Aspect-Based Unsupervised Negative Sentiment Analysis

Mainak Ghosh, Kirtika Gupta, and Seba Susan

Abstract Twitter is a social media platform where users post their opinions on various events, products, services and celebrities. Automated analysis of these public posts is useful for tapping into public opinion and sentiment. Identifying negative public sentiment assumes importance when national security issues are at stake or when critical analysis of a product or policy is required. In this paper, a method is introduced that classifies tweets based on their negative content, without any prior training. Specifically, an unsupervised negative sentiment analysis is presented using an aspect-based approach. Phrase and keyword selection criteria are devised after identifying fourteen valid combinations of part-of-speech tags listed in a prioritized order, that are defined as phrase patterns. A sliding text window is passed through each sentence of the tweet to detect the longest valid phrase pattern. The keyword indicating the aspect information is detected using a dependency parser. SentiWordNet lexicon is used for scoring the terms in the detected keyword and phrase combination. The scores are summed up for each sentence of the tweet and transformed nonlinearly by a modified sigmoid function whose output is in the range [−2, 2]; this value comes out to be negative for negative tweets. The utility of our method is proved by superior results as compared to the state of the art on the benchmark SemEval 2013 twitter dataset.

Keywords Sentiment analysis · Negative sentiment · Aspect-based sentiment analysis · Unsupervised method · Dependency parser

1 Introduction

There is a continuous growth of textual content in social media platforms which has led to research on finding automatic ways for processing this information. Analysing the sentiments of social media posts allows us to understand public opinion behind

M. Ghosh (✉) · K. Gupta · S. Susan
Department of Information Technology, Delhi Technological University, Bawana Road, Delhi 110042, India
e-mail: mainakgh1@gmail.com

J. Hemanth et al. (eds.), *Intelligent Data Communication Technologies and Internet of Things*, Lecture Notes on Data Engineering and Communications Technologies 57,
https://doi.org/10.1007/978-981-15-9509-7_29

crucial topics [1]. Sentiment analysis involves the classification of tweets as negative, positive and neutral. In this paper, exclusively focusing on negative sentiment analysis because, in recent times, an increase in the number of online posts that promote hatred and discord in society is observed.

There are two types of learning methodologies employed for sentiment analysis, namely supervised and unsupervised. Examples of supervised machine learning tools are naïve Bayes, logistic regression, support vector machines (SVM) and long short-term memory (LSTM) [2–5]. Examples of unsupervised methods are fuzzy logic-based approaches that interpret the SentiWordNet scores of tweets to make a decision [6, 7], and aspect-based analysis [8–11] which is the approach undertaken in this paper. A brief discussion on aspect-based works in the literature is included in Sect. 2 prior to introducing the proposed model that is based on deriving phrase patterns and keywords from tweets and scoring them with the help of the SentiWordNet lexicon. This approach generally involves determining the head aspect or issue subject of opinion phrases in a sentence. The organization of this paper is as follows. The proposed work is introduced in Sect. 2, the experimentation and the results are discussed in Sect. 3, and the conclusions are given in Sect. 4.

2 Proposed Work

The aspect-based analysis aims to determine the subject issue or head aspect in the given text to understand the topic of discussion. One of the significant works in this regard is that of Canonical Conditional Random Fields (CRF) [12] that locates an opinion phrase in every sentence where the head aspect occurs. However, it is not necessary that all opinion phrases, detected in this manner, would contribute to useful information. Some other related works are: [13] that extracts subjective verb expressions, and [14] that extracts keyphrases in a supervised manner. Mukherjee [11] tried to work around the shortcomings by adopting a feature-based approach for representing the head aspect. The semantic features derived in the generative phase are fed into the discriminative phase for issue-based sentence classification. A labelled dataset of aspect-based opinion phrases was generated for the purpose. In [8], the sentiment score of a sentence was obtained by averaging the individual sentiment scores of each aspect at the clause level. This was done under the assumption that a single sentence may express different sentiments towards different aspects of the movie. The focus in [9] was to determine user sentiments of different aspects of a product from tweets. A method that finds special mention is the unsupervised aspect analysis for sentiment detection in [10] since the theme is closely related to our method. They used a specially labelled dataset with annotations of both aspect and sentiment. The top-ranked words for each aspect were shortlisted based on the probabilities generated by a latent Dirichlet allocation (LDA) model. A polarity score was assigned to each noun–adjective pair to compute the overall sentiment. In our aspect-based approach, a sentiment-scoring function for tweets is used that is explained in more detail below.

