# Automatic Unsupervised Polarity Detection on a Twitter Data Stream

Diego Terrana, Agnese Augello, Giovanni Pilato
ICAR - Istituto di Calcolo e Reti ad Alte Prestazioni
CNR - Consiglio Nazionale delle Ricerche
Viale delle Scienze - Edicio 11 - 90128 Palermo, Italy
Email: {terrana, augello, pilato}@pa.icar.cnr.it

***Abstract*—In this paper we propose a simple and completely automatic methodology for analyzing sentiment of users in Twitter. Firstly, we built a Twitter corpus by grouping tweets expressing positive and negative polarity through a completely automatic procedure by using only emoticons in tweets. Then, we have built a simple sentiment classifier where an actual stream of tweets from Twitter is processed and its content classified as positive, negative or neutral. The classification is made without the use of any pre-defined polarity lexicon. The lexicon is automatically inferred from the streaming of tweets. Experimental results show that our method reduces human intervention and, consequently, the cost of the whole classification process. We observe that our simple system captures polarity distinctions matching reasonably well the classification done by human judges.**

***Index Terms*—Sentiment Analysis, Text Classification, Twitter, Opinion Mining, Polarity.**

## I. Introduction

Social networks have become an interesting repository of information; they represent popular platforms where users can discuss a wide variety of topics with other people. Users posts can be analysed to obtain useful information, for example they can represent a useful source for psychologists and marketers interested in the extraction and mining of opinions and attitudes about events, news and products. Compared to blogs, microblogging users upload messages more frequently and instantly. One of the most representative microblogging examples is Twitter [1]. Twitter users generate about 200 million tweets (short messages of up to 140 characters) per day. Textual information in tweets can be divided in two main categories: *facts* and *opinions*. Facts are objective news about entities and events while opinions reflect people's sentiments. Sentiment and Opinion mining methodologies, broadly used for the analysis of news and products reviews, recently have also been employed to process tweets, even if the nature of these data lead to an high level of difficulty in identifying the sentiments expressed in tweets. In fact, while reviews are longer, more verbose and well formed, people in writing tweets frequently use shortened forms, slangs and abbreviations, due the limitation of 140 characters. Another difference is that reviews usually focus on a entity while tweets often are ambiguous. Tweets often contain significant spelling errors and a sentiment is conveyed in one or two sentence passages which are rather informal. Moreover, a considerably large fraction of tweets convey no sentiment but advertisements and links to news articles, which provide some difficulties in data gathering, training and testing.

In this paper, we propose a totally unsupervised methodology for tweets sentiment analysis. The methodology is based on two main phases: an automatic generation of a training dataset and a polarity classification phase. The proposed methodology does not use any external resources such as lexicons, dictionaries or other manually tagged datasets but it uses only the sentiment expressed by emoticons in the training dataset as a source of information. Emoticons are graphic representations of users moods, obtained by proper combinations of characters. When web users use an emoticon, they are effectively marking up their own text with an emotional state. Considering as "positive" and "negative" the tweets containing respectively positive and negative emoticons, the proposed approach is based on the assumption that a word appearing frequently in positive and rarely in negative tweets, should have a high polarity positive score and analogously that a word appearing frequently in negative rather than in positive tweets, should have a low polarity score. Therefore, according to this assumption the method performs a classification of tweets analysing their composing words. We use this corpus to create a polarity dictionary to recognize positive, neutral and negative posts without using any standard classifier, language translators or manually tagged texts. Since the approach is unsupervised is adaptable for other languages without any manual intervention.

The remaining part of the paper is organized as follows: Section 2 illustrates a brief overview of related works in literature. Sections 3 and 4 describe the proposed system. Section 5 presents the experimental results. Finally, section 6 specifies possible future directions for this study.

