A Probabilistic Approach to Tweets' Sentiment Classification

Conference Paper · September 2013

DOI: 10.1109/ACII.2013.13

3 authors:

F. Colace
Università degli Studi di Salerno

Massimo De Santo
Università degli Studi di Salerno

Luca Greco
Università degli Studi di Salerno

2013 Humaine Association Conference on Affective Computing and Intelligent Interaction

A probabilistic approach to Tweets' Sentiment Classification

Francesco Colace, *Member, IEEE,* Massimo De Santo, *Member, IEEE,* and Luca Greco

Abstract—Prior to 2003, mankind generated a total of about 5 Exabyte's of contents. Now, we generate this amount of contents in about two days! The spread of generic (as Twitter, Facebook or Google+) or specialized (as LinkedIn or Viadeo) social networks allows sharing opinions on different aspects of life every day. Therefore this information is a rich source of data for opinion mining and sentiment analysis. This paper introduces a novel approach to the sentiment analysis based on the Weighted Word Pairs obtained by the use of the Latent Dirichlet Allocation (LDA) approach. The proposed methodology aims at identifying a word-based graphical model for depicting and mining a positive or negative attitude towards a topic. For the evaluation of the proposed approach a challenging scenario has been set: the real-time analysis of tweets. The experimental evaluation shows how the proposed approach is effective and satisfactory.

Index Terms—Sentiment Analysis, Latent Dirichlet Allocation, Information Extraction Management.

1 Introduction

Daily millions of messages appear on the web thanks to blogs, microblogs, social networks or review collector sites. This textual information can be divided in two main categories: facts and opinions [17]. Facts are objective statements while opinions reflect people's sentiments about products, personalities and events. The latter are extremely important for helping someone to take a decision [16]. The interest, that potential customers show in online opinions and reviews about products, is something that vendors are gradually paying more and more attention to. In this scenario, a promising approach is *sentiment analysis* - the computational study of opinions, sentiments and emotions expressed in a text [10]. Its main aim is the identification of agreement or disagreement statements to capture positive or negative feelings in comments or reviews. In this paper, we investigate the adoption of an approach to sentiment analysis based on the Latent Dirichlet Allocation (LDA). In LDA, each document may be viewed as composed by a mixture of various topics. This is similar to probabilistic latent semantic analysis (pLSA), except that in LDA the topic distribution is assumed to have a Dirichlet prior. By the use of the LDA approach on a set of documents belonging to a same knowledge domain, a *Mixed Graph of Terms* can be automatically extracted [11] [5]. Such a graph contains a set of weighted word pairs, which we demonstrate to be discriminative for sentiment classification. The rationale of this paper is the following: in section 2 related works are discussed; section 3 discusses the extraction of a Mixed Graphs of Terms from a document corpus and their sentiment discriminative power. Section 4 introduces the proposed approach for the sentiment extraction while section 5 discusses the experimental results.

• *F. Colace, M. De Santo and L. Greco are with the Department of Information Technology and Electric Engineering of the University of Salerno, Fisciano, Italy*
E-mail: {fcolace, desanto, lgreco}@unisa.it

2 Related Work

In literature, there are many approaches to the sentiment analysis [13][19][4]. In particular, some approaches attempt to classify the sentiment at a document level. In [6] authors introduce an approach based on the algebraic sum of the orientation terms (positive or negative) for document classification. Starting from this approach other techniques have been developed [20]. Baroni [2] proposed to rank a large list of adjectives according to a subjectivity score by employing a small set of manually selected adjectives and computing the mutual information of pairs of them using frequency and co-occurrence frequency counts on the web. Starting from this approach many researchers developed "sentiment" lexicon. The work of Turney [18] proposes an approach to measure the semantic orientation of a given word based on the strength of its association with a set of context insensitive positive words minus the strength of its association with a set of negative words. By this approach sentiment lexicon can be built and a sentiment polarity score can be assigned to each word [12][7]. Artificial intelligence and probabilistic approaches have been adopted for the sentiment mining. In [14] three machine learning approaches (Naive Bayes, Maximum Entropy and Support Vector Machines) have been adopted to label the polarity of movie reviews. A promising approach has been developed in [15] where a novel methodology has been obtained by the combination of rule based classification, supervised learning and machine learning. Another interesting approach is in [21] where a probabilistic model, the Sentiment Probabilistic Latent Semantic Analysis (S-PLSA), has been adopted [9]. The

