Stopwords Aware Emotion-Based Sentiment Analysis of News Articles

Chhaya Yadav and Tirthankar Gayen

1 Introduction

Sentiment analysis is procedurally distinguishing, extricating, evaluating, and contemplating abstract data by devices of computational phonetics, computational linguistics, biometrics, and text analysis. The sentiment analyzer assesses what constitutes a positive, neutral, or negative piece of writing. Sentiment analysis is helpful for data analysts within large enterprises for conducting research on market, public opinion, brand reputation, and understanding customer experiences [1]. The already existing product or a service is the object considered for sentiment analysis. Sentiment analysis can be broadly classified into four types, namely, fine-grained, emotion-based, aspect-based, and intent-based sentiment analysis. In fine-grained sentiment analysis, the polarity of the opinion is determined by simple binary classification of positive and negative sentiment. Depending on the use case, this type may also fit within the higher specification [2]. To find indications of particular emotional states mentioned in the text, emotion detection is used. Advanced sentiment analysis is done using aspect-based sentiment analysis, and the objective is to convey opinion with respect to the certain part of the input. The intent-based sentiment analysis ascertains the type of intention that the message is expressing. Opinion mining is synonymous to sentiment analysis despite the general understanding that sentiments are emotionally loaded opinions [3].

Analyzing enormous amount of unstructured text into a more specific news articles, devising suitable algorithms to understand the opinion from text and finding positive and negative score out of it is a challenging task. Thus, there is a need to incorporate suitable techniques to improve the accuracy of the results obtained from

C. Yadav (✉) · T. Gayen
Jawaharlal Nehru University, New Delhi, India

A. Haldorai et al. (eds.), *5th EAI International Conference on Big Data Innovation for Sustainable Cognitive Computing*, EAI/Springer Innovations in Communication and Computing, https://doi.org/10.1007/978-3-031-28324-6_15

sentiment analysis of the news articles [4]. Although there are several approaches concerned with sentiment analysis of news articles, the outputs provided by these approaches lack accuracy to a considerable extent [5].

Negative stopwords include important information about the sentiment of the sentence, yet it has been discovered that most sentiment analysis pre-processing techniques discard these stopwords [6]. As a result, several times semantic information gets lost, resulting in inaccurate sentiment analysis. The contributions of importance of this article are as follows:

1. This article presents an approach where every sentence is processed and examined at the sentence level to establish its polarity. In order to minimize the loss of significant information for labelling news articles, the proposed approach does the pre-processing considering the negative stopwords and labels the sentiments of the article using Support Vector Machine (SVM).
2. The results obtained after applying the proposed approach on the dataset of different categories of news articles obtained from BBC are usually found to yield a comparatively higher accuracy for providing the sentiment polarity of various news items.
3. A regression analysis is presented to confirm that the proposed approach provides comparatively better accuracy for sentiment analysis of news articles.

A survey of relevant research on sentiment analysis of text data is described in Sect. 2 of this article. The proposed approach for sentiment analysis of news items is described in Sect. 3, which is followed by an implementation utilizing news articles from the BBC dataset in Sect. 4. This section also provides an evaluation of the proposed approach. Section 5 concludes with a discussion on the benefits and limitations of the proposed approach as well as outlines the goals and enhancements for future work.