## II. Related WorkS

Generally, the approaches proposed in literature rely on the use of polarity and sentiment lexicons containing lists of words with semantic orientations. As an example, Bracewell et al. [2] propose a system to create an emotion dictionary and judge the emotion of a news article based on emotion word, idiom and modier. The dictionary is semi-automatically created by using WordNet, a standard English thesaurus and a set of articles manually tagged with emotion, phrases

978-1-4799-4003-5/14 $31.00 © 2014 IEEE
DOI 10.1109/ICSC.2014.17

and idioms. They calculate an emotion score subtracting the number of negative emotion words from the number of positive emotion words to give an emotion score.
On the other end, both Maarten et al. [3] and Fellbaum et al. [4] used WordNet [5] for measuring semantic orientations of adjectives.
Furthermore, Strapparava et al. [6] presented WordNet-Affect[6] while Esuli et al. [7] presented SentiWordNet as an additional sentiment resource.
In [8], the authors developed techniques that algorithmically identifying large number of adjectives, each one with a score of polarity, starting around a dozen of seed adjectives. Their methods enlarge positive and negative clusters of adjectives by recursively querying the synonyms and antonyms from WordNet.
Many of the resources used in literature are available in English and they are mostly limited to a small set of terms and very expensive to build and extend, since the work of expert is required. However it is worthwhile to highlight that the online community is very creative in coining new terms with semantic orientation which could be not considered in the lexicons.
Other approaches exploit annotated data set, where the annotation is made mostly manually. Roberts et al. [9] introduce a corpus of manually annotated $tweets$ with seven emotions: $anger$, $disgust$, $fear$, $joy$, $lov$, $sadness$ and $surprise$. They use the annotated corpus to train a classier that automatically discovers the emotions in tweets.
Pak and Paroubek [10] collect a corpus for sentiment analysis and opinion mining purposes using Twitter API. They query Twitter for two types of emoticons: happy and sad emoticons. The two types of collected corpora are then used to train a multinomial Naïve Bayes classier to recognize positive and negative sentiments. They query accounts of $44$ newspapers to collect a training set of objective texts.
Agarwal et al. [11] acquire 11,875 manually annotated Twitter data from a commercial source and use Google translate to convert it into English. Each tweet is labeled by a human annotator as positive, negative, neutral or junk. For obtaining the prior polarity of words, they use $Dictionary\ of\ Affect\ in\ Language\ (DAL)$.
Nasukawa et al. [12] combine semantic analysis with a syntactic parser at statement level to capture opposite sentiments in the same expressions. They include a manually defined sentiment lexicon and use a Markov-modal based tagger for recognizing part of speech to identify sentiments related to subject.
Zhang et al. [13] combine emoticons, negation word position, and domain-specific words.
These approaches are limited to specific domains and the process is very time-consuming, subjective and reducing its real-time applicability to big data. The work presented in this paper is similar but it follows a different approach: the tweets are collected in streaming and therefore represent a true sample of actual tweets in terms of language use and content.

## III. Tweets Polarity Evaluation

A simple methodology for the evaluation of tweets polarities is presented. The approach exploits the presence of emoticons into tweets. We assume that the presence of an emoticon is a clue about the sentiment that the microblogger wants to express in the post. This assumption is strengthened by the consideration that, due the limitation of 140 characters per tweet, the emotion expressed by that emoticon, may generally suggest the sentiment of the whole text in the tweet. Starting from this observation we illustrate an automatic procedure for the creation of a lexical resource. The procedure does not require any kind of tagged datasets or dictionaries, but it only requires the process of tweets containing emoticons. The advantage is that in this manner we are able to map and enrich common expressions with newly created words, slangs, grammatical errors.
Our system catches polarity distinctions and classifies tweet messages according to their emotional content as being positive, negative or neutral.
In a training phase (Figure 1), we automatically generate a training dataset without using external lexicons, dictionaries or any other manually tagged documents.
Tweets messages containing emoticons are retrieved by using the Twitter APIs ("$Data\ Collection\ phase$") and then they are grouped into two sets: positive set and negative set, containing positive and negative emoticons respectively ("$Positive/Negative\ Emoticon\ Class\ Creation\ phase$"). The tweets belonging to each one of the sets are then selected according to a specific language ("$Language\ Recognition\ phase$") and then they are pre-processed ("$Preporcessing\ phase$"). A polarity score is assigned to each word of these tweets ("$Word\ Polarity\ Calculation\ phase$").
We assign:

- a high positive score to a word if it frequently appears in the positive set and rarely, at the same time, in the negative set;
- a low polarity score if the word appears more often in the negative set rather than in the positive.

