Towards Sentiment and Emotion Analysis of User Feedback for Digital Libraries

Stefano Ferilli[1](✉), Berardina De Carolis[1],
Domenico Redavid[2], and Floriana Esposito[1]

[1] Dipartimento di Informatica, Università di Bari, Bari, Italy
{stefano.ferilli,berardina.carolis,floriana.esposito}@uniba.it
[2] Artificial Brain S.r.l., Bari, Italy
redavid@abrain.it

Abstract. The possibility for people to leave comments in blogs and forums on the Internet allows to study their attitude (in terms of valence or even of specific feelings) on various topics. For some digital libraries this may be a precious opportunity to understand how their content is perceived by their users and, as a consequence, to suitably direct their future strategic choices. So, libraries might want to enrich their sites with the possibility, for their users, to provide feedback on the items they have consulted. Of course, manually analyzing all the available comments would be infeasible. Sentiment Analysis, Opinion Mining and Emotion Analysis denote the area of research in Computer Science aimed at automatically analyzing and classifying text documents based on the underlying opinions expressed by their authors.

Significant problems in building an automatic system for this purpose are given by the complexity of natural language, by the need of dealing with several languages, and by the choice of relevant features and of good approaches to building the models. Following the interesting results obtained for Italian by a system based on a Text Categorization approach, this paper proposes further experiments to check whether reliable predictions can be obtained, both for opinions and for feelings.

1 Introduction

For some digital libraries, knowing the attitude of their users toward their content may be very important to understand how it is perceived by their (actual or potential) audience and, as a consequence, to suitably direct their future strategic choices. They might be interested in just the valence of the attitude, or more specifically in the feelings that some items have raised in their users. This may be particularly true for libraries containing works of art (movies, music, leisure literature, etc.), but also for libraries more oriented toward scientific contents. So, libraries might want to enrich their sites with the possibility, for their users, to provide feedback on the items they have consulted, e.g. in the form of forums or blogs. Although they are not libraries technically speaking, most websites of online shops for buying items that might well end up in libraries already provide

M. Agosti et al. (Eds.): IRCDL 2016, CCIS 701, pp. 137–149, 2017.
DOI: 10.1007/978-3-319-56300-8_13

this option (e.g., Amazon). By going through the users' messages, the library managers might gain precious information. Of course, manually analyzing all the available comments would be infeasible. Hence, the interest in automatic techniques to extract the users' attitude from their textual comments.

In fact, opinions play a fundamental role in our everyday life, directing or affecting our decisions in all contexts. People often look for the opinions of others before deciding about their own actions. Companies and politicians want to know what people thinks about their products or actions. So, obvious applications of SA have been to recommender systems, marketing, brand analysis, business and government intelligence, Web monitoring, terrorism prevention, etc. Market-oriented applications, in particular, may take great advantage from the availability of websites (such as epinions.com and rateitall.com), that collect feedback, opinions and reviews of users about all kinds of products and services. E.g., the possible correlations between the dominant sentiment in a film's reviews and its income was analyzed in [12].

Opinion Mining was originally aimed at "process[ing] a set of search results for a given item, generating a list of product attributes (quality, features, etc.) and aggregating opinions about each of them (poor, mixed, good)" [7]. It is now considered as a synonym of *Sentiment Analysis* (SA), appeared in [6,18] (where the meaning was borrowed from economics) and in [14,15,20,25]. The area of interest of SA was subsequently extended to the study, analysis and classification of text documents based on the underlying opinions expressed by their authors (e.g., about a product, a service, an event, an organization, or a person). While early works in the field date back to 1979 [3] and 1984 [23], thorough research started only in the new millennium, thanks to the availability of huge amounts of data to be processed in the World Wide Web and, in particular, in Social Networks, where people exchange ideas and comments on any branch of human interests [6,13,18,20,21]. While SA is interested just in the polarity of the opinion (positive or negative, or maybe neutral), *Emotion Analysis* aims at classifying the specific kind of emotion expressed by the text.