978-0-7695-5048-0/13 $26.00 © 2013 IEEE
DOI 10.1109/ACII.2013.13

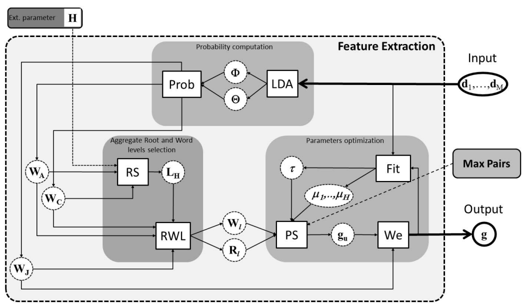

Fig. 1. Proposed mGT extraction method.

S-PLSA is an extension of the PLSA where it is assumed that there are a set of hidden semantic factors or aspects in the documents related to each other according to a probabilistic framework.

3 Extracting a Mixed Graph of Terms

The aim of this section is to explain how a Mixed Graph of Terms (mGT) can be extracted from a corpus of documents. The extraction process is shown in Fig. 1. The input of the system is the set of documents $\Omega_r = (\mathbf{d}_1, \cdots, \mathbf{d}_M)$ and the output is a vector of weighted word pairs $\mathbf{g} = \{w'_1, \cdots, w'_{|\mathcal{T}_p|}\}$, where $\mathcal{T}_p$ is the number of pairs and w'_n is the weight associated to each pair (feature) $t_n = (v_i, v_j)$.

The mGT is made of several clusters, each containing a set of words v_s (*aggregates*) related to an *aggregate root* (r_i), a special word which represents the centroid of the cluster. The weight ρ_{is} can measure how a word is related to an aggregate root and can be expressed as a probability: $\rho_{is} = P(r_i|v_s)$. The resulting structure is a subgraph rooted on r_i. Moreover, *aggregate roots* can be linked together building a centroids subgraph. The weight ψ_{ij} can be considered as the degree of correlation between two aggregate roots and can also be expressed as a probability: $\psi_{ij} = P(r_i, r_j)$.

Given the training set Ω_r of documents, the term extraction procedure is obtained first by computing all the probabilistic relations between words and aggregate roots (ρ_{is} and ψ_{ij}), and then selecting the right subset of pairs $\mathcal{T}_p$ from all the possible ones.

A mGT graph $\mathbf{g}$ is learnt from a corpus of documents as a result of two important phases: the *Relations Learning* stage, where graph relation weights are learnt by computing probabilities between word pairs (see Fig. 1); the *Structure Learning* stage, where the shape of an initial mGT graph, composed by all possible aggregate root and word levels, is optimized by performing an iterative procedure which, given the number of aggregate roots H and the desired max number of pairs as constraints, chooses the best parameter settings τ and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_H)$ defined as follows:

1) τ: the threshold that establishes the number of *aggregate root/aggregate root* pairs of the graph. A relationship between the aggregate root v_i and aggregate root r_j is relevant if $\psi_{ij} \geq \tau$.
2) μ_i: the threshold that establishes, for each aggregate root i, the number of *aggregate root/word* pairs of the graph. A relationship between the word v_s and the aggregate root r_i is relevant if $\rho_{is} \geq \mu_i$.

3.1 Relations Learning

Since each aggregate root is lexically represented by a word of the vocabulary, we can write $\rho_{is} = P(r_i|v_s) = P(v_i|v_s)$, and $\psi_{ij} = P(r_i, r_j) = P(v_i, v_j)$. Considering that $P(v_i, v_j) = P(v_i|v_j)P(v_j)$, all the relations between words result from the computation of the joint or the conditional probability $\forall i, j \in \{1, \cdots, |\mathcal{T}|\}$ (where $|\mathcal{T}|$ is the size of the vocabulary $\mathcal{T}$ which contains all the indexed words from the corpus) and $P(v_j)$ $\forall j$. An exact calculation of $P(v_j)$ and an approximation of the joint, or conditional, probability can be obtained through a smoothed version of the generative model introduced in [3] called Latent Dirichlet Allocation (LDA), which makes use of Gibbs sampling [8]. The output obtained by performing Gibbs sampling on a set of documents Ω_r consists of two matrices:

1) the *words-topics* matrix that contains $|\mathcal{T}| \times K$ elements representing the probability that a word v_i of the vocabulary is assigned to topic k: $P(u = v_i|z = k, \beta_k)$;
2) the *topics-documents* matrix that contains $K \times |\Omega_r|$ elements representing the probability that a topic k is assigned to some word token within a document $\mathbf{d}_m$: $P(z = k|\theta_m)$.

