

# An Emotion-driven Approach for Aspect-based Opinion Mining

Marco Polignano (✉), Pierpaolo Basile, Marco de Gemmis,  
and Giovanni Semeraro

University of Bari “Aldo Moro”, Dept. of Computer Science  
marco.polignano@uniba.it, pierpaolo.basile@uniba.it,  
marco.degemmis@uniba.it, giovanni.semeraro@uniba.it

**Abstract.** The remarkable ability to understand the opinion of a user about a specific topic of discussion allows intelligent systems to provide more specific and personalized suggestions especially when no other information is available. The strategies for opinion mining, also known as sentiment analysis, are in last years topic of in-depth studies. In this work, we present an approach of text mining for detecting the topic of discussion for textual contents and the emotion that the writer feels while writing it. Conversely to the classic strategies of sentiment analysis, we enrich the standard polarity prediction task with more fine-grained information about user’s emotion. By using this information, the final behavior of the personalized system could be designed by taking into account the view about the topic of the specific user. For performing this task, we adopted a hybrid approach which uses both lexicons and semantic representation of sentences for the operation of aspect classification. Training data for the aspects detection module have been extracted from already categorized last year world news. The emotional labeling approach is, instead, based on the posts left by users on Facebook, which have been annotated using the emoticon encountered. The evaluation has been conducted on a dataset of tweets opportunely collected using hash-tags which refer both to the topic of discussion and the emotional opinion.

**Keywords:** Opinion Mining, Sentiment Analysis, Emotions, Aspects Detection, User Modeling, Social Network Sites, Natural Language Processing

## 1 Introduction

The high diffusion of many social platforms has made possible the collection of information about different aspects of the user’s life. In this scenario, starting from user-generated contents, the system should be able to detect the topic of the discussion and the opinion of the user about it. This operation, recently, has become popular with the name of opinion mining and sentiment analysis. A

---

IIR 2018, May 28-30, 2018, Rome, Italy. Copyright held by the author(s).

prominent area of application is the analysis of e-commerce user reviews. Sentiment analysis for the construction of a holistic user profile can take advantages of a more fine-grained emotional annotation of relevant aspects of a topic. In a general domain, such as "travels", it could be interesting to note that the information "the user is scared by traveling", is completely different from "the user is angry about traveling". In the first case, a personalized system will never suggest to the user to take an airplane to far holiday countries. In the second case, it can start to recommend many offers and discounts for traveling again maybe with a different airline company. According to this idea, we suggest the adoption of a general approach for annotating user-generated content with a topic of discussion (if possible, with a specific aspect the user is talking about), and an emotional label. In particular, we focused on eighth general areas of application: technology, travels, style, sports, politics, music, movies, and art. In Sec. 3, we propose a strategy for automatically annotate a sentence with one topic of the discussion, by learning a dynamic semantic representation of words from newspapers. Moreover, in Sec. 5 we use a word2Vec semantic annotation [10] of words for detecting the sentiment of text associated with the aspects recognized. Finally, in Sec. 6, the approach has been evaluated on a set of tweets annotated with hash-tags corresponding to the considered topics and emotions.

## 2 Related Work

The opinion mining is a complex task which commonly merges the results of two sub-tasks which are resolved independently and combined in a single result only at the end of the process. The first is the aspect opinion extraction; the second is the sentiment annotation.

### 2.1 Approaches for aspect opinion annotation

The opinion mining on a textual content has the purpose of discovering one or more aspects or entities which the writer is talking about. A common approach for dealing with opinion extraction is *topic modeling* [3] based on unsupervised probabilistic models such as pLSA (Probabilistic Latent Semantic Indexing) [5] and LDA (Latent Dirichlet Allocation) [1]. In each document, a latent aspect is chosen according to a multinomial distribution, controlled by a Dirichlet prior  $\alpha$  [14]. In that way it is possible to estimate the probability that a word is used in a specific topic of discussion, providing the information required for the task of words classification and aspects extraction. The weakness of the process is based on the lack of precision and internal cohesion of the topic words which in a domain of opinion mining can be an important missing characteristic. Despite this classic approach, recently supervised techniques, based on Neural Network are showing interesting performance improvements over the task. A common procedure is the use of word embeddings [10] and NLP features, such as words and bi-grams for feeding that algorithms. Liu [8] uses the RNNs (Recurrent Neural Networks) and amazon embeddings for successfully identify the opinion

target over the SemEval2014 competition<sup>1</sup>. In this work, we will follow the idea of using word-embeddings and natural language elaboration of text for obtaining relevant features for annotating sentences with the correspondent topic of discussion using an SVM classifier.

