# DAEDALUS at SemEval-2014 Task 9: Comparing Approaches for Sentiment Analysis in Twitter

**Julio Villena-Román**
Daedalus, S.A.
jvillena@daedalus.es

**Janine García-Morera**
Daedalus, S.A.
jgarcia@daedalus.es

**José Carlos González-Cristóbal**
Universidad Politécnica de Madrid
josecarlos.gonzalez@upm.es

## Abstract

This paper describes our participation at SemEval-2014 sentiment analysis task, in both contextual and message polarity classification. Our idea was to compare two different techniques for sentiment analysis. First, a machine learning classifier specifically built for the task using the provided training corpus. On the other hand, a lexicon-based approach using natural language processing techniques, developed for a generic sentiment analysis task with no adaptation to the provided training corpus. Results, though far from the best runs, prove that the generic model is more robust as it achieves a more balanced evaluation for message polarity along the different test sets.

## 1 Introduction

SemEval[1] is an international competitive evaluation workshop on semantic related tasks. Among the ten different tasks that have been proposed in 2014, Task 9 at SemEval-2014[2] focuses on sentiment analysis in Twitter.

Sentiment analysis could be described as the application of natural language processing and text analytics to identify and extract subjective information from texts. Given a message in English, the objective is to determine if the text expresses a positive, negative or neutral sentiment in that context.

It is a major technological challenge and the task is so hard that even humans often disagree on the sentiment of a given text, as issues that one individual may find acceptable or relevant may not be the same to others, along with multilingual aspects and different cultural factors.

The task defines two subtasks, where the difference is that whereas the output in subtask B must be the message polarity classification, i.e., the global polarity of the whole message, subtask A is focused on contextual polarity disambiguation, i.e., the message contains a marked instance of a word or phrase and the expected output must be the polarity of that specific instance within the whole message.

Daedalus (2014) is a leading provider of language-based solutions in Spain, and long-time participants in different research conferences and evaluation workshops such as CLEF (2014) and NTCIR (2014), in many different tasks including sentiment analysis (Villena-Román et al., 2008; Villena-Román et al., 2012).

This paper describes our participation in both contextual (subtask A) and message (subtask B) polarity classification. The main idea behind our participation is to compare two different techniques for sentiment analysis: a machine learning approach using the provided corpus to train a model specifically adapted to that scenario against a lexicon-based approach using advanced natural language processing techniques for capturing the meaning of the text, developed prior to the task and obviously without using the provided corpus.

Our point of view is that although machine learning classifiers generally achieve better results in competitive evaluations that provide a training corpus, when these same models are applied to a different scenario, the precision and recall metrics are drastically reduced, thus affecting to the perception and confidence of stakeholders in sentiment analysis technologies.

Our different approaches, experiments and results achieved are presented and discussed in the following sections.

[1] http://alt.qcri.org/semeval2014/

[2] http://alt.qcri.org/semeval2014/task9/

*Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 218–222,
Dublin, Ireland, August 23-24 2014.

## 2 Constrained Runs: Machine Learning Classifier

The first approach is a simple quite naive machine learning classifier trained exclusively with the provided training corpus. This is the approach adopted for constrained runs in both subtask A and B.

First, based on the Vector Space Model (Salton et al., 1975), the text of each tweet is converted into a term vector where terms are assumed to represent the semantic content of the message. Textalytics parsing API (Textalytics, 2014a) offered through a REST-based web service is used to get the lemma of each word and filter part-of-speech categories: currently nouns, verbs, adjectives and adverbs are selected as terms. A weighted term vector based on the classical TF-IDF is used. Both the training and the test set are preprocessed in this same way.

After this preprocessing, a classifier trained on the training corpus is used to classify the test corpus. Many different supervised learning algorithms where evaluated with 10-fold cross validation, using Weka (Hall et al., 2009). We finally selected Multinomial Naive Bayes algorithm, training three different binary classifiers: positive/not_positive, negative/not_negative and neutral/not_neutral. To select the final global message polarity, a simple rule-based decision is made:

```
if positive and not_negative and
  not_neutral then positive
else if negative and not_positive and
  not_neutral then negative
else neutral
```

This is directly the output for subtask B. For subtask A, this same global polarity is assigned to each text fragment, i.e., subtask A and B are treated in the same way.

