REDVET - Revista electrónica de Veterinaria – ISSN 1695-7504
Vol 25, No. 2 (2024)
http://www.veterinaria.org
Article Received- Revised- Accepted-

Bidirectional Recurrent Capsule Networks with Transformer Based Learning for Twitter Sentiment Classification

Simran Rawat Gaurav Suratwala[1*], Aman Singh Jagat Jeet Biswas[2], Jeffrey Obiri Boahen Hanumant Jagdish Chauhan[3], Mudit Parihar[4], Yashwardhan Atre[5], Bhavani Sangani[6], Prof. Hardik Parmar[7]

[1*]Student, Parul Institute of Engineering and Technology, Parul University, Waghodia Road, 391760Vadodara, Gujarat
[2]Student, Parul Institute of Engineering and Technology, Parul University, Waghodia Road, 391760 Vadodara, Gujarat.
[3]Student, Parul Institute of Engineering and Technology, Parul University, Waghodia Road, 391760Vadodara, Gujarat.
[4]Student, Parul Institute of Engineering and Technology, Parul University, Waghodia Road, 391760, Vadodara,Gujarat
[5]Student, Parul Institute of Engineering and Technology, Parul University, Waghodia Road, 391760, Vadodara,Gujarat
[6]Student, Parul Institute of Engineering and Technology, Parul University, Waghodia Road, 391760, Vadodara,Gujarat
[7]Assistant Professor, Parul Institute of Engineering and Technology, Parul University, Waghodia Road 391760, Vadodara, Gujarat.

ABSTRACT

Rapid developments have been happening in recent years in terms of classifying aspect-level sentiment analysis (ASA). Several research gaps have been identified with the existing approaches like aspect extraction, opinion target aspects, and handling long-term dependencies between the contextual aspects. In this work, we focus on addressing the above limitations by proposing Bidirectional Recurrent Capsule Networks (BRCN) framework with a pre-trained BERTweet transformer and sequence-based self-attention mechanism. This combination relatively improves the task of ASA. Furthermore, we also contributed to the development of one dataset on Covid-19 based on public tweets collected from social media to analyze its intensity. Based on the experimental results, it is evident that the proposed model surpasses other traditional approaches by identifying the hidden contextual information using sequence position information. In addition, we evaluated the performance of our proposed model against other approaches using three benchmark datasets, namely IMDB, HNT, and MR. Our findings reveal that our model achieved higher accuracy and had better time complexity than the existing approaches.

Keywords: Bidirectional Recurrent Capsule Networks, Sentiment Analysis, Transformer, Capsule Network, BiLSTM, GRU, Self-attention, Deep Learning.

1. Introduction

People are encouraged to express their own experiences of understanding a product/issue /places/persons, etc., in the form of opinions and feelings, which strengthens the analysis (Yoon et al. 2013; Yin et al. 2015; Park et al. 2016). To analyze these opinions, Sentiment Analysis (SA) was conducted, which is also known as Opinion Mining (OM). It is a process of interpreting and classifying various emotions expressed by different people as positive, negative, or neutral towards a piece of text based on text analysis techniques (Cambria et al. 2013; Khan et al. 2014; Ravi & Ravi, 2015; Sun et al. 2017; Yadollahi et al. 2017). For decades, several developments have been made to capture, measure, quantify, and classify public sentiments from social media by utilizing various methods. This allows SA as one of the most popular research fields. The applications of SA have been extended widely to several domains like financial services, political elections, health care, and social events. There have been several developments in recent years in terms of classifying sentiments in social media. Tweets are most common in social media posts. Researchers have typically categorized tweets based on the sentiments expressed in their messages. These sentiments are classified at various levels of granularity, including document level, sentence level, and aspect level, with the classification of emotions taking place over three stages. Usually, sentiment analysis conducted at the document level focuses on the overall polarity of the document, ensuring that the entire document expresses a single polarity. On the other hand, when conducting sentiment analysis at the sentence level, the analysis aims to determine the sentiment expressed towards a specific entity or aspect. In such cases, similar expressions may have varying polarities within the text. In contrast, aspect-level sentiment analysis does not have such a requirement. Its objective is to determine the polarity of a single phrase in a sentence, rather than focusing on the overall polarity of the document or sentiment towards a specific entity. In comparison, ASA (Do et al. 2019; Schouten & Frasincar, 2016) aims to understand the concerns of traditional SA by straightly focusing on the sentiments rather than language composition.

Earlier studies on ASA typically employ several machine learning (ML) methods to develop a sentiment classifier and received great success in ASA tasks. However, it still has some limitations in determining quality aspects based on their semantics, extracting emotional aspects, and dependencies. Recently, deep learning (DL) models made several advancements to overcome the limitations of ASA. DL models are the potential to discover the aspects with high representation. For instance, the Conditional Random Fields (CRF) model is used for exploiting the dependencies among the aspects based on conjunctions using. The limitation identified from this model is that it can extract only one aspect within a single sentence. A new model has been developed to deal with multiple aspects per sentence. Further, LSTM networks have also been extensively employed for extracting multiple aspects per sentence. Though DL is capable, the performance is still degraded due to the loss of syntactic and semantic relations among the aspects. In this work, we have developed one corpus of opinions on the Corona Virus to evaluate the proposed model. In early 2020, the Coronavirus outbreak revealed a cluster of cases of unknown cause in China when they informed the World Health Organization (WHO) on 31 December 2019. They identified it as SARS-CoV-2, a new kind of Coronavirus that goes virally by affecting nasal severe, sinus, or deep throat infections (Kouzy et al. 2020). Efforts to cure the disease of pneumonia also contributed to nationwide lockdowns (Barkur et al. 2020), extensive international transportation suspensions, massive shortages, and disrupted stock markets(Varalakshmi & Swetha, 2020). Much is still unknown about the infection but reported that about 1% of affected patients might die as per projections. That makes it ten times deadlier than seasonal flu, but much less dangerous than SARS(Yongshi Yang et al. 2020), MERS, or Ebola (Beryl Joylin et al. 2016; Ichev & Marinč, 2018; John et al. 2019; E. H. J. Kim et al. 2015). The Covid-19 outbreak (Naeim, 2020; Vibha et al. 2020)has caused public panic. People express their responses and reactions on social media like Twitter and Facebook. There are many medical blogs and communities available on social media, where people chat and communicate with regards to their illnesses, symptoms, and medications. All these opinions on the Coronavirus are collected. Several research gaps were identified from the ASA using DL models like implicit and explicit aspect extraction, determining the opinion targets from aspects, handling long-term dependencies based on the semantic and syntactic relationships between aspects. Therefore, we focus on addressing the above limitations by proposing a new methodology to identify and extract explicit, implicit, and opinion target aspects based on their contextual information through effective ways. In addition to this, relationships between high-level semantic and syntactic aspects are also extracted by capturing the long-term dependencies based on sequential information. Finally, our proposed model demonstrates better accuracy and time complexity than traditional deep learning models.

The rest of the paper is organized as follows: Section II presents an in-depth literature review. The proposed methodology is outlined in Section III, while Section IV presents the experimental results. The findings are discussed, and the paper concludes in Section V.

2. Related work

The focus of the present study is on ASA. Recent studies in sentiment analysis have utilized state-of-the-art deep learning models such as CNN (Shin et al. 2018), LSTM (Y. Ma et al. 2018), RCNN (R. Kumar et al. 2020; Rehman et al. 2019; Rhanoui et al. 2019), BiLSTM(Abdi et al. 2019; Zhou et al. 2019), GRU (Luo, 2019; Tran et al. 2019), ConvLSTMConv(Ghorbani et al., 2020), and Capsule Networks (Kruthika et al. 2019). These models have garnered significant attention in recent times within the field of SA (Habimana et al. 2020). Several of these models have also been employed for challenging natural language processing (NLP) tasks such as question answering systems (Jain et al., 2020), text classification (M. Yang, Zhao, et al. 2019), and sentiment classification (Tao & Fang, 2020). In addition to this, several RNN variants like BiLSTM and BiGRU networks, which are primarily used in the field of sentiment classification have been revealed by (Ait Hammou et al. 2020). (Shuang et al. 2019) proposed AE-DLSTM, which uses two LSTMs to capture the correlations among context words and aspect-words, and AELA-LSTM which can use context-word location information. In addition, they employed a unique attention mechanism that enables the extraction of contextual information for the aspects, resulting in a better comprehension of aspect correlations and their contexts. Moreover, the experimental evaluation was conducted on two English datasets and one Chinese dataset, and the proposed approach yielded superior performance compared to the standard approaches. Automatic opinion and aspect term extraction have become challenging tasks due to the sequence labeling problems.

(Yu et al.2019) developed a global inference-based multi-tasking framework for extracting various aspects and opinion terms by leveraging Bi-LSTM and multi-layer attention networks. The model used in their study successfully captures the semantic relations by handling various syntactic constraints and extracts opinion-related aspects. The proposed model was successful in outperforming existing models on various benchmark datasets. Traditional deep learning models have limitations like losing the semantical and syntactical information in the classification of text, particularly while dealing with large-size text corpus with a huge vocabulary. (X. Ma et al. 2019) highly effective term weighting method for improving the performance of text classification with multi-channel CNN. The proposed method tackles the difficulties of the conventional weighting methods in aspect term extraction based on multiple weighted word embedding's. The results show that the baseline models lack behind the proposed model in terms of efficiency on benchmark datasets. ASA with deep learning models based on attention mechanisms obtains an effective representation of features when performing the task of sentiment classification. However, it might produce some unfair results due to the incorporation of average pooling mechanisms. The average function used in the pooling mechanism can develop noise and results in imbalance problems among the target aspects in the classification. To overcome such limitations, (C. Yang et al. 2019) had developed an LSTM network with Co-attention that extracts more significant sentiment features. The proposed model learns the context and target level representations by sequentially computing the attention scores of each target aspect instead of learning the context and targets with non-linear representations.