The probability distribution of a word within a document $\mathbf{d}_m$ of the corpus can be then obtained as:

$$P(u_m) = \sum_{k=1}^{K} P(u_m|z = k, \beta_k)P(z = k|\theta_m). \qquad (1)$$

In the same way, the joint probability between two words u_m and y_m of a document $\mathbf{d}_m$ of the corpus can be obtained by assuming that each pair of words is represented in terms of a set of topics z and then:

$$P(u_m, y_m) = \sum_{k=1}^{K} P(u_m, y_m|z = k, \beta_k)P(z = k|\theta_m) \qquad (2)$$

Note that the exact calculation of Eq. 2 depends on the exact calculation of $P(u_m, y_m|z = k, \beta_k)$ that cannot be directly obtained through LDA. If we assume that words in a document are conditionally independent given a topic, an approximation for Eq. 2 can be written as:

$$P(u_m, y_m) \simeq \sum_{k=1}^{K} P(u_m|z = k, \beta_k)P(y_m|z = k, \beta_k)P(z = k|\theta_m). \qquad (3)$$

Moreover, Eq. 1 gives the probability distribution of a word u_m within a document $\mathbf{d}_m$ of the corpus. To obtain the probability distribution of a word u independently of the document we need to sum over the entire corpus:

$$P(u) = \sum_{m=1}^{M} P(u_m)\delta_m \quad (4)$$

where δ_m is the prior probability for each document ($\sum_{m=1}^{|\Omega r|} \delta_m = 1$). In the same way, if we consider the joint probability distribution of two words u and y, we obtain:

$$P(u, y) = \sum_{m=1}^{M} P(u_m, y_v)\delta_m \quad (5)$$

Concluding, once we have $P(u)$ and $P(u, y)$ we can compute $P(v_i) = P(u = v_i)$ and $P(v_i, v_j) = P(u = v_i, y = v_j)$, $\forall i, j \in \{1, \cdots, |\mathcal{T}|\}$ and so the relations learning can be totally accomplished.

3.2 Structure Learning

Once each ψ_{ij} and ρ_{is} is known $\forall i, j, s$, aggregate root and word levels have to be identified in order to build a starting mGT structure to be optimized as discussed later. The first step is to select from the words of the indexed corpus a set of aggregate roots $\mathbf{r} = (r_1, \ldots, r_H)$, which will be the nodes of the centroids subgraph. Aggregate roots are meant to be the words whose occurrence is most implied by the occurrence of other words of the corpus, so they can be chosen as follows:

$$r_i = \text{argmax}_{v_i} \prod_{j \neq i} P(v_i | v_j) \quad (6)$$

Since relationships' strengths between aggregate roots can be directly obtained from ψ_{ij}, the centroids subgraph can be easily determined. Note that not all possible relationships between aggregate roots are relevant: the threshold τ can be used as a free parameter for optimization purposes. As discussed before, several words (aggregates) can be related to each aggregate root, obtaining H aggregates' subgraphs. The threshold set $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_H)$ can be used to select the number of relevant pairs for each aggregates' subgraph. Note that a relationship between the word v_s and the aggregate root r_i is relevant if $\rho_{is} \geq \mu_i$, but the value ρ_{is} cannot be directly used to express relationships' strengths between aggregate roots and words. In fact, being ρ_{is} a conditional probability, it is always bigger than ψ_{is} which is a joint probability. Therefore, once pairs for the aggregates' subgraph are selected using ρ_{is}, relationships' strength are represented on the mGT structure through ψ_{is}.

Given H and the maximum number of pairs as constraints (i.e. fixed by the user), several mGT structure $\mathbf{g}_t$ can be obtained by varying the parameters $\Lambda_t = (\tau, \mu)_t$. As shown in Fig.1, an optimization phase is carried out in order to search the set of parameters Λ_t which produces the best mGT graph. This process relies on a scoring function and a searching strategy that will be now explained.