2 Related Work

In Natural Language Processing (NLP), the field of sentiment analysis has been explored with a variety of approaches. There are plenty of researchers who are working on Sentiment Analysis of Texts. The researchers have extensively contributed to its development and enhancing its applications in various fields. There are various methods, ranging from dictionary-based methods to machine learning methods. In 2018, Urologin [7] proposed techniques for extracting and displaying text data. It performs combined text summarization and sentiment analysis. A text summarization technique based on pronoun replacement is created, and sentiment data is gathered using the VADER sentiment analyzer. However, their summarization approach may cause loss in semantic information, which can lead to a wrong sentiment analysis of the document. In their work, they used a standard sentiment analysis repository (Github repo VADER). Taj et al. [8] characterized news articles as positive, negative, and unbiased classes by gathering the all-out opinion scores of the sentences in the article using lexical-based methodology. It implements Lexicon-based sentiment assessment of news stories, but it has insufficient or restricted word inclusion. As a result, numerous new lexical items with distinct semantics should be refreshed in lexical data set. It solely employs news articles in English from one hotspot for sentiment analysis. Vilasrao et al. [9] intended to develop a system with emotion dataset and training dataset to obtain valency (in the form of emotional and neutral). They have used Lexicon-based approach and Deep Learning Technology. They presented their estimation and investigation utilizing dictionary-based methodology and deep learning techniques to deal with emotion classes. The output of their proposed framework (which perceives the presence of assumptions extremity) can be used to enhance the sentiment analysis framework but their approach is only lexical based and uses very less amount of data for training using traditional machine learning (ML) algorithm. As a result, the accuracy of their output is not very high. Souma et al.'s [10] work was to forecast the financial news sentiments. They used Simple Sequential LSTM network architecture for the analysis, but in their work, an assumption (which may be error prone) is made that if the stock log return value is negative then the sentiment is negative and vice versa. No standard dataset was used to perform the experiment and obtain the results. No normalization of the statements in the news articles was done since same weightage was given to all the statements. Shirsat et al.'s [11] work was concerned with sentence level negation identification from news articles. They used a dictionary-based approach with ML techniques but no standard dataset was used (scraped dataset was used) and the used dataset was quite small. Their approach was dictionary based so no semantic information was used. For obtaining word-level emotion distribution Li et al. [12] considered the use of dictionary with word-level emotion distribution (known as NRC-Valence arousal dominance) for assigning emotions along with intensities to the sentiment words as efficient. Two models were proposed by Basiri et al. [13] in their study that employed a three-way decision theory and proposed two models. The three-way fusion of one deep learning model and the conventional learning method was used in the first model (3W1DT), whereas, three-way fusion of three deep learning models was used in the second model (3W3DT). The results obtained using Drugs.com dataset showed that both frameworks outperformed the traditional deep learning methods. In addition, it was noted that the first fusion model performed significantly better than the second model in terms of accuracy and F1-metric. Using the Rotten Tomato movie review dataset, Tiwari et al. [14] have implemented 3 ML algorithms (Maximum Entropy, Naive Bayes, and SVM) with the n-gram feature extraction technique. They noted a drop in accuracy for *n*-grams with larger values of n, such as $n = 4, 5$, and 6. Using various feature vectors like Bag of Words (BOW), Unigram with Sentiwordnet Soumya et al's [15] work divided 3184 Malayalam tweets into negative and positive opinions. They used the ML algorithms Naive Bayes, Random Forest and found that Random Forest performed better (with an accuracy of 95.6%) with Unigram Sentiwordnet when negation words were taken into account. Rao et al. [16] used Long Short-Term Memory (LSTM) to improve sentiment analysis by first cleaning the datasets and removing the sentences with weaker emotional polarity. On three publicly accessible document-level review datasets, their model outperforms the state-of-the-art models.

From the srvey, it is found that although there are several approaches concerned with sentiment analysis, but the output provided by these approaches lacks accuracy to a considerable extent. In many approaches, it is found that no standard dataset was used to perform the experiment and obtain the results. In some approaches, no normalization of the statements in the news articles were done since same weightage was given to all the statements [17]. The dataset used in some of the approaches were quite small. In some approaches, no semantic information was used. In few approaches, there was loss in semantic information which lead to wrong sentiment analysis of the document. Analyzing enormous amount of unstructured text into a more specific news articles, devising suitable algorithms to understand the opinion from text and finding positive and negative score out of it is a challenging task. In essence, it is the process of determining the emotional tone behind a series of words, used to gain an understanding of the attitudes, opinions, and emotions expressed within an online mention. Negative stopwords carry significant information about the sentiment of the sentence, but it is found that most of the approaches concerned with sentiment analysis removed these stopwords during preprocessing stage [18]. As a result, there can be loss in semantic information leading to incorrect sentiment analysis [19]. Thus, there is a need to incorporate suitable techniques to improve the accuracy of the results obtained from the sentiment analysis [20]. Hence, the intention is to develop a suitable approach to improve the accuracy of sentiment analysis by considering the negative stopwords [21].