2.1 Process Flow for the Negative Sentiment Analysis Task

The block diagram in Fig. 1 shows an overview of the generic negative sentiment analysis task for the SemEval 2013 twitter dataset. Text preprocessing, feature extraction and sentiment computing are the major chunks of the generic negative sentiment analysis task as noted from Fig. 1. The text preprocessing shown, prior to the feature extraction, is common to our task. Synsets from the SentiWordNet lexicon [15] provide positive, negative and neutral scores for each word in a tweet. The maximum of these three scores has been taken for each word assuming that the maximum score reflects the real nature and context of a word. The labels of tweets have been considered as 'negative' and non-negative' with the 'neutral' category considered as 'negative'. The first phase is the preprocessing phase, and the first step is slang and abbreviation removal from each sentence. A list of abbreviations along with their full forms is maintained. This list was taken from the slangs and abbreviation list of webopedia and can be accessed online at [16].

Each tweet is scanned for words in the abbreviation list that are replaced by its full form. Once all the abbreviations are replaced, each word is converted to lower-case followed by tokenizing the text. The next step is spelling correction. The *symspell* checker using the *symspell* inbuilt dictionary [17] with a maximum distance of 2

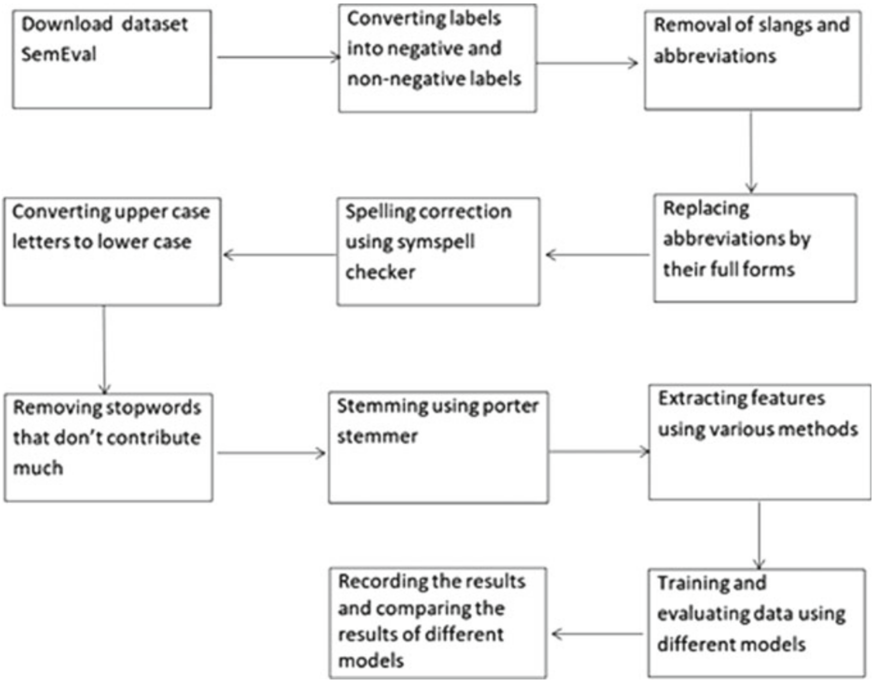
Download dataset SemEval
Converting labels into negative and non-negative labels
Removal of slangs and abbreviations
Replacing abbreviations by their full forms
Spelling correction using symspell checker
Converting upper case letters to lower case
Removing stopwords that don't contribute much
Stemming using porter stemmer
Extracting features using various methods
Training and evaluating data using different models
Recording the results and comparing the results of different models