## 2.2 Emotional labeling for opinion mining

The emotional labeling of aspects, detected in spans of text, is commonly approached by strategies of polarity annotation (positive, negative, neutral / objective). Most of them are based on the polarity score associated with each word in a standard lexicon or thesaurus [6]. There are many types of lexicons at state-of-the-art that contain nouns, adverbs and verbs, uni-grams, bi-grams and also misspelling and morphological variants annotated with a numeric value which can be in a range  $[-1, 1]$ . In all that cases the main limitation is the presence of only most-used frequent words in a standard domain (news documents). In real-world domains, such as that of reviews or social media, they are not able to correctly detect all the different contractions and variations in words which are encounter. A strategy of acting is, conversely, provided by machine learning algorithms which have been commonly used for the task in the last decade. One of the pioneers has been Pang [12] that classifies reviews utilizing an SVM and a Naive Bayes algorithm for the binary classification of subjective sentences. Approaches based on an ensemble of classifiers have also been proposed. Wang [16], combines five different learners: Naive Bayes, Maximum Entropy, Decision tree, K-NN, and SVM. Most of these approaches, formalize the sentences considering only the syntactic aspect of the words, ignoring the semantic. Conversely, Tang [15] proposed a strategy for learning sentiment-specific words embeddings able to formalize the semantic differences between opposite polarity words using three neural networks for learning sentiment-specific word embedding (SSWE). The approach was evaluated on a dataset of Tweets, comparing it with some of the most famous classical approaches based on SVM and Naive Bayes obtaining breaking results. These outcomes demonstrated the importance to include word embeddings in strategies of text classification, supporting the interest showed in this work for that approach.

## 3 Opinion mining based on user emotions

The state of the art described in Sec. 2 points out many issues about opinion mining, including the limitations derived by annotating a single review with only a binary polarity label. In some cases, to overcome the limit more fine-grained intervals of polarity have been added such as "almost positive" or "almost negative", but the information about what kind of emotion the user is trying to expose is still missing.

The opinion mining strategy proposed in this work aims to define a general

---

<sup>1</sup> <http://alt.qcri.org/semeval2014/task4/>

model for annotating textual sentences, including textual sources which are not reviews, with a macro topic of discussion and an emotional label. We decided to adopt the Ekman model of emotions [4] which include six primary emotional reactions: joy, sadness, fear, disgust, surprise, anger. We decided, furthermore, to assume that each specific sentence to process has been already analyzed and removed from the processing pipeline if it has been considered neutral/objective. Each sentence, in our discussion, is deemed to be self-reliant and about only one topic of discussion with a main emotional opinion. These assumptions help us to do not focus the work on common issues already deeply approached in state-of-the-art, and, on the contrary, to exhaustively describe the two annotation strategies proposed. The process starts with some self-reliant sentences provided as an input of the "Emotion-driven Opinion Mining" pipeline. The sentences provided are pre-processed for removing additional extra characters which can produce noise for the analysis. In particular, we removed "HTML" characters, emails and not UTF-8 letters. Hashtags, commonly used in Twitter, are replaced with the special sequence *\_TAG\_* concatenated with the original piece of text. Users cited through @ are, instead, replaced with the *\_ENTITY\_* sequence. A similar approach is used for links which are replaced with *\_URL\_*. The "emoji" in the text have been opportunely annotated using the library "Java-Emoji" <sup>2</sup> in the format " :< *nameOfEmoji* >:". The text opportunely pre-processed is then analyzed independently by the two annotation modules which are detailed in following sections. The result provided by the system is then a topic label among technology, politics, travel, style, sport, music, movies, art; and an emotional tag among joy, fear, sadness, anger, surprise, disgust. These labels can be consequently used for further operations of profiling or personalization.