3 Proposed Approach

The polarity of the text data is determined or expressed by the sentiment analysis approach. Essentially, there are three layers of sentiment analysis: A sentiment analysis at the document level will be performed first to ascertain the polarity of the document. In the case of a text file containing only product reviews, the algorithm decides the polarity of all the content in the document. It is because of this that the document only conveys opinions about one specific subject and cannot be used to evaluate other products. Every sentence is processed and examined at the sentence level to establish its polarity. Finding emotions about things and their characteristics are made possible by aspect-level sentiment analysis. The proposed algorithm *Negative Stopwords Aware Sentiment Analysis* (NSASA) does the preprocessing considering the negative stopwords and labels the sentiments of the article using SVM. The steps of the proposed algorithm NSASA are as follows:

Algorithm Negative Stopwords Aware Sentiment Analysis (NSASA)

Step 01 : Initialisation of negation_words ;
Step 02: pos_list= set(opinion_lexicon.positive()) ; /* generating a set of positive words */
Step 03: neg_list=set(opinion_lexicon.negative()) ; /* generating a set of negative words */
Step 04: stop_words = stopwords.words('english') ; /* storing english stop words */
Step 05: lemma = nltk.wordnet.WordNetLemmatizer() ; /* lemmatizer object */
Step 06: opinionDict={}; /* dictionary which has key 'pos' , 'neg' which contains corresponding words */
Step 07: opinionDict['pos']= pos_list ;
Step 08: neg_list.update(negation_words) ; /* updating negative list with inclusion of negation_words */
Step 09: opinionDict['neg'] = neg_list ;
Step 10: **for** word in negation_words:
stop_words.remove(word) ; /* removes words in negation_words dictionary from stop_words dictionary */
end for
Step 11: sentiment_vectorizer = CountVectorizer(input = "content", encoding = "utf-8", decode_error = "replace", ngram_range = (1, 1), preprocessor = preprocessText, analyzer = "word", vocabulary = sentiment_words, tokenizer = None); /* creating the vector */
Step 12: x_train = sentiment_vectorizer.transform(xtrain); /* dividing the dataset into train dataset */
Step 13: x_test = sentiment_vectorizer.transform(xtest); /* dividing the dataset into test dataset */
Step 14: x_train.shape
(360, 6823); /* fitting the model over the dataset*/
Step 15: model = LinearSVC(penalty = "l2", loss = "squared_hinge", dual = True, tol = 0.0001, C = 1.0, multi_class = "ovr", fit_intercept = True, intercept_scaling = 1, max_iter = 20000); /* fitting the model over the dataset*/
Step 16: model.fit(x_train, ytrain); /* fitting the model over the dataset */
Step 17: predictions = model.predict(x_test); /* evaluative tool */
Step 18: print(classification_report(ytest, predictions)); /* outputting the evaluative scores */

Algorithm NSASA begins with the initialization and storing of negative words in the negation_words variable depicted in step 1. Generation of a set of positive and negative words takes place in steps 2 and 3. English stopwords are stored in step 3 and lemmatizer object is created in step 4. A dictionary is created for storing separately the positive and negative set of words as depicted in steps 6–9. The dictionary containing stopwords is updated by removing negative words list from it as depicted in step 10. Training and testing of dataset takes place from steps 11 to 13. Step 14 fits the model over the dataset. SVM is applied on the model and the predictions are made and final label is printed from steps 14 to 18. The model used in this proposed algorithm corresponds to LinearSVC.

Various user-defined functions which are used in the algorithm NSASA are defined in Table 1. Table 2 provides the descriptions of various predefined functions and parameters which are used in the algorithm NSASA.

4 Implementation Details and Evaluation

The proposed approach has been implemented using Python 3.7 with Google Colaboratory and the NLTK package. From the NLTK package nltk.download('punkt'), nltk.download('stopwords'), nltk.download('opinion_lexicon') were imported. Table 3 below displays the article's categorization and the proportion of neutral, favorable, and unfavorable terms in it. This study makes use of the Bing Liu dictionary, which has 4783 negative words and 2006 positive terms [11].

Table 4 shows the article's categorization and the proportion of neutral, favorable, and unfavorable terms in it after using the proposed algorithm NSASA. Additionally, it makes use of the Bing Liu dictionary, which has 4783 negative words and 2006 positive terms.

Figure 1 shows the percentage accuracy values obtained from the approach proposed by Shirsat et al. and the proposed NSASA for four different types of datasets.

For the purpose of evaluation, a comparison of the proposed approach using NSASA has been made with the approach provided by Shirsat et al. [11], and the results obtained are summarized in Table 5. Based on the results obtained after the experimentation it is found that the highest accuracy achieved by the approach proposed by Shirsat et al. in the Tech. data is 86%, and the lowest accuracy achieved by the approach proposed by Shirsat et al. in the business data is 75%. Whereas the highest accuracy achieved by the proposed NSASA in Tech. data is 98%, and the lowest accuracy achieved by the proposed NSASA in business data is 73%.

