A Latent Dirichlet Allocation Approach using Mixed Graph of Terms for Sentiment Analysis

Mario Casillo †, Fabio Clarizia ‡, Francesco Colace ‡, Massimo De Santo ‡, Marco Lombardi ‡, Francesco Pascale ‡
† University of Naples "Federico II", Italy
mario.casillo@unina.it
‡ DIIn, University of Salerno, Italy
{fcolace — desanto — malombardi — fpascale}@unisa.it

Abstract

The spread of generic (as Twitter, Facebook or Google+) or specialized (as LinkedIn or Viadeo) social networks allows to millions of users to share 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 presents a novel approach to the sentiment analysis based on the Latent Dirichlet Allocation (LDA) approach. The proposed methodology aims to identify a word-based graphical model (we call it a mixed graph of terms) for depicting a positive or negative attitude towards a topic. By the use of this model it will be possible to automatically mine from documents positive and negative sentiments. Experimental evaluation, on standard and real datasets, shows that the proposed approach is effective and furnishes good and reliable results.

1. Introduction

In the 2010 Eric Schmidt, CEO of Google, said: Between the birth of the world and 2003, there were five exabytes of information. Now, we create five exabytes every two days. See why its so painful to operate in information markets?. Millions of messages appear daily thanks to blogs, microblogs, social networks or reviews collector sites. In general, this textual information can be divided in two main categories: facts and opinions [22]. Facts are objective statements while opinions reflect peoples sentiments about products, other person and events and are extremely important when someone needs to evaluate the feelings of other people before taking a decision. Before the wide diffusion of the Internet and Web 2.0, people used to share opinions and recommendations with traditional approaches: asking friends, talking to experts and reading documents. The Internet and web made possible to find out opinions and experiences from people being neither our personal acquaintances nor well known professional critics. 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 [29]. Companies are interested in what customers say about their products as politicians are interested in how different news media are portraying them. Therefore there is a lot of information on the web that have to be properly managed in order to provide vendors with highly valuable network intelligence and social intelligence to facilitate the improvement of their business. In this scenario, a promising approach is the sentiment analysis: the computational study of opinions, sentiments and emotions expressed in a text [13]. Its main aim is the identification of the agreement or dis-agreement statements that deal with positive or negative feelings in comments or reviews. In the literature, there are many approaches to the sentiment analysis. A very broad overview of the existing work was presented in [19]. The authors describe in detail the main techniques and approaches for an opinion oriented information retrieval. Early work in this area was focused on determining the semantic orientation of documents. In particular some approaches attempt to learn a positive-negative classifier at a document level. [26] introduces the results of review classification by considering the algebraic sum of the orientation of terms as respective of the orientation of the documents. Starting from this approach other techniques have been developed by focusing on some specific tasks as finding the sentiment of words [27]. Baroni [3] proposed to rank a large list of adjectives according to a subjectivity score by employing a small set of manually selected subjective adjectives and computing the mutual information of pairs of adjectives using frequency and co-occurrence frequency counts on the web. The work of Turney [25] proposes an approach to measure the semantic orientation of a given word based on the strengths of its association with a set of context insensitive positive words minus the strengths of its association with a set of negative words. By this approach sentiment lexicons can be built and a sentiment polarity score can be