## 4 Topic detection module

The module of topic detection task usually involves approaches to span analysis or aspect detection. In our approach, we consider only one topic of discussion at a sentence level. Consequently, it is possible to formalize the topic detection as a multi-class classification problem. We decided to use a supervised approach based on SVM as consequence of the optimal results obtained by this algorithm in literature. Eight main topics of discussion have been considered relevant: **technology, politics, travel, style, sport, music, movies, art**; but the set is easily extensible when new data about different topics are provided as a training set of the model. In order to collect enough data for training the model, we developed a crawler of web news already classified by humans in that categories. In particular, we chose "flipboard.com" <sup>3</sup> as a platform candidate for the task. The website provides a set of thematic news classified in a huge number of categories by the user which shares or writes it. They are free to be read, and they are available through "RSS". When invoked, it provides a list of the last ten news about a specific topic published on the social platform. We collected

<sup>2</sup> <https://github.com/vdurmont/emoji-java>

<sup>3</sup> <https://flipboard.com/>

all the articles about the defined topics from the *first of June 2017 to the 31st of December 2017*. For each one, we stored the abstract provided by the RSS system, and the content of the article obtained by visiting its published link and stripped by "HTML" tags using Jsoup library <sup>4</sup>.

**Dataset.** The dataset collected contains in total more than 400.000 articles subdivided as follow, and it is available at: [http://bit.ly/EOM\\_dataset](http://bit.ly/EOM_dataset)

**Table 1.** Composition of the Flip2017 dataset

	<i>n.news</i>	%	<i>resized</i>	<i>training</i>	<i>dev</i>	<i>test</i>
<i>art</i>	24.900	6,12%	6.895	4.137	1.379	1.379
<i>movie</i>	36.280	8,91%	10.046	6.028	2.009	2.009
<i>music</i>	43.263	10,63%	11.980	7.188	2.396	2.396
<i>politic</i>	40.096	9,85%	11.103	6.662	2.221	2.221
<i>sport</i>	116.474	28,61%	32.253	19.352	6.451	6.451
<i>style</i>	71.232	17,50%	19.725	11.835	3.945	3.945
<i>tech</i>	21.444	5,27%	5.938	3.563	1.188	1.188
<i>travel</i>	53.410	13,12%	14.790	8.874	2.958	2.958
	<b>407.099</b>	<b>100%</b>	<b>112.730</b> <b>27,7% tot.</b>	<b>67.639</b> <b>60% of res.</b>	<b>22.547</b> <b>20% of res.</b>	<b>22.547</b> <b>20% of res.</b>

Each news collected is formalized through its *link*, *title*, *summary* and *textual content* in English language. The table 1 describes the distribution of the articles among the categories and the section of the dataset used for training the algorithm. It is important to note that it has been resized for dealing with the computational cost necessary for processing them. In particular, only the 27,69% of the original dataset has been used, keeping intact the original ratio which identifies the sports category as the most discussed on the web. Moreover, the 60% of the resized dataset has been used as a training set of the method. The last 40% is used half as dev dataset for the optimization of the parameters and the last half as a training set.

**Methodology.** The implementation of the SVM algorithm used in this work is the one provided by LibSVM <sup>5</sup> [2]. We used the C-Support Vector Classification for solving a Multi-class Classification problem approached with a 'one-against-one' implementation. For each pair of classes a C-SVM is constructed for optimizing the error minimization problem:

$$\min_{w,b,\varepsilon} \frac{1}{2} w^t w + C \sum_{i=1}^l \varepsilon_i \quad (1)$$

<sup>4</sup> <https://jsoup.org/>

<sup>5</sup> <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>

where  $w$  are the features vectors for each  $x_i, i = 1, 2, \dots, l$  training vectors,  $\epsilon$  is the error and  $C > 0$  is the regularization parameter. This observation points out the necessity to optimize the choice of  $C$  for obtaining accurate results. Moreover, changing the dimensionality and the content of the features vectors  $w$  it is possible to obtain completely different results.