(Edara et al. 2019) performed sentiment analysis and text categorization on various tweets collected from Twitter online cancer communities. The work was carried out by applying several conventional machine learning techniques along with LSTM networks. The obtained results proved that LSTM outperforms conventional models in both accuracy and running time. (Abdi et al. 2019) developed a direct approach by incorporating both RNN and LSTM networks to address the limitations related to the lexicon, sentiment, and semantical problems faced by the conventional models. The proposed model in their study effectively extracts and utilizes the word and sentence level aspects based on the sentence type, contextual polarities, and sentiment shifts by embedding previous knowledge based on sentiment lexicons and parts-ofspeech (POS) Tagging. The experimentation was conducted on various large-scale datasets and achieved better classification performance at SSA when compared to existing methods. (Kastrati et al. 2019)proposed an enhanced framework by incorporating the topic modeling approaches by addressing the semantical problems faced in ontology extraction from the documents. With the help of this model, it is possible to have a more complete understanding of ontologies that describe documents. The experiments were conducted on INFUSE dataset with various deep learning models and achieved 78.10% of accuracy in document classification.

(Alaei et al. 2019) conducted SA to explore the emotions experienced by tourists around their desire to travel and subsequently post about this on social media. Sentiment classification with high efficiency is one of the major concerns while working with massive and irregular data. In this situation, learning the model by extracting emotional words from the large-sized text corpus may result in better performance of the models. (Xie et al. 2019) developed a novel framework for addressing the challenge related to the improvement of sentiment classification accuracy. The emotion words are initially extracted by using the latent probabilistic model known as PLSA from the training corpus along with Wikipedia. Later, a set of semantic, syntactic, lexical, and morphological features were extracted to train models as part of TSA using k-fold method. (Majumder et al. 2019) developed a novel approach based on multi-task learning to address both sarcasm detection and sentiment classification tasks. The findings proved that the sentiment classification with sarcasm detection could relatively increase the performance of the model by multi-task learning. Besides, a novel segment-level joint topic sentiment model was developed (Q. Yang et al. 2019) to represent the sentiment and topics based on the correlations between topics and sentiments. Nowadays, SA is considered a sequence model and faces many problems like sequence labeling, misjudgment of text, and loss of input information. (Li et al., 2020) addressed such issues by proposing SAMFBiLSTM variants to perform sentiment classification using multi-channel features. The proposed approach utilized a selfattention mechanism to help enhance the performance of sentiment and document level text classifications. The obtained results proved that both models had attained better results in the comparison of existing models. (Han et al. 2020) performed aspect level sentiment analysis on Drug reviews using a multi-task learning model with two BiGRU networks. They also proposed the Senti-Drugs dataset for analyzing aspect-level sentiment classification on drug reviews. The achieved results show that the performance of the proposed model improved when compared with modern approaches in the classification of drug reviews at the aspect-level.

Recently, several pre-trained language models like BERT(Devlin et al., 2019), ALBERT (Z. Lan et al., 2019), DistillBERT (Y.-C. Chen et al., 2020), BioBERT (Lee et al., 2020), RoBERTa(Liu et al., 2019), XLNet(Z. Yang et al., 2019) and etc… has shown great success in most of the NLP tasks.(Malhotra et al., 2021)proposed a novel transfer learning approach for the sentiment classification task to overcome the problems like contextualization and regularization. The proposed model combines the ULMFit (Howard &amp; Ruder, 2018) transfer learning method with forward and backward language models for highly effective results. The ensemble representation of this model is further utilized to extract features from embedding features based on attention, pooling, and concatenation mechanism with and without zeta parameter. In this paper, we propose a novel deep learning architecture to address the aforementioned limitations identified from the literature along with research gaps like implicit and explicit aspect extraction, determining the opinion targets from aspects, handling long-term dependencies based on the semantic and syntactic relationships between aspects. The main contributions of our work are as follows:

i. Sentiment analysis of covid reviews is conducted using the deep learning-based model.
ii. Six steps of pre-processing are performed to measure the influence of pre-processing on the efficiency of the sentiment classification task.
iii. The topic model (N-LDA) is constructed to discover the abstract topics by extracting the most dominant topics with significant aspects to Unigrams, Bigrams, and Trigrams.
iv. The polarity is computed on the dominant aspects using the VADER lexicon and categorized the aspects into positive, neutral, and negative classes.

v.A new pre-trained transformer is utilized to generate the contextual vectors of each aspect by incorporating deep learning knowledge to improve the generalization capability of sentiment identification by modeling the semantic and syntactic relations.

3. METHODOLOGY

The architecture of the developed model is illustrated in Figure 2. This framework is divided into six main components: (1) Data Collection (2) Pre-processing (3) Aspect Term Extraction (4) Topic Modeling (5) Sentiment Analysis, and (6) Sentiment Classification. The first component consists of the data collection, which deals with tweets' acquisition using Twitter API from Twitter. The second component deals with the preparation and preprocessing of the corpus by cleaning and sanitizing the tweets acquired from the previous component. The third component involves the identification of aspects through Bag-of-n-grams (BON) (Cho, 2018; Li et al. 2017) by creating a matrix with TF-IDF measure. The fourth component deals with the LDA topic model (Bagheri et al. 2014; Inouye et al. 2014; Jin et al. 2018; Lin et al. 2012; Pang et al. 2016) to discover the most dominant topics in the tweets based on n-gram aspects. Sentiment Analysis is carried out in the fifth component by computing the polarity of each n-gram aspect derived from the most dominant topics using the VADER lexicon (Adarsh et al. 2019; Amin et al. 2019; Harish Rao M, Shashikumar D.R, 2017; Kumaresh et al. 2019; Pandey, 2018; Thu & Aung, 2018). Therefore, the last component deals with the classification of sentiments of each aspect at the N-gram level using various deep learning approaches. An in-depth explanation of each component contained in the architecture is presented in the following sections.

3.1. Data Collection & Pre-processing

The dataset utilized in this research is concerning the tweets related to the Coronavirus. This dataset was initially designed and introduced by (Lamsal, 2020). A total of 954,927,708tweets were collected using search API provided by Twitterfrom20March 2020 to 23 January 2021. These search terms retrieve the user Tweets based on the hashtags, as represented in Table 1. Due to some technical issues, the data between 27 January 2020 and 20 March 2020 has been unpublished. Therefore, we collected a total of 205,459 public tweets, with 158,109 replied tweets from 133,007 users between 27 January 2020 and 20 March 2020. Out of these collected tweets, 125,744 tweets are identified as duplicated tweets, and 200 tweets contain less than 15 characters.

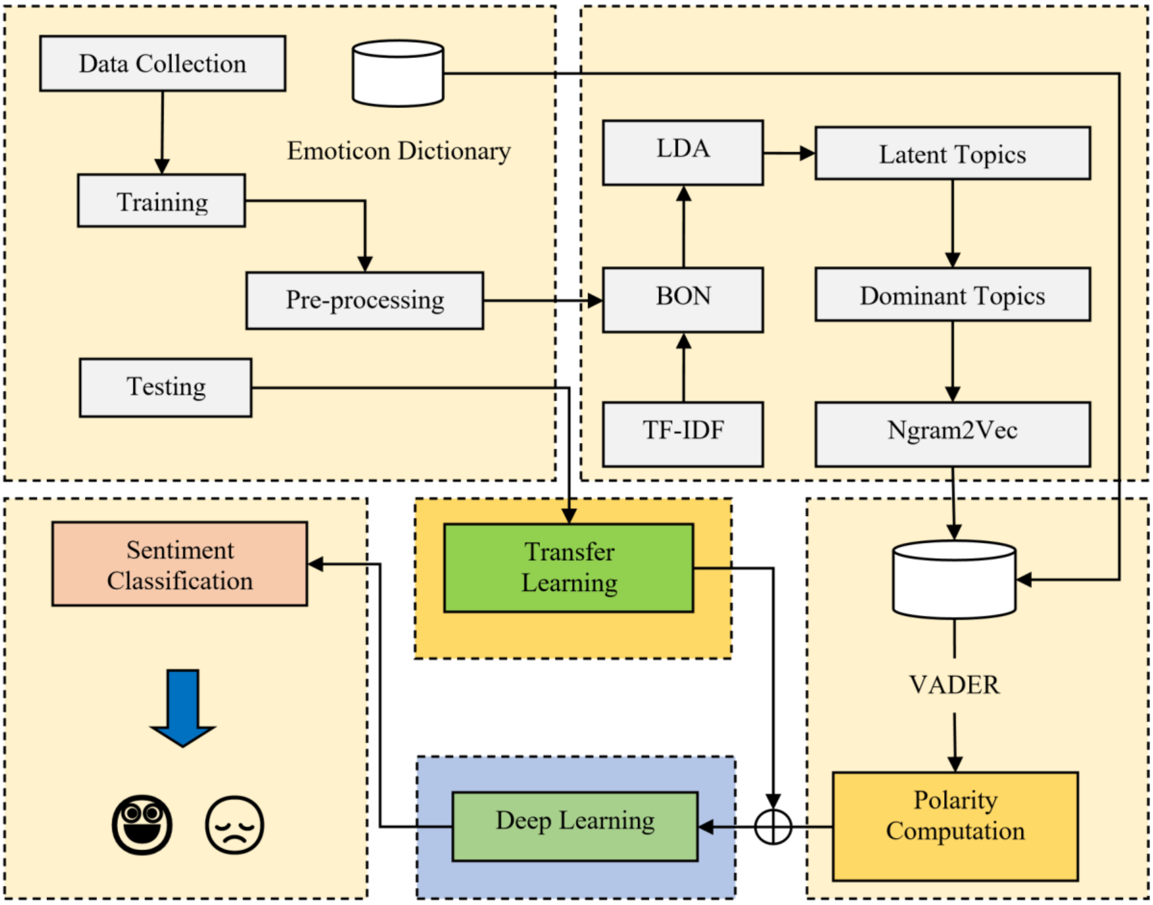


Fig 2. Proposed Workflow

As a primary step, the task of preprocessing is to provide data cleaning before moving into the further stages of analysis. The preprocessing component in Figure.2 consists of six stages; Tokenization, Data Cleaning, Spelling Correction, Stop Word Removal, Lemmatization, and Normalization. The output of this task will be a refined and quality corpus by applying these stages to the data characteristics present within the tweets/reviews. The flow of preprocessing is described in steps as follows:

i. Initially, all the collected tweets are tokenized to create an array of pre-processed tweets for further analysis.

ii. Data Cleaning is performed to obtain refined data by removing irrelevant information from the corpus. For each unique tweet in the corpus, the subsequent steps are performed;

a. We are removing identical tweets to obtain a set of unique tweets.