As we have previously seen, a $\mathbf{g}_t$ is a vector of features $\mathbf{g}_t = \{b_{1t}, \ldots, b_{|\mathcal{T}_{sp}|t}\}$ in the space $\mathcal{T}_{sp}$ and each document of the training set Ω_r can be represented as a vector $\mathbf{d}_m = (w_{1m}, \ldots, w_{|\mathcal{T}_{sp}|m})$ in the space $\mathcal{T}_{sp}$. A possible scoring function is the cosine similarity between these two vectors:

$$\mathcal{S}(\mathbf{g}_t, \mathbf{d}_m) = \frac{\sum_{n=1}^{|\mathcal{T}_{sp}|} b_{nt} \cdot w_{nm}}{\sqrt{\sum_{n=1}^{|\mathcal{T}_{sp}|} b_{nt}^2} \cdot \sqrt{\sum_{n=1}^{|\mathcal{T}_{sp}|} w_{nm}^2}} \quad (7)$$

and thus the optimization procedure would consist in searching for the best set of parameters Λ_t such that the cosine similarity is maximized $\forall \mathbf{d}_m$. Therefore, the best $\mathbf{g}_t$ for the set of documents Ω_r is the one that produces the maximum score attainable for each document when used to rank Ω_r documents. Since a score for each document $\mathbf{d}_m$ is obtained, we have:

$$\mathbf{S}_t = \{\mathcal{S}(\mathbf{g}_t, \mathbf{d}_1), \cdots, \mathcal{S}(\mathbf{g}_t, \mathbf{d}_{|\Omega_r|})\},$$

where each score depends on the specific set $\Lambda_t = (\tau, \mu)_t$. To compute the best value of Λ we can maximize the score value for each document, which means that we are looking for the graph which best describes each document of the repository from which it has been learned. It should be noted that such an optimization maximizes at the same time all $|\Omega_r|$ elements of $\mathbf{S}_t$. Alternatively, in order to reduce the number of the objectives being optimized, we can at the same time maximize the mean value of the scores and minimize their standard deviation, which turns a multi-objective problem into a two-objective one. Additionally, the latter problem can be reformulated by means of a linear combination of its objectives, thus obtaining a single objective function, i.e., *Fitness* ($\mathcal{F}$), which depends on Λ_t,

$$\mathcal{F}(\Lambda_t) = E[\mathbf{S}_t] - \sigma[\mathbf{S}_t],$$

where E is the mean value of all the elements of $\mathbf{S}_t$ and σ_m is the standard deviation. By summing up, the parameters learning procedure is represented as follows,

$$\Lambda^* = \text{argmax}_t\{\mathcal{F}(\Lambda_t)\} \quad (8)$$

Since the space of possible solutions could grow exponentially, a clustering method, that is the *K-means* algorithm, has been applied to all ψ_{ij} and ρ_{is} values, so that the optimum solution can be exactly obtained after the exploration of the entire space.

4 Searching the Sentiment by the use of the Mixed Graph of Terms

As described in the previous section, a Mixed Graph of Terms gives a compact representation of a set of documents related to a well-defined knowledge domain. In this way the obtained graph can be considered as a filter to be employed in document classification problems. The main aim of this paper is to show how mGT

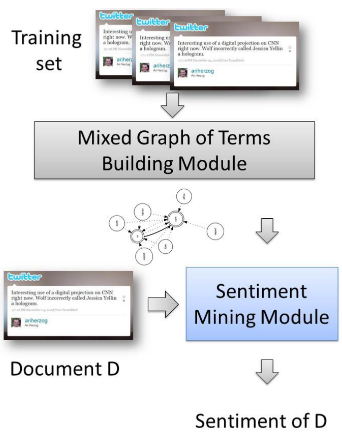

Fig. 2. The Sentiment Analysis Classification System Architecture.

can be effectively applied for sentiment mining from texts: the proposed method can be used to build a sentiment detector able to label a document according its sentiment. Our system is composed by the following modules:

- *Mixed Graph of Terms building module*: this module builds a mixed graph of terms starting from a set of documents belonging to a well-defined knowledge domain and previously labeled according the sentiment expressed in them. In this way the obtained mixed graph of terms contains information about the words and their co-occurrences so representing a certain sentiment in a well-defined knowledge domain. As described in section 2 thanks to the LDA approach such a graph can be obtained by the use of a set of few documents. In figure 2 the module architecture and its main functional steps are depicted. The output of this module is a mixed graph of terms representing the documents and their sentiment. By feeding this module with positive or negative training sets, it will be possible to build mixed graphs of terms for documents that express positive or negative sentiment in a well-defined domain.
- *Sentiment Mining Module*: this module extracts the sentiment from a document thanks to the use of the Mixed Graph of Term as a sentiment filter. The input of this module is a generic document, the mixed graph of terms representing positive and negative sentiment in a knowledge domain and the output is the sentiment detected in the input document.