Opinions are expressed as text. The text carrying an opinion is called an *opinionated text*, and the person or organization who expresses the opinion is called the *opinion holder*. The target of the opinion is called *object* or *entity*. The opinion may be about specific features of an object, rather than (or in addition to) the object as a whole. This is the domain of *Feature-Based Sentiment Analysis*, and requires suitable processing to extract the features about which opinions are expressed, and the associated portion of text. Somehow tricky are features implicitly expressed by some kinds of adjectives, adverbs or verbs (e.g., 'costly' implicitly identifies the feature 'price').

Classifying the polarity of a text based only on the terms that make it up is not easy [15], especially for machines. There are several reasons for this:

- Intrinsic complexity of natural language (a well-known example is Mark Twain's review of a book by Jane Austen: "Jane Austen's books madden me so that I can't conceal my frenzy from the reader. Everytime I read 'Pride

and Prejudice' I want to dig her up and beat her over the skull with her own shin-bone.")
- Subjectivity of opinions. Even worse, some subjective sentences do not express any opinion (e.g., "I think I will go there"), while some objective sentence do (e.g., "the phone I bought stopped working in three days").
- The opinion holder might be different than the author of the message (e.g., in a quoted sentence).
- The context of an utterance may change the polarity of an opinion (e.g., "should never be missing" is positive if referred to an object, or negative if referred to a feature).
- Differently from normal Text Categorization, the order in which the terms appear in the text may be very relevant.
- While 'direct' opinions concern a single object and/or feature, 'comparative' ones highlight the similarities, differences or preferences between many objects.

To properly handle this complex landscape, a SA system must accomplish several sub-tasks, such as the identification of the object of the opinion (when many objects are compared), of the evaluated features, sometimes of the opinion holder and even of the moment in which the opinion is expressed.

This paper aims at evaluating whether useful indications about the opinion and feelings of users toward library items might be drawn from an analysis of their comments on such items, if available. In particular, we wanted to focus on the Italian language, whose grammar is more complex and for which less advanced pre-processing techniques are available with respect to English. The next section overviews related works. Then, the proposed approach is described in Sect. 3 and evaluated in Sect. 4. Finally, Sect. 5 draws some conclusions and outlines future work.

2 Background and Related Work

There are several sub-problems to be faced to carry out SA. First of all, some pre-processing steps may be needed to clean the input from formatting information (e.g., in Web pages). Also, it may be useful to filter out all sentences that do not carry opinions (a problem known as *subjectivity classification*) [21,22,26].

Given the plain text of opinionated sentences, full natural language is still out of reach for automatic procedures. The lexical level is often considered a fair trade-off between expressiveness and computational complexity. At this level, the items of interest are just tokens (words or other elements having an atomic meaning). To reach further simplification, tokens that are considered meaningless for the task at hand are removed, and inflected forms of words are normalized. Stemming reduces each term to its stem, with the risk of merging, as a side effect, terms having different meaning but the same stem. Lemmatization reduces a term to its base form, but requires additional linguistic knowledge to be able to do this. It is unclear whether exploiting phrases or n-grams (i.e., sequences of n terms in a text) brings a real advantage over just using single words [11].

Although the position of terms may be relevant for SA, the given corpus is usually represented as a *Vector Space*, i.e. as a matrix where rows are indexed with the filtered and normalized terms and columns represent the documents. Each cell expresses the relevance of a term to a document using a weighting scheme (e.g., TF*IDF) that is typically directly proportional to the number of occurrences of the term in that document and inversely proportional to its spread across the whole corpus. The vector space is often used for learning predictive models. So, each document can be seen as a vector, that identifies a point in a space whose dimensions are the terms in the vocabulary. Since the very large dimensionality may cause various problems, among which inefficiency and overfitting, dimensionality reduction may be obtained by eliminating some terms (as in [24]) or by considering a transposed space (as in Latent Semantic Indexing [8]).

Considering the Part-of-Speech (PoS) tag of terms, i.e. their lexical category, may be useful for some purposes. Indeed, it is a common feeling that nouns and verbs express objective concepts, while adjectives or adverbs may be more indicative of subjectivity. PoS tagging can nowadays be carried out automatically with satisfactory results. However, problems may arise due to ambiguous words (e.g., as in "We can can the can") or unknown ones. Also, using a large set of tags usually results in tags that have very close meaning, which makes it difficult to distinguish them using automatic taggers (or even human ones).