**Vectorial representation of text.** We modeled the representation of the content of articles using both a binary n-grams representation and word embeddings features vectors. The transformation of the plain-text into unigrams and bigrams have been implemented through the Apache Lucene Standard Analyzer class <sup>6</sup>. The class uses a default English set of stop-words and a grammar-based tokenizer with a max length of tokens set to 2. For each n-gram, we use its frequency in the document as a numerical feature. The formalization through a vector of word embeddings has been approached averaging the singular embedding vectors of the words encountered in the document into only one per document. In particular, the word-embedding procedure used is word2vec [9] introduced by Mikolov. In particular, we used the whole corpus of 400K news for training, through CBOW (Continuous Bag-of-Words) and ten epochs of learning, 100 dimensionality vectors of words with a minimum number of occurrences in the collection equal to 5. Moreover, for including the influence of the writing style used on Twitter we used a corpus of 1.3 millions of generic tweets collected during the week from the 22nd to the 28th of January 2018 for training, at the same way, 50 dimensionality vectors. The final vectorial representation of each news is then the concatenation of three groups of features: bag-of-words (unigrams and bigrams), 100 features of word2vec space learned on the news, 50 features of word2vec space learned on tweets.

**Optimization of parameters.** The model has been optimized considering two parameters: the  $C$  regularization parameter of the SVM and the features used for representing the information. We started testing four settings for the  $C$  value: 1, 2, 4, 8 fixing the representation of the data as the concatenation of the all features available. After we found the better  $C$  value, we tested the same algorithm on the news represented respectively by bag-of-words, bag-of-words + embeddings news, bag-of-words + embeddings tweets, bag-of-words + embeddings news + embeddings tweets. With this experimental setting, we aim to understand if the inclusion of embeddings can support a better classification process. The evaluation, in both the optimization experiments, has been conducted on the "dev" dataset of the news extracted from Flipboard (Tab. 1). We used as evaluation metrics the accuracy of the classifier and the *macro* Precision, Recall and F1-measure. The macro metrics are obtained just averaging the values of evaluation obtained for each class. In particular, for the F1 measure, its macro version is obtained calculating it on averaged precision and recall. On the contrary, micro metrics are calculated summing all the true positives, true negatives, false positives and false negatives obtained for each class and finally calculating a sin-

---

<sup>6</sup> <https://lucene.apache.org/core/>

gular metric value. The results show that a C value equal to 2, allows obtaining better results considering accuracy, precision, recall and F1 measure. Moreover, the best formalization of information is obtained using bag-of-words and both news, tweets word2vec embeddings. The final configuration used for the topic detection module is then the one with a C value equal to 2 and all the three group of features available.

**Internal evaluation of performance.** The subset of news kept as "test" dataset have been evaluated with this obtained configuration of the classifier in order to be sure about its performance also on the news unrelated to the optimization process. It is interesting to note that the results (Accuracy: 0.9195, P: 0.9137, R: 0.8965, F1: 0.905) preserve the performance obtained in the optimization tests. Moreover, the values of the accuracy obtained on that dataset are statistically better for  $p < 0.05$ , using a two-tailed Chi<sup>2</sup> test, than the obtainable predicting the topic label randomly (0,125) or using the most frequent class in the dataset (Sport 0,286). The results allow us to confirm the validity of the approach.

## 5 Emotion detection module

The emotion detection module aims to detect the emotion expressed by each piece of text according to what writer wants to communicate. As consequences of the assumption of self-reliant and mono-thematic made in this work, we detect the presence of only one main emotion associated with the sentence. The task has been approached with the methodology of dynamic lexicon describe in [13]. In this work, we extended the already presented model, going to include a pre-processing emoji/emoticons classifier and enriching the set of dynamic words used for each emotional class with those available in a state-of-the-art NRC lexicon [11]. Moreover, we trained our word2vec probabilistic space also over the generic Tweeter dataset, concatenating the two set of features in a way similar to them used in the topic detection module (Sec. 3).