The sentiment extraction is obtained by a comparison between document and the mixed graph of terms according to algorithm 1.

The proposed algorithm requires the use of an annotated lexicon, as for example WordNet [1] or ItalWordNet, for the retrieval of synonyms of the words contained

Algorithm 1 Sentiment Mining Algorithm

Input: $W = [w_1, w_2, \ldots, w_n]$ the words that are in a Document D belonging a knowledge domain K; the mixed graph of terms $mGT+$ and $mGT-$ obtained analyzing documents related to the knowledge domain K expressing positive and negative sentiment; $RW+ = [rw_1, rw_2, \ldots, rw_t]$ the aggregator words that are in $mGT+$; $AW+ = [aw_1, aw_2, \ldots, aw_m]$ the aggregated words that are in $mGT+$; $RW- = [rw_1, rw_2, \ldots, rw_n]$ the aggregator words that are in $mGT-$; $AW- = [aw_1, aw_2, \ldots, rwp]$ the aggregated words that are in mGT, L an annotated lexicon.
Output: $Sentiment_D = \{Positive, Negative, Neutral\}$ the sentiment expressed in the document D.
Algorithm Description
$fp = 0$;
$fn = 0$;
Determining the synonyms for each word belonging to the vector W
for $i = 0 \rightarrow Length[W]$ **do**
  $WS = WS + Synset[L, W[i]]$;
end for
$W = W + WS$;
Mining the sentiment from the document
for $i = 0 \rightarrow Length[W]$ **do**
  for $k = 0 \rightarrow Length[RW_+]$ **do**
    if ($RW_+[k] == W[i]$) **then**
      $f_p = f_p + 2$;
    end if
  end for
  for $k = 0 \rightarrow Length[RW_-]$ **do**
    if ($RW_-[k] == W[i]$) **then**
      $vf_n = f_n + 2$;
    end if
  end for
  for $k = 0 \rightarrow Length[AW_+]$ **do**
    if ($AW_+[k] == W[i]$) **then**
      $f_p = f_p + 1$;
    end if
  end for
  for $k = 0 \rightarrow Length[AW_-]$ **do**
    if ($AW_-[k] == W[i]$) **then**
      $f_n = f_n + 1$;
    end if
  end for
end for
Determining the Sentiment
if ($f_p > f_n$) **then**
  $Sentiment_D = Positive$;
else
  if ($f_p < f_n$) **then**
    $Sentiment_D = Negative$;
  else
    $Sentiment_D = Neutral$;
  end if
end if

in the document D and not included in the reference mGT. The retrieved synonyms are added to the vector W and analyzed according to the classification strategy. The proposed approach is effective in an asynchronous sentiment classification, but can work also in a synchronous way. In figure 3 the synchronous sentiment real time classificatory architecture is depicted. For real time working two new modules have been introduced:

- *Document Grabber*. This module aims to collect documents from web sources (social networks, blogs and so on). These documents can be collected both for updating the training set and for their classification according to the sentiment. The training set update is an important feature of the proposed approach. In this way, in fact, the various mGTs can be contin-

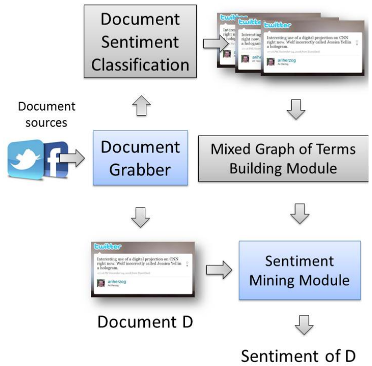

Fig. 3. System Architecture for Synchronous Classification.

Reference Paper	Methodology	Accuracy
[14]	Support Vector Machines	82.90%
	Naïve Bayes	81.50%
	Maximum Entropy	81.00%
mGT Approach	LDA	88.50%

TABLE 1
The accuracy obtained by the various methods on the considered standard dataset.

Total Tweets	36165
Average Accuracy	87.1 %

TABLE 2
Obtained results for Tweets' analysis.

uously updated and improve their discriminating power introducing new words and relations and deleting inconsistent ones.