The created dictionary of words is subsequently used in the production phase (Figure 2) for the detection of emotional content in a generic twitter stream. We assign a polarity score to each tweet as the average of the sum of the polarity scores of its words ("$Tweet\ Polarity\ Calculation\ phase$"). If the polarity of a tweet exceeds a given positive threshold value we classify it as positive; if its score is below a negative threshold value we classify it as negative, otherwise it is considered being neutral. The following sections describe each one of the steps in detail.

### A. Dictionary of polarized words

To obtain our dictionary of polarized words, we use emoticons to select messages and associate a sentiment mood

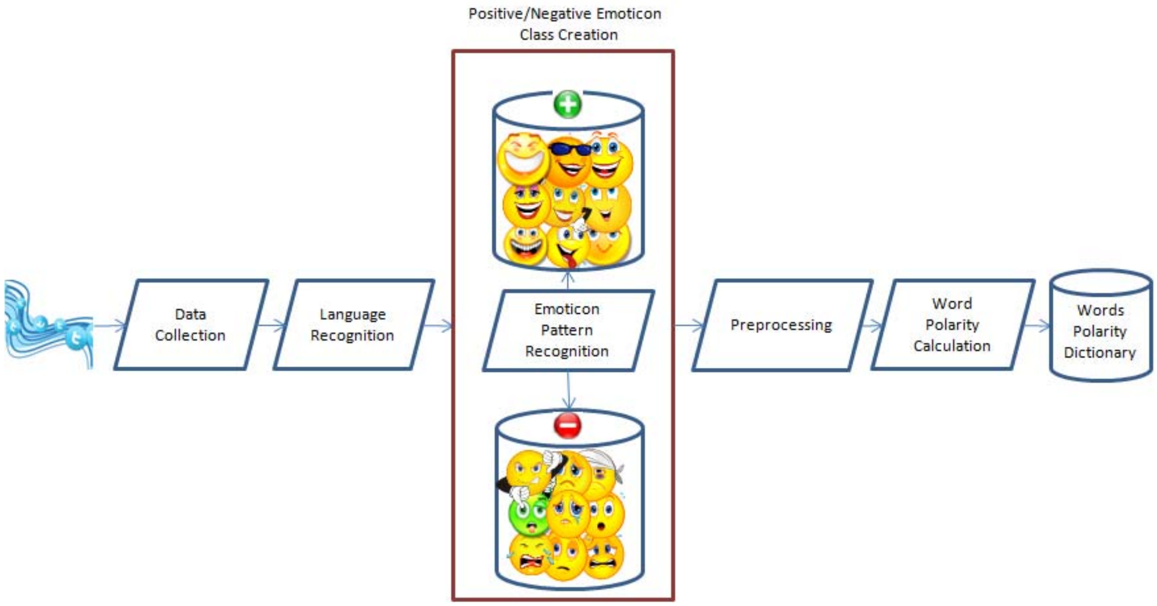


Fig. 1: Sentiment Analysis Procedure - Training Phase

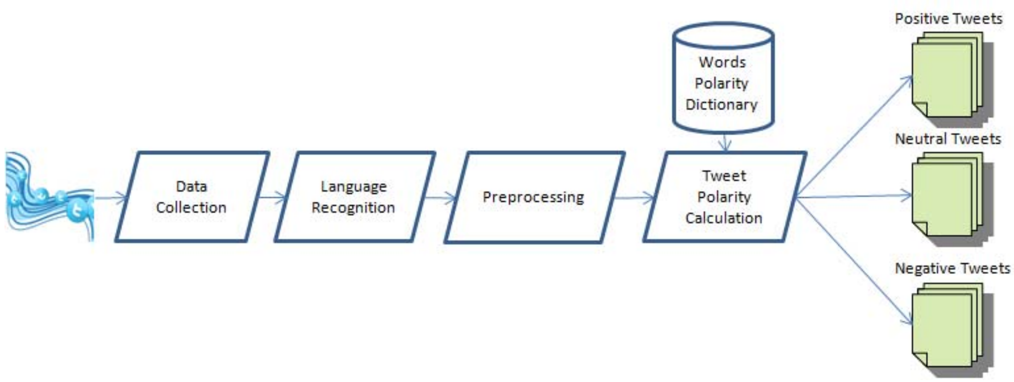


Fig. 2: Sentiment Analysis Procedure - Production Phase

<table>
<tr><th>ICON</th><th>MEANING</th></tr>
<tr><td>:-) =) :) 8) :] =] => 8-) :-> :-] :”) :’)</td><td>smiley</td></tr>
<tr><td>:3 :> :^) :-3 => :-> :-V =v :-1</td><td>happy face</td></tr>
<tr><td>^_^ ^L^ ^)^</td><td>happiness</td></tr>
<tr><td>:* :-*</td><td>kiss, couple kissing</td></tr>
<tr><td>;-) ;) ;-] ;] ;-> ;> %-}</td><td>wink, smirk</td></tr>
<tr><td><3</td><td>heart</td></tr>
<tr><td>:-D :D =D :-P =3 xD</td><td>laughing</td></tr>
<tr><td>:P =P</td><td>tongue sticking out, playful</td></tr>
<tr><td>O.o o.O</td><td>surprised</td></tr>
<tr><td>:v</td><td>gape</td></tr>
<tr><td>B) B-) B| 8|</td><td>feel cool</td></tr>
<tr><td>:’-)</td><td>tears of happiness</td></tr>
<tr><td>:!</td><td>exclamation</td></tr>