Fig. 1 Process flow of the negative sentiment analysis task

Table 1 Our fourteen POS tag categories for detecting sentiment phrases

1	VERB
2	NOUN
3	ADJECTIVE
4	ADVERB + NOUN
5	ADVERB + ADJECTIVE
6	ADVERB + VERB
7	MODAL + ADJECTIVE
8	MODAL + VERB
9	ADJECTIVE + NOUN
10	ADJECTIVE + ADJECTIVE
11	ADVERB + ADVERB + ADJECTIVE
12	ADVERB + ADVERB + VERB
13	VERB + ADVERB + ADVERB
14	COORDINATING CONJUNCTION

is used, to edit changes in the spelling. The words are then tagged along with their part-of-speech (POS) tags using the NLTK library [18]. POS tags that do not give any sentiment information are identified and remove them from our text corpus. These tags are identified as [‘EX’, ‘FW’, ‘LS’, ‘NNP’, ‘WP’, ‘WP$’, ‘NNPS’, ‘POS’, ‘PRP’, ‘PRP$’, ‘TO’, ‘WDT’, ‘WRB’] according to [19]. The remaining POS tags are the ones that are based on which the sentences can be scored. An aspect is a subject that is spoken about in the sentence. For example, the sentence ‘The food was very good but the ambience was not very nice’, here the aspect is ‘food’ and ‘ambience’ and the words describing them are ‘very good’ and ‘not very nice’, respectively. Since a single sentence contains both positive and negative sentiments, a prioritized scoring system is devised by identifying 14 types of phrases containing combinations of POS tags ADJECTIVES, VERBS, NOUNS, MODALS and ADVERBS, as shown in Table 1.

The idea is to capture the above phrases in a sentence and then score it. A sliding text window is iterated over the tokenized sentence until the variable token reaches the last token in the sentence. There is a possibility that multiple tag patterns may be associated with the same keyphrase in the given sentence. If more than one pattern in Table 1 exists in a sentence then the longest one is chosen, as the phrase ‘not very nice’ in the context of ‘ambience’ keyword, in the example cited above. Preference is given in the order from bottom to top in Table 1. In case, the length and preference order of two patterns is the same, and the first pattern is chosen.

For example, consider the sentence *She was not very good at basketball*. The phrase patterns derived are—(i) *very good* → ADVERB+ADJECTIVE (ii) *not very good* → ADVERB + ADVERB + ADJECTIVE. Since both (i) and (ii) are valid tag patterns present in Table 1, our system selects the longest possible tag pattern in the sentence, i.e. ADVERB + ADVERB + ADJECTIVE (*not very good*) as the phrase pattern to be scored. The SentiWordNet scores of the terms (*not, very, good*) are multiplied to get a sentiment score for the phrase pattern which comes out to be a negative value due to the presence of *not*.

A corpus of verbs, adverbs, modal, conjunctions, noun and adjectives were compiled using wordnet [20]. An ordered dictionary was kept for each category of verbs, adverbs, modal, conjunctions, noun and adjectives so that the searching can be performed faster. The category of a word is identified by examining the allowed POS tags in that category. For POS tags have been referred to the English Penn Treebank [21]. The allowed POS tags for each category are compiled in Table 2 for verbs and Table 3 for adjectives and adverbs, respectively.

The dependency parser [22] has been used to get the syntactic dependency between the tokens in each statement. The resulting aspects need to belong to this class of dependency parsing tags shown in Table 4. For more reading, the readers are referred to [23] that enlists examples for these dependency parsing tags. Finding links between tokens and relationship-tagging gives more meaningful classification than single keyword-based classification systems [24] or bag-of-word features where the order in which words occur is not known [25]. A sentence when processed by the dependency parser may detect zero or more of these dependency tags.

The preference of selection of these tags, as the keyword, is from top to bottom in Table 4. Further sub-categories or extended POS tagging for each of these dependency tags are kept in our accept states as shown in Table 5. The keyword that matches a category in both Tables 4 and 5 qualifies as the aspect information and is included for sentiment-scoring along with the detected phrase.

Table 2 Acceptable POS tags for the verb category

VB	BASE FORM OF A VERB
VBD	PAST TENSE OF BASE FORM OF VERB
VBG	PRESENT PARTICIPLE OF BASE FORM OF VERB
VBN	PAST PARTICIPLE VERB
VBP	VERB, NON-3RD PERSON SINGULAR PRESENT
VBZ	VERB, 3RD PERSON SINGULAR PRESENT
MD	MODAL AUXILIARY (could, cannot etc.)