The presence of specific words may be very indicative for SA. Opinion words, i.e. words used to express opinions, are strictly related to PoS. There are also typical phrases that are commonly understood as expressing definite sentiments independently of their strict semantics. Sentiment shifters are phrases used to change the polarity of an opinion from positive to negative or *vice versa*. Negations are an outstanding example, but again the issue may be tricky (e.g., in the correlation 'not only ... but also').

Sentiment Polarity Classification consists in assigning the text to a category that represents a value in a given scale. In 'Binary Sentiment Classification' the scale includes just the two extremes (to be interpreted as positive/negative, or in favor/against, etc.). Document-level Sentiment Classification focuses on the assessment of the opinion of an opinion holder on a single entity in a whole opinionated document. The underlying assumption is that the document was written by a single author, and that the author expressed opinions on a single object. While product reviews usually fulfill this assumption, blog posts or forum discussions often do not. In these cases, one needs to preliminarily decompose the text in different pieces, each referred to a single object. If the pieces corresponds to sentences, one gets Sentence-Level Sentiment Classification.

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'. Producing these sets manually can be very costly, but opinions on the Web are often associated with a numeric evaluation (e.g., in terms of 'stars') that can be used to derive the associated sentiment. In the unsupervised case, the system takes unlabeled data and tries to find meaningful correlations among them. Supervised learning is more interesting here, because it somehow constrains the systems to reproduce in the learned models the same behavior as the expert who labeled the data, which is very important in our case. In the supervised setting, [15] 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. Different settings led to precision slightly above 80%, but the results of ME based only on adjectives reached just 77.7%.

Emotions and opinions are strictly related. The intensity of opinion is related to the intensity of some emotions, such as happiness and anger. Ekman [9] identified a set of 'primary' emotions that are universal (i.e., not determined by the culture or place where one lives): anger, disgust, sadness, joy, fear, and surprise. 'Secondary' emotions derive from them, but depend on the culture and are developed during growth. Emotions can be exploited to understand the behavior of people on social media, or of individuals (e.g., understanding suicides based on letters written before the event) [4]. Emotions are inherently multi-modal, involving text, sound and images. However, most works focused on text, due to its explicit encoding of information. Feeler [5], an emotion-based document classifier, exploits stopword removal (excluding emotional words), negations, question and exclamation marks (replaced by explicit labels) but no PoS information. The Vector Space Model-based classifier proved to be as effective as a Support Vector Machines-based one and a Naive Bayes approach on short sentences. It also emerged that the use of stemming improves accuracy.

3 Proposed Approach

In a previous work [10], we developed a system for Sentiment Analysis/Opinion Mining and Emotion Analysis that obtained interesting results on the task of determining the polarity of opinions concerning movies and expressed in Italian. This is especially relevant because Italian is a more complex language than English, and so many and so reliable linguistic resources and systems are not available for it as for English. This allows to hypothesize that good results can be obtained also for several other languages. To be general and context-independent, the system relies on supervised Machine Learning approaches. For the sake of flexibility, it allows to select different combinations of features to be used for learning the predictive models. In the following, we recall the system's technical features.

Our system casts the Sentiment Classification problem as a TC task, where the categories represent the polarity (or the emotions). However, several differences exist with respect to classical topic-based TC: topics are objective, while sentiments are subjective; there may be hundreds (or even thousands) of topics, but just a few sentiments (at the extreme, just two polarities, positive and negative); topics are usually application-dependent, while sentiment is general; topics may be independent from each other, while sentiments typically are not (e.g., in the evaluation of an object based on a number of 'stars' the categories are different degrees of a single scale).

Text Categorization (TC) is the activity aimed at mapping documents in natural language to a pre-defined set of categories. Formally, given a set of documents D and a set of categories C, a text *classifier* implements a function $\Phi : D \times C \rightarrow \{True, False\}$ that for each document-category pair says whether the document belongs to the category. The 'hard' categorization can be replaced by a degree of belonging ($\Phi_i : D \times C \rightarrow [0, 1]$). Often, the target function Φ is unknown, and must be approximated by another function Φ' with the same pattern as Φ. Manually creating logic rules for each category, to be used to classify documents, is costly, difficult (both for creation and for update) and allows limited reuse of the rules in different domains. Supervised Machine Learning approaches learn Φ' inductively based on the observation of the features of a 'training set' of documents manually classified by experts as belonging ('positive examples') or not ('negative examples') to specific categories. The learned classifier can be applied on an additional 'test set' of documents whose category is known to check whether its predictions are correct.