<tr><td>:-X</td><td>Sealed lips, wearing braces</td></tr>
<tr><td>=* :-* :*</td><td>kiss</td></tr>
</table>

TABLE I: Examples of Positive Emoticon Icons with its meanings.

| *ICON* | *MEANING* |
|---|---|
| -_- | boredom, sleep, sarcasm |
| :-( :( :[ :-< :-[ =( :-@ :-& :-t :-z :<) }-( | angry, sadness, sulky expression, contrariness |
| :o :O :-o :-O | amazement, speechless |
| :-\ :/ :-/ :-. :\ | troubled, annoyed, embarrassed |
| :'( :,( :'-( :,-( :˜(˜˜ :˜-( | cry |

TABLE II: Examples of Negative Emoticon Icons with its meanings.

within a message. Since each tweet cannot be longer than 140 characters we can conjecture that the emoticon represents the sentiment of the whole message [10]. This is true except for a very few number of cases (ironic or sarcastic tweets) which are difficult to classify even for a human expert. Thus, we state that if a word occurs more frequently in one class, it is more strongly related to the according emotion.
We exploit a set of emoticons classified as being positive or negative. $Table\ I$ and $Table\ II$ show samples of positive and negative emoticons. As a consequence, we split the tweets messages into two class:

- positive class: tweets containing positive emoticons;
- negative class: tweets containing negative emoticons.

### B. Word Polarity Calculation

For each word $w_i$, we count the number of occurrences in positive and negative emoticons sets and we propose the following formula to estimate the polarity score of it:

$$polarity(w) = \frac{occ_{+}(w_i) - occ_{-}(w_i)}{occ_{+}(w_i) + occ_{-}(w_i)} \tag{1}$$

where
$occ_{+}(w_i)$ = word occurrences in the positive class.
$occ_{-}(w_i)$ = word occurrences in the negative class.

