A Sentiment Polarity Analyser based on a Lexical-Probabilistic Approach

Berardina De Carolis, Domenico Redavid, Angelo Bruno

Department of Computer Science, University of Bari
berardina.decarolis@uniba.it, redavid@abrain.it, angelo.bruno89@gmail.com

Abstract. In this paper, we propose an unsupervised approach to automatically classify the sentiment polarity of texts that can be documents or tweets related to the user's favorite hashtags. The system is based on a combination of probabilistic and lexicon-based approaches. We first apply the Latent Dirichlet Allocation (LDA) model to discover two vectors of terms relevant for two topics (presumably positive and negative) and then we calculate the polarity of the associated sentiment using the SentiWordnet resource. Experiments have been conducted first on an English dataset and then the system has been associated to an application and tested for Italian. Results show that the system can partition the polarity with a good accuracy.

1 Introduction

Information from online social network and micro-blogging platforms, such as Twitter, is of interest for many research fields from social to computer science. In particular, in the linguistic analysis field, several frameworks for detecting sentiments in social media have been developed for different application purposes. For instance, tweets have been used for opinions mining about products, for monitoring political sentiment [1], for detecting moods in a given geographical area [2], and so on. The recent integration of social media with Digital Libraries (DL) will open the way for new types of applications. One of these concerns the application of the sentiment analysis to digital documents in order to understand relations between opinions and other factors (i.e. location, gender, etc.) in order to support the administrator of the DL in the phase of social marketing and advertising.

The main goal of the work presented in this paper is to develop an unsupervised approach to analyze the sentiment polarity of a set of text messages that can be for instance reviews about items or a set of tweets corresponding to a set hashtags. In addition we add to deal with another constraint regarding the language. In our application the tweets to be analyzed were written both in Italian and in English. To come up with a technique to find sentiment polarity of a set of texts that could be of different nature we use a combination of probabilistic with a lexicon-based approach. As a first step we apply Latent Dirichlet Allocation (LDA), a probabilistic graphical model, which mines hidden semantics from a set of documents [3]. It is a "topical" model that represents documents as bags of words, and looks to find semantic dependencies between words. In our approach we use LDA over tweeter collections, so as to get two topics, which probably correspond to two different sentiments. In

particular we discover two vectors of terms characterizing the two topics, presumably positive and negative. These vectors are then analyzed from the polarity point of view using the SentiWordnet resource [4]. The resulting vectors polarity is then analyzed to determine the global sentiment polarity of the set of hashtags. In order to determine the accuracy of results obtained with this approach we conducted first some experiments on an English dataset and then the system has been associated to an application and tested for Italian. Results show that the system can partition the topic/polarity with a good accuracy.

The paper is structured as follows. Section 2 presents the motivation for this work. Then, in Section 3, we describe how Sentiment Polarity Analyzer has been developed. Then, Section 4 reports results of experiments that have been conducted on both English and Italian. Finally, conclusions and directions for future work are illustrated in Section 5.

2 Motivations for the proposed approach

Machine Learning-based techniques for sentiment classification can use supervised or unsupervised approaches. In the former case, a 'training set' of documents annotated with the correct sentiment is needed, and performance can be evaluated using a different 'test set'. In the supervised setting, [5] profitably used Naive Bayes (NB), Maximum Entropy (ME) and Support Vector Machines (SVM) to classify film reviews as positive or negative. As features they use term vectors obtained without stemming or stopword removal, and considered only single terms appearing at least 4 times in the corpus and bi-grams appearing at least 7 times. They also implemented a simple mechanism to recognize the presence of negations that invert the polarity. In the unsupervised setting, [6] proposed an algorithm to classify reviews based on the average polarity of sentences containing adjectives or adverbs. After carrying out PoS tagging, pairs of words including one adjective or adverb are extracted, checking their correspondence with pre-selected PoS patterns. Then, the polarity of the extracted expression is estimated, using a formula based on Pointwise Mutual Information (PMI) applied to the results obtained by an Internet search engine. The final outcome is 'recommended' if the sign of the average polarity is positive, or 'not recommended' if it is negative. A similar approach was used for Sentence-level Sentiment Classification in [18], leveraging the co-occurrence of terms of known polarity in the sentence, but using a different likelihood index. Instead of comparing a word with a single known term, subsets of manually classified adjectives are used, and the polarity of the sentence is determined based on its score: positive if above a given threshold, or negative if below another threshold. [7] determine the polarity of the sentence based on the polarity of the single opinion words it includes, using a set of adjectives of known polarity and WordNet. If an adjective in the sentence has unknown polarity, the system tries with its synonyms and opposites. The list of known adjectives is expanded if the search is successful.