Table 2. Example List of Labeled Emoticon Dictionary

Emoticon	Label	Emoticon	Label
🙌	raising	👍	thumbsup
😨	fearful	🥺	pleading
😕	confused	😂	laughing
😐	neutral	😠	angry
☺	smiling	😒	dissatisfied

b. Generally, people habitually type tweets with emoticons. Such Emoticons help in determining the moods of the people. So, we do not intend to remove such emoticons from the tweets. Therefore, we develop a dictionary with a list of emoticons collected from a publicly available domain called https://emojipedia.org/. By specifying a label for each lexicon, we mapped each emoticon with its label among the tweet based on the developed emoticon dictionary. Table-1 indicates the example list of labeled emoticon dictionaries.

c. Removing URLs with "https://, Tag Mentions with 'RT @',"@", '#', and other Non-Ascii characters associated with the tweets.

d. De-Capitalization was performed on each tweet to transform all the Upper-Case characters from the tweet into Lower Case characters.

e. We are removing all the irrelevant numerical information like digits, roman numbers associated with the tweet using regular expressions. **iii.** Spell checking is done to fix the grammar mistakes within the tweets based on Hunspell dictionary.

iv. Stop-word Removal is done by removing the most-used English words like 'if,' 'the,' 'who,' etc., which do not carry any necessary information for the analysis and are usually discarded from the tweets.

v. Lemmatization, like Stemming, reduces the inflectional endings of a word to its base form. Stemming trims, the inflectional endings of a word, whereas Lemmatization uses the lexical knowledge of a word to get its base form. **vi.** Normalization in this work deals with the removal of extra whitespaces, short words, and long words from all the lemmatized reviews. We have identified a list of 14,370 irrelevant words along with all extra whitespaces, which could make our analysis more critical. Such words have been considered as own stop-words and were removed by adding them to the stop-word dictionary. **vii.** Finally, the arrays of tokenized tweets are normalized with 86,521 tokens that can be used for further text analysis.

3.2. Aspect Extraction from pre-processed tweets

This component involves the extraction of significant aspects by constructing an effective Bag-Of-Ngrams (BON) model utilizing an n-gram. To accurately capture sentiment from a review, this model utilizes an n-gram system to record how many times each mention appears in the preprocessed text. This concept was first explained by H. K. Kim et al. (2017), and further developed by R. Zhao and Mao (2018). Both BOW and BON do not split review text into words directly. BOW creates a bag by extracting individual words from the review text based on the frequent words and results in a considerable vocabulary list from the large-scale review corpus without any spatial information that could make the analysis more difficult. Based on these limitations, the BON model is used with the support of TF-IDF(G. Chen & Xiao, 2016; Erra et al. 2015; Qaiser & Ali, 2018; "TF-IDF," 2012) for extracting unique n-gram aspects. This BON model does not split text directly from the preprocessed review. Figures 3 shows the top n-grams extracted from the TF-IDF matrix. Initially, an empty bag is created with a different pair of words by extracting unique n-gram aspects based on its frequency counts. The frequency count of each unique n-gram aspect is calculated by constructing a TF-IDF matrix from the model bag. If the unique n-gram aspect in the model bag contains any missing values, then the corresponding values will be ignored.

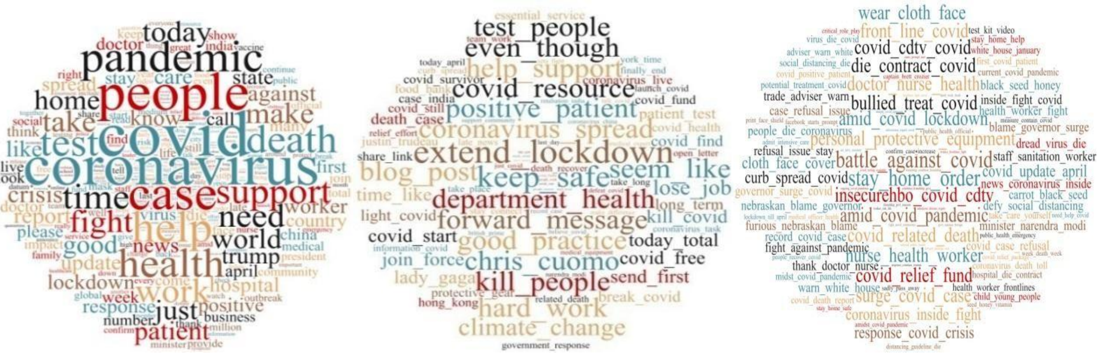


Fig 3. Example Representations of Top n-grams identified from TF-IDF Matrix

3.3. Topic Modeling using N-LDA

Table 3. Notations used for N-LDA

*N**	*Description*	*N**	*Description*
$\boldsymbol{R}$	Reviews	n_{rw}	Word frequencies in each review
$\boldsymbol{V}$	Vocabulary	n_{rk}	Set of frequent n-grams from each review
$\boldsymbol{K}$	Similar N-Grams	α_t, β_w	Parameters
$\boldsymbol{T}$	Number of Topics	w_a, w_b	Top-k-Words in a topic
$\boldsymbol{P}$	Perplexity	$p(t)$	Probability of appearing words in each topic
$\boldsymbol{C}$	Coherence	$p(r)$	Probability of appearing topics in each review
$\boldsymbol{e}$	exponent	$P(w_a), P(w_b)$	Probability of Top Words in a Topic
$\boldsymbol{t}$	Each Individual Topic	$P(w_a, w_b)$	Probabilities of n-grams in a topic
$\boldsymbol{w}$	Word	$p(ubt)$	Probability of appearing words in n-grams
$\boldsymbol{k}$	Set of frequent n-grams	$p(r)$	Probability of appearing n-grams from each review
$\boldsymbol{X_t}$	Set of top-k- n-grams	$P(w_a), P(w_b)$	Probability of Top Words in a Topic
$\boldsymbol{Y_t}$	Set of all possible n-grams	$P(w_a, w_b)$	Probabilities of n-grams in a topic
$\boldsymbol{K_w}$	Set of ngrams like words	$p(r, w)$	Probability of each topic appearing in review with words
$\boldsymbol{sc}$	Stop Condition	$P(w_a), P(w_b)$	Probability of Top Words in a Topic
$\boldsymbol{ubt}$	n-grams	$P(w_a, w_b)$	Probabilities of n-grams in a topic
N* = Notations used in LDA model			

Based on the frequent n-grams that occurred from the TD-IDF matrix, the co-occurred n-grams were modeled into a topical n-gram model using LDA (Elberrichi, 2006; Hardt et al., 2020; Jameel & Lam, 2013; Mandhula et al., 2019; Nikolenko et al., 2017). The topical n-gram model in this study uses multi-words of each n-gram length into a variety of topics. These topics consist of similar n-grams allocating the same terms frequently belong to similar topics. During the implementation, we use a set of K similar n-grams and words: $K = \{K_w\}$, where K_w represents the set of n-grams that are identical to t, which is denoted as follows:

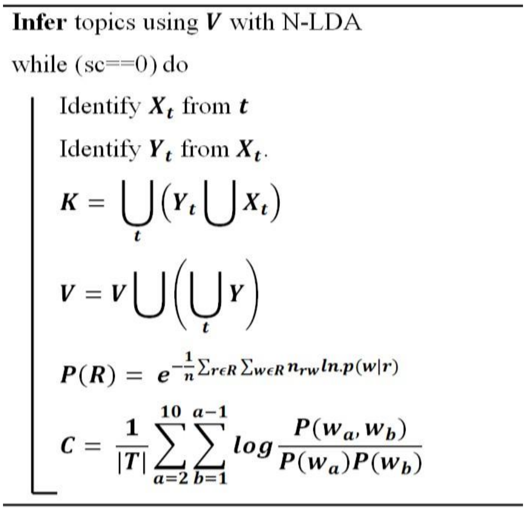
$$K_w = \{w \cup_n (\cup_{w^1 \dots w_n : \exists i: w_i = w} \; {}^1 w \dots w_n)\} \tag{1}$$

where n represents a word, and $w_1 \dots w_n$ represents the n-grams obtained from the preprocessed tokens. After, the vocabulary has been developed by adding the n-grams as single terms to each topic by identifying the top$-k$ n-grams by computing frequencies among various n-grams from each review r. The resulted frequencies are represented as n_{rw}. The below algorithm describes the process of LDA modeled with T = 100latent n-gram topics. It is quite like the traditional LDA model, but the only difference is to infer the n-grams can be under a similar topic or not. The probabilities of similar n-grams are computed based on their frequencies from the TF-IDF matrix (Devi et al. 2018). Later, we hypothesized that it is feasible to select the most suitable n-grams to fit into the topic models. All the possible top-k n-grams are modeled into the topic model based on the topics that have been inferred previously. Later, sets with top-K-terms from each topic and sets with all possible n-grams from those previous sets were selected at every iteration. As a result, a set of similar ngrams was produced using N-LDA. In this study, topic modeling is applied with 100 topics consisting of the top 100 terms in each topic for extraction of most dominant features using perplexity (P) and topic coherence scores to obtain the final optimal number of topics. These metrics judge how well the topic modeling has been done from Unigrams, Bigrams, and Trigrams. Finally, 62 topics with dominant features were obtained (Wang et al. 2007). Finally, the quality of each dominant topic is estimated by computing the topic probabilities of the n-grams from each topic are extracted from the dominant topics based on the highest probability score. Furthermore, the individual vocabulary list with the top words from the dominant topics has been constructed for each Unigrams, Bigrams, and Trigrams and given as input to the sentiment analysis component. At the same time, we have also developed the n-gram word embedding's from the vocabulary based on the Ngram2Vec approach (Z. Zhao et al. 2017). Figures 4-5 represent the optimal topics with top keywords identified from the N-LDA topic model for Unigrams, Bigrams, and Trigrams.

