

# ***Minions* at SemEval-2016 Task 4: or how to build a sentiment analyzer using off-the-shelf resources?**

**Călin-Cristian Ciubotariu, Marius-Valentin Hrișca, Mihail Gliga, Diana Darabană,  
Diana Trandabăț and Adrian Iftene**

University “Alexandru Ioan Cuza” of Iași, Romania

{calin.ciubotariu, marius.hrisca, mihail.gliga, diana.darabana,  
diana.trandabat, adiftene}@info.uaic.ro

## **Abstract**

*Minions*, a team formed of first year students in the Master of Computational Linguistics, started the participation at Semeval-2016 as a semester project, aiming to build a model for analyzing and classifying “tweets” into positive, neutral and negative, according to the evoked sentiment, while getting familiar with Natural Language Processing tools and methods. Therefore, the backbone of our sentiment analyzer consists in several off-the-shelf, freely available resources, enhanced with a classifier trained on the SemEval-2016 data.

## **1 Introduction**

Texts live around us just as we live around them. At any instant, there are texts that people write, share, use to get informed, etc. (starting with an advertisement heard on the radio every morning and finishing with the contract of sale signed before a notary). Combining this with the concept of language economy (or the principle of least effort) – a tendency shared by all humans – consisting in minimizing the needed amount of effort to achieve the maximum result, it is no wonder why the social media, with its short, informal and context-dependent texts, achieved such a high popularity.

SemEval-2016 task 4 had several subtasks, but since our team consists mainly of members beginning to learn about Natural Language Processing, we only felt comfortable in participating in Subtask A. This subtask involved the classification of a message polarity, i.e. classify a given

tweet in three categories: positive, negative or neutral, according to the identified sentiment (Nakov et al., 2016).

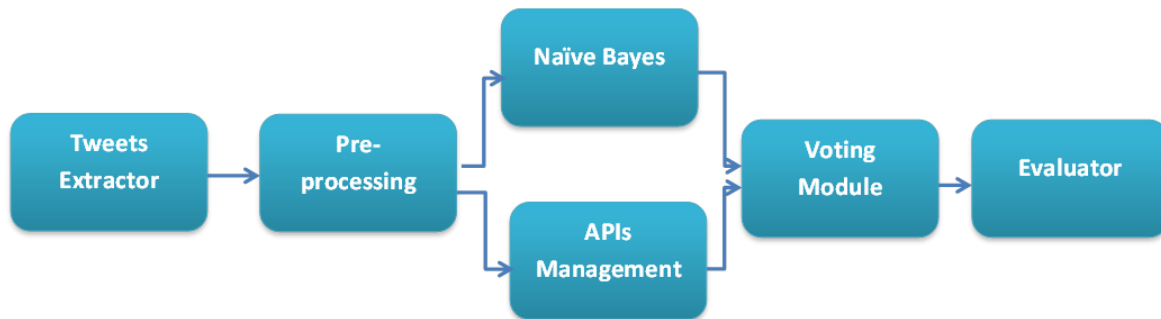
The remaining of this paper is structured as follows: Section 2 provides an overview of available online applications for analyzing social media sentiments; Section 3 presents our own sentiment analyzer, before the final section presenting the evaluation of the system and drawing some conclusions.

## **2 State of the art**

Specific processing tools (such as POS taggers or anaphora resolution systems), score a higher performance if used on the same text type as the ones they were trained on. In other words, we will have better results if using a POS tagger trained on news corpora to analyze news texts, rather than speech transcripts.

Thus, the short dimension of tweets and their creative informal spelling have raised a new set of challenges to the natural language processing field. How to handle such challenges while automatically mining and understanding the opinions and sentiments that people are communicating has been the subject of several research (Jansen et al., 2009; Barbosa and Feng, 2010; Bifet and Frank, 2010; Davidov et al., 2010; O’Connor et al., 2010; Pak and Paroubek, 2010; Tumasjan et al., 2010; Kouloumpis et al., 2011).

We have investigated existing online applications for sentiment extraction of social media from Twitter, briefly discussed below, and integrated some of them in our sentiment analyzer.



**Figure 1.** Diagram of our tweet sentiment analyzer

Trackur<sup>1</sup> is an online application, allowing the display of opinions on a particular search criterion, trained on datasets from various social networks such as Facebook, Google Plus, Instagram, etc.

Social Mention<sup>2</sup> application monitors over 100 social networks, blogs or forums such as Twitter, Facebook, Youtube, Digg, Google, etc. in an attempt to identify emerging hot topics.

AlchemyAPI<sup>3</sup> (Turian, 2013), provider of artificial intelligence cloud services, offers multiple modes of sentiment analysis: document-level, entity-level, and keyword-level sentiment mining is provided, in addition to support for advanced features such as negation handling, sentiment amplifiers / diminishers, slang, and typos, all based on the company's deep-learning classifier, trained on an impressive social media corpus.