Even if supervised learning is commonly used in text categorization, and then in Sentiment Analysis – recently there has been an increased used of unsupervised or semi-supervised approaches to sentiment classification in order to solve the problem of domain dependency and the need for annotated dataset [6]. In the unsupervised case, the system takes unlabeled data and tries to find meaningful correlations among them. To this aim various techniques, both probabilistic and non-probabilistic have been used, few of which include Latent Semantic Indexing (LSI) [8], probabilistic LSI [9], Latent Dirichlet Allocation (LDA), etc.

Among the different unsupervised approaches proposed in the literature, those based on topic models seem to be appropriate to addressing the sentiment classification problem. In particular, among them, LDA is the most recently developed and widely used technique that has been working well in capturing these semantics [3]. It is a probabilistic generative topic model that is very often used for this task. It is based on the assumption that each document is a mixture of latent topics and each topic is a probability distribution over different words. Then, for each latent topic T, the model learns a conditional distribution $p(w|T)$ for the probability that word w occurs in T. One can obtain a k-dimensional vector representation of words by first training a k-topic model and then filling the matrix with the $p(w|T)$ values (normalized to unit length). The result is a word–topic matrix in which the rows are taken to represent word meanings. However, since LDA is used to model topics and it is not related to word meanings, there is no guarantee that the discovered word vectors identify words denoting the polarity of the sentiment. Some recent work introduces extensions of LDA to capture sentiment in addition to topical information [10, 11].

In our approach we use LDA to extract two word vectors that ideally should represent words characterizing two topics corresponding to the polarity of the considered set of tweets. Then, in order to identify the sentiment content of the discovered vectors we rely on the SentiWordNet [4] affective lexicon with the aim of giving an affective weight to words in the vectors.

2 Sentiment Polarity Analyzer

Sentiment Polarity Analyzer (SPA) is a system able to analyze the sentiment of a set of text messages (tweets, posts, etc.) using an approach that combines a topic model, LDA, and SentiWordNet. In particular:

i) LDA is used to extract the word vectors relative to two topics, that ideally should represent words relevant to the two different polarities of the dataset;

ii) SentiWordNet is used to give a weight to the sentiment polarity of each single word.

Figure 1 illustrates the workflow of task executed by SPA.

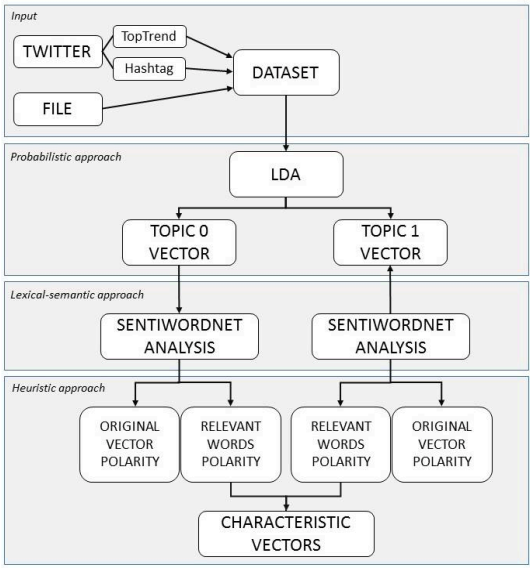

Figure 1. Sentiment Polarity Analyzer Tasks Flow

The process is composed of four main phases:

1. **Input**: the input to the system can be a dataset extracted by *Twitter* according to the selected hashtag(s) or a text file that contains text messages of any type in English and/or in Italian. To this aim we use *Twitter4J*[1] API. For each execution, the application extracts up to 8000 tweets to which we apply the following filters: a) re-tweet delete, b) detection of the tweet language (English and Italian), c) deletion of tweets already extracted in a previous execution run, d) pre-processing of each tweet by removing the hashtag, URLs and tweet shorter than 10 chars.

2. **Word vectors extraction**: using the LingPipe framework [12] we use LDA with the number of topic set to two on the given input. In this way we extract the two word vectors. It is possible to set some parameters such as the minimum occurrence of considered token in the document. This parameter is important in case of very large datasets in order to avoid that the polarity is influenced by very rare terms (in this case this parameters has to be set to an high value), on the contrary considering an input with a little quantity of text is important to set a low threshold in order to avoid skipping words that are important for the considered topic.