The polarity value of a word is a value between $-1$ and $1$ ($-1$ means strongly negative, $0$ means neutral, $1$ means strongly positive).
In our system, a word has a high polarity score if it frequently appears in the positive class and rarely, at the same time, in the negative class. On the contrary, a word that appears more often in the negative class rather than in the positive one, has a low polarity score. The list of all the polarized words automatically extracted from the training corpus constitutes the $opinion\ words$ of the dictionary to be used in the classification step of a new tweet. Since tweets from Twitter usually contain $noisy\ text$, i.e. text that does not comply having the standard rules of orthography, syntax and semantics, we filter out not frequent words with less than $k$ characters, where $k$ is an integer experimentally fixed to $k = 3$.

### C. Tweet Polarity Calculation

The polarity score of a tweet is given by the average of the sum of the polarity scores of its words. We consider all the words as sentiment indicators also verbs and nouns in addition to adjectives. Given a tweet message $m$ which can be regarded as a collection of $n$ words $w_1, w_2, ..., w_n$, we define its polarity score as the mean of polarity scores of all the terms having more than $k$ characters:

$$polarity(m) = \frac{\sum_{i=1}^{N} polarity(w_i)}{N} \tag{2}$$

where
$w_i$ = is an opinion word.
$polarity(w_i)$ = is the polarity score of a term $w_i$ calculated by using the constructed affective lexicon in the previous step according to the positive and negative tweets in the training corpora.
$N$ = is the number of words in the tweet $s$.

Sentiment classification of message $m$ is obtained exploiting the $polarity(m)$ as value:

- If $polarity(m) \geq polarityAvg + \epsilon$ then the text is considered to have a positive polarity;
- If $polarity(m) < polarityAvg + \epsilon$ and $polarity(m) > polarityAvg - \epsilon$ then the text is considered neutral;
- If $polarity(m) \leq polarityAvg - \epsilon$ then the text is considered to have a negative polarity.

where
$polarityAvg$ is the mean polarity of all the words in the lexicon.
$\epsilon$ is the threshold value experimentally calculated. It defines the polarity of the neutral tweets.

## IV. Implementation of the Proposed Procedure

### A. Data Collection

The dataset object of analysis has been retrieved by using the Twitter APIs (statuses/samplemethod) with the $default\ access$ level. The default access level ($'Spritzer'$) returns a random sample of all public tweets [14]. This level of access provides a small proportion of all public tweets (1%). The Twitter APIs provide two other levels of access the $'Firehose'$ (100%) and $'Gardenhose'$ (10%) using special account. The twitter4j [15] JAVA library has been used to capture tweets and make queries in real time. The returned data is a set of documents, one for each tweet. We stored tweets into text files according to the Java Script Object Notation (JSON) format.
Each retrieved tweet, in addition to the plain text of the tweet contains many other fields, including $userid$, $date$, $source$, $type$, $profile$, $location$, $number\ of\ favorites$, $friends$, $followers$, $URL$, $hashtag$, etc.. The JSON tweet document contains attributes describing the tweet, user information,

tweet relations with other tweets, a lists of urls, hashtags and user ids contained in the tweet. In some cases information related to the location of the user is also provided in the document. Given the average number of 170 million tweets sent per day [16] it is expected that the size of the data retrieved by the Streaming APIs (1%), in a 24 hour will be around 1.700.000 tweets.

### B. Language Recognition

The stream of incoming tweets is filtered for language. The language has been checked for every incoming tweet and we have considered only the tweets of a specific language. For the correct recognition of the language of a tweet the Cybozu library[1] was used. That library detects language of a text using naive Bayesian filter with 99% over precision for 53 languages. In our tests, the recognition accuracy is not always proved to be so high because of the limited number of characters in a tweet, not more than 140 characters. In particular, the problem has been reported for the Latin languages such as Spanish, Portuguese, Romanian, Italian. Therefore, we exploit both the recognized language and the declared nationality of the Twitter user in order to be sure that the examined tweet is expressed in language of interest.