URI: https://hdl.handle.net/10125/59661
ISBN: 978-0-9981331-2-6

assigned to each word [16][9]. Sentiment polarity score means the strengths or degree of sentiment in a defined sentence pattern. Artificial Intelligence and probabilistic approaches have also been adopted for sentiment mining. In [20] three machine learning approaches ($Na\ddot{i}ve$ Bayes, Maximum Entropy and Support Vector Machines) has been adopted to label the polarity of a movie reviews datasets. A promising approach is presented in [21] where a novel methodology has been obtained by the combination of rule based classification, supervised learning and machine learning. In [23] a SVM based technique has been introduced for classifying the sentiment in a collection of documents. In [18], instead, a $Na\ddot{i}ve$ Bayes classifier is used for the sentiment classification of tweets corpora. Other approaches are inferring the sentiment orientation of social media content and estimate sentiment orientations of a collection of documents as a text classification problem [7]. More in general, sentiment related information can be encoded within the actual words of the sentence through changes in attitudinal shades of word meaning using suffixes as discussed in [8]. This has been investigated in [17] where a lexicon for sentiment analysis has been obtained. In [28] a probabilistic approach to sentiment mining is adopted. In particular this paper uses a probabilistic model called Sentiment Probabilistic Latent Semantic Analysis (S-PLSA) in which a review, and more in general a document, can be considered as generated under the influence of a number of hidden sentiment factors [11]. The 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 documents and words under a probabilistic framework. In [4] an approach combining the ontological formalism and a machine learning technique has been introduced. In particular the proposed system uses domain ontology to extract the related concepts and attributes starting from a sentence and then labels the sentence itself as positive, negative or neutral by means of the Support Vector Machine (SVM) classifier. In this paper, we investigate the adoption of a similar approach 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, 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 [6] [15] . Such a graph contains a set of weighted word pairs, which we demonstrate to be discriminative for sentiment classification. The main reason of such discriminative power is that LDA based topic modeling is essentially an effective conceptual clustering process and it helps discover semantically rich concepts describing the respective sentimental relationships. By means of applying these semantically rich concepts, that contain more useful relationship indicators to identify the sentiment from messages, the proposed system can accurately discover more latent relationships and make less errors in its predictions. The rationale of this paper is the following: section 2 discusses the searching the sentiment by the use of a Mixed Graphs of Terms from a document corpus. The section 3 introduces the proposed approach for the sentiment extraction and the section 4 discusses the experimental results. Finally, conclusions and further works are discussed.

2. Searching the sentiment by the use of the Mixed Graph of Terms

In this paper we explain how a complex structure, that we call a Mixed Graph of Terms (mGT), allows to capture and represent the information contained in a set of documents that belong to a well-defined knowledge domain. Such a graph can be automatically extracted from a document corpus and can be effectively used as a filter to employ in document classification as well as in sentiment extraction problems. Formally, a Mixed Graph of Terms can be defined as a graph $g = \langle N, E \rangle$ where:

- $N = \{R, W\}$ is a finite set of nodes, covered by the set $R = \{r_1, ..., r_H\}$ whose elements are the aggregate roots and by the set $W = \{w_1, ..., w_M\}$ containing the aggregates. Aggregate roots can be defined as the words whose occurrence is most implied from the occurrence of all other words in the training corpus. Aggregates are defined as the words most related to aggregate roots from a probabilistic point of view.

- $E = \{E_{RR}, E_{RW}\}$ is a set of edges, covered by the set $E_{RR} = \{e_{r1r2}, ..., e_{rH-1rH}\}$ whose elements are links between aggregate roots and by the set $E_{RW} = \{e_{r1W1}, ..., e_{eHwM}\}$ whose elements are links between aggregate roots and aggregates. As better explained further, two aggregate roots are linked if strongly correlated (in a probabilistic sense):

$$e_{r_i r_j} = \left\{ \begin{array}{ll} 1 & if \psi_{ij} \geq \tau \\ 0 & otherwise \end{array} \right\} \qquad (1)$$

Aggregate roots can be also linked to aggregates

if a relevant probabilistic correlation is present:

$$e_{r_i w_s} = \left\{ \begin{array}{ll} 1 & if \rho_{ij} \geq \mu_i \\ 0 & otherwise \end{array} \right\} \quad (2)$$

Details about mGT building and thresholds τ and μ_i will be cleared in the next section. First we show how mGT can be effectively applied for the sentiment mining from texts. The proposed method adopts the Mixed Graph of Terms for building a sentiment detector able to label a document according its sentiment. We propose an architecture 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 1 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 the algorithm 1. The proposed algorithm requires the use of an annotated lexicon, as for example WordNet [2] o ItalWord- Net [1], for the retrieval of synonyms of the words contained in the document D and not 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.