3. **Polarity Analyzer**: each of the extracted vector is analyzed in terms of sentiment polarity using SentiWordNet. At the end of this phase the system returns

[1] Twitter4J: http://twitter4j.org/en/index.html

the *positive/negative* polarity for each topic (word vector) identified by LDA. In order to deal with text written either in English or Italian we used an automatic translation service (*Java Google Translate Text-to-Speech*[2]). For determining the polarity of each word/term, we considered each possible use of the word in the SentiWordNet classes - name, verb, adverb, adjective – and summed each class score for computing a global polarity of the word/term (1 and 2):

$$\text{class\_score}(w_c) = \frac{\sum_{w_a \in S_c} \text{score}(w_a)}{n} \quad (1)$$

$$score(w) = \frac{\sum_{c \in C} class\_score(w_c)}{j} \quad (2)$$

(1) expresses the average of the polarity of a word *w* for a class *c*, where:

- w_c is the word *w* in the class *c*;
- w_a is the word *w* in the meaning *a*;
- S_c is the set of synset of w_a in *c*;
- $score(w_a)$ is the polarity score of w_a;
- *n* is the number of w_a in S_c.

(2) denotes the global average of the polarity of *w*, where *C* is the set of classes and *j* is the number of classes.

4. **Heuristic Evaluation of Results:** using SentiWordNet the system extracts other two word vectors using terms that have a strong polarity weight. These are mixed both in one single vector which is divided in two new vectors by polarity, obtaining the true positive and negative score of the dataset. To evaluate the performance of the proposed approach, three evaluations are performed on results.

The First Heuristic aims at "evaluating whether LDA is suitable to determine the two topic word vectors as denoting two opposite sentiment polaraties". To this aim the two vectors v_0 and v_1 are analyzed in terms of polarity with SentiWordNet in order to determining a *positive* and a *negative* score for each vector. This evaluation has been performed using the condition that the vectors should have an opposite polarity: $score\_pos(v_0) - score\_neg(v_0) > 0$ *AND* $score\_pos(v_1) - score\_neg(v_1) < 0$.

The *Second Heuristic* aims "evaluating whether the performance of LDA on a single topic may be improved". We created two vectors composed by relevant words in order to increase the semantic consistency of terms. The rule for determining the relevance of a term is the following for selecting a positive word: *(pos_score ≥ 0.5 AND pos_score > neg_score) OR (pos_score - neg_score ≥ 0.25)*. We apply an analogous rule for the selection of a negative word. Table 1 summarizes an example of application of these rules (words are translated from Italian).

[2] Java Google Translate Text-to-Speech: gtranslateapi-1.0.jar

TOPIC VECTORS	TERMS
Topic0	father - shame - situation - donate - hospitalized - good - suck - sick - prison - outburst
Topic0 Relevant words	shame - situation - good - suck - sick - affected
Topic1	solution - victims - supreme - stop - bad - free - priority – never today - international - cure
Topic1 Relevant words	supreme - stop - bad - free - priorità – never today - cure

Table 1. Application of the relevance rules on the dataset: #marò.

The *Third Heuristic* aims "*evaluating the distance of the dataset polarity extracted with SentiWordNet compared to the polarity of the LDA vectors*." To this aim we merged the two vectors of relevant words and to create automatically two vectors containing the set of terms characterizing the polarity of the dataset (see Table 2).

TOPIC VECTOR	TERMS
Topic0 Relevant words	shame - situation - good - suck - sick - affected
Topic1 Relevant words	supreme - stop - bad - free - priorità – never today - cure
Characterizing Positive Vector	supreme - good - stop - free - today - cure
Characterizing Negative Vector	shame - situation - suck - sick - affected - bad - priority - never

Table 2. Relevant word vectors are merged and then split according to polarity in order to get two characterizing vectors.

In this way it is possible to determine the polarity of each vector. In particular, considering the example reported in Table 2, for the Topic0 (T_0) we have:

$$\text{pos\_polarity}(T_0) = \frac{\text{n° positive words}}{\text{n° total words}} = 0{,}17$$

$$neg\_polarity(T_0) = \frac{n° \, negative \, words}{n° \, total \, words} = 0{,}83$$

In the same way the vector T_1 expresses a positive polarity for 62,5% and a negative one for 37,5%. Then, we can say that LDA extracts relevant words that allow distinguishing the sentiment polarity since, in this example T_0 can be denoted as the *negative word vector*, since is polarity is 83% negative and T_1 as the *positive one*.