**Methodology.** The approach used for the task is articulated in four steps: classification of sentences which come from the training dataset with emotional labels using emoji/emoticons; generation of the emotional lexicon from the annotated sentences; creation of word embedding centroids with the most frequent words for each class; use of a similarity function for detecting the most similar centroid for the piece of text to annotate. The approach used for text labeling is based on the idea that it is possible to automatically annotate user's post on Social Media Sites (SMSs) using *emoticons*. Each smile has been associated with a main emotional sense with a straightforward strategy. We identify the most common emotions<sup>7</sup> included in UNICODE standard version 9 and we manually classify them (also considering their main writing variations). Moreover,

---

<sup>7</sup> <https://it.wikipedia.org/wiki/Emoticon>

we extended this list with most of the new emoji codes <sup>8</sup> which are, as previously, associable with a main emotional class. Posts which contains discordant emoji have been excluded. Due to the page limits of the article, the full list of emoji/emoticons used is not here presented but the list is available in [13]. The sentences annotated by the approach just described, are divided into six independent lists, one for emotion. The sentences within each list are tokenized using TweetNLP, and the frequency list of each token is calculated. The frequencies are normalized over the maximum frequency detected in the list, and the tokens are ranked according to the values obtained. Only the tokens whose frequency is higher than 1% of the number of whole sentences are considered relevant. Finally, words falling in more than one list are removed, to keep only the distinctive words for each class. The weighted list of words is used as a lexicon which fully represents each emotional class. The learning phase consists in computing the word2vec centroid of each emotional class. The six lists of words previously generated are the sources used as input lexicon. The lexicon is then subdivided into six meta-documents, and each one is transformed in a word2vec centroid by averaging the vectors associated with each word belonging to the meta-document. Given a class (e.g., *joy*), each word in this list is associated with word an embedding vector and then averaged with others. For removing the dependency of each word from the centroid of the whole distributional space, we subtract it at every step of the sum. The average is weighed with the relative frequency associated with each word [13]. The annotation phase is based on the Tanimoto Similarity between the numerical vector of the considered piece of text and the numerical centroid of each class. The label of the corresponding highest score will be provided as output.

**Dataset.** The datasets used for the approach is *myPersonality* <sup>9</sup> [7]. The *myPersonality* dataset contains information about 4 millions Facebook users and 22 millions posts over their timeline. The dataset annotated have been subdivided into training and test. We kept 20.000 sentences for each class randomly chosen as posts for creating the six emotional meta-documents. Moreover, 1.000 phrases for each emotional class have been used for evaluating the internal validity of the approach. We modeled the representation of SMSs posts using word embeddings features vectors. In particular, we used the whole corpus of 2.4M annotated posts in myPersonality for training, through CBOW and ten epochs of learning, 100 dimensionality vectors of words with a minimum number of occurrences in the collection equal to 5. Moreover, as already done in Sec. 3 for the topic detection module, we extended the data representation used in [13], concatenating, for each word, another word2vec vector of 50 features learned on the dataset of 1.3M generic tweets.

**Extension of centroids words with a standard lexicon.** In the approach used for this emotion detection module, we manipulated the list of words in each

---

<sup>8</sup> <https://unicode.org/emoji/charts/full-emoji-list.html>

<sup>9</sup> <http://mypersonality.org/>



meta-document. First of all, we limited the words obtained by the analysis of annotated posts, to the 100 most frequent. This decision has been taken considering that the huge number of words can create centroid vectors too much similar each other, losing part of its semantic information. Moreover, we decided to enrich each list with the 20 words extracted from a standard lexicon most frequent in our training dataset. The lexicon adopted is NRC Word-Emotion Association Lexicon. It is an extensive English dictionary in which each word is linked to one or more emotions among the eight available (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust).

**Emoji-based classifier.** This simple classifier uses the presence of smiles inside the text as the main element to consider for assigning a primary emotion to the phrase. A counter for the correspondent emotion is incremented every time that an emoji/emoticon, previously manual classified in the six fixed emotional classes, is found in the text. This counting is used for defining an order of the most probable emotion associated with the phrase. If two or more emotions obtain the same number of increments, then the component will be not able to classify the text correctly, and the sentence will be processed by the approach previously described. On the contrary, the emotional label provided as output by the emotion detection module will be the class predicted by the Emoji-based classifier.

**Internal evaluation of performance.** Unlike what presented in [13], in this work the semantic representation of posts has been extended, the lexicon used has been integrated, and the emoji classifier has been added as a pre-classification module. These changes have been internally evaluated on the portion of myPersonality dataset reserved for testing purposes. It has been observed a shallow increment of performances for the new version of the approach called DYN-TH+ (DYN-TH Average F1: 0,3549; DYN-TH+ Average F1: 0,3611).