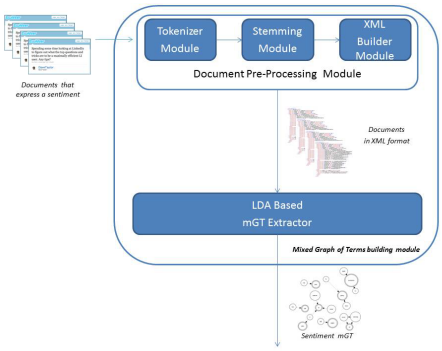

Figure 1.
Mixed Graph of Terms Building Module.

In figure 2 and figure 3 the synchronous sentiment real time classificatory architecture is depicted. For real time working two new modules have been introduced:

Algorithm 1 Sentiment Mining Algorithm

Input: $W = [w_1, w_2, ..., w_n]$ 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; $RW+ = [rw_1; rw_2, ..., rw_t]$ the aggregator words that are in mGT+; $AW+ = [aw_1, aw_2, ..., aw_m]$ the aggregated words that are in mGT+; $RW- = [rw_1, rw_2, ..., rw_n]$ the aggregator words that are in mGT-; $AW- = [aw_1, aw_2, ..., rw_p]$ 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 ! Length[W] **do**
    WS = WS + Synset[L;W[i]];
end for
W = W +WS;
Mining the sentiment from the document
for i = 0 ! Length[W] **do**
    for k = 0 ! Length[RW+] **do**
        if (RW+[k] == W[i]) **then**
            fp = fp + 2;
        end if
    end for

for k = 0 ! Length[RW+] **do**
if (RW+[k] == W[i]) **then**
vfn = fn + 2;
end if
end for
for k = 0 ! Length[AW+] **do**
if (AW+[k] == W[i]) **then**
fp = fp + 1;
end if
end for
for k = 0 ! Length[AW+] **do**
if (AW+[k] = W[i]) **then**
fn = fn + 1;
end if
end for
end for
Determining the Sentiment
if (fp ¿ fn) **then**
$Sentiment_D = Positive$;
else
if (fp ¡ fn) **then**
$Sentiment_D = Negative$;
else
$Sentiment_D = Neutral$;
end if
end if

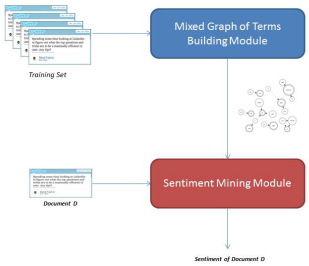

Figure 2.
Sentiment Analysis Classification System Architecture.

- 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 continuously 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.

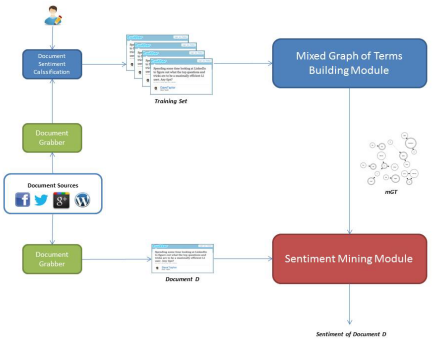

Figure 3.
System Architecture for Synchronous Classification.

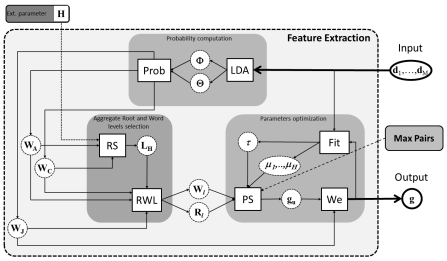

Figure 4.
Proposed mGT extraction method.