Sentiment Polarity Analyser (SPA) has been implemented in Java both as an application and as a webserver to be used by any other application that may need this service. Its interface is illustrated Figure 2 and it is composed by 4 main sections:

- Selection of the dataset and starting of the analysis;

- Log of execution steps;
- Polarity score of the dataset – the TAG CLOUD buttons allows reading the word vectors characterizing the topic;
- Graph section illustrating the trend of the topic polarity.

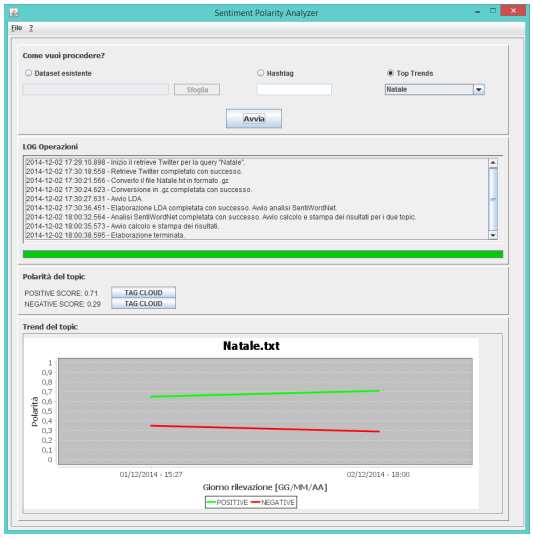

Figure 2. Main Interface of SPA.

SPA can be used not only to extract the polarity of the dataset but also for monitoring an hashtag or a set of hashtags in time. In this case results are presented as a graph that shows the trend of the sentiment around that topic. Figure 3 reports an example of the monitoring of the hashtag "Renzi" (the Italian premier) in from the 1st to the 30th of October 2014. You can notice that the positive trend goes down after the 15th of October the day in which the "legge di Stabilita'" was issued (new taxes).

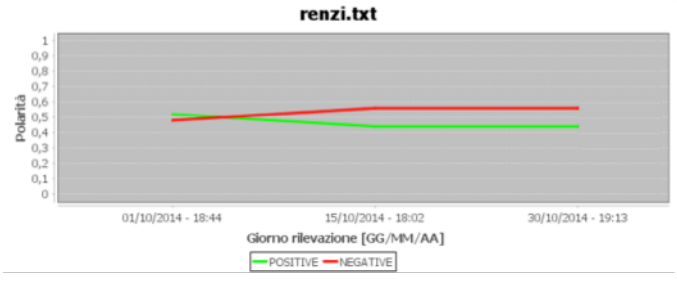

Figure 3. Trend of the polarity of the tweets regarding "Renzi" for one month.

3 Evaluation

Our approach does not aim at classifying a single post or short text message as a *positive* or *negative* one but, given the goal of our application, aims at analyzing and monitoring the polarity trend of a topic or a set of topics and therefore it can be seen as a tool for determining the degree of liking about a certain topic. For determining how well SPA performed this task we have evaluated the tool on 4 datasets for which we know the sentiment polarity *mixture* and the results are shown in Table 3.

#Test	Dataset	%dataset polarity		% relevant vector polarity LDA		%Characterizing vectors	
		POS	NEG	POS	NEG	POS	NEG
1	*sentiment140*[3]	55.19	44.81	59.45	40.55	50.63	49.37
2	*filmup*[4]	61.54	38.46	64.14	38.86	57.65	42.35
3	*cornell_polarity*[5]	50	50	59.85	40.15	46.87	53.13
4	*large_movie_review*[6]	30	70	30.65	69.35	37.23	62.77

Table 3. Evaluation results on the 4 different datasets.

In Table 4, test #1 shows an error of about 4% using the vectors extracted with LDA, while for the characterizing vectors the error is about the same but the polarity of the dataset is not defined. This can be caused by the number of words in the vectors that depends on the number of minimum occurrence of the tokens in the LDA that has been set as a default to 5. Results of test #2 are encouraging since the in both types of vectors is about 3% and LDA identifies the negative topic with the about the same polarity of the original dataset - 38.86% vs. 38.46%. We have similar results in the tests #3 and #4. After these results we made some experiments by varying the number of minimum occurrence of the tokens by increasing it opportunely (up to 500) and while this unbalanced the polarity of the LDA extracted vectors (by increasing the error to 9%), the characterizing vectors reached the correct mixture or topic polarity in particular for the *large_movie_review* dataset.