Algorithm 1: N-gram LDA

```
Input: $\boldsymbol{R}, \boldsymbol{V}, \boldsymbol{T}, \boldsymbol{p(w|t)}, \boldsymbol{p(t|r)}, \boldsymbol{\beta_w}$ and $\boldsymbol{\alpha_t}$ .
Output: $\boldsymbol{p(w|t)}$ and $\boldsymbol{p(t|r)}$
while (sc==0) do
    for $\boldsymbol{r \in R, w \in V, t \in T}$ do
        $p(t|r,w) = \frac{p(w|t)p(t|r)}{\sum_{ubt \in T} p(w|ubt)p(ubt|r)}$
    for $\boldsymbol{r \in R, w \in V, t \in T}$ do
        $n'_{rw} = n_{rw} + \sum_{k \in K_w} n_{rk}$
        $p(w|t) = \frac{\sum_{r \in R} n'_{rw} p(t|r,w) + \beta_w}{\sum_{r \in R} \sum_{w \in R} n'_{rw} p(t|r,w) + \sum_{w \in V} \beta_w}$
        $p(t|r) = \frac{\sum_{w \in R} n'_{rw} p(t|r,w) + \alpha_t}{\sum_{w \in V} \sum_{t \in T} n'_{rw} p(t|r,w) + \sum_{t \in T} \alpha_t}$
```

Algorithm 2: Iterative-LDA



```
Infer topics using $\boldsymbol{V}$ with N-LDA
while (sc==0) do
    Identify $\boldsymbol{X_t}$ from $\boldsymbol{t}$
    Identify $\boldsymbol{Y_t}$ from $\boldsymbol{X_t}$.
    $K = \bigcup_t \left(Y_t \bigcup X_t\right)$
    $V = V \bigcup \left(\bigcup_t Y\right)$
    $P(R) = e^{-\frac{1}{n}\sum_{r \epsilon R} \sum_{w \epsilon R} n_{rw} ln.p(w|r)}$
    $C = \frac{1}{|T|} \sum_{a=2}^{10} \sum_{b=1}^{a-1} log \frac{P(w_a, w_b)}{P(w_a)P(w_b)}$
```

3.4. Aspect-level Sentiment Analysis from n-grams

At this stage, sentiment analysis is performed at an aspect level, each word obtained from the N-LDA model's optimal topics. The topics with the highest perplexity and coherence scores were considered as significant topics, and the keywords from these topics were labeled by computing the polarity scores of each n-gram aspect using Valence Aware Dictionary and Sentiment Reasoner (VADER) algorithm (Crossley et al. 2017; Hiremath & Patil, 2020; C. S. P. Kumar & Babu, 2020; Ribeiro et al. 2016). The algorithm computes compound scores by aggregating individual n-gram scores that are updated according to the algorithm rules. Then the scores are normalized between -1, 0 and 1. Finally, we developed a Senti-Ngram based lexicon (Dey et al. 2018) with the obtained final sentiment list for our further analysis. The following Figure 4 presents a positive word cloud for Unigrams, Bigrams, and Trigrams and Figure 5 illustrates the negative word cloud.

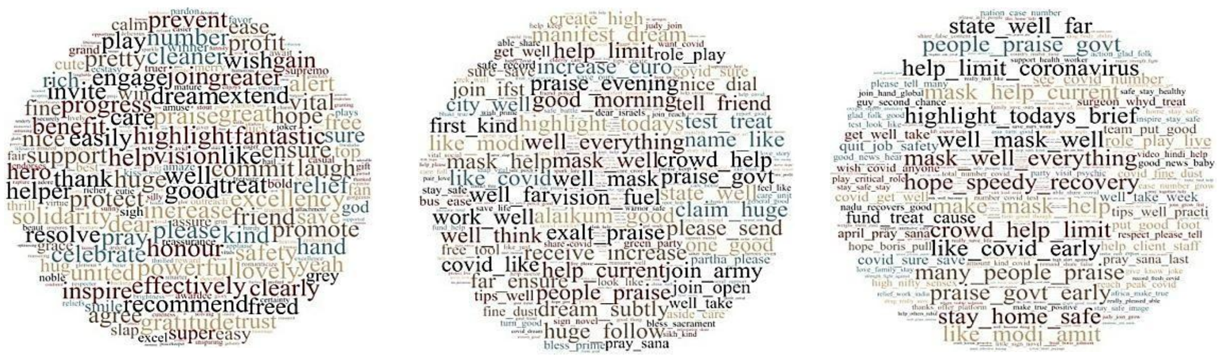


Fig 4. Positive word cloud for Unigram, Bigram, and Trigram feature

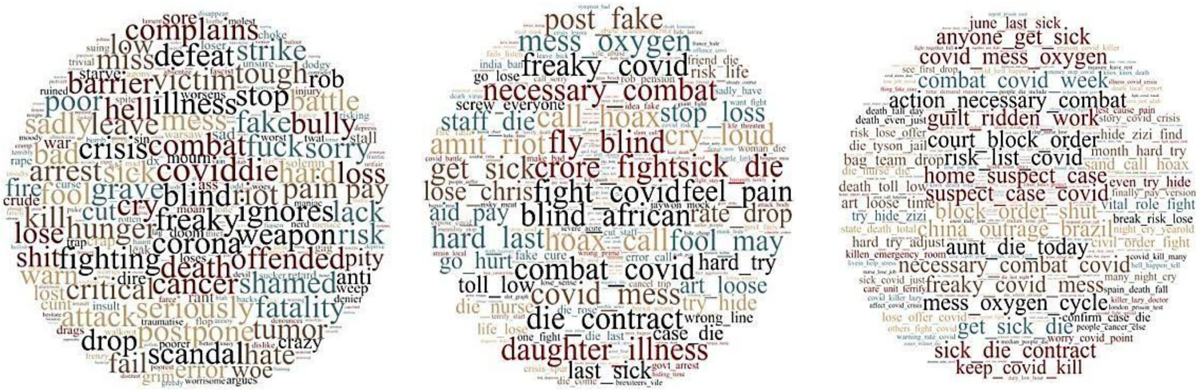


Fig 5. Negative word cloud for Unigram, Bigram, and Trigram feature

3.5. Aspect-level Sentiment Classification using Self attention-based BRCN

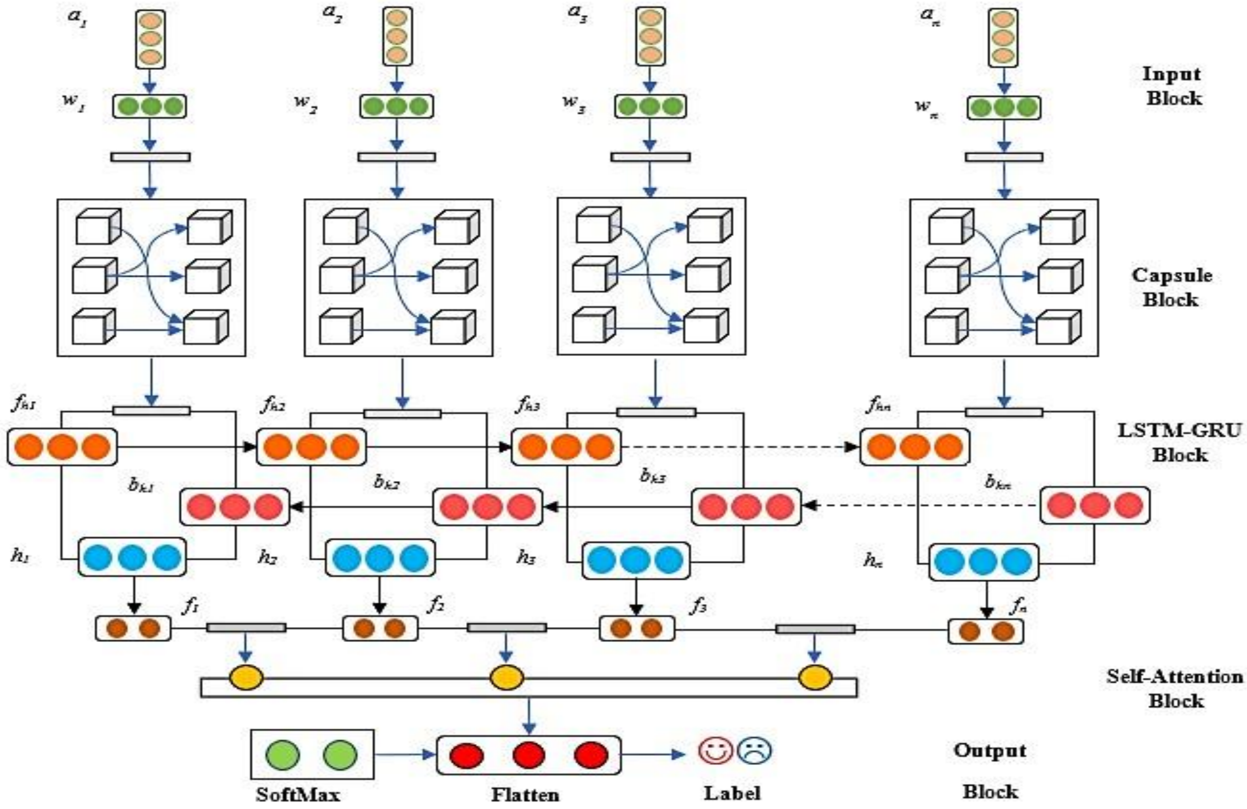


Fig 6. The framework of the Proposed BRCN model with Latent Knowledge Transfer

In this section, we proposed a sequence-based self-attentive recurrent BiLSTM – GRU neural network model for classifying the determined polarity aspects from the above sentiment analysis component. The benefit of using the LSTM network is that it improves over the RNN model by continuously updating the data in its memory through input and forget gates. In this case, the long-term dependencies can be effectively obtained by solving the problems like exploding gradient and vanishing gradient. In general, this network comprises of one input gate, forget gate, and output gate. Each GRU cell unit consists of final hidden vectors h_t obtained from BiLSTM, which are pooled into one vector o_t. The overall Framework of the proposed approach is presented in Figure 6. Initially, the proposed model is comprised with different layers like word embedding layer, capsule block, bidirectional LSTM-GRU block with self-attention, and softmax layer.

i.Word Embedding Layer:

Let A_i represents a sequence of aspects $A_{i1}, A_{i2}, \dots, A_{ik}, \dots, A_{in}$that has been learned into a Word Embedding Layer.We employed BERTweet (Nguyen et al., 2020) transformer at the Embedding Layer to generate the contextual vectors of each aspect in the given input.

This transformer utilizes the base architecture of conventional BERT which is trained with an objective of masked language modeling. The process of pre-training the BERTweet is based on RoBERTa that improves the performance of BERT by optimization. This paper does not discuss the architecture of BERT and RoBERTa transformers since they are widely used. Later, a pre-trained word embedding matrix $M^{n \times d}$ is constructed based on the contextual vectors generated by the BERTweet transformer and then each token is learned from A_i into a d- dimension vector. Therefore, $M^{n \times d}$ is considered as both parameters and input for the other layers of the BRCN.

ii.Capsule Block:

This block is a typical convolution block used for capturing the spatial relationships between outputs aspects derived from the embedding layer. An embedding layer with a transformer connects the capsule layer to the transformer so it can capture spatial informative features. Therefore, it enables the inputs in the form of scalar vectors and generates the output vectors called a capsule that preserves the semantic patterns among aspects based on their local order. The communication among these capsules is established by the dynamic routing based on the "routing-by-agreement" policy.