3. Extracting a Mixed Graph of Terms

The aim of this section is to explain how a Mixed Graph of Terms, which contains the most significant word pairs, can be extracted from a corpus of documents. The extraction process is shown in figure 4 he input of the system is the set of documents $\Omega_r(d_1, ..., d_M)$ and the output is a vector of weighted word pairs $g = \{w'_1, ..., w'_{|\tau_\rho|}\}$ where τ_ρ is the number of pairs and w'_n s the weight associated to each pair (feature) $t_n = (\upsilon_i, \upsilon_j)$. Such a weighted word pairs structure can be suitably represented as a mixed graph g of terms (mGT) figure 5 The mGT is made of several clusters, each containing a set of

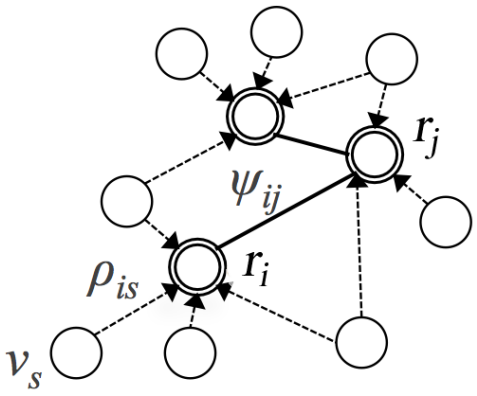

Figure 5.
mGT graphical representation.

words υ_s (aggregates) related to an aggregate root τ_i a special word which represents the centroid of the cluster. How aggregate roots are selected will be clear further. The weight ρ_{is} can measure how a word is related to an aggregate root and can be expressed as a probability: $\rho_{is} = P(\tau_i|\upsilon_s)$ The resulting structure is a subgraph rooted on τ_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: $\psi_{ij} = P(\tau_i, \tau_j)$. Given the training set Ω_r 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 τ_p from all the possible ones. A mGT graph 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. 4); 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 $\mu = (\mu_1, ..., \mu_H)$ 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 υ_i and aggregate root τ_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 υ_s and the aggregate root τ_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 $\rho_{is} = P(\tau_i|\upsilon_i) = P(\upsilon_i|\upsilon_s)$ and $\psi_{ij} = P(\tau_i|\tau_j) = P(\upsilon_i|\upsilon_s)$. Considering that $P(\upsilon_i, \upsilon_j) = P(\upsilon_i|\upsilon_j)P(\upsilon_j)$ all the relations between words result from the computation of the joint or the conditional probability $\forall i, j, \in \{1, ..., |\tau|\}$ (where $|\tau|$ is the size of the vocabulary τ which contains all the indexed words from the corpus) and $P(upsilon_j)$. An exact calculation of $P(upsilon_j)$ and an approximation of the joint, or conditional, probability can be obtained through a smoothed version of the generative model introduced in [4] called Latent Dirichlet Allocation (LDA), which makes use of Gibbs sampling [10]. The original theory mainly proposes a semantic representation in which documents are represented in terms of a set of probabilistic topics z . Formally, we consider a word u_m of the document d_m as a random variable on the vocabulary τ and z as a random variable representing a topic between {1,...,K}. A document d_m results from generating each of its words. To obtain a word, the model considers three parameters assigned: α, η and the number of topics K. Given these parameters, the model chooses θ_m through $P(\theta|\alpha) \sim Dirichlet(\alpha)$, the topic k through $P(z|\theta_m) \sim Multinomial(\theta_m)$ and $\beta_k \sim Dirichlet(\eta)$.Finally, the distribution of each word given a topic is $P(u_m|z, \beta_z) \sim Multinomial(\beta_z)$. The output obtained by performing Gibbs sampling on a set of documents Ω_r consists of two matrixes:

1) the words-topics matrix that contains $|\tau|$ X K elements representing the probability that a word υ_i of the vocabulary is assigned to topic k : $P(u = \upsilon_i|z = k, \beta_k)$;

2) the topics-documents matrix that contains K X Ω_r elements representing the probability that a topic k is assigned to some word token within a document d_m : $P(z = k|\theta_m)$.