### C. The Pre-processing

The tweet text content is preprocessed to extract the relevant content and leave out the irrelevant ones. We first deleted duplicate tweets like some re-tweets ("$RT$") and the tweets containing mainly abnormal sequences of characters (i.e. "?? $2/14$ $8$ : $09$ ?? $57$ ???????????? $+$ $36$?? $countkun$ @$mnw$_031869"). Tokenization has been used to identify all words in a given text. Stop words have been filtered out in order to eliminate those words that occur too frequently, such as articles, prepositions, conjunctions. Meta information and special characters have been excluded from processing. We captured a set of tweet patterns using regular expressions, link tokens (starting with "$http://$", "$https://$", and "$www.$" ), hashtag tokens(starting with "#" character) user identifier tokens (starting with "@" character) and emoticons. All of them are removed before processing the text.

## V. Experimental Results

In our experiments, we have created a large dataset[2] of tweets using Twitter search API from December 2012 to June 2013, to be used in training and testing of our system. We have collected more than 500 million tweets partitioned into documents of ten thousand lines within folders, one for each day of scanning.

Each file contains one row for each tweet in JSON (JavaScript Object Notation) format, which stores in addition to text other information such as date, type, source, user profile, location, followers number, friends number, favorites number. All the information is stored by using a key / value encoding for each data.

Some data such as location and profile information have not considered in this approach and they will be used in future work.

Tests were carried out by analyzing tweets written in Italian language. We have selected only tweets which contain emoticons. To capture several types of emoticons, we have manually built a set of regular expressions.

$Table\ III$ lists the number of tweets containing the main emoticon type. We have split these tweets in a positive and a negative class with respectively $2132116$ and $31516$ tweets. The positive and negative retrieved corpora are the real proportions of positive and negative tweets found on twitter and they are used to set up the classification parameters with the aim of recognizing positive and negative sentiments. The created dictionary of words, contains 98445 polarity terms with more than $K = 3$ characters. To validate our approach, we calculated the mean polarity ($polarityAvg = 0.70189$). For data testing, we randomly collected $1000$ tweets from 10 October 2013 to 11 October 2013 manually annotated by three independent experts. The value to be assigned to each tweet is one out of positive, neutral, or negative. The neutral value includes both objective statements or equally positive and negative at the same time. As shown in $Table\ IV$, the annotators detected more than half of the tweets as being neutral. To measure the performance of our sentiment classifier the accuracy of the classifier was calculated. It represents the success rate of the classification and corresponds to the number of correct classifications divided by the number of tweets:

$$accuracy(class_i, \epsilon) = \frac{numCorr(class_i, \epsilon)}{numTot(class_i)} \tag{3}$$

where

$\epsilon$ = is the threshold value experimentally calculated.

$accuracy(class_i, \epsilon)$ = is the accuracy for $class_i$ with $i$ = positive, negative or neutral.

$numCorr(class_i, \epsilon)$ = is the number of tweets of $class_i$ correctly labeled by the system.

$numTot(class_i)$ = is the total number of tweets of $class_i$ in a test set.

The classification of each individual tweet has been compared with its manually assigned tag. In $Table\ V$, we report the best scores obtained. The $3rd$ column represents the best accuracy for, respectively, positive, negative and neutral class. We also report the label accuracy for all three classes simultaneously. Figure 3 shows the trend of accuracy function of $\epsilon$. The number of tweets correctly tagged with each class is represented as a percentage. We have obtained an accuracy of $97.67\%$ for neutral tweets, $86,26\%$ for negative tweets and $76,06\%$ for positive tweets. This different can be explained considering that the icons are associated with the user's feeling but he/she can use sarcasm and irony to express it.