Algorithm 3: Modified Dynamic Routing Process

```
Input:$u_{Pj|i}$,            $r$;
Output:$v_{pj}$;  for all i^th and
j^th capsule:
$b_{pj|i} = 0$      for  $r$
iterations do;
    for all capsule $i^{th}$ and capsule $j^{th}$do;
     $c_{pj|i} \leftarrow \text{swish}(b_{pj|i})$;
    end for
    for all $j^{th}$ capsule do;
     $S_{pj} \leftarrow \sum_i c_{pij} u_{pj|i}$;
    end for
    for all $j^{th}$ capsule do;
     $v_{pj} \leftarrow \text{swift}(S_{pj})$ ;
    end for
    for all $i$ and capsule $j$do ;
      $b_{pij} \leftarrow b_{pij} + u_{pj|i} . v_{pj}$;
        end
for     end
for
        retur
n $v_{Dj}$;
```

The dynamic routing process simply updates the weights of each nearby capsule by computing the weight of the coupling coefficient. Therefore, similar capsules with greater coupling coefficients are considered based on their similarity. At this stage, the lower-level capsules in the consequent layer transfer the output to the higher-level capsules. The following Algorithm 3 describes the pseudo-code of the modified dynamic routing process in the capsule layer. From Algorithm 3, $u_{Rj|i}$ is the predictive vector transferred to the j^{th} digit capsule from the output of the i^{th} primary capsule. u_{Pi} is the primary capsule and w_{Pij} is the weight matrix. $b_{Pij}$is the prior logarithmic probability of both i^{th} and j^{th} capsules that have been set to 0. During the iteration process, $c_{Pj|i}$is the probability achieved by normalizing b_{Pij} with Swish function from the i^{th} primary capsule connects to the j^{th} capsule. Secondly, S_{Pj} of the digital capsule was obtained by computing the products of all the predictive vectors along with their connection probabilities as represented in the following equation 4.

$$u_{Pj|i} = w_{Pij} U_{Pi} \tag{2}$$

where $U_{Pi} = f(F_{N,d} W_{i2} + b_2)$ (3)

$$S_{Pj} = \sum_{j} c_{Pij} u_{Pj|i} \tag{4}$$

Thirdly, to obtain the output v_{Pj} of the digit capsule, we apply the swift function to S_{pj}as shown in equation 5.

$$v_{Pj} = \frac{||S_{Pj}||_2}{1 + exp\ (||S_{Pj}||)_{||S_{Pj}||}} . \frac{S_{Pj}}{||S_{Pj}||} \tag{5}$$

Finally, b_{Pij} is updated according to equation 6, and recur r the procedure until convergence.

$$b_{Pij} \leftarrow b_{Pij} + u_{Pj|i} . v_{Pj} \tag{6}$$

iii.BiLSTM – GRU Block:

A customized BiLSTM network is developed by constituting two independent LSTM networks with a GRU. Both of these approaches are the special kinds of RNN. LSTM can overcome the limitations of a standard RNN. The forget gate f_t determines which portion of the long-term state c_t should be dropped.

$$f_t = \sigma(w_{xfT} . x_t + w_{hTf} . h_{t-1} + b_f) \tag{7}$$

The input gate i_t influences which portion of the $\tilde{c_t}$ should be added to the long-term state c_t.

$$i_t = \sigma(w_x^T\ i . x_t + w_h^T\ i . h_{t-1} + b_i) \tag{8}$$

The final output gate g_t regulates which part of c_t should be interpreted and outputs to h_t and o_t.

$$o_t = \sigma(w_x^T\ o . x_t + w_h^T\ o . h_{t-1} + b_i) \tag{9}$$

$$g_t = \sigma(w_x^T\ g . x_t + w_h^T\ g . h_{t-1} + b_i) \tag{10}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c_t} \tag{11}$$

$$o_t, h_t = g_t * tanh\ (c_t) \tag{12}$$

where σ represents the sigmoid function, $w_{xf}{}^T$, $w_x{}^T\ i$, $w_x{}^T\ o$, $w_x{}^T\ g$ represents the weighted matrices and b_f, b_i biases are considered as parameters of the gates in an LSTM cell during training. For our model, we have incorporated two independent LSTM networks for handling both forward and backward dependencies between the features. Firstly, both the independent LSTMs feeds the knowledge in both directions of a word pairs and sums the knowledge coming out from both the directions. However, this model shares the same input word embedding in both the directions, though the parameters are independent. At each time t, thehidden vector $f_{ht}$is computed by the forward LSTM based on its prior hidden vector f_{ht-1} with input x_t. Similarly, the backward LSTM also computes the hidden vector b_{ht} at each time t based on its input x_t and prior hidden vector b_{ht-1}. Further, both the forward and backward hidden vectors f_{ht} and b_{ht} are concatenated to produce the final vectors of the BiLSTM model, which is represented as follows.

$$h_t = [f_{ht}, b_{ht}] \tag{13}$$

Later, a GRU is incorporate into the BiLSTM by utilizing the final vectors obtained from both the directions tracks of the BiLSTM model to develop a BRCN model. GRU is another kind of RNN and like the LSTM network. The only key difference between GRU and LSTM network is that LSTM employs three gates, whereas GRU employs only two gates, namely the 'reset' gate and 'update' gate. The following equations describe the key differences between LSTM and GRU.

$$r_t = \sigma(w_{xr}^T . x_t + w_{or}{}^T . o_{t-1} + b_r) \quad (14)$$

$$z_t = \sigma(w_{xz}^T . x_t + w_{oz}{}^T . o_{t-1} + b_z) \quad (15)$$

$$\tilde{o}_t = tanh(w_{x\tilde{o}}^T . x_t + w_{o\tilde{o}}^T . (r_t * o_{t-1}) + b_{\tilde{o}}) \quad (16)$$

$$o_t = z_t * o_{t-1} + (1 - z_t) * \tilde{o}_t \quad (17)$$

A GRU cell unit consists of final hidden vectors h_t obtained from BiLSTM, which are pooled into one vector o_t. There presents a gate controller whose job is to control both input and forget gates. The forget gate is opened when z_t outputs '1' and closes when z_t is '0'. Similarly, the input gate also opens when the forget gate is closed and closes when the forget gate is opened. In this scenario, the input of the time step is erased whenever the prior memory $(t - 1)$ is accumulated. In the exclusion of an output gate, it can be understood that the GRU is a distinct employment of the distribution and arrangement of the knowledge that LSTM needs to employ. Spontaneously, the reset gate decides how to amalgamate the new input with the prior memory. The computes the A new state is computed by update gate to determine the amount of retained knowledge in the prior memory. Thus, the obtained final vectors from the BRCN cells are measured as a final hidden vector f_t is presented in the equation (18):

$$f_t = [h_t, o_t] \quad (18)$$

iv.Self-Attention Block:

Traditionally, attention mechanisms have been used in image processing to train models based on definite feature information (Letarte et al. 2019; Yi Yang & Eisenstein, 2017; Zou et al. 2018). Such a mechanism is employed in the last state of the hidden layer to improve the efficiency of the BRCN by extracting more significant features based on the computation of higher weight. Particularly, the BRCN model will output a final hidden vector f_t as represented in Figure 7, the final hidden vector f_t is initially learned as an input into the self-attention layer. This layer extracts the internal correlation of the sentiment-related aspects from f_t by computing the weights of each sentiment aspect with the help of MLP and hence obtains a new hidden representation u_t. The importance of each aspect is then calculated based on the weight values for f_t given u_t and aspect level context vector u_w. Hence, u_w is judged as a higher dimensional representation to evaluate the importance of various aspects that are initialized randomly and learned jointly during training.

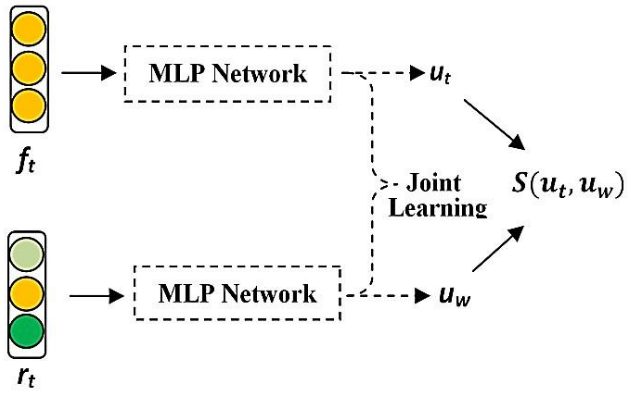


Fig 7. Joint Learning Process of Self-Attention Layer using MLP Network

Finally, the weighted mean of the final hidden vector f_t is therefore estimated by a Softmax function. Figure 7 illustrates the basic mechanism of self-attention, and the following equations (19), (20) & (21) describe each step of the process.

$$u_t = tanh\ tanh\ (W_w f_t + b_w) \quad (19)$$

$$\partial_t \quad \sum_t \quad e = \frac{e(u_t T u_w)}{\quad} T u_w) \quad (u_t \quad (20)$$

$$s = \sum_t \partial_t h_t \quad (21)$$

v.SoftMax Layer:

This layer outputs a higher-level representation of aspects and obtains sentiment aspect vector s based on the weight computed in the final hidden vector. Therefore, the polarity of each aspect is classified as:

$$\tilde{p} = softmax(W_s s + b_s) \tag{22}$$

4. EXPERIMENTAL RESULTS & DISCUSSIONS

4.1. Experimental Setup

Due to high timeliness and complexities that occurred while training the above models on the concerned datasets, the proposed work utilizes existing approaches like CNN, LSTM, RCNN, BiLSTM, and GRU (Biswas et al., 2015; Lien Minh et al., 2018; Luo, 2019; Penghua & Dingyi, 2019; Rozental & Fleischer, 2018; Tran et al., 2019; Wu et al., 2018) along with proposed approach are implemented in Google Colaboratory (Colab). The models used for this analysis were developed in the cloud based Jupyter Notebook with the Tensor flow as background.