To learn a classifier, one must first choose what features to consider to describe the documents, and what is the learning method to be exploited. An analysis of the state-of-the-art, as reported in previous sections, suggested that no single approach can be considered as the absolute winner, and that different approaches, based on different perspectives, may reach interesting results on different features. Assuming that these perspectives are sufficiently complementary to mutually provide strengths and support weaknesses, our proposal is to set up a subset of approaches and features to be brought to cooperation.

3.1 Features

As said, most NLP approaches and applications focus on the lexical/grammatical level as a good tradeoff for expressiveness and complexity, effectiveness and efficiency. Accordingly, we have decided to take into account the following kinds of descriptors:

- single, normalized words (ignoring dates, numbers and the like), that we believe convey most informational content in a text;
- abbreviations, acronyms, and colloquial expressions, especially those that are often found in informal texts such as blog posts on the Internet and phone messages;
- n-grams (groups of n consecutive terms) whose frequency of occurrence in the corpus is above a pre-defined threshold, that sometimes may be particularly meaningful;
- PoS tags, that are intuitively discriminant for subjectivity;
- expressive punctuation (dots, exclamation and question marks), that may be indicative of subjectivity and emotional involvement;
- emoticons, due to their direct and explicit relationship to emotions and moods.

For NLP pre-processing, we used the TreeTagger [17] for PoS-tagging and the Snowball suite [16] for stemming.

All the selected features are collectively represented in a single vector space based on the real-valued weighting scheme of Term Frequency - Inverse Document Frequency (TF-IDF):

$$tfidf(t_i, d_j) = \#(t_i, d_j) \cdot \log_2 \frac{|T|}{\#_T(t_i)}$$

where $\#(t_i, d_j)$ is the number of occurrences of term t_i in document d_j, and $\#_T(t_i)$ is the number of documents in the training set T that include term t_i. To have values into $[0, 1]$ we use cosine normalization:

$$w_{ij} = \frac{tfidf(t_i, d_j)}{\sqrt{\sum_{k=1}^{n} tfidf(t_k, d_j)^2}} \tag{1}$$

where n is the number of terms occurring at least once in the training set documents. To reduce the dimensionality of the vector space, Document Frequency (i.e., removing terms that do not pass a pre-defined frequency threshold) was used as a good tradeoff between simplicity and effectiveness.

3.2 Algorithms

To build the classification model we focused on two complementary approaches that have been proved effective in the literature: a similarity-based one (Rocchio) and a probabilistic one (Naive Bayes).

For each category $c_k \in C$, Rocchio's algorithm creates an explicit profile, reporting the weight of each term in the training set vocabulary, in the form of a 'prototype vector' $p_k = \langle p_{1k}, \ldots, p_{nk} \rangle$:

$$p_{ik} = \beta \cdot \sum_{d_j \in P_k} \frac{w_{ij}}{|P_k|} - \gamma \cdot \sum_{d_j \in N_k} \frac{w_{ij}}{|N_k|}$$

where w_{ij} is the weight reported in the vector space, in our case as defined in (1), P_k is the subset of documents in the training set that belong to category c_k, N_k is the subset of documents in the training set that do not belong to category c_k, and β, γ are parameters that allow to balance the importance of positive and negative instances on the classifier (e.g., taking $\beta = 1, \gamma = 0$ ignores negative examples and returns as a prototype the centroid of the positive ones). A new document is classified simply by comparing its associated vector to all prototype vectors, and taking the category associated to the most similar. Cosine similarity, measuring the angle between two vectors, can be used for this purpose:

$$sim(d_j, p_k) = \frac{\overline{d_j} \cdot \overline{p_k}}{|\overline{d_j}| \cdot |\overline{p_k}|} = \frac{\sum_{i=1}^{n} w_{ij} \cdot w_{ik}}{\sqrt{\sum_{i=1}^{n} w_{ij}^2 \cdot \sum_{i=1}^{n} w_{ik}^2}}$$

It has the advantage of being less affected by the dimensionality of the space and by the normalization applied to the TF*IDF value. Note that consistency in this approach a training example might be classified differently than its known label used for learning.