[1]http://developer.cybozu.co.jp/oss/2013/01/16/internet-2012-innumbers/

[2]The dataset will be available at http://lithium.pa.icar.cnr.it/twitter_dataset/.

| *ICON* | *COUNT* | *ICON* | *COUNT* | *ICON* | *COUNT* | *ICON* | *COUNT* |
|---|---|---|---|---|---|---|---|
| :) | 75133 | :( | 10319 | :* | 3560 | :-* | 1045 |
| <3 | 43005 | :') | 7063 | :-D | 1620 | :-P | 832 |
| :-) | 22570 | ;-) | 6817 | :'( | 1615 | -_- | 446 |
| ;) | 21636 | :P | 5453 | =) | 1303 | =P | 370 |
| :3 | 20053 | :O | 4995 | ^_^ | 1297 | :-/ | 309 |
| :D | 17559 | :/ | 4462 | :-( | 1290 | =( | 158 |

TABLE III: Number of Italian tweets retrieved for the main emoticons.

| *CLASS* | *NUMBER* |
|---|---|
| Positive | 188 |
| Negative | 211 |
| Neutral | 601 |

TABLE IV: Test dataset manually annotated by three experts.

| *CLASS* | *EPSILON* | *ACCURACY* |
|---|---|---|
| Positive | 0.0 | 0.761 |
| Negative | 0.0 | 0.863 |
| Neutral | 0.28 | 0.977 |
| ALL | 0.13 | 0.723 |

TABLE V: Test dataset manually annotated by three experts.