4.2. Datasets

- **TCV-19:** This is a small dataset that contains a total of 79,811 tweets about coronavirus collected from various online communities on Twitter.
- **IMDB:** Internet Movie Database (IMDB) Dataset includes 50,000 polar movie comments, equally split to 25,000 positives and 25,000 negatives.
- **HNT:** This dataset covers 58,000 health news tweets collected from various health news organizations like BBC, CNN, and NYT.
- **MR:** This dataset comprises a total of 10,662 movie reviews, which are evenly divided into 5331 positive and 5331 negative reviews.

4.3. ANALYSIS & DISCUSSIONS

Initially, we performed all the six stages as defined in the above Figure 2on the collected datasets. The most dominant Ngrams are obtained using LDA topic modeling, and ASA is performed to obtain sentiment-related N-gram aspects. We do not consider the aspects with neutral polarity for this experimentation. Because neutral aspects are not contributing to results and unnecessarily increasing the training time, we have removed all the aspects with the neutral sentiment. Later, we split the obtained sentiment lists into training and testing for each dataset. All the N-gram aspects, along with its label, are vectorized into a list of vectors by developing n-gram word embedding's using the Ngram2Vec approach. After, we combined these word embedding's with a transformer-based multilingual masked language pre-trained model called BERTweet to extract the embedding's based on the context of word pairs and further trained with several deep learning models and the proposed model.

Firstly, a CNN model is created with three different kernel sizes, and a filter size of length 256 devised with a global maxpooling layer, a dropout layer, and a dense layer. The following graphs show the performance of the CNN model on different datasets with respective Unigrams, Bigrams, and Trigrams based on various evaluation metrics like precision, recall, f-score, and accuracy. Secondly, a single-layer LSTM is created with an embedding layer, a hidden layer with 256 input dimensions with a recurrent dropout condition with 0.3, flatten layer and a dense layer with sigmoid activation. Next, an RCNN is developed with an embedding layer, convolution layer with 256 filters, max-pooling layer, and single LSTM layer with 128 input dimensions followed by a sigmoid layer. Furthermore, a GRU is developed with the same parameters used for the LSTM network.

Finally, the BRCN is developed with an embedding layer, two independent forward and backward LSTM layers with 256 input dimensions enabled with return sequences and recurrent dropout at 0.3. The layers added next to BiLSTM are a GRU layer with 128 dimensions, a sequence-based self-attention layer with a sigmoid layer and a flatten layer, and a dense output layer with a softmax activation function. Once the model is successfully trained, it starts learning the input to all the layers and then produces the result on the predicted test set. Throughout the training procedure, the various hyper-parameters like batch size, loss, optimizer, and epochs are customized to optimize the final model's performance. Therefore, it is noticed that the efficiency is moderately increased, and time has been reduced when the batch size is increased from 128 to 256 with binary cross-entropy and an optimizer loss function with a learning rate of 0.001. Further, the below section describes the achieved results and comparative performance of the experiments conducted on different datasets with various models based on the following hyper-parameters, as presented in Table-4.

Table 4. Details of Hyper-Parameters

Parameters	Values	Parameters	Values
Vocabulary Size	20000	No. of Epochs	25

Train & Test Split	70:30	Batch Size	256
Maximum word length	3	Optimizer	Adam
Embedding Dimensions	300	Loss function	Cross-Entropy

Figure 8 illustrates the performance evaluation and comparison of various classifiers on the collected datasets. On DB-1, the CNN model performs better with Bigrams (83%) than Unigrams (78.26%) and Trigrams (81.33%). The LSTM model achieved better performance with Trigrams (88.40%) than Unigrams (83.92%) and Bigrams (87%). The GRU model also outperforms with Trigrams (87%) when compared to both Unigrams (83.46%) and Trigrams (86.10%). The BiLSTM model achieved better results with Bigrams (91.33%) than Unigrams (87.30%) and Trigrams (90.36%). Similarly, the RCNN model outperforms better with Trigrams (88.90%) when compared with both Unigrams (85.64%) and Bigrams (87%). On DB-2, the CNN model obtains better accuracy with Bigrams (86.24%) than Unigrams (83%) and Trigrams (85.16%). A simple LSTM model performs better with Trigrams (89.47%) than Unigrams (85.54%) and Bigrams (88%). The GRU model also achieved a better result with Trigrams (89%) when compared to both Unigrams (86.38%) and Bigrams (89%). The BiLSTM model outperforms with Bigrams (93.67%) than Unigrams (90.23%) and Trigrams (92.15%). Similarly, the RCNN model outperforms with Trigrams (90.31%) when compared with both Unigrams (88.24%) and Bigrams (90.31%). On DB-3, the CNN model outperforms with Trigrams (73.33%) than Unigrams (70.21%) and Bigrams (75.60%). A simple LSTM model achieved better accuracy with Bigrams (80.37%) than Unigrams (76.54%) and Trigrams (79.43%). The GRU model also performs better in Bigrams (79.83%) when compared to both Unigrams (75.36%) and Trigrams (77.32%). The BiLSTM model performs better with Trigrams (85.26%) than Unigrams (80.82%) and Bigrams (83.20%).

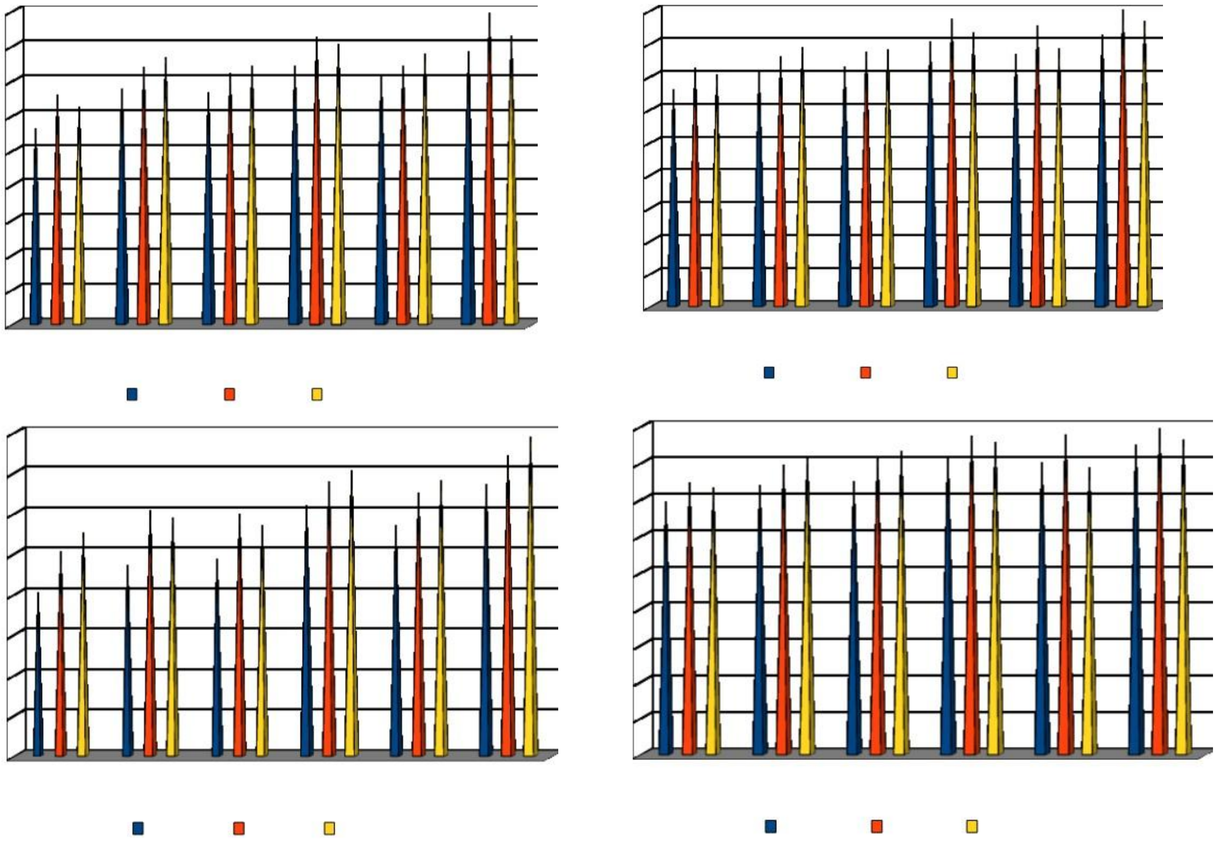


Fig 8. Performance comparison of various classifiers on benchmark datasets

Similarly, the RCNN model outperforms better with Trigrams (84%) when compared with both Unigrams (78.43%) and Bigrams (82.10%). On DB-4, as shown in Figure, the CNN model outperforms with Bigrams (87.21%) than Unigrams (84.67%) and Trigrams (86.52%). LSTM model performs better with Trigrams (90.87%) than Unigrams (86.93%) and Bigrams (89.62%). The GRU model also achieved better performance with Trigrams (91.56%) when compared to both Unigrams (87.48%) and Bigrams (90.87%). The BiLSTM model performs better with Bigrams (94.66%) than Unigrams (91.12%) and Trigrams (93.72%). Similarly, the RCNN model outperforms with Bigrams (93.88%) when compared with both Unigrams (90%) and Trigrams (89.38%). Finally, we also conducted several experiments with our proposed model on various datasets concerning Unigrams, Bigrams, and Trigrams. On DB-1, the proposed model achieves better accuracy with Bigrams (94.62%) over Unigrams (89.18%) and Trigrams (91.48%).