Table 3 Acceptable POS tags for the adverb and adjective categories

JJ	ADJECTIVE
RB	ADVERB
RBR	COMPARATIVE ADVERB
RBS	SUPERLATIVE ADVERB

Table 4 Acceptable syntactic dependency tags

nsubj	NOMINAL SUBJECT
dobj	DIRECT OBJECT
iobj	INDIRECT OBJECT
pobj	PREPOSITION OBJECT

Table 5 Results on the SemEval 2013 dataset

Model	Accuracy (%)	Precision	Recall	F-score
Logistic regression + bag of words [28]	52	0.29	0.65	0.4066
SVM-+ bag of words [29]	49.6	0.51	0.49	0.487
SVM-NGRAM (mix of uni- and bi-gram) [30]	57.5	0.58	0.49	0.51
LSTM with embedding, convolution layer and max pooling [31]	77.5	0.5	0.5	0.5
Aspect-based method (unsupervised) [ours]	80.0	0.67	0.535	0.52

2.2 *Sentiment-Scoring of Tweets*

Once our phrase pattern from Table 1 and the keyword from Table 5 have been selected, the phrase + keyword combination for sentiment-scoring and classification is used. *Synsets* from the SentiWordNet lexicon provide us with the positive, negative and neutral scores for each word in the phrase + keyword combination. The maximum of these three scores has been taken for each word assuming that maximum score reflects the real nature and context of a word. The polarity is (+) for positive scores and (−) for negative scores. The labels have been considered as ‘negative’ and non-negative’ with all non-positive labels including ‘neutral’ labelled as ‘negative’. The procedure of sentiment-scoring for keywords is the same as for the phrases. The software implementation of our code is available for reference at https://github.com/mkaii/bug-/tree/gh-pages. All valid scores derived from sentences in the twitter post are added up to form the variable x that is nonlinearly transformed into a decision variable $f(x)$ as explained below.

The polarized SentiWordNet score is used to replace each word of our phrase + keyword combination. The summation of all these scores, denoted by x, is fed as input into our modified sigmoid function shown in Eq. (1).

$$f(x) = \frac{4}{1 + e^{-x}} - 2 \tag{1}$$

The sigmoid function in Eq. (1) is plotted in Fig. 2. The range of the sigmoid function in (1) is from -2 to 2. When the sentiment score x is 0, $f(x) = 0$. If the score x is greater than 0, then $f(x) > 0$ and the overall sentiment is determined to be positive. Likewise, if the score x is less than or equal to 0, then $f(x) \leq 0$ and the overall sentiment is determined to be negative. So the final sentiment score of any

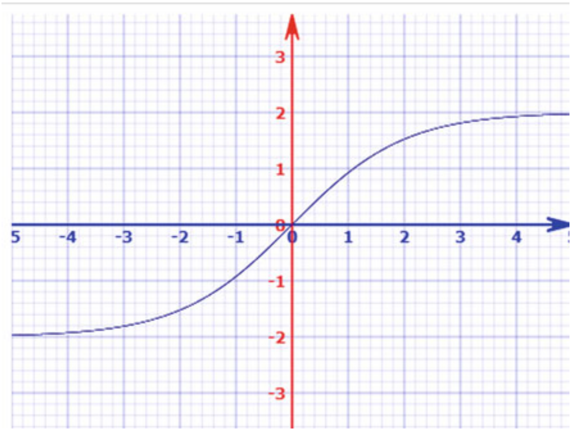

Fig. 2 Sigmoid function $f(x)$ versus x

phrase + keyword combination would be between -2 (for negative sentiment) and 2 (for non-negative sentiment).

3 Experimentation and Results

The dataset that has been used in this project is SemEval 2013 [26]. It is a twitter-based dataset having 1650 samples. It contains 3 labels, positive, negative and neutral. '0' is for neutral, '-1' is for negative and '1' is for positive sentiment. As the dataset consists of three labels ('1' for positive, '0' for neutral and '-1' for negative), for our application of negative sentiment analysis, two labels have been used (converted from three labels by considering neutral tweets as negative). In the two-label approach, label '1' represents non-negative or positive sentiment and label '-1' represents negative sentiment. The experiments were performed in Python 3 software on an Intel i-5 processor with a 2.6 GHz clock. Our code took less than one minute to execute. The software implementation of our code is available online at [27].