A Naive Bayes classifier allows to infer the posterior probability $p(c_k|d)$ of a document $d = \langle d_1, \ldots, d_n \rangle$ belonging to a category c_k based on the likelihood of its terms being found in documents that are known to be in that category:

$$p(c_k|d) \propto p(c_k) \cdot p(d|c_k) \approx \frac{|T_k|}{|T|} \cdot \prod_{j=1}^{n} \frac{n_{kj}+1}{n_k+n}$$

$p(c_k)$ is the a priori likelihood of category c_k. Assuming that the categories are disjoint (i.e., each document may belong to only one category), it can be computed as $p(c_k) = \frac{|T_k|}{|T|}$, where T_k is the subset of the training set T that belongs to class c_k. Assuming that the terms in the document are statistically independent from each other (a clearly false assumption, but one that significantly reduces computational demands—whence the term 'naive'), one gets $p(d|c_k) = p(d_1 \wedge \cdots \wedge d_n|c_k)) = \prod_{j=1}^{n} p(d_j|c_k)$ where the posterior probability of terms can be computed as $p(d_j|c_k) = \frac{n_{ij}}{n_i}$, with n_{kj} the number of occurrences of term d_j in documents belonging to category c_k and n_k the sum of all occurrences of all terms in documents of category c_k. The Laplace correction to the relative frequency $p(d_j|c_k) \approx \frac{n_{kj}+1}{n_k+n}$ avoids that $p(t|c) = 0$ if a term t is not present in the documents of a category c, which would yield 0 for the whole product. The category of an unknown document d is computed as the one that maximizes the posterior probability:

$$\arg\max_{c_k \in C} p(c_k) \cdot \prod_{j=1}^{t} p(d_j|c_k)$$

where t is the number of terms that are present in d, and d_j is the j-th document in d.

Our system combines the above approaches in a committee, where each classifier $i = 1, 2$ plays the role of a different domain expert that assigns a score s_k^i to category c_k for each document to be classified. The final prediction is obtained as class $c = \arg\max_k S_k$, considering a function $S_k = f(s_k^1, s_k^2)$ [19]. This approach has the advantage of allowing easy extension with additional classifiers when needed. There is a wide range of options for function f. In our case we use a weighted sum, which requires that the values returned by the single approaches are comparable, i.e. they refer to the same scale. In fact, while the Naive Bayes approach returns probability values, Rocchio's classifier returns similarity values, both in $[0, 1]$.

4 Experiments

Experiments on Opinion Mining were run in [10] on a dataset of 2000 reviews in Italian language, concerning 558 movies, taken from http://filmup.leonardo.it/. The evaluation, expressed as a number of 'stars' (from 1 to 10), associated to reviews was used to distinguish positive (6 to 10 stars) from negative (1 to 5 stars) examples. The corpus included half positive reviews and half negative ones. On a quite mediocre platform (a PC endowed with an Intel Core 2 Duo E6750 working at 2.66 GHz and 2 GB RAM, running Windows 8), using different sets of features, runtime ranged between 3'25" (for 5892 features) and 13'08" (for 9001 features, of which 2784 n-grams). This should ensure applicability of our method even using very cheap resources. The use of n-grams significantly increases the number of features, and runtime as a consequence. Classification performance was evaluated on 17 different feature settings using a 5-fold cross-validation procedure. Equal weight was assigned to all classifiers in the committee. Overall accuracy reported in [10] was always above 81%, and always above 82% for the committee. These are very good results, compared to the state-of-the-art for English and especially for Italian. When Rocchio outperformed Naive Bayes, accuracy of the committee was greater than that of the components; in the other cases, corresponding to settings that used n-grams, Naive Bayes alone was the winner. Even if balanced between positive and negative cases, accuracy on the former was always better than that on the latter. This is somehow surprising, because it is commonly believed that negative emotions are stronger, and hence easier to recognize.