The probability distribution of a word within a document d_m of the corpus can be then obtained as:

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

In the same way, the joint probability between two words u_m and y_m of a document 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:

$$P(u_m, y_m) = \sum_{k=1}^{K} P(u_m, y_m | z = k, \beta_k) P(z = k | \theta_m) \quad (4)$$

Note that the exact calculation of Eq. 4 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. 4 can be written as:

$$P(u_m, y_m) \simeq \quad (5)$$

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

Moreover, Eq. 3 gives the probability distribution of a word u_m within a document 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:

$$P(u) = \sum_{m=1}^{M} P(u_m) \delta_m \quad (6)$$

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:

$$P(u, y) = \sum_{m=1}^{M} P(u_m, y_v) \delta_m \quad (7)$$

Concluding, once we have P(u) and P(u,y) we can compute $P(\upsilon_i) = P(u = \upsilon_i)$ and $P(\upsilon_i, \upsilon_j) = P(u = \upsilon_i; y = \upsilon_j)$, $\forall i, j, \in \{1, ..., |\tau|\}$ 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 $r = (r_1, ..., r_H)$, 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:

$$r_1 = argmax \prod_{j \neq 1} P(\upsilon_i | \upsilon_j) \quad (8)$$

Since relationships' strenghts 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 $\mu = (\mu_1, ..., \mu_H)$ can be used to select the number of relevant pairs for each aggregates' subgraph. Note that a relationship between the word υ_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' strenghts between aggregate roots and words. In fact, being ρ_{is} 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' strenght 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 g_t can be obtained by varying the parameters $\Lambda_t = (\tau, \mu)_t$. As shown in Fig. 4 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 g_t is a vector of features $g_t = \{b_{1t}, ..., b_{|\tau_{sp}|t}\}$ in the space τ_{sp} and each document of the training set Ω_r can be represented as a vector $d_m = (\omega_{1m}, ..., \omega_{|\tau_{sp}|t})$ in the space τ_{sp}. A possible scoring function is the cosine similarity between these two vectors:

$$S(g_t, d_m) = \frac{\sum_{n=1}^{|\tau_{sp}|} b_{nt} * \omega_{nm}}{\sqrt{\sum_{n=1}^{|\tau_{sp}|} b_{nt}^2} * \sqrt{\sum_{n=1}^{|\tau_{sp}|} \omega_{nm}^2}} \quad (9)$$

and thus the optimization procedure would consist in searching for the best set of parameters Λ_t such that the cosine similarity is maximized $\forall d_m$. Therefore, the best 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 d_m is obtained, we have:

$$S_t = \{S(g_t, d_1), ..., S(g_t, d_{|\Omega_r|})\} \quad (10)$$

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 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 multiobjective 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 (F), which depends on Λ_t,

$$F(\Lambda_t) = E|S_t| - \sigma_m|S_t| \tag{11}$$

where E is the mean value of all the elements of S_t and σ_m is the standard deviation. By summing up, the parameters learning procedure is represented as follows,

$$\Lambda^* = argmax_t\{F(\Lambda_t)\} \tag{12}$$

Since the space of possible solutions could grow exponentially, $|\tau_{sp}| \leq 300$ has been considered. Furthermore, the remaining space of possible solutions has been reduced by applying a clustering method, that is the K-means algorithm, to all ψ_{ij} and $\rho_i s$ values, so that the optimum solution can be exactly obtained after the exploration of the entire space.

4. Experimental Results

In order to evaluate the performance of the proposed approach, two experimental phases have been conducted. The first one has been carried out using a standard dataset while the second one has been applied to "real life" datasets collected from Twitter and Facebook; results obtained by the proposed method have been compared with the others in literature. The first dataset used for the experimentation is the Movie Reviews Dataset provided by Pang et al. in [20]. This dataset consist of 1000 positive and 1000 negative reviews from the Internet Movie Database. Positive labels were assigned to reviews that had a rating above 3.5 stars and negative labels were assigned to the rest. The first step of the experimental campaign was aimed to find the best size for the training set. For achieving this task nine training sets have been built selecting in a random way from the 10% to 90% of the positive and negative comments that are in the full dataset. By the use of these training sets, the positive and negative mixed Graphs of Terms have been built and the sentiment classification on the remaining comments has been conducted. The process of training sets and mixed graph of terms building and documents classification has been conducted ten times. The obtained results, in terms of average accuracy, are depicted in figure 6. As depicted in the figure the value of accuracy improves with the increase of the training set but the change is very low after the adoption of a training set composed from the 50% of comments that are in the dataset. After this phase a comparison with the results obtained on the same dataset by other approaches that are in literature has been conducted (table 1). The proposed approach shows the best results in comparison with the other ones when the 50% of dataset is used as training set. The other approaches usually adopt a larger dataset and in real cases the training phase could be critical and time consuming.