Our method is the aspect-based method for negative sentiment analysis which is unsupervised, and in this approach, there is no need for training the model. In the preprocessing step, spelling correction was done, slangs and abbreviations are removed. Preprocessing includes text normalization, stemming, removal of stop-words, spelling correction. After the preprocessing step, the words are tagged along with their correct POS tags using the NLTK library. POS tags that do not give any sentiment information are identified and removed them from our text corpus. The remaining tags are the ones based on which the sentences can be scored. Since a single sentence contains both positive and negative sentiments, a prioritized scoring system has been made by identifying the aspect as well as the phrases containing the acceptable POS tags, in a priority order selected by us. k-fold cross-validation has also been performed for all our experiments, where $k = 10$. In this process, the input is divided into *k* subsets. The model is trained on *k*-1 subsets and evaluated on the remaining 1 subset. This experiment is repeated *k* times, each time choosing different $k - 1$ subsets, and the final results are averaged. Apart from accuracy (in %), the precision, recall and F-score metrics are also computed. Table 5 shows the comparison between accuracy, precision, recall and F-score values of different classification models for the SemEval 2013 dataset. The highest accuracy and F-score values are got by using this method, as observed from the results summarized in Table 5.

For comparison, several baseline approaches are also implemented for sentiment analysis and detection that are supervised approaches. The least mean square (LMS) loss function is used for the training phase of all the supervised techniques. The grid search was used for determining support vector machine (SVM) parameter settings. Using n-gram, as a feature extraction method, delivered better results as compared to using the bag-of-word (BOW) features. Long short-term memory (LSTM) model also performed well. The Adam optimizer, with a batch size of 50 and 100 epochs, is the hyperparameter settings for LSTM. The GLOVE word embedding was used. The best result, however, was delivered by our unsupervised aspect-based method for detecting negative sentiment tweets.

4 Conclusions

Hatred in society has spread, in recent times, to social media platforms that are now required to be monitored continuously to detect negative online posts that may trigger mishappenings like riots. In this work, twitter posts have been classified based on their negative content, using an aspect-based approach based on the sentiment scoring of a phrase and keyword combination, detected separately. Various text preprocessing techniques are initially used such as normalization, stemming, removal of stop words, spelling correction and replacing abbreviations with their full forms. Phrases and keywords are selected from a prioritized list involving POS tags, from each sentence. The SentiWordNet lexicon provides us with the sentiment scores of phrases and keywords that are summed up for each sentence of the tweet and nonlinearly transformed into a decision score for detecting the sentiment of the tweet. A negative decision score indicates a negative tweet. Higher classification accuracies on the benchmark SemEval 2013 dataset prove the superiority of our approach as compared to the state of the art. Inclusion of multilingual slang to detect negative content of tweets is the future scope of our work.

References

1. Giachanou A, Crestani F (2016) Like it or not: a survey of twitter sentiment analysis methods. ACM Comput Surv (CSUR) 49(2):1–41