Sentiment140<sup>4</sup> (formerly known as "Twitter Sentiment") allows the discovery of the sentiment associated to a brand, product, or topic on Twitter. It uses a maximum entropy classifier, trained on a set of automatically extracted tweets. The training corpus of 1.600.000 tweets was created relying solely on the use of emoticons (tweets with happy smileys suggest a positive contents, while tweets with sad/anger smileys refer to negative contents). The API lets users classify tweets and integrate sentiment analysis functionality into their own websites or applications, using RESTful calls and responses formatted in JSON.

NLTK<sup>5</sup> (Bird et al., 2009) is a platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources (including WordNet), along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, etc.

After analyzing the available applications for sentiment analysis, we decided to build our own analyzer based on several existing services (Alchemy, Sentiment 140 and NLTK's Sentiment analyzer), enhanced with a Naïve Bayes classifier trained on the SemEval-2016 data.

### 3 System Architecture

The system developed for SemEval-2016 can be divided in the following modules:

1. **Tweets Extractor:** a module which extracts tweets based on a list of ID's provided by the task organizers;
2. **Pre-processing:** cleaning operations needed to remove from tweets symbols unsupported by the sentiment analysis services;
3. **APIs Management:** development of a web service able to manage the calls to the three sentiment analysis APIs: Alchemy, Sentiment 140, NLTK;
4. **Naïve Bayes Classifier:** trains a Naïve Bayes classifier from the NLTK toolkit for identification of positive, negative and neutral sentiments.

<sup>1</sup> <http://www.trackur.com/about-trackur>

<sup>2</sup> <http://socialmention.com/>

<sup>3</sup> <http://www.alchemyapi.com/>

<sup>4</sup> <http://help.sentiment140.com/home>

<sup>5</sup> <http://www.nltk.org/>

5. **Voting Module** receives sentiment scores from the three APIs and decides, in case of mismatch, which one to further obey;
6. **Evaluator**: Analysis the output file and creates statistics used to improve the system's performances.

The system's architecture is presented in figure 1 above and the modules are discussed in more details in the next subsections.

### 3.1 Tweet Extractor

This module receives as input a set of tweets IDs and extracts the text of tweets using Twitter API. One of the challenges with regard to this module was to overcome the limitations set by the Twitter API (a limit of 100 tweets in a response for any request). Therefore, the tweet extractor has a parameter that allows us to specify the frequency of crawling. We found that an interval of 2 minutes is a reasonable polling parameter.

Several tweet IDs returned errors when processing, the content of the tweets being no longer available. For example, for the train data, out of the 6000 ID's, 632 were not found.

### 3.2 Pre-processing

After obtaining the texts from tweets, a cleaning phase was performed, in order to standardize the data. Thus, regular expressions have been built to: convert the texts to lowercase, discard words shorter than two characters, remove special diacritic signs, URLs, as well as symbols unsupported by the sentiment analyzers' APIs (such as "?"). Users often include Twitter usernames in their tweets in order to direct their messages, using the @ symbol before the username (e.g. @radut), therefore a regex replaces all words that start with the @ symbol. Another modification proved to significantly reduce feature space, inspired by (Pang et al., 2002), is removing duplicated vowels in the middle of the words (e.g. coooooool). Any letter occurring more than two times in a row is replaced with exactly two occurrences.

### 3.3 APIs Management

This module was intended to manage the calls to the sentiment analysis APIs used in this project:

Alchemy, Sentiment 140 and NLTK's Sentiment analyzer.

### 3.4 Naïve Bayes Classifier

Similar to (Go et al, 2009 and Pang et al., 2002), we trained a Naïve Bayes classifier, using the NLTK's training facility, with the following features: tokenized unigrams, emoticons, hashtags. We used the train and development data made available by the SemEval-2016 organizers for training.

Ultimately, our internal evaluation on test development data from the SemEval-2016 competition revealed the fact that our Naïve Bayes classifier was introducing more errors than correct cases, most probably due to a bug. We therefore introduced a parameter allowing us to run the system with a customized series of analyzers. For the submitted runs, we only considered the outputs of the three sentiment analysis APIs.

### 3.5 Voting Module and Evaluator

These modules are used to analyze the output of the sentiment analysis APIs aiming to identify, in case different labels are issued, which sentiment analyzer is most reliable. The Evaluator outputs a set of statistics using the test development data provided by the SemEval-2016 organizers.