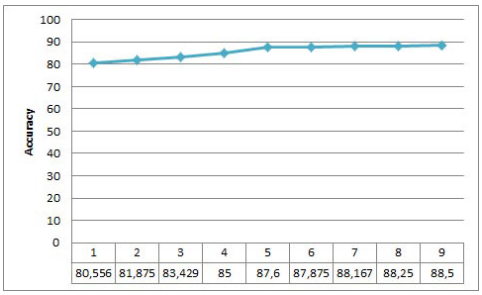

Figure 6.
The variation of the accuracy compared to the size of the training set (on the x-axis 1 means 10set and so on)

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

Reference Paper	Methodology	Accuracy
(20)	Support Vector Machines	82.90%
	Naiive Bayes	81.50%
	Maximum Entropy	81.00%
(12)	Support Vector Machines	86,20%
(5)	Ontology Supported Polarity Mining	72,20%
(14)	Bayesian Classification with the support of lexicons	81,42%
(24)	Formal Concept Analysis	77,75%
mGT Approach	LDA	88,50%

The classification phase average lasts about a couple of minutes using a Linux Ubuntu platform running on a 8GB RAM single CPU while the training phase lasts from about 5 minute to ten minutes depending from the size of the training set. As previously said our approach gives effective results also in real time scenarios. To demonstrate this aspect, an experimental campaign on posts coming from social networks has been conducted. In particular, 2.500 posts from Facebook and 10.000 tweets from twitter have been collected. The posts of Facebook has been collected from an official page of a well-known mobile phone producer while the tweets have been collected from the hashtags related to Pompeii Archaeological Park. A group of five experts labeled the posts and the tweets according their sentiments using a majority vote rule and deleting the neutral comments (table 2).

The methodologies based on the Naiive Bayes and Support Vector Machine introduced in the paper [20]

Table 2. Considered datasets.

Dataset	Source	Positive	Negative	Neutral
Mobile Phone Producer	Facebook	864	759	877
Pompei$_{Tweets}$	Twitter	4783	3498	1719

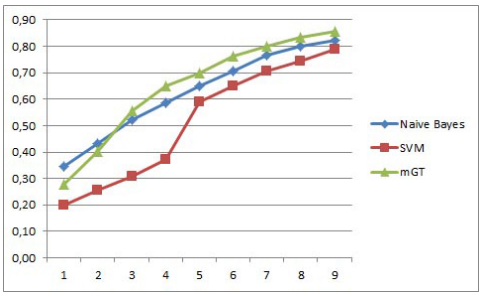

Figure 7.
Pompei Tweets: the variation of the accuracy compared to the size of the training set (on the x-axis 1 means 10% of training set and so on)

and [12] have been implemented and applied to the collected data in order to compare results with the ones obtained by the proposed method. The first step was aimed at building the training set. As previously said for achieving this task nine training sets have been built selecting in a random way from the 10% to 90% of the positive and negative comments that are in the full dataset. At the end of the training phase the selected approaches has been tested on the on the test sets obtaining the results depicted in figure 7 (dataset Pompei Tweets), figure 8 (Mobile Phone Producer) and table 3.