2. Prabhat A, Khullar V (2017) Sentiment classification on big data using Naïve Bayes and logistic regression. In: 2017 international conference on computer communication and informatics (ICCCI). IEEE, pp 1–5
3. Devi DVN, Kumar CK, Prasad S (2016) A feature based approach for sentiment analysis by using support vector machine. In: 2016 IEEE 6th international conference on advanced computing (IACC). IEEE, pp 3–8
4. Vashishtha S, Susan S (2019) Sentiment cognition from words shortlisted by fuzzy entropy. IEEE Trans Cogn Dev Syst
5. Kumar KLS, Desai J, Majumdar J (2016) Opinion mining and sentiment analysis on online customer review. In: 2016 IEEE international conference on computational intelligence and computing research (ICCIC). IEEE, pp 1–4
6. Vashishtha S, Susan S (2019) Fuzzy rule based unsupervised sentiment analysis from social media posts. Expert Syst Appl 138:112834
7. Vashishtha S, Susan S (2018) Fuzzy logic based dynamic plotting of mood swings from tweets. In: International conference on innovations in bio-inspired computing and applications. Springer, Cham, pp 129–139
8. Thet TT, Na J-C, Khoo CSG (2010) Aspect-based sentiment analysis of movie reviews on discussion boards. J Inf Sci 36(6):823–848
9. Lek HH, Poo DCC (2013) Aspect-based twitter sentiment classification. In: 2013 IEEE 25th international conference on tools with artificial intelligence. IEEE, pp 366–373
10. Brody S, Elhadad N (2010) An unsupervised aspect-sentiment model for online reviews. In: Human language technologies: the 2010 annual conference of the North American chapter of the association for computational linguistics. Association for Computational Linguistics, pp 804–812
11. Mukherjee A (2016) Extracting aspect specific sentiment expressions implying negative opinions. In: International conference on intelligent text processing and computational linguistics. Springer, Cham, pp 194–210
12. Yang B, Cardie C (2014) Joint modeling of opinion expression extraction and attribute classification. Trans Assoc Comput Linguist 2:505–516
13. Li H, Mukherjee A, Si J, Liu B (2015) Extracting verb expressions implying negative opinions. In: Twenty-ninth AAAI conference on artificial intelligence
14. Berend G (2011) Opinion expression mining by exploiting keyphrase extraction. In: Proceedings of 5th international joint conference on natural language processing, pp 1162–1170
15. Esuli A, Sebastiani F (2006) Sentiwordnet: a publicly available lexical resource for opinion mining. LREC 6:417–422
16. https://www.webopedia.com/quick_ref/textmessageabbreviations.asp. Accessed 22 Mar 2020
17. Python port of SymSpell (2019) [online]. Available at: https://github.com/mammothb/symspellpy. Accessed 22 Mar 2020
18. Loper E, Bird S (2002) NLTK: the natural language toolkit. In: Proceedings of the ACL-02 workshop on effective tools and methodologies for teaching natural language processing and computational linguistics, pp 63–70
19. Singh S, Rout JK, Jena SK (2016) Construct-based sentiment analysis model. In: Proceedings of the international conference on signal, networks, computing, and systems. Springer, New Delhi, pp 171–178
20. Miller GA (1998) WordNet: an electronic lexical database. MIT Press, Cambridge, MA
21. Marcus M, Kim G, Marcinkiewicz MA, MacIntyre R, Bies A, Ferguson M, Katz K, Schasberger B (1994) The Penn Treebank: annotating predicate argument structure. In: Proceedings of the workshop on human language technology. Association for Computational Linguistics, pp 114–119
22. Eisner JM (1996) Three new probabilistic models for dependency parsing: an exploration. In: Proceedings of the 16th conference on computational linguistics, vol 1. Association for Computational Linguistics, pp 340–345
23. Lamontagne L, Plaza E (eds) (2014) Case-based reasoning research and development. In: 22nd international conference, ICCBR 2014, Cork, Ireland, 29 Sept 2014–1 Oct 2014. Proceedings, vol 8765. Springer

24. Susan S, Zespal S, Sharma N, Malhotra S (2018) Single-keyword based document segregation using logistic regression regularized by bacterial foraging. In: 2018 4th international conference on computing communication and automation (ICCCA). IEEE, pp 1–4
25. Susan S, Keshari J (2019) Finding significant keywords for document databases by two-phase Maximum Entropy Partitioning. Pattern Recogn Lett 125:195–205
26. Wilson T, Kozareva Z, Nakov P, Rosenthal S, Stoyanov V, Ritter A (2013) SemEval-2013 Task 2: sentiment analysis in Twitter. In: Proceedings of the international workshop on semantic evaluation SemEval ‘13, Atlanta, Georgia, June 2013
27. https://github.com/mkaii/bug-/tree/gh-pages. Accessed 23 Mar 2020
28. Salazar DA, Vélez JI, Salazar JC (2012) Comparison between SVM and logistic regression: which one is better to discriminate? Revista Colombiana de Estadística 35(SPE2):223–237
29. Dias C, Jangid M (2020) Vulgarity classification in comments using SVM and LSTM. In: Smart systems and IoT: innovations in computing. Springer, Singapore, pp 543–553
30. Abbass Z, Ali Z, Ali M, Akbar B, Saleem A (2020) A framework to predict social crime through Twitter tweets by using machine learning. In: 2020 IEEE 14th international conference on semantic computing (ICSC). IEEE, pp 363–368
31. Chen Q, Ling Z-H, Zhu X (2018) Enhancing sentence embedding with generalized pooling. In: Proceedings of the 27th international conference on computational linguistics, pp 1815–1826