- *Document Sentiment Classification*. The new documents inserted into the training set have to be classified by the support of an expert. The aim of this module is to provide a user friendly environment for the classification, according to their sentiment, of the retrieved documents.

5 Experimental Results

The evaluation of the proposed method has been conducted through two steps. Firstly the proposed approach has been applied on a standard dataset: the Movie Reviews Dataset [14]. The main aim of this experimentation was to evaluate method's performance and make a comparison with the other approaches well known in literature. The experimentation has been conducted considering the 25% of the dataset as training set and the remaining 75% as test set. The obtained results and their comparison with other approaches are depicted in table 1. From the table 1 it can be observed that the proposed approach shows the best results from the point of view of accuracy.

The second experimental phase has been carried out using a *real life* dataset. The experimental scenario involved the analysis of posts coming from the popular microblog environment Twitter. These posts, also known as tweets, are characterized by a short length (about 140 characters) and can be easily grouped considering their "hashtags" that define the topics of the discussions. In particular, "hashtags" related to the various Italian politic parties were collected, starting one month before the last administrative election (February the 24th and the 25th, 2013). During the first week, tweets were collected for building the training set . In particular, a group of ten users labeled the various tweets according to positive, negative and neutral sentiment. A majority rule evaluation approach has been adopted. Using the training set (composed by about 4000 tweets) reference mGTs were built and the classification task started. The grabber module of the system started to collect the tweets from the various selected "hashtags" and, in real time, the classifier module classified them according to the proposed algorithm. Each day the same group of the ten users, that built the training set, evaluated the classification results for a real evaluation of the accuracy of the system. The daily accuracy of the system, for each Italian party involved in the election, has been reported on a web-site[1]. Every two days mGTs were updated by adding new correctly classified post to the training set. So, the mGTs could continuously improve their classification power. During the three weeks about 35,000 tweets have been classified and in figure 5 the results obtained in terms of daily accuracy are displayed. Table 2 shows the average accuracy. The obtained results show that the system improved its performance after updating mGTs. We expect this improvement since more relevant terms are added to the initial graph as a result of the updating process. In figure 5 we also notice some performance reduction at certain points: it is due to the unsupervised learning approach, where ambiguous words (which could be considered both positive and negative) are also taken into account. Thanks to a continous update, such noise can be controlled and eventually reduced.

6 Conclusion

This paper proposes the use of the mixed graph of terms, obtained by the use of Latent Dirchlet Allocation

1. http://193.205.190.209/elezioni2013/

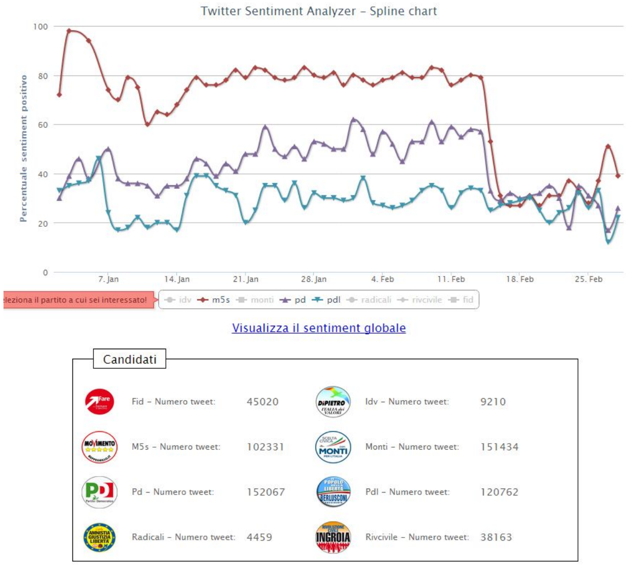

Fig. 4. Twitter Sentiment Analyzer snapshot.