## 3 Unconstrained Runs: Lexicon-Based Model

Our second approach, used in the unconstrained runs in both subtasks, is based on 1) the information provided by a semantic model that includes rules and resources (polarity units, modifiers, stopwords) annotated for sentiment analysis, 2) a detailed morphosyntactic analysis of the tweet to lemmatize and split the text into segments, useful to control the scope of semantic units and perform a fine-grained detection of negation in clauses, and 3) the use of an aggregation algorithm to calculate the global polarity value of the text based on the local polarity values of the different segments, including an outlier detection.

We consider this approach to be unconstrained because the lexicon in the semantic model (which would be valid itself for a constrained run) has been generated, tested and validated using additional training data.

All this functionality is encapsulated and provided by our Textalytics API for multilingual sentiment analysis (Textalytics, 2014b) in several languages, including English. Apart from the text itself, a required input parameter is the semantic model to use in the sentiment evaluation. This semantic model defines the domain of the text (the analysis scenario) and is mainly based on an extensive set of dictionaries and rules that incorporate both the well-known "domain-independent" polarity values (for instance, in general, in all contexts, *good* is positive and *awful* is negative) and also the specificities of each analysis scenario (for instance, an *increase* in the *interest rate* is probably positive for financial companies but negative for the rest).

First the local polarity of the different clauses in the text (segments) is identified based on the sentence syntactic tree and then the relation among them is evaluated in order to obtain a global polarity value for the whole given text. The detailed process may be shortly described as follows:

1. **Segment detection**. The text is parsed and split into segments, based on the presence of punctuation marks and capitalization of words.

2. **Linguistic processing**: each segment is tokenized (considering multiword units) and then each token is analyzed to extract its lemma(s). In addition, a morphosyntactic analysis divides the segment into proposition or clauses and builds the sentence syntactic tree. This division is useful, as described later, for detecting the negation and analyzing the effect of modifiers on the polarity values.

3. **Detection of negation**. The next step is to iterate over every token of each segment to tag whether the token is affected by negation or not. If a given token is affected by negation, the eventual polarity level is reversed (turns from positive to negative and the other round). For this purpose, the semantic model includes a

list of negation units, such as the obvious negation particles (adverbs) such as *not* (and contracted forms), *neither* and also expressions such as *against*, *far from*, *no room for*, etc.

4. **Detection of modifiers**. Some special units do not assign a specific polarity value but operate as modifiers of this value, incrementing or decrementing it. These units included in the semantic model can be assigned a + (positive), ++ (strong positive), - (negative) or -- (strong negative) value. For instance, if *good* is positive (P), *very good* is be strong positive (P+), thus *very* would be a positive modifier (+). Other examples are *additional*, *a lot*, *completely* (positive) or *descend*, *almost* (negative).

5. **Polarity tagging**. The next step is to detect polarity units in the segments. The semantic model assigns one of the following values, ranging from the most positive to the most negative: P++, P+, P, P-, P--, N--, N-, N, N+ and N++. Moreover, these units can include a context filter, i.e., one or several words or expressions that must appear or not in the segment so that the unit is considered in the sentiment analysis. The final value for each token is calculated from the polarity value of the unit in the semantic model, adding or subtracting the polarity value of the modifier (if thresholds are fulfilled) and considering the negation (again, if thresholds are fulfilled).

6. **Segment scoring.** To calculate the overall polarity of each segment, an aggregation algorithm is applied to the set of polarity values detected in the segment. The average of polarity values is calculated and assigned as the score of the segment, ranging from -1 (strong negative) to +1 (strong positive). In addition to this numeric score, discrete nominal values are also assigned (N+, N, NEU, P, P+). When there are no polarity units, the segment is assigned with a polarity value of NONE. The aggregation algorithm performs an outlier filtering to try to reduce the effect of wrong detections, based on a threshold over the standard deviation from the average.