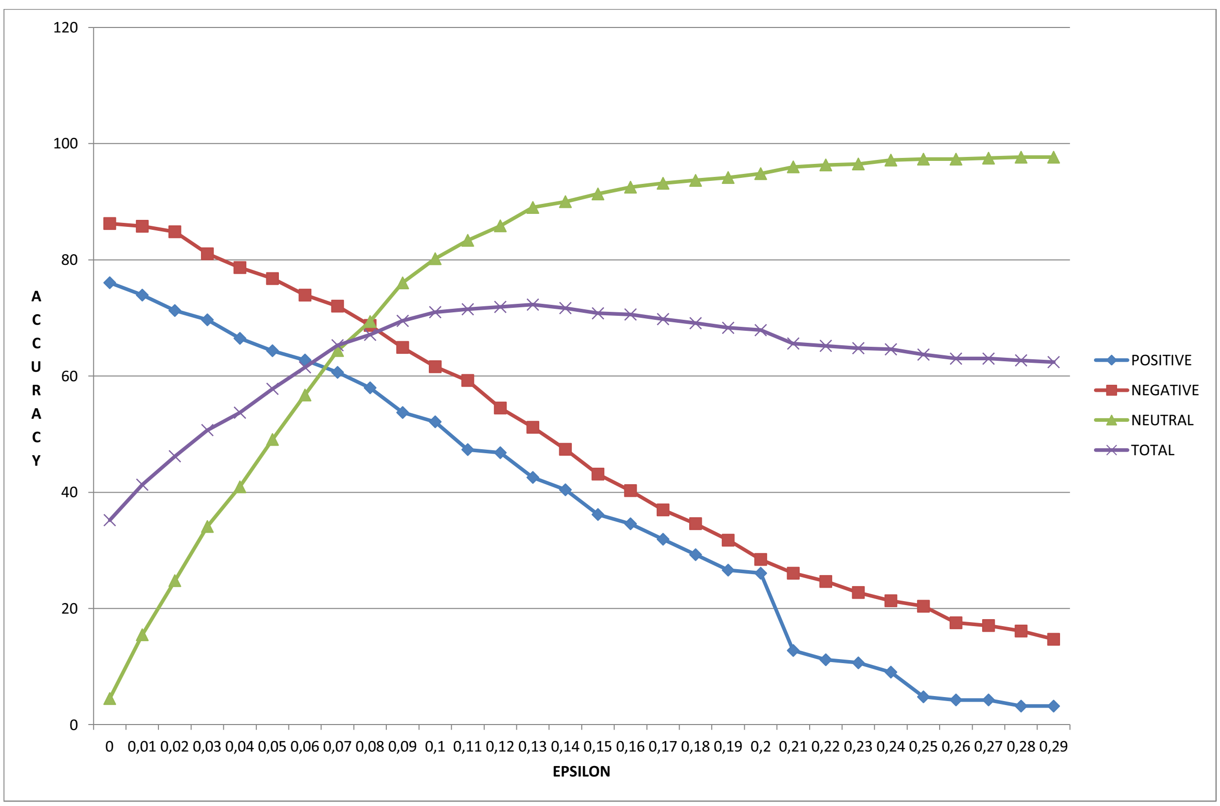


Fig. 3: Trend of accuracy function of epsilon

## VI. Conclusions and Future Work

In this paper, we have discussed a method to automatically analyze sentiment of users in Twitter. The sentiment analysis of Twitter data is significantly different from other sentiment classification on structured text. We have shown that is not necessary to use external dictionaries of polarized words to capture the sentiment polarity in everyday colloquial tweets. Our method reduces at minimum human intervention and it is able to automatically generate a training dataset by inferring sentiment present in tweets containing emoticons. It is able to map colloquial expressions with new words, slangs and errors. Experimental tests showed reasonably results in polarity distinctions matching, without using external limited resources such as lexicons or dictionaries. Our system can be applied to any language. As a future work, we plan to explore the phenomenon of irony or sarcasm and to build a model to identify it automatically.

## VII. Acknowledgements

This work has been partially supported by the PON01_01687 - SINTESYS (Security and INTElligence SYSstem) Research Project.

## References

[1] "Twitter for business - resources available at:https://business.twitter.com/."

[2] D. B. Bracewell, J. Minato, F. Ren, and S. Kuroiwa, "Determining the emotion of news articles." in *ICIC (2)*, ser. Lecture Notes in Computer Science, D.-S. Huang, K. Li, and G. W. Irwin, Eds., vol. 4114. Springer, 2006, pp. 918–923. [Online]. Available: http://dblp.uni-trier.de/db/conf/icic/icic2006-2.htmlBracewellMRK06

[3] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," 1997, pp. 174–181.

[4] J. Kamps, M. Marx, R. J. Mokken, and M. D. Rijke, "Using wordnet to measure semantic orientation of adjectives," in *National Institute for*, 2004, pp. 1115–1118.

[5] "C. fellbaum, ed., wordnet. an electronic lexical database. the mit press, 1998."

[6] R. Valitutti, "Wordnet-affect: an affective extension of wordnet," in *In Proceedings of the 4th International Conference on Language Resources and Evaluation*, 2004, pp. 1083–1086.

[7] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06*, 2006, pp. 417–422.

[8] N. Godbole, M. Srinivasaiah, and S. Skiena, "Large-scale sentiment analysis for news and blogs," in *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, 2007.

[9] K. Roberts, M. A. Roach, J. Johnson, J. Guthrie, and S. M. Harabagiu, "Empatweet: Annotating and detecting emotions on twitter." in *LREC*, N. Calzolari, K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, Eds. European Language Resources Association (ELRA), 2012, pp. 3806–3813. [Online]. Available: http://dblp.uni-trier.de/db/conf/lrec/lrec2012.htmlRobertsRJGH12

[10] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, N. C. C. Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, Eds. Valletta, Malta: European Language Resources Association (ELRA), may 2010.

[11] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in *Proceedings of the Workshop on Languages in Social Media*, ser. LSM '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 30–38. [Online]. Available: http://dl.acm.org/citation.cfm?id=2021109.2021114

[12] T. Nasukawa and J. Yi, "Sentiment analysis: capturing favorability using natural language processing." in *K-CAP*, J. H. Gennari, B. W. Porter, and Y. Gil, Eds. ACM, 2003, pp. 70–77.

[13] K. Zhang, Y. Cheng, Y. Xie, D. Honbo, A. Agrawal, D. Palsetia, K. Lee, W. keng Liao, and A. N. Choudhary, "Ses: Sentiment elicitation system for social media data," in *ICDM Workshops*, 2011, pp. 129–136.

[14] "Twitter developers: Streaming api methods - resources available at:https://dev.twitter.com/docs/streaming-api/methods/."

[15] "Twitter4j - a java library for the twitter api - resources available at:http://twitter4j.org/en/index.html/."

[16] "Internet 2012 numbers - resources available at:http://royal.pingdom.com/2013/01/16/internet-2012-in-numbers/."