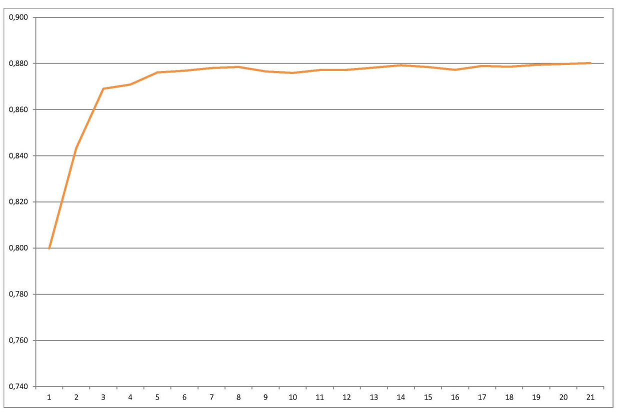

Fig. 5. Daily accuracy of the system.

approach, as tool for the sentiment classification of documents. The method relies on building the reference mGTs from documents labeled according their sentiment. The classification of a document can be conducted by using the reference mGTs. The proposed method was compared to the main methods in literature using standard and real datasets. In both cases the obtained results are better than those obtained by other approaches. Further development of this approach will include the introduction of annotated lexicon, as SentiWordnet, for a better evaluation of the words and the sentence structures.

REFERENCES

[1] Wordnet - http://wordnet.princeton.edu/.

[2] Marco Baroni and Stefano Vegnaduzzo. Identifying subjective adjectives through web-based mutual information. In *In Proceedings of the 7th Konferenz zur Verarbeitung Natrlicher Sprache (German Conference on Natural Language Processing KONVENS04*, pages 613–619, 2004.

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[4] E. Cambria, Y. Song, H. Wang, and N. Howard. Semantic multidimensional scaling for open-domain sentiment analysis, 2013.

[5] F. Clarizia, F. Colace, M. De Santo, L. Greco, and P. Napoletano. Mixed graph of terms for query expansion. In *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on*, pages 581–586, 2011.

[6] Richard Colbaugh and Kristin Glass. Estimating sentiment orientation in social media for intelligence monitoring and analysis. In *Intelligence and Security Informatics (ISI), 2010 IEEE International Conference on*, pages 135 –137, may 2010.

[7] Michael Gamon and Anthony Aue. Automatic identification of sentiment vocabulary: Exploiting low association with known sentiment terms. In *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing*, pages 57–64, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[8] Thomas L. Griffiths, Joshua B. Tenenbaum, and Mark Steyvers. Topics in semantic representation. *Psychological Review*, 114:2007, 2007.

[9] Thomas Hofmann. Probabilistic latent semantic analysis. In *In Proc. of Uncertainty in Artificial Intelligence, UAI99*, pages 289–296, 1999.

[10] Bing Liu. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing, Second Edition. Taylor and Francis Group, Boca*, 2010.

[11] P. Napoletano, F. Colace, M. De Santo, and L. Greco. Text classification using a graph of terms. In *Complex, Intelligent and Software Intensive Systems (CISIS), 2012 Sixth International Conference on*, pages 1030–1035, July.

[12] A. Neviarouskaya, H. Prendinger, and M. Ishizuka. Sentiful: A lexicon for sentiment analysis. *Affective Computing, IEEE Transactions on*, 2(1):22 –36, jan.-june 2011.

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

[14] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86, 2002.

[15] Rudy Prabowo and Mike Thelwall. Sentiment analysis: A combined approach. *Journal of Informetrics*, 3:143–157, 2009.

[16] Fabrizio Sebastiani and Consiglio Nazionale Delle Ricerche. Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1–47, 2002.

[17] Khin Phyu Phyu Shein. Ontology based combined approach for sentiment classification. In *Proceedings of the 3rd International Conference on Communications and information technology*, CIT'09, pages 112–115, Stevens Point, Wisconsin, USA, 2009. World Scientific and Engineering Academy and Society (WSEAS).

[18] P. Turney and M. Littman. Unsupervised learning of semantic orientation from a hundred-billion-word corpus. Technical report nrc technical report erb-1094, Institute for Information Technology, National Research Council Canada, 2002.

[19] Hao Wang, Dogan Can, Abe Kazemzadeh, François Bar, and Shrikanth Narayanan. A system for real-time twitter sentiment analysis of 2012 u.s. presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, pages 115–120, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[20] Theresa Wilson, Janyce Wiebe, and Rebecca Hwa. Just how mad are you? finding strong and weak opinion clauses. In *Proceedings of the 19th national conference on Artifical intelligence*, AAAI'04, pages 761–767. AAAI Press, 2004.

[21] Xiaohui Yu, Yang Liu, Xiangji Huang, and Aijun An. Mining online reviews for predicting sales performance: A case study in the movie domain. *IEEE Trans. on Knowl. and Data Eng.*, 24(4):720–734, April 2012.