7. **Global text scoring**. The same aggregation algorithm is applied to the local polarity values of each segment to calculate the global polarity value of the text, represented by an average value (both numeric and nominal values).

Although unconstrained runs were allowed to use the training corpus for improving the model, we were interested on not doing so, as we pointed out in the introduction, to compare the robustness of both models.

For the purpose of both subtasks, the provided output was adapted so that P+ and P were grouped into positive and similarly N+ and N into negative. In subtask B, the global polarity was directly used as the output, whereas in subtask A, the polarity assigned to each text fragment was the polarity value of the segment in which this text fragment is located. As compared to the constrained task, this allows a more fine-grained assignment of polarity and, expectedly, achieve a better evaluation.

Although we had different models available, some developed for specific domains such as the financial, telecommunications and tourism domains, for this task, a general-purpose model for English was used. This model was initially based on the linguistic resources provided by General Inquirer[3] in English. Some information about the model is shown in Table 1.

| Unit Type | Count |
|---|---:|
| Negation (NEG) | 31 |
| Modifiers (MOD) | 117 |
| -- | 3 |
| - | 16 |
| + | 75 |
| ++ | 23 |
| Polarity (POL) | 4 606 |
| N++ | 81 |
| N+ | 297 |
| N | 2 222 |
| N- | 221 |
| N-- | 13 |
| P-- | 6 |
| P- | 82 |
| P | 1 340 |
| P+ | 316 |
| P++ | 28 |
| Stopwords (SW) | 59 |
| Macros | 19 |
| TOTAL UNITS | 4 832 |

Table 1. English semantic model.

[3] http://www.wjh.harvard.edu/~inquirer

## 4 Results

We submitted two runs, constrained and unconstrained, for each subtask, so four runs in all. As defined by the organization, the evaluation metric was the average F-measure (averaged F-positive and F-negative, ignoring F-neutral). Separate rankings for several test dataset were also produced for comparing different scenarios.

Results achieved for runs in subtask A are shown in Table 2.

| Run | A | B | C | D | E | Avg |
|---|---|---|---|---|---|---|
| *DAEDALUS-A-constrained* | 61.0 | 63.9 | **67.4** | 61.0 | 45.3 | 59.7 |
| *DAEDALUS-A-unconstrained* | 58.7 | 56.0 | **62.0** | 58.1 | 49.2 | 56.7 |
| Average | 77.1 | 77.4 | **80.0** | 76.8 | 68.3 | 75.9 |
| *NRC-Canada-A-constrained* (best run) | 85.5 | 88.0 | **90.1** | 86.6 | 77.1 | 85.5 |

A=LiveJournal 2014, B=SMS 2013, C=Twitter 2013
D=Twitter 2014, E=Twitter 2014 Sarcasm

Table 2. Results for subtask A.

We did not specifically the contextual polarity classification in subtask A, so results are not good. The machine learning classifier achieved a slightly better result on average for all test corpus than the lexicon-based model, as expected, about a 5% improvement. As compared to other participants, we rank the second-to-last group (19 out of 20) and our best experiment is 27% below the average, and 43% below the best run.

The best test set for our experiments is the Twitter 2013 corpus, as it is the most similar to the training corpus. If Twitter 2014 Sarcasm corpus is removed from the evaluation, which clearly is the most difficult set for all runs, the constrained run is only 22% below the average and 38% below the best run, so a relative improvement against the others.

| Run | A | B | C | D | E | Avg |
|---|---|---|---|---|---|---|
| *DAEDALUS-B-constrained* | 40.8 | **40.9** | 36.6 | 33.0 | 29.0 | 36.1 |
| *DAEDALUS-B-unconstrained* | **61.0** | 55.0 | 59.0 | 57.6 | 35.2 | 53.6 |
| Average | **63.5** | 55.6 | 59.8 | 60.4 | 45.4 | 57.0 |
| *TeamX-B-constrained* (best run) | 69.4 | 57.4 | **72.1** | 71.0 | 56.5 | 65.3 |

A=LiveJournal 2014, B=SMS 2013, C=Twitter 2013
D=Twitter 2014, E=Twitter 2014 Sarcasm

Table 3. Results for subtask B.