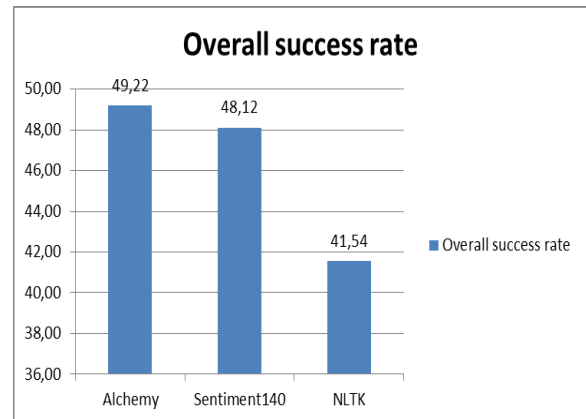


Figure 2. Comparison between the three sentiment identification services

Thus, this module checks how many agreements/disagreements are found in the results offered by different sentiment analyzers (see figure 2 for an overview and figure 3 for a detailed analysis of matched labels). We found that in

only 14.9% of the cases, all three services gave the same good result. For 9.4% of the cases, the three services gave similar label, but failed to find the good one. Out of these situations, almost 14% were mislabeled negative cases, and only 1.5% mislabeled positive tweets.

However, in 30.6% of the cases, two of the services gave the same label, the good one and in 78% of cases at least one classifier gave the right answer.

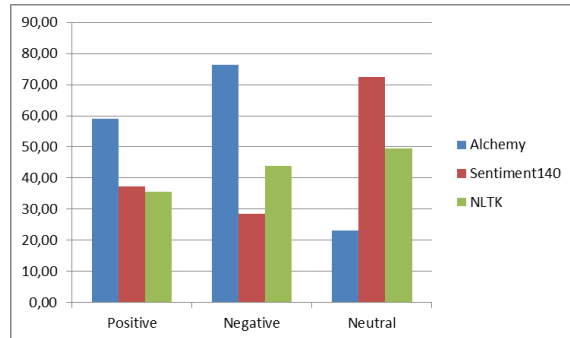


Figure 3. Percentage of correctly identified sentiments on SemEval-2016 training data

These analyses lead us to the decision to implement a simple voting module, which is based on the following empirically derived rules:

- If at least two services give the same label, this label is chosen;
- Otherwise, based on the internal evaluation (see figure 3), the priority was given as follows:
  - o if Alchemy gives a negative result, select it;
  - o else if Sentiment140 gave a neutral result, select it;
  - o otherwise, if Alchemy gave a positive result, select it,
  - o otherwise select the neutral label.

#### 4 Official evaluation and discussions

The official evaluation (Nakov et al., 2016), presented in table 1, placed our system on the 28th place (out of 34 places).

Minions	2013		2014			2015 tweet	2016 Tweet
	Tweet	SMS	Tweet	Tweet sarcasm	Live Journal		
	0.48	0.52	0.55	0.42	0.47	0.48	0.41

Table 1 Official results for the Minions team

In this version of the system, we did not use part-of-speeches, since initial tests showed that in this configuration, they bring more noise than relevant information, conclusion shared (for part-of-speeches) also by (Pang et al., 2002). However, as further improvements, we intend to lemmatize the tweets before feeding them to our classifier, and use an external dictionary of sentiment valences in the voting module, to enhance our system's performance.

#### References

Barbosa, L. and Feng, J. (2010). *Robust sentiment detection on twitter from biased and noisy data*. Proceedings of Coling 2010 .

Bifet, A. and Frank, E. (2010). *Sentiment knowledge discovery in twitter streaming data*. Proceedings of 14th Int. Conference on Discovery Science.

Bird Steven, Klein E., Loper E., (2009) *Natural Language Processing with Python*, O'Reilly Media.

Davidov, D., Tsur, O., Rappoport, A. (2010). *Enhanced sentiment learning using twitter hashtags and smileys*. Proceedings of Coling 2010.

Go A., Bhayani R., Huang. L. (2009) Twitter Sentiment Classification using Distant Supervision, *Technical Report*.

Jansen, B.J., Zhang, M., Sobel, K., Chowdury, A. (2009). *Twitter power: Tweets as electronic word of mouth*. Journal of the American Society for Information Science and Technology 60(11):2169-2188.

Kouloumpis, E., Wilson, T., and Moore, J. (2011). *Twitter Sentiment Analysis: The Good the Bad and the OMG!* Proceedings of ICWSM. 2011

Nakov P., Ritter A., Rosenthal S., Stoyanov V., Sebastiani F.(2016) *SemEval-2016 Task 4: Sentiment Analysis in Twitter*, Proc. of SemEval '16.

O'Connor, B., Balasubramanian, R., Routledge, B., and Smith, N. 2010. From tweets to polls: Linking text sentiment to public opinion time series. Proceedings of ICWSM.

Pak, A. and Paroubek, P. 2010. *Twitter as a corpus for sentiment analysis and opinion mining*. Proceedings of LREC 2010.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. (2002) "Thumbs up?: sentiment classification using machine learning techniques." Proceedings of EMNLP, pp. 79-86.

Tumasjan, A., Sprenger, T.O., Sandner, P., Welpe, I. (2010). *Predicting elections with twitter: What 140 characters reveal about political sentiment*. Proceedings of ICWSM 2010.

Turian Joseph, (2013) *Using AlchemyAPI for Enterprise-Grade Text Analysis*, PhD Thesis.