Table 5. Comparison of various classification metrics of the proposed model with other classifiers on all benchmark datasets

		Unigrams			Bigrams			Trigrams		
Datasets	Model	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
DB 1	CNN	0.782	0.781	0.781	0.830	0.830	0.830	0.813	0.811	0.812
	LSTM	0.839	0.837	0.838	0.870	0.868	0.869	0.884	0.882	0.883
	GRU	0.834	0.832	0.833	0.860	0.860	0.860	0.870	0.870	0.870
	Bi-LSTM	0.873	0.871	0.872	0.913	0.911	0.912	0.903	0.901	0.902
	RCNN	0.856	0.854	0.855	0.870	0.870	0.870	0.889	0.887	0.888
	Proposed	0.892	0.891	0.893	0.945	0.944	0.946	0.913	0.912	0.914
DB 2	CNN	0.830	0.830	0.830	0.862	0.860	0.861	0.852	0.850	0.851
	LSTM	0.855	0.853	0.854	0.880	0.880	0.880	0.894	0.895	0.893
	GRU	0.863	0.861	0.862	0.885	0.883	0.884	0.890	0.890	0.890
	Bi-LSTM	0.902	0.900	0.901	0.936	0.934	0.935	0.917	0.915	0.916
	RCNN	0.882	0.880	0.881	0.926	0.924	0.925	0.893	0.892	0.891
	Proposed	0.911	0.910	0.912	0.950	0.950	0.950	0.932	0.931	0.933
DB 3	CNN	0.702	0.700	0.701	0.753	0.751	0.752	0.776	0.774	0.775
	LSTM	0.735	0.733	0.734	0.803	0.801	0.802	0.794	0.792	0.793
	GRU	0.743	0.741	0.742	0.798	0.796	0.797	0.785	0.783	0.784
	Bi-LSTM	0.808	0.806	0.807	0.838	0.836	0.837	0.852	0.850	0.851
	RCNN	0.784	0.782	0.783	0.825	0.823	0.824	0.840	0.840	0.840
	Proposed	0.835	0.834	0.836	0.871	0.870	0.872	0.893	0.891	0.894
DB 4	CNN	0.846	0.844	0.845	0.872	0.870	0.871	0.865	0.863	0.864
	LSTM	0.869	0.867	0.868	0.896	0.894	0.895	0.908	0.906	0.907
	GRU	0.874	0.872	0.873	0.909	0.907	0.908	0.915	0.913	0.914
	Bi-LSTM	0.908	0.906	0.907	0.936	0.934	0.935	0.927	0.925	0.926
	RCNN	0.900	0.900	0.900	0.938	0.936	0.937	0.893	0.891	0.892
	Proposed	0.925	0.924	0.926	0.947	0.946	0.948	0.936	0.935	0.937

Correspondingly, it also achieved better performance on DB-2 with Trigrams (95%) than Unigrams (91.18%) and Bigrams (93.25%). On DB-3, the proposed model also outperforms with Trigrams (89.42%) when compared to both Unigrams

(83.56%) and Bigrams (87.18%). Finally, on DB-4, the proposed model achieved better performance with Bigrams (94.72%) than Unigrams (92.53%) and Trigrams (93.10%). Later, each model's running time complexities are analyzed separately on each dataset for a better and more accurate understanding of model performance in terms of efficiency. Due to the number of layers in the network, the proposed model displays increased time complexity. Table 5 compares the proposed model with existing models based on Precision, Recall, and F-Score evaluation metrics. From these observations, we came to know that the models trained with Unigrams have high time complexity and do not improve accuracy due to the high range of sparseness among the feature space. Such sparseness is relatively reduced from the feature space by learning the model with Bigrams and Trigrams. Bigrams improve both classification accuracy and time complexity by obtaining quality features. Models trained with quality features gradually improve efficiency by adding the phrasal features based on various POS combinations. Models trained with Trigrams also improved a lot in some cases by reducing sparse features and extracting the phrases with semantic information that could relatively increases the efficiency of the models. Based on these observations, when compared to other models, the proposed BRCN significantly improves ASA's performance.

5. CONCLUSION AND FUTURE SCOPE

The BRCN approach is proposed to analyze the public reactions and responses to tweets about Covid-19. According to the architectural flow, the collected tweets were well-processed by discarding irrelevant information. The emoticons associated with the tweets were manually annotated and labeled by adding them to the emoticon dictionary. Later, aspectlevel feature extraction was done by extracting the n-gram features using Unigrams, Bigrams, and Trigrams with TF-IDF measures. To better understand the contextual aspects, we employed LDA with n-gram features and determined the topmost n-gram features from several topics by computing perplexity and coherence scores. Moreover, sentiment analysis was performed on the topmost n-gram features by computing the compound polarity using VADER Lexicon. The final sentiment list obtained with n-gram features was incorporated into word embedding's by developing the vocabulary with the Ngram2Vec approach. Further, these embedding's were combined with a pre-trained BERTweet transformer model and learned as an input to the various existing models, like CNN, LSTM, BiLSTM, GRU, and RCNN, with several parameters used in this study. The experimentation with these models, along with BRCN, was performed on various standard datasets, and the performances of each of these models were compared by various evaluation metrics. The achieved results state that the proposed model outperforms all the traditional models from the literature and gains better computational efficiency with all Unigrams, Bigrams, and Trigrams. Besides, the computational time complexities were also analyzed to assess the performance of the existing aproaches. The advantage of the proposed model is that it produces a better classification rate by handling the dependencies among features by determining their semantic and syntactical relations. Our proposed model's only limitation is that it delivers the weak generalization ability on the new datasets containing fewer polarity aspects with huge sparseness among the feature space. Similarly, the running time complexity also increases in some cases while working with hyper-parameters to improve accuracy. In this case, we performed some average successful runs by adjusting the parameters that improve our model's accuracy with increased running time. Consequently, to improve the generalization ability, we will try to work on feature extraction techniques in the immediate future to diminish the sparseness among the feature space in new datasets.

6. REFERENCES

1. Abdi, A., Shamsuddin, S. M., Hasan, S., & Piran, J. (2019). Deep learning-based sentiment classification of evaluative text based on Multi-feature fusion. *Information Processing and Management*, *56*(4), 1245–1259. https://doi.org/10.1016/j.ipm.2019.02.018
2. Adarsh, R., Patil, A., Rayar, S., & Veena, K. M. (2019). Comparison of VADER and LSTM for sentiment analysis. *International Journal of Recent Technology and Engineering*.
3. Ait Hammou, B., Ait Lahcen, A., & Mouline, S. (2020). Towards a real-time processing framework based on improved distributed recurrent neural network variants with fastText for social big data analytics. *Information Processing and Management*, *57*(1), 102122. https://doi.org/10.1016/j.ipm.2019.102122
4. Al-Smadi, M., Talafha, B., Al-Ayyoub, M., & Jararweh, Y. (2019). Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews. *International Journal of Machine Learning and Cybernetics*. https://doi.org/10.1007/s13042-018-0799-4
5. Alaei, A. R., Becken, S., & Stantic, B. (2019). Sentiment Analysis in Tourism: Capitalizing on Big Data. *Journal of Travel Research*, *58*(2), 175–191. https://doi.org/10.1177/0047287517747753
6. Amin, A., Hossain, I., Akther, A., & Alam, K. M. (2019). Bengali VADER: A Sentiment Analysis Approach Using Modified VADER. *2nd International Conference on Electrical, Computer and Communication Engineering, ECCE 2019*. https://doi.org/10.1109/ECACE.2019.8679144
7. Bagheri, A., Saraee, M., & de Jong, F. (2014). ADM-LDA: An aspect detection model based on topic modeling using the structure of review sentences. *Journal of Information Science*, *40*(5), 621–636. https://doi.org/10.1177/0165551514538744
8. Barkur, G., Vibha, & Kamath, G. B. (2020). Sentiment analysis of nationwide lockdown due to COVID 19 outbreak: Evidence from India. *Asian Journal of Psychiatry*, *51*(April), 102089. https://doi.org/10.1016/j.ajp.2020.102089
9. Baziotis, C., Pelekis, N., & Doulkeridis, C. (2018). *DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis*. https://doi.org/10.18653/v1/s17-2126

10. Beryl Joylin, A., Aswathi, T., Suma, P., & Victor, N. (2016). Sentiment analysis on "ebola" outbreak using twitter data. *International Journal of Pharmacy and Technology*.
11. Biswas, S., Chadda, E., & Ahmad, F. (2015). Sentiment analysis with gated recurrent units. *Advances in Computer Science and Information Technology (ACSIT)*.
12. Cambria, E., Schuller, B., Xia, Y., & Havasi, C. (2013). New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*. https://doi.org/10.1109/MIS.2013.30
13. Chen, G., & Xiao, L. (2016). Selecting publication keywords for domain analysis in bibliometrics: A comparison of three methods. *Journal of Informetrics*. https://doi.org/10.1016/j.joi.2016.01.006
14. Chen, H., Li, S., Wu, P., Yi, N., Li, S., & Huang, X. (2018). Fine-grained sentiment analysis of Chinese reviews using LSTM network. *Journal of Engineering Science and Technology Review*, *11*(1), 174–179. https://doi.org/10.25103/jestr.111.21
15. Chen, T., Xu, R., He, Y., & Wang, X. (2017). Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems with Applications*, *72*, 221–230. https://doi.org/10.1016/j.eswa.2016.10.065
16. Chen, Y.-C., Gan, Z., Cheng, Y., Liu, J., & Liu, J. (2020). *Distilling Knowledge Learned in BERT for Text Generation*. https://doi.org/10.18653/v1/2020.acl-main.705
17. Cho, K. (2018). *Strawman: An Ensemble of Deep Bag-of-Ngrams for Sentiment Analysis*. https://doi.org/10.18653/v1/w17-5409
18. Crossley, S. A., Kyle, K., & McNamara, D. S. (2017). Sentiment Analysis and Social Cognition Engine (SEANCE): An automatic tool for sentiment, social cognition, and social-order analysis. *Behavior Research Methods*, *49*(3), 803–821. https://doi.org/10.3758/s13428-016-0743-z
19. Devi, K. A., Edara, D. C., Sistla, V. P. K., & Kolli, V. K. K. (2018). Extended correlated principal component analysis with SVM-PUK in opinion mining. *Turkish Journal of Electrical Engineering and Computer Sciences*, *26*(5), 2570–2582. https://doi.org/10.3906/elk-1704-178
20. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*.
21. Dey, A., Jenamani, M., & Thakkar, J. J. (2018). Senti-N-Gram: An n-gram lexicon for sentiment analysis. *Expert Systems with Applications*, *103*, 92–105.
22. Do, H. H., Prasad, P. W. C., Maag, A., & Alsadoon, A. (2019). Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review. In *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2018.10.003
23. Edara, D. C., Vanukuri, L. P., Sistla, V., & Kolli, V. K. K. (2019). Sentiment analysis and text categorization of cancer medical records with LSTM. *Journal of Ambient Intelligence and Humanized Computing*, *0123456789*. https://doi.org/10.1007/s12652-019-01399-8
24. Elberrichi, Z. (2006). Text mining using N-grams. *Proceedings of the International Conference on Computer Science and Its Applications*.
25. Erra, U., Senatore, S., Minnella, F., & Caggianese, G. (2015). Approximate TF-IDF based on topic extraction from massive message stream using the GPU. *Information Sciences*. https://doi.org/10.1016/j.ins.2014.08.062
26. Feizollah, A., Ainin, S., Anuar, N. B., Abdullah, N. A. B., & Hazim, M. (2019). Halal Products on Twitter: Data Extraction and Sentiment Analysis Using Stack of Deep Learning Algorithms. *IEEE Access*. https://doi.org/10.1109/ACCESS.2019.2923275
27. Fu, X., Yang, J., Li, J., Fang, M., & Wang, H. (2018). Lexicon-Enhanced LSTM with Attention for General Sentiment Analysis. *IEEE Access*. https://doi.org/10.1109/ACCESS.2018.2878425
28. Ghorbani, M., Bahaghighat, M., Xin, Q., & Özen, F. (2020). ConvLSTMConv network: a deep learning approach for sentiment analysis in cloud computing. *Journal of Cloud Computing*, *9*(1), 1–12. https://doi.org/10.1186/s13677020-00162-1
29. Habimana, O., Li, Y., Li, R., Gu, X., & Yu, G. (2020). Sentiment analysis using deep learning approaches: an overview. *Science China Information Sciences*, *63*(1), 1–36. https://doi.org/10.1007/s11432-018-9941-6
30. Han, Y., Liu, M., & Jing, W. (2020). Aspect-Level Drug Reviews Sentiment Analysis Based on Double BiGRU and Knowledge Transfer. *IEEE Access*, *8*, 21314–21325. https://doi.org/10.1109/ACCESS.2020.2969473
31. Hardt, D., Hovy, D., & Lamprinidis, S. (2020). Predicting news headline popularity with syntactic and semantic knowledge using multi-task learning. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*.
32. Harish Rao M , Shashikumar D.R, H. R. M. , S. D. . (2017). Automatic Product Review Sentiment Analysis Using Vader and Feature Visulaization. *International Journal of Computer Science Engineering and Information Technology Research*. https://doi.org/10.24247/ijcseitraug20178