Thus, the approach of Shirsat et al. is found to provide results with an average accuracy of 80.25 whereas the proposed NSASA is found to provide results with an average accuracy of 85.75. The accuracy values obtained from the technique of Shirsat et al. and the proposed NSASA were also employed in a regression analysis, with the outputs reported in Table 6. Upon data analysis, it is discovered that the proposed NSASA algorithm's p-value is 0.009156.

Regression coefficient r (Multiple R) = 0.9908 and $p < 0.05$ are obtained in Table 6. This suggests that the accuracy values acquired from Shirsat et al. [11], and the accuracy values obtained from the proposed NSASA algorithm have a positive relationship. Thus, the proposed NSASA seems to provide better accuracy in sentiment analysis of news articles as compared to the approach specified by Shirsat et al.

Table 1 User-defined functions and their definitions

Function	Definition
removeStopWords(text)	removeStopWords(text) { word_tokens = word_tokenize(text.lower()); filtered_sentence = " ".join([w for w in word_tokens if not w.lower() in stop_words]); return filtered_sentence; }
removeDigits(text)	removeDigits(text) { res = ''.join([i for i in text if not i.isdigit()]); return res;}
remove_punctuation(text)	remove_punctuation(text) { punctuationfree = "".join([i for i in text if i not in string.punctuation]); return punctuationfree;}
removeExtraSpaces(text)	removeExtraSpaces(text) {return re.sub(' +', ' ', text);}
textStemmer(text)	textStemmer(text) {ps = PorterStemmer(); res = " ".join([ps.stem(i) for i in text.split()]); return res;}
OpinionOfWord(word)	OpinionOfWord(word) {if word in opinionDict['pos']; return 1; elif word in opinionDict['neg']; return -1; else; return 0;}
getOpinionListofSen-tence(text)	getOpinionListofSentence(text) {res = [OpinionOfWord(word) for word in text.split()]; return res;}
getPosScore(opinionList)	getPosScore(opinionList) { return opinionList.count(1)/len(opinionList);}
getNegScore(opinionList)	getNegScore(opinionList) {return opinionList.count(-1)/len(opinionList);}
getNeuScore(opinionList)	getNeuScore(opinionList) {return opinionList.count(0)/len(opinionList);}

getGT(list)	getGT(list)
	{if getPosScore(list)>getNegScore(list);
	return 1;
	elif getPosScore(list)<getNegScore(list);
	return -1;
	else;
	return 0;}

getDiff(list)	getDiff(list) {return getPosScore(list) - getNegScore(list);}

preprocessText(text)	preprocessText(text)
	{filtered_sentence = removeStopWords(text);
	removedPunctuationVar = remove_punctuation(filtered_sentence);
	removedDigitsVar =removeDigits(removedPunctuationVar);
	removedEx-trSpacesVar=removeExtraSpaces(removedDigitsVar);
	stemmedSentencetextStem-mer(removedExtraSpacesVar);
	return stemmedSentence;}

Table 2 Predefined functions, parameters, and their description

Functions/parameters	Descriptions
Set()	Function can be used to create sets
Opinion_lexicon.positive()	Return all positive words in alphabetical order
Opinion_lexicon.negative()	Return all negative words in alphabetical order
Stopwords.words('english')	Return list of stopwords stored in English
nltk.wordnet.WordNetLemmatizer()	Returns the input word unchanged if it cannot be found in WordNet
Update()	Inserts the specified items to the dictionary
Stopwords.remove()	Remove stop words from string
penalty{'l1', 'l2'}	It specifies the standard that was applied in the penalty. The benchmark used in SVC, also known as Ridge Regression, is the "l2" penalty
loss = "squared_hinge"	For "maximum margin" binary classification issues, the squared hinge loss is a loss function that is employed
dual = True	This chooses the algorithm to solve the dual optimization problem or the primary optimization problem
tol = 0.0001	It is tolerance of limiting standards
C = 1.0	The Regulator C is parameter. The strength of the regularization is inversely connected to C. It can only be positive
multi_class = "ovr"	"ovr" trains n_classes one-vs-rest classifiers
fit_intercept = True	Calculations will use the intercept if it is set to True
fit_intercept = True	If it is set to True, calculations will use the intercept
intercept_scaling = 1	Intercept scaling lessens regularization's effects on feature weight and hence on the intercept
max_iter = 20000	The most iterations that can be performed
negation_words	['no','against','nor','not',"aren't",'couldn',"couldn't",'didn',"didn't",'doesn',"doesn't",'hadn',"hadn't",'hasn',"hasn't",'haven',"haven't",'isn',"isn't",'ma','mightn',"mightn't",'mustn',"mustn't",'needn',"needn't",'shan',"shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]

Table 3 Category wise document polarity using Shirsat et al.