# Test	Dataset	min Token count	% error variation with LDA	% error variation characterizing vectors
1	sentiment140	Default → 500	$4{,}85^{-}$	$2{,}2^{+}$
2	cornell_polarity	Default → 300	$5{,}77^{-}$	$2{,}17^{+}$
3	l_movie_review	Default → 1000	$10{,}03^{+}$	$0{,}5^{-}$

Table 4. Percentage error of the evaluation results.

[3] Sentiment140 dataset: http://help.sentiment140.com/for-students/
[4] http://filmup.it[13]
[5] Cornell Polarity Dataset 1.0: http://www.cs.cornell.edu/people/pabo/movie-review-data/
[6] Large Movie Review Dataset: http://ai.stanford.edu/~amaas/data/sentiment/

4 Conclusions and Future Work Directions

We conclude that the methodology presented in this paper is a feasible approach to model how trends of sentiments about a particular topic or a set of topics could be monitored. SPA uses an unsupervised approach to automatically classify the sentiment polarity of text messages, documents and tweets. The flexibility of SPA allows its use in different application domains where there is the need of determining or monitoring the polarity of a dataset. The system is based on a combination of probabilistic and lexicon-based approaches. We first apply the Latent Dirichlet Allocation (LDA) model to discover two vectors of terms relevant for two topics (presumably positive and negative) and then we calculate the polarity of the associated sentiment using the SentiWordnet resource. Experiments have been conducted first on an English dataset and then the system has been associated to an application and tested for Italian. Results show that the system can partition the polarity with a good accuracy.

The presented work represents the implementation of the first prototype of the system and we are aware of its limitations. For improving the performance of the proposed approach an affective lexical resource for Italian is necessary in order to avoid problems due to the translation. Another important issue regards the negation problem that needs particular attention. Most of the proposed solutions are based on heuristics similar to those used to handle the *AND* and *BUT* connectors. A possible solution could be represented by a switch to an approach based on *semantics (bag-of-concepts* [14]) although these would request another methodology for sentiment classification. In our future work we plan to integrate our implementation in a Digital Library Management System and to perform some experiment on a dataset of Italian tweets [15] in order to compare our results with those obtained in the EVALITA context[7].

Acknowledgment

This work fulfils the research objectives of the PON02_00563_3489339 project "PUGLIA@SERVICE - funded by the Italian Ministry of University and Research (MIUR).

References

[1] Tumasjan, A.; Sprenger, T.; Sandner, P. G.; and Welpe, I. M. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. In Proc. of 4th ICWSM, 178–185. AAAI Press.

[7] EVALITA - Evaluation of NLP and Speech Tools for Italian, www.evalita.it

[2] Mitchell, L., Frank, M. R., Harris, K. D. , Dodds, P. S. and Danforth C. M. (2013). The Geography of Happiness: Connecting Twitter Sentiment and Expression, Demographics, and Objective Characteristics of Place. PLoS ONE, 8(5), 05.

[3] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," Journal of Machine Learning Research, vol. 3, pp. 993–1022, 2003.

[4] Baccianella, S., Esuli, A. and Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Nicoletta Calzolari et al., editor, Proceedings of LREC.

[5] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002, pp. 79–86.

[6] P.D. Turney et al. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pages 417–424, 2002.

[7] M. Hu and B. Liu, "Mining and summarizing customer reviews," in Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2004, pp. 168–177.

[8] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. J AM SOC INFORM SCI, 41:391–407, 1990.

[9] Hofmann, T.:Probabilistic Latent Semantic Indexing. In: Proceedings of the 22nd International ACM Conference on Research and Development in Information Retrieval, pp. 50–57 (1999)

[10] C. Lin and Y. He. 2009. Joint sentiment/topic model for sentiment analysis. In Proceeding of the 18th ACM Conference on Information and Knowledge Management, pages 375–384

[11] F. Li, M. Huang, and X. Zhu. 2010. Sentiment analysis with global topics and local dependency. In Proceedings of AAAI, pages 1371–1376.

[12] Alias-i. 2008. LingPipe 4.1.0. http://alias-i.com/lingpipe.

[13] Casoto P., Dattolo A., Omero P., Pudota N., Tasso C.. 2008. A new machine learning based approach for sentiment classi- fication of italian documents. In M. Agosti, F. Esposito, and C. Thanos, editors, IRCDL, pages 77–82. DELOS: an Association for Digital Libraries.

[14] Jay Kuan-Chieh Chung, Chi-En Wu and Richard Tzong-Han Tsai. Improve Polarity Detection of Online Reviews with Bag-of-Sentimental-Concepts. ESWC 2014.

[15] Basile V. and Nissim M. (2013). Sentiment Analysis on Italian Tweets. Proc. of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 100–107.