIC-2014 Conference*, Guangzhou, CHINA. In Press.

Anita Alicante, Anna Corazza, Francesco Isgrò, and Stefano Silvestri. 2016. Unsupervised entity and relation extraction from clinical records in Italian. *Computers in Biology and Medicine*.

Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 36–44, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jesús Giménez and Lluís Màrquez. 2004. SVMTool: A general POS tagger generator based on Support Vector Machines. In *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC)*, pages 43–46, Lisbon, Portugal.

Pollyanna Gonçalves, Daniel Hasan Dalip, Helen Costa, Marcos André Gonçalves, and Fabrício Benevenuto. 2016. On the combination of off-the-shelf sentiment analysis methods. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pages 1158–1165. ACM.

Fred Karlsson, Atro Voutilainen, Juha Heikkila, and Arto Anttila, editors. 1995. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter.

Dan Klein and Christopher D. Manning. 2003a. Accurate unlexicalized parsing. In *Proc. of ACL 03 of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430, Morristown, NJ, USA. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2003b. Fast exact inference with a factored model for natural language parsing. In *Proc of NIPS03: In Advances in Neural Information Processing Systems*, pages 3–10. MIT Press.

Olga Kolchyna, Thársis TP Souza, Philip Treleaven, and Tomaso Aste. 2015. Twitter sentiment analysis: Lexicon method, machine learning method and their combination. *arXiv preprint arXiv:1507.00955*.

E. Kouloumpis, T. Wilson, and J. Moore. 2011. Twitter Sentiment Analysis: The Good the Bad and the OMG! In *Fifth International AAAI Conference on Weblogs and Social Media*.

Roger Levy and Galen Andrew. 2006. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the fifth international conference on Language Resources and Evaluation*, pages 2231–2234. Citeseer.

Yasuhide Miura, Shigeyuki Sakaki, Keigo Hattori, and Tomoko Ohkuma. 2014. Teamx: A sentiment analyzer with enhanced lexicon mapping and weighting scheme for unbalanced data. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 628–632, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Alessandro Moschitti. 2006. Making tree kernels practical for natural language learning. In *EACL*.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. Semeval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval 2016), San Diego, US (forthcoming)*.

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

Kumar Ravi and Vadlamani Ravi. 2015. A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. *Knowledge-Based Systems*, 89:14–46.

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 73–80, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. Semeval-2015 task 10: Sentiment analysis in twitter. *Proceedings of SemEval-2015*.

Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani. 2016. Contextual semantics for sentiment analysis of twitter. *Information Processing & Management*, 52(1):5–19.

Nadia Felix F Da Silva, Luiz FS Coletta, and Eduardo R Hruschka. 2016. A survey and comparative study of tweet sentiment analysis via semi-supervised learning. *ACM Computing Surveys (CSUR)*, 49(1):15.

Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. 2011. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First Workshop on Unsupervised Learning in NLP*, EMNLP '11, pages 53–63, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiaodan Zhu, Svetlana Kiritchenko, and Saif Mohammad. 2014. Nrc-canada-2014: Recent improvements in the sentiment analysis of tweets. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 443–447, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.