Sr no	Name of category	Neutral	Negative	Positive	Total
1	Business	34	214	262	510
2	Entertainment	21	244	136	401
3	Politics	17	190	210	417
4	Sport	33	327	151	511
5	Tech	21	244	136	401

Table 4 Category wise document polarity using NSASA

Sr no	Name of category	Neutral	Negative	Positive	Total
1	Business	35	219	256	510
2	Entertainment	15	87	284	401
3	Politics	34	180	203	417
4	Sport	35	117	359	511
5	Tech	16	128	257	401

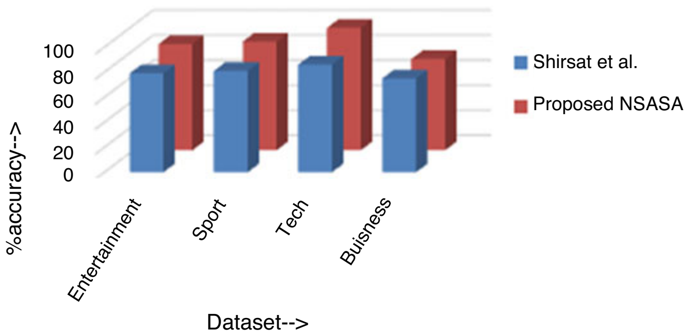

Fig. 1 Obtained accuracy values for four different datasets

Table 5 Evaluation results

Dataset	Shirsat et al.			Proposed NSASA		
	Accuracy	Precision	*F*-score	Accuracy	Precision	*F*-score
Entertainment	79	81	79	85	88	84
Sport	81	78	79	87	88	87
Tech	86	89	87	98	98	98
Business	75	69	72	73	69	77

5 Conclusion

The amount of text continually expanding is an invaluable source of knowledge and information that must be effectively retrieved to reap its benefits. It may be quite challenging to analyze the vast amount of unstructured content into more specialized news items to create the proper algorithms to extract opinions from text

Table 6 Regression analysis results

Regression statistics	
Multiple R	0.990844423
R square	0.981772671
Adjusted R square	0.972659007
Standard error	1.693672311
Observations	4

	df	SS	MS	*F*	Significance *F*
Regression	1	309.0129482	309.0129482	107.7253	0.0091555
Residual	2	5.737051793	2.868525896		
Total	3	314.75			

	Coefficient	Standard error	*t* Stat	*p*-Value
Intercept	−92.33466135	17.17892068	−5.374881408	0.032915
SVM	2.219123506	0.213807297	10.3790822	0.009156

and assign positive and negative scores to it. Problems are encountered with the existing techniques for sentiment analysis in the presence of punctuations, ironical sentences, etc., which results in incoherent sentiment. In the proposed approach, every sentence is processed and examined at the sentence level to establish its polarity. Aspect-level sentiment analysis is used for finding emotions about things and their characteristics. The proposed algorithm NSASA does the preprocessing considering the negative stopwords and labels the sentiments of the article using SVM. The inclusion of negative stopwords in the proposed approach ensures that there will be minimum loss of significant information for labeling news articles. The proposed approach using SVM can be considered to be an improvement over Shirsat et al.'s approach (which has not considered the negative stopwords) to provide more accurate results. The proposed approach can be considered as an approach for sentiment analysis using negative stopwords to provide more accurate results for obtaining labeled positive, negative, and neutral sentiments from news articles. When the target classes are overlapping and the data set includes more noise, SVM does not perform very well. In the future, the proposed approach of using negative stopwords can be used to enhance the accuracy of the Naïve Bayes approach and other Machine Learning algorithms. Presently, the proposed approach has been tested on the news article dataset from BBC. This approach can be further applied to other datasets traditionally used in sentiment analysis, such as DUC-2002 and DUC-2004. The proposed approach is anticipated to be very helpful in determining the sentiment polarity of various news items and in retaining the information with the least amount of exclusion of important terms in the article. Apart from creating a sentiment analysis of news articles, the proposed approach can also benefit the governments in determining public opinion related to policies and program implementation expressed on various social networking platforms with better accuracy.