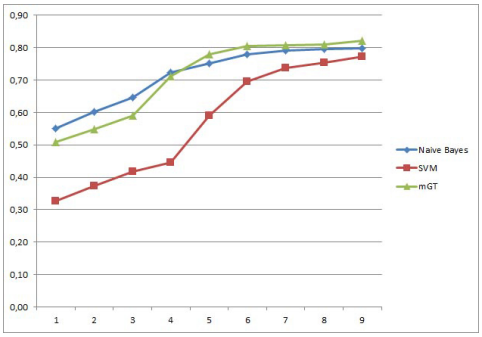

Figure 8.
Mobile Phone Producer: the variation of the accuracy compared to the size of the training set (on the x-axis 1 means 10% of training set and so on)

In the case of the dataset coming from twitter, the experimental results show how the proposed method offers the best performance starting from a training set composed by the 30% of the collected dataset. In general the performances start to be interesting with a training set composed by the 70% of the dataset: it

Table 3. The obtained results.

	Pompei Tweets Dataset			Mobile Phone Producer Dataset		
perc.	Naiive Bayes	SVM	mGT	Naiive Bayes	SVM	mGT
10%	0.55	0.33	0.51	0.35	0.2	0.28
20%	0.6	0.37	0.55	0.43	0.25	0.4
30%	0.65	0.42	0.59	0.52	0.31	0.56
40%	0.72	0.45	0.71	0.58	0.37	0.65
50%	0.75	0.59	0.78	0.65	0.59	0.7
60%	0.78	0.7	0.81	0.71	0.65	0.76
70%	0.79	0.74	0.81	0.77	0.7	0.8
80%	0.8	0.75	0.81	0.8	0.74	0.83
90%	0.8	0.77	0.82	0.82	0.79	0.85

is an expected result because the tweets are composed by a short number of words and so systems based on $Na\ddot{i}ve$ Bayes and mGT has to learn from a great number of examples. For the SVM we observe acceptable performances when employing at least the 50% of the dataset. The poor results of the SVM, compared tpo other methods, are due to the difficult to find a well-defined pattern for the correct classification and this task is very difficult in the case of the tweets (where there is not a welldefined structure). In the case of the dataset collected from Facebook, the same approach has been adopted for the classification of the posts. Also in this case the proposed approach shows the best results starting from a training set composed by the 30% of the dataset. In this case the performance of our system improved faster than the twitter case because Facebook's posts contain a greater number of words so that the built mGTs are more effective.

5. Conclusion

This paper proposes the use of the mixed graph of terms, obtained by the use of Latent Dirchlet Allocation approach, as tool for the sentiment classification of documents. The method relies on building complex structures called mGTs from documents labeled according their sentiment. Then, the classification of a new 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 regards the introduction of annotated lexicon, as SentiWordnet, for a better evaluation of the words and the sentence structures.

References

[1] Italwordnet - http://www.ilc.cnr.it/iwndb/iwndb php/.

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

[3] 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.

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

[5] Pimwadee Chaovalit and Lina Zhou. Movie review mining: a comparison between supervised and unsupervised classification approaches. In Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 4 - Volume 04, HICSS '05, pages 112.3-, Washington, DC, USA, 2005. IEEE Computer Society.

[6] 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.

[7] 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.

[8] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06, pages 417-422, 2006.

[9] 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.

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

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

[12] F. Colace, M. De Santo, S. Lemma, M. Lombardi, A. Rossi, A. Santoriello, A. Terribile, and M. Vigorito, "How to Describe Cultural Heritage Resources in the Web 2.0 Era?", 11th International Conference on Signal-Image Technology and Internet-Based Systems (SITIS), pp. 809-815, 2015. DOI:10.1109/SITIS.2015.50

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

[14] Prem Melville, Wojciech Gryc, and Richard D. Lawrence. Sentiment analysis of blogs by combining lexical knowledge with text classification. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '09, pages 1275-1284, New York, NY, USA, 2009. ACM.

[15] 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.

[16] 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.

[17] 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.

[18] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta, May 2010. European Language Resources Association (ELRA).

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

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

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

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

[23] 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).

[24] 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).

[25] 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.

[26] Peter D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, pages 417-424, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[27] 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.

[28] 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.

[29] Vivek Rajput and Sarika Bobde, "Stock market prediction using hybrid approach", International Conference on Computing, Communication and Automation (ICCCA), 2016, DOI: 10.1109/CCAA.2016.7813694