## 6 Evaluation of the approach

The complete pipeline of "Emotion-driven Opinion Mining" has been evaluated through a dataset of 4.898 tweets collected using predefined hash-tags. In particular, messages which contain both hash-tags about emotions and has-tags about the topic of interest have been crawled. For the emotional hash-tags we used: #happy, #sad, #anger, #fear, #disgust, #surprise. For the topic related hash-tags, we defined for each one a list of words which refers to micro-topic relevant for it. As an example, for the topic technology, we used the hashtags: #tech, #technology, #phone, #sony, #iphone, #nokia, #samsung, #microsoft, #linux, #mac, #ios, #windows, etc. The crawling task started on the 20th of January 2018 and ended the 20th of February 2018.

Observing the Tab. 2 it is immediately evident what is the main problem of the dataset collected. The "joy" tweets are 71.74% of the whole dataset which results very unbalanced. Moreover, messages associated with disgust and surprise

**Table 2.** Composition of the evaluation dataset

	<i>joy</i>	<i>sadness</i>	<i>fear</i>	<i>anger</i>	<i>disg.</i>	<i>surp.</i>	<b>Tot.</b>	<b>%</b>
<i>art</i>	733	195	50	15	0	29	<b>1022</b>	20,87%
<i>movie</i>	129	42	22	9	2	34	<b>238</b>	4,86%
<i>music</i>	819	148	24	4	0	38	<b>1033</b>	21,09%
<i>politic</i>	87	363	78	21	2	15	<b>566</b>	11,56%
<i>sport</i>	802	28	122	20	0	15	<b>987</b>	20,15%
<i>style</i>	262	8	2	0	1	26	<b>299</b>	6,10%
<i>tech</i>	301	8	14	1	0	11	<b>335</b>	6,84%
<i>travel</i>	381	13	11	0	0	13	<b>418</b>	8,53%
<b>Tot.</b>	<b>3514</b>	<b>805</b>	<b>323</b>	<b>70</b>	<b>5</b>	<b>181</b>	<b>4898</b>	100%
<b>%</b>	71,74%	16,44%	6,59%	1,43%	0,10%	3,70%	100%	-

are very low and inevitably challenging to detect. For this reason, we decided to do not consider them in the evaluation. Despite the highlighted issues, we decided to use it for the evaluation due to the difficulties to find another dataset already annotated with both topics and emotional labels. Besides, the dataset has been collected directly from the web, and it reflects what is the real behavior of the users on SMSs. They commonly tend to share more often positive things than negative when communicating on that platforms. This characteristic has been considered an added value more than an issue which on the contrary is often present in artificial or ad-hoc datasets.

**Table 3.** Results obtained by the Emotion-driven Opinion Mining approach

<b>Module</b>	<b>Accuracy</b>	<b>P</b>	<b>R</b>	<b>F1</b>
Topic Detection	0,6762	0,6582	0,6364	0,6472
Emotion Detection	0,5587	0,2688	0,2613	0,2650
Emotional Opinion Mining	0,3498	0,1691	0,1394	0,1528

The results resumed in Tab. 3 confirm our hypothesis about the difficulties of the task in a real scenario. The topic detection module performed an accuracy of 0,6772 that is about the 30% less than the results obtained in internal evaluations, as the same of the other measures. Considering that the classification task has to work with eight different classes and spontaneous messages, we can hold to be true the result obtained. An issue observed during the annotation is the difficulty of the classifier to match domain relevant entities, as consequence of the pre-processing step performed on the text. Entities such as "Barack Obama", often encountered during the training phase, are never matched in tweets because they are often recalled using hash-tags or "@ citations", which are automatically transformed in a unique sequence of characters entirely different from the

real entity name. In a similar way, the emotion detection module obtained an accuracy of 0,5587 and a macro F1 measure of 0,2650 which, also in this case, is about the 30% less than the internal evaluation. In that case, we note that the module does not work well on emotions with associated few messages which result challenging to recall. We consequently encourage the further studies to use larger datasets for increasing that statistic. The complete pipeline performed on Twitter messages produces results which could be interpreted as lower than expected, but they are obtained considering as "true positive" only the annotation which correctly predicts both the labels. This strong assumption consequently caused a significant reduction of the performance measured because the module should be able to detect one pair {topic,emotion} among the possible 48 (eight aspects \* six emotions). The accuracy of 0,3498 is in line with the difficulties of the task and, undoubtedly better than a classic random strategy which can obtain only a result of 0,0250 and the use of the most frequent pair (joy; music) that can obtain an accuracy of 0,1672. The differences in accuracy obtained are statistically significant using a two-tailed Chi<sup>2</sup> test for  $p < 0.05$ . The results are consequently very encouraging and can be considered as one of the first baselines for further studies about Emotional-driven Opinion mining approaches.