To further evaluate the proposed approach in the perspective of using it for digital libraries, we devised two experiments. One still concerned the Opinion Mining task, but involved the Evalita Sentipolc 2014 dataset. It consists of 4513 tweets, collected by harnessing Twitter messages in Italian with mainly a politic content, encoded as described in [2]. Compared with the previous dataset, it has two peculiarities. First, it is standard in the literature, and was used as a benchmark for state-of-the-art competitions. Second, it involves tweets, that are shorter than the movie reviews, so that it can check the performance of the system on a different ground. Since we were again interested in just distinguishing positive messages from negative ones, neutral items in the dataset were removed, yielding a reduced dataset made up of 2091 tweets (1412 negative and 679 positive ones). In this case we used only the system configuration that provided the best results in the previous experiments:

Normalization	PoS tags	Punct./Abbrev.	n-grams
lemmas	nouns, verbs, adjectives, adverbs, emoticons	Yes	–

We carried out a 10-fold cross validation whose results in terms of Precision (P), Recall (R) and F1-measure ($F1$) are as follows:

Positive			Negative			Average		
P	R	F1	P	R	F1	P	R	F1
0.752	0.498	0.599	0.750	0.901	0.819	0.751	0.700	0.724

In this case we used Precision (P), Recall (R), and F1-measure ($F1$), since the dataset was imbalanced toward negative examples. These are figures that compare well to the state-of-the-art best system in the competition [1]. Recall on positive cases is worse than on negative ones, possibly due to the difference in the number of examples in the two classes of the dataset. Anyway, this may be a useful outcome, because library managers (differently from e-business site holders) may be more interested in identifying and analyzing criticisms than on reading positive comments.

Analyzing the emotions expressed by the users' comments may also be of interest, since it may provide a more precise and detailed account about which sentiment the digital content (document, movie, song, etc.) triggered in the user. To this aim we trained the classifier on three classes, selected to represent the more standard and relevant emotions that items in a library might cause in a user. So, we included one positive emotion (happiness) and two negative ones (for the moment we focused on a lightly negative one, sadness, and a strongly negative one, anger). We used a dataset purposely collected for this experiment by taking 800 comments about movies from filmup and showing them randomly to 11 human raters. The raters were asked to evaluate whether the opinion about the movie expressed one of the three feelings of interest, and in such a case which one. A label was given to each comment according to the majority agreement criterion. Those comments for which majority was not reached were discarded. At the end of this process the dataset included 752 entries (namely: 406 for happiness, 175 for sadness, and 171 for anger). The features for this experiment involved an extended set of Pos tags:

Normalization	PoS tags	Punct./Abbrev.	n-grams
lemmas	nouns, verbs, adjectives, adverbs, articles, pronouns, emoticons	Yes	–

Then we ran a 10-fold cross validation whose results in terms of Precision (P), Recall (R) and F1-measure ($F1$) are as follows:

Anger			Happiness			Sadness			Average		
P	R	F1	P	R	F1	P	R	F1	P	R	F1
0.698	0.408	0.514	0.742	0.870	0.801	0.630	0.575	0.600	0.690	0.617	0.651

The emotion analyzer performs well on Happiness, while its performance is less accurate for the other two classes. This might be due to the fact that the dataset was imbalanced. However, positive emotions are typically harder to recognize than negative ones. This makes us confident that, in any case, combining our classifier with other state-of-the-art ones might improve the overall results.

5 Conclusions

The possibility for people to leave comments in blogs and forums on the Internet allows to study their attitude (in terms of valence or even of specific feelings) on various topics. For some digital libraries this may be a precious opportunity to understand how their content is perceived by their users and, as a consequence, to suitably direct their future strategic choices. So, libraries might want to enrich their sites with the possibility, for their users, to provide feedback on the items they have consulted. Of course, manually analyzing all the available comments would be infeasible. Sentiment Analysis, Opinion Mining and Emotion Analysis denote the area of research in Computer Science aimed at automatically analyzing and classifying text documents based on the underlying opinions expressed by their authors.

Significant problems in building an automatic system for this purpose are given by the complexity of natural language, by the need of dealing with several languages, and by the choice of relevant features and of good approaches to building the models. Following the interesting results obtained for Italian by a system based on a Text Categorization approach, this paper proposed further experiments to check whether reliable predictions can be obtained, both for opinions and for feelings. Experimental results compare well to state-of-the-art tools, suggesting that the proposed approach might be profitably exploited in the target application domain.

To test this hypothesis, future work will include experiments on use cases specifically concerning digital libraries dedicated to art, provided that the users comments are collected and made available by such libraries. We also plan to extend the set of emotions to be recognized, including at least all primary ones.