References

1. MonkeyLearn. Sentiment analysis: A definitive guide, https://monkeylearn.com/sentiment-analysis/. Last accessed 5 Dec 2022
2. A. Zhao, Y. Yu, Knowledge-enabled BERT for aspect-based sentiment analysis. Knowledge-Based Syst. **227**, 107220 (2021) ISSN 0950-7051
3. K. Roebuck, *Sentiment Analysis: High-Impact Strategies What You Need to Now: Definitions, Techniques and Applications for Sentiment Analysis, Adoptions, Impact, Benefits, Maturity* (Emereo Publishing, 2012)
4. P. Pooja, G. Sharvari, A survey of sentiment classification techniques used for Indian regional languages. Int. J. Comput. Sci. Appl. **5**(2), 13–26 (2015)
5. A. Shoukry, Sentence-level Arabic sentiment analysis, in *2012 International Conference on Collaboration Technologies and Systems (CTS)*, (IEEE, 2012), pp. 546–550
6. P. Melville, W. Gryc, R.D. Lawrence, Sentiment analysis of blogs by combining lexical knowledge with text classification, in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (ACM, 2009), pp. 1275–1284
7. S. Urologin, Sentiment analysis, visualization and classification of summarized news articles: A novel approach. Int. J. Adv. Comput. Sci. Appl. **9**(8), 616–625 (2018)
8. S. Taj, B.B. Shaikh, A.F. Meghji, Sentiment analysis of news articles: A lexicon based approach, in *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, (IEEE, 2019), pp. 1–5
9. G.S. Vilasrao, P.D. Sathya, Lexical approach for sentiment analysis on news articles with deep learning method. Int. J. Sci. Res. **8**(12), 725–730 (2019)
10. W. Souma, I. Vodenska, H. Aoyama, Enhanced news sentiment analysis using deep learning methods. J. Comput. Soc. Sci. **2**, 33–46 (2019)
11. V.S. Shirsat, R.S. Jagdale, S.N. Deshmukh, Sentence level sentiment identification and calculation from news articles using machine learning techniques, in *Computing, Communication and Signal Processing*, (Springer, 2019), pp. 371–376
12. Z. Li, H. Xie, G. Cheng, Q. Li, Word-level emotion distribution with two schemas for short text emotion classification. Knowledge-Based Syst., Elsevier **227**, 107163 (2021)
13. M.E. Basiri, M. Abdar, M.A. Cifci, S. Nemati, U.R. Acharya, A novel method for sentiment classification of drug reviews using fusion of deep and machine learning techniques. Knowledge-Based Syst., Elsevier **198**, 105949 (2020)
14. P. Tiwari, B.K. Mishra, S. Kumar, V. Kumar, Implementation of n-gram methodology for rotten tomatoes review dataset sentiment analysis. Int. J. Knowl. Discovery Bioinf., IGI Global **7**(1), 30–41 (2017)
15. S. Soumya, K.V. Pramod, Sentiment analysis of Malayalam tweets using machine learning techniques. ICT Express, Elsevier **6**(4), 300–305 (2020)
16. G. Rao, W. Huang, Z. Feng, Q. Cong, LSTM with sentence representations for document-level sentiment classification. Neurocomputing, Elsevier **308**, 49–57 (2018)
17. R. Mishra, T. Gayen, Automatic lossless-summarization of news articles with abstract meaning representation. Procedia Comput. Sci., Elsevier **135**, 178–185 (2018)
18. M. Birjali, M. Kasri, A. Beni-Hssane, A comprehensive survey on sentiment analysis: Approaches, challenges and trends. Knowledge-Based Syst. **226**, 107134 (2021) ISSN 0950-7051
19. P. Wang, J. Li, J. Hou, S2SAN: A sentence-to-sentence attention network for sentiment analysis of online reviews. Decis. Support Syst. **149**, 113603 (2021) ISSN 0167-9236
20. W. Liao, B. Zeng, J. Liu, P. Wei, X. Cheng, W. Zhang, Multi-level graph neural network for text sentiment analysis. Comput. Electr. Eng. **92**, 107096 (2021) ISSN 0045-7906
21. S. Zitnik, S. Blagus, M. Bajec, Target-level sentiment analysis for news articles. Knowledge-Based Syst. **249**, 108939 (2022) ISSN 0950-7051