## 7 Conclusion

The proposed approach of Emotion-driven Opinion Mining has demonstrated to be an effective strategy for:

- discovering the topic of discussion of a message from Social Media Sites
- detecting the primary emotion related with the social textual content

Traditional opinion mining strategies are commonly applied in the domain of reviews where fixed sets of aspects and internal structures of the comments can support a more accurate detection. Moreover, the user opinion is annotated using a polarity level which does not provide a fine-grain overview of the emotional state of the user. Producing more detailed emotional annotations is a challenging task, but nowadays they are essential features for enriching the user profile and to adopt in personalized systems. In this work, we provide a complete pipeline which, starting from the plain text, can accomplish this task. The proposed module of topic detection is based on an SVM algorithm trained on a dataset of more than 400k real news classified according to eight topics: technology, politics, travel, style, sport, music, movies, art. They have been processed and formalized through a conjunct vector of unigrams, bigrams, and word embeddings. The emotions detection module is based on word embedding centroids of the most significant words of a specific emotional class dynamically detected by a corpus of spontaneous messages: ~2M of the Facebook post and ~1.3M of tweets. The result of the final Emotion-driven Opinion mining has shown baseline results for the challenging task on a dataset of 1k annotated tweets. To encourage further studies in this area of Opinion Mining and to ensure reproducibility of our experiments, we distributed all the datasets which have been collected and used in this work. They are available at: [http://bit.ly/EOM\\_dataset](http://bit.ly/EOM_dataset)

## References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *Journal of machine Learning research* 3(Jan), 993–1022 (2003)
2. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 27:1–27:27 (2011), software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
3. Crain, S.P., Zhou, K., Yang, S.H., Zha, H.: Dimensionality reduction and topic modeling: From latent semantic indexing to latent dirichlet allocation and beyond. In: *Mining text data*, pp. 129–161. Springer (2012)
4. Ekman, P.: An argument for basic emotions. *Cognition & emotion* 6(3-4), 169–200 (1992)
5. Hofmann, T.: Probabilistic latent semantic analysis. In: *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. pp. 289–296. Morgan Kaufmann Publishers Inc. (1999)
6. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 168–177. ACM (2004)
7. Kosinski, M., Matz, S.C., Gosling, S.D., Popov, V., Stillwell, D.: Facebook as a research tool for the social sciences: Opportunities, challenges, ethical considerations, and practical guidelines. *American Psychologist* 70(6), 543 (2015)
8. Liu, P., Joty, S., Meng, H.: Fine-grained opinion mining with recurrent neural networks and word embeddings. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pp. 1433–1443 (2015)
9. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013)
10. Mikolov, T., Yih, W.t., Zweig, G.: Linguistic regularities in continuous space word representations. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pp. 746–751 (2013)
11. Mohammad, S.M., Turney, P.D.: Crowdsourcing a word-emotion association lexicon 29(3), 436–465 (2013)
12. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. pp. 79–86. Association for Computational Linguistics (2002)
13. Polignano, M., de Gemmis, M., Narducci, F., Semeraro, G.: Do you feel blue? detection of negative feeling from social media. In: *Conference of the Italian Association for Artificial Intelligence*. pp. 321–333. Springer (2017)
14. Poria, S., Cambria, E., Gelbukh, A.: Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems* 108, 42–49 (2016)
15. Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., Qin, B.: Learning sentiment-specific word embedding for twitter sentiment classification. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. vol. 1, pp. 1555–1565 (2014)
16. Wang, G., Sun, J., Ma, J., Xu, K., Gu, J.: Sentiment classification: The contribution of ensemble learning. *Decision support systems* 57, 77–93 (2014)