Acknowledgment. This work was partially funded by the Italian PON 2007–2013 project PON02_00563_3489339 'Puglia@Service'.

References

1. Basile, V., Nissim, M.: Sentiment analysis on Italian tweets. In: Proceeding of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 100–107 (2013)
2. Basile, V., Bolioli, A., Nissim, M., Patti, V., Rosso, P.: Overview of the evalita 2014 sentiment polarity classification task. In: Proceedings of the 4th Evaluation Campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2014), pp. 50–57. Pisa University Press (2014)
3. Carbonell, J.: Subjective Understanding: Computer Models of Belief Systems (1979)
4. Cherry, C., Mohammad, S.M., de Bruijn, B.: Binary classifiers and latent sequence models for emotion detection in suicide notes. **5**, 147–154 (2012)
5. Danisman, T., Alpkocak, A.: Feeler: emotion classification of text using vector space model. In: Proceedings of the AISB 2008 Convention, Communication, Interaction and Social Intelligence (2008)

6. Das, S., Chen, M.: Yahoo! for amazon: extracting market sentiment from stock message boards. In: Proceedings of the Asia Pacific Finance Association Annual Conference (APFA) (2001)
7. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: Proceedings of WWW, pp. 519–528 (2003)
8. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic indexing. J. Am. Soc. Inform. Sci. Technol. **41**, 391–407 (1990)
9. Ekman, P.: Sixteen enjoyable emotions. Emotion Researcher **18**, 6–7 (2003)
10. Ferilli, S., De Carolis, B., Esposito, F., Redavid, D.: Sentiment analysis as a text categorization task: a study on feature and algorithm selection for Italian language. In: IEEE International Conference on Data Science and Advanced Analytics (DSAA), 36678 2015, pp. 1–10 (2015)
11. Lewis, D.D.: An evaluation of phrasal and clustered representations on a text categorization task. In: Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval, pp. 37–50 (1992)
12. Mishne, G., Glance, N.: Predicting movie sales from blogger sentiment. In: Proceedings ofAAAI-CAAW-2006, the Spring Symposia on Computational Approaches to Analyzing Weblogs (2006)
13. Morinaga, S., Yamanishi, K., Tateishi, K., Fukushima, T.: Mining product reputations on the web. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002) (2002)
14. Nasukawa, T., Yi, J.: Sentiment analysis: capturing favorability using natural language processing. In: Proceedings of the Conference on Knowledge Capture (K-CAP) (2003)
15. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 79–86 (2002)
16. Porter, M.F.: Snowball A language for stemming algorithms, October 2001
17. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of International Conference on New Methods in Language Processing, pp. 44–49 (1994)
18. Tong, R.M.: An operational system for detecting and tracking opinions in on-line discussion. In: Proceedings of the SIGIR Workshop on Operational Text Classification (OTC) (2001)
19. Tulyakov, S., Jaeger, S., Govindaraju, V., Doermann, D.: Review of classifier combination methods. In: Marinai, S., Fujisawa, H. (eds.) Machine Learning in Document Analysis and Recognition. SCI, vol. 90, pp. 361–386. Springer, Heidelberg (2008)
20. Turney, P.: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the Association for Computational Linguistics (ACL), pp. 417–424 (2002)
21. Wiebe, J.M.: Learning subjective adjectives from corpora. In: Proceedings of National Conference on Artificial Intelligence (AAAI-2000) (2000)
22. Wiebe, J.M., Bruce, R.F., O'Hara, T.P.: Development and use of a gold-standard data set for subjectivity classifications. In: Proceedings of the Association for Computational Linguistics (ACL-1999) (1999)
23. Wilks, Y., Bien, J.: Beliefs, points of view and multiple environments. In: Proceedings of the International NATO Symposium on Artificial and Human Intelligence, pp. 147–171. Elsevier North-Holland Inc. (1984)

24. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proceedings of ICML-97, 14th International Conference on Machine Learning, pp. 412–420 (1997)
25. Yi, J., Nasukawa, T., Bunescu, R., Niblack, W.: Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques. In: Proceedings of the IEEE International Conference on Data Mining (ICDM) (2003)
26. Yu, H., Hatzivassiloglou, V.: Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2003) (2003)