On the other hand, results achieved for runs in subtask B are shown in Table 3. The subtask was a bit more difficult than the first one, and results are in general worse than in the first subtask, as more difficult aspects arise in the global polarity assignment, such as the appearance of coordinated or subordinated clauses or a higher impact of negation.

We think that the specific consideration of these issues is the main reason why in this case our best run is the lexicon-based model, with an improvement of 48 % over the constrained run.

Also results are more robust as they are more consistent for the different test sets. The best results are achieved for the LiveJournal 2014 corpus, which presumably contains longer texts with more formal writing corpus, so benefiting with the use of the advanced linguistic preprocessing.

Comparing to other participants, we rank 29 out of 42 groups, and our best experiment is just 6% below the average, and 22% below the best run. If, again, the worst set, the Twitter 2014 Sarcasm corpus, is removed from the evaluation, our unconstrained run is around the average (2% below), and, a bit surprisingly, the best group changes to the one that submitted the best run in subtask A, and our experiment is just 23% below (comparing to 38% below in subtask A).

## 5 Conclusions and Future Work

Our main conclusion after our first participation in SemEval is that, although results are not good compared to the best ranked participants, our lexicon-based model, externally developed for a generic sentiment analysis task, without any adaptation to the provided training corpus, and currently in production, is robust and achieves a balanced evaluation result for message polarity along the different test corpus analyzed. Despite of the difficulty of the task, results are valuable and validate the fact that this technology is ready to be included into an automated workflow process for social media mining.

Due to lack of time, no error analysis has been carried out yet by studying the confusion matrix for the different categories, which is left as short-term future work. We expect to get a better understanding of the miss classifications of our system and find a way to solve the issues that may arise. Probably there is still much to do in both the enlargement of the semantic resources and also the improvement of the linguistic processing (specially building the sentence syntactic tree) in a general domain for a non-formal writing style.

## Acknowledgements

This work has been supported by several Spanish R&D projects: *Ciudad2020: Hacia un nuevo modelo de ciudad inteligente sostenible* (INNPRONTA IPT-20111006), MA2VICMR: *Improving the Access, Analysis and Visibility of Multilingual and Multimedia Information in Web* (S2009/TIC-1542) and MULTIMEDICA: *Multilingual Information Extraction in Health Domain and Application to Scientific and Informative Documents* (TIN2010-20644-C03-01).

## References

CLEF. 2014. CLEF Initiative (Conference and Labs of the Evaluation Forum). http://www.clef-initiative.eu/

NTCIR. 2014. NII Testbeds and Community for Information Access Research. http://research.nii.ac.jp/ntcir/

Julio Villena-Román, Sara Lana-Serrano and José C. González-Cristóbal. 2008. MIRACLE at NTCIR-7 MOAT: First Experiments on Multilingual Opinion Analysis. 7th NTCIR Workshop Meeting. Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access. Tokyo, Japan, December 2008.

Julio Villena-Román, Sara Lana-Serrano, Cristina Moreno-García, Janine García-Morera, José Carlos González-Cristóbal. 2012. DAEDALUS at RepLab 2012: Polarity Classification and Filtering on Twitter Data. CLEF 2012 Labs and Workshop Notebook Papers. Rome, Italy, September 2012.

Daedalus. 2014. http://www.daedalus.es/

G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing, Communications of the ACM, v.18 n.11, p.613-620, Nov. 1975.

Textalytics. 2014. Meaning as a service. http://textalytics.com/home.

Textalytics Parser API. 2014. Lemmatization, PoS and Parsing v1.2. http://textalytics.com/core/parser-info

Textalytics Sentiment API. 2014. Sentiment Analysis v1.1. http://textalytics.com/core/sentiment-info

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. 2009. The WEKA Data Mining Software: An Update. SIGKDD Explorations, Volume 11, Issue 1.