33. Hassan, A., & Mahmood, A. (2018). Convolutional Recurrent Deep Learning Model for Sentence Classification. *IEEE Access*. https://doi.org/10.1109/ACCESS.2018.2814818
34. Heikal, M., Torki, M., & El-Makky, N. (2018). Sentiment Analysis of Arabic Tweets using Deep Learning. *Procedia Computer Science*. https://doi.org/10.1016/j.procs.2018.10.466
35. Hiremath, B. N., & Patil, M. M. (2020). Enhancing Optimized Personalized Therapy in Clinical Decision Support System using Natural Language Processing. *Journal of King Saud University - Computer and Information Sciences*, *xxxx*. https://doi.org/10.1016/j.jksuci.2020.03.006
36. Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*. https://doi.org/10.18653/v1/p18-1031
37. Ichev, R., & Marinč, M. (2018). Stock prices and geographic proximity of information: Evidence from the Ebola outbreak. *International Review of Financial Analysis*. https://doi.org/10.1016/j.irfa.2017.12.004
38. Inouye, D., Ravikumar, P., & Dhillon, I. (2014). Admixture of Poisson MRFs: A Topic Model with Word Dependencies. *Journal of Machine Learning Research*, *Icml*, 683–691. http://jmlr.org/proceedings/papers/v32/inouye14.html
39. Jain, D. K., Jain, R., Upadhyay, Y., Kathuria, A., & Lan, X. (2020). Deep Refinement: capsule network with attention mechanism-based system for text classification. *Neural Computing and Applications*, *32*(7), 1839–1856. https://doi.org/10.1007/s00521-019-04620-z
40. Jameel, S., & Lam, W. (2013). An n-gram topic model for time-stamped documents. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-642-36973-5_25
41. Jin, M., Luo, X., Zhu, H., & Zhuo, H. H. (2018). Combining deep learning and topic modeling for review understanding in context-Aware recommendation. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*, 1605–1614. https://doi.org/10.18653/v1/n18-1145
42. John, B., Baulch, B., & Wickramasinghe, N. (2019). *A Sentiment Analysis of the 2014-15 Ebola Outbreak in the Media and Social Media*. https://doi.org/10.4018/978-1-7998-1371-2.ch014
43. Kastrati, Z., Imran, A. S., & Yayilgan, S. Y. (2019). The impact of deep learning on document classification using semantically rich representations. *Information Processing and Management*, *56*(5), 1618–1632. https://doi.org/10.1016/j.ipm.2019.05.003
44. Khan, K., Baharudin, B., Khan, A., & Ullah, A. (2014). Mining opinion components from unstructured reviews: A review. In *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2014.03.009
45. Kim, E. H. J., Jeong, Y. K., Kim, Y., Kang, K. Y., & Song, M. (2015). Topic-based content and sentiment analysis of Ebola virus on Twitter and in the news. *Journal of Information Science*. https://doi.org/10.1177/0165551515608733
46. Kim, H. K., Kim, H., & Cho, S. (2017). Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. *Neurocomputing*. https://doi.org/10.1016/j.neucom.2017.05.046
47. Kouzy, R., Abi Jaoude, J., Kraitem, A., El Alam, M. B., Karam, B., Adib, E., Zarka, J., Traboulsi, C., Akl, E., & Baddour, K. (2020). Coronavirus Goes Viral: Quantifying the COVID-19 Misinformation Epidemic on Twitter. *Cureus*. https://doi.org/10.7759/cureus.7255
48. Kruthika, K. R., Rajeswari, & Maheshappa, H. D. (2019). CBIR system using Capsule Networks and 3D CNN for Alzheimer's disease diagnosis. *Informatics in Medicine Unlocked*, *14*(November 2018), 59–68. https://doi.org/10.1016/j.imu.2018.12.001
49. Kumar, C. S. P., & Babu, L. D. D. (2020). Evolving dictionary based sentiment scoring framework for patient authored text. *Evolutionary Intelligence*, *september 2011*. https://doi.org/10.1007/s12065-020-00366-z
50. Kumar, R., Pannu, H. S., & Malhi, A. K. (2020). Aspect-based sentiment analysis using deep networks and stochastic optimization. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-019-04105-z
51. Kumaresh, N., Bonta, V., & Janardhan, N. (2019). A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis. In *Asian Journal of Computer Science and Technology*.
52. Lamsal, R. (2020). Design and analysis of a large-scale COVID-19 tweets dataset. *Applied Intelligence*. https://doi.org/10.1007/s10489-020-02029-z
53. Lan, M., Zhang, Z., Lu, Y., & Wu, J. (2016). Three Convolutional Neural Network-based models for learning Sentiment Word Vectors towards sentiment analysis. *Proceedings of the International Joint Conference on Neural Networks*, *2016-Octob*, 3172–3179. https://doi.org/10.1109/IJCNN.2016.7727604
54. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. In *arXiv*.

55. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btz682
56. Letarte, G., Paradis, F., Giguère, P., & Laviolette, F. (2019). *Importance of Self-Attention for Sentiment Analysis*. https://doi.org/10.18653/v1/w18-5429
57. Li, B., Liu, T., Zhao, Z., Wang, P., & Du, X. (2017). Neural bag-of-ngrams. *31st AAAI Conference on Artificial Intelligence, AAAI 2017*.
58. Li, W., Qi, F., Tang, M., & Yu, Z. (2020). Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification. *Neurocomputing*, *387*(xxxx), 63–77. https://doi.org/10.1016/j.neucom.2020.01.006
59. Liao, S., Wang, J., Yu, R., Sato, K., & Cheng, Z. (2017). CNN for situations understanding based on sentiment analysis of twitter data. *Procedia Computer Science*. https://doi.org/10.1016/j.procs.2017.06.037
60. Lien Minh, D., Sadeghi-Niaraki, A., Huy, H. D., Min, K., & Moon, H. (2018). Deep learning approach for shortterm stock trends prediction based on two-stream gated recurrent unit network. *IEEE Access*. https://doi.org/10.1109/ACCESS.2018.2868970
61. Lin, C., He, Y., Everson, R., & Rüger, S. (2012). Weakly supervised joint sentiment-topic detection from text. *IEEE Transactions on Knowledge and Data Engineering*, *24*(6), 1134–1145. https://doi.org/10.1109/TKDE.2011.48
62. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. In *arXiv*.
63. Luo, L. xia. (2019). Network text sentiment analysis method combining LDA text representation and GRU-CNN. *Personal and Ubiquitous Computing*. https://doi.org/10.1007/s00779-018-1183-9
64. Ma, X., Zeng, J., Peng, L., Fortino, G., & Zhang, Y. (2019). Modeling multi-aspects within one opinionated sentence simultaneously for aspect-level sentiment analysis. *Future Generation Computer Systems*, *93*, 304–311. https://doi.org/10.1016/j.future.2018.10.041
65. Ma, Y., Peng, H., Khan, T., Cambria, E., & Hussain, A. (2018). Sentic LSTM: a Hybrid Network for Targeted Aspect-Based Sentiment Analysis. *Cognitive Computation*, *10*(4), 639–650. https://doi.org/10.1007/s12559-0189549-x
66. Majumder, N., Poria, S., Peng, H., Chhaya, N., Cambria, E., & Gelbukh, A. (2019). Sentiment and Sarcasm Classification with Multitask Learning. *IEEE Intelligent Systems*, *34*(3), 38–43. https://doi.org/10.1109/MIS.2019.2904691
67. Malhotra, S., Kumar, V., & Agarwal, A. (2021). Bidirectional transfer learning model for sentiment analysis of natural language. *Journal of Ambient Intelligence and Humanized Computing*, *2018*. https://doi.org/10.1007/s12652020-02800-7
68. Mandhula, T., Pabboju, S., & Gugulotu, N. (2019). Predicting the customer's opinion on amazon products using selective memory architecture-based convolutional neural network. *Journal of Supercomputing*, *0123456789*. https://doi.org/10.1007/s11227-019-03081-4
69. Naeim, M. (2020). Coronavirus disease (COVID-19) outbreak provides a unique platform to review behavioral changes in Iran. *Asian Journal of Psychiatry*, *51*, 102090. https://doi.org/10.1016/j.ajp.2020.102090
70. Nguyen, D. Q., Vu, T., & Nguyen, A. T. (2020). BERTweet: A pre-trained language model for english tweets. In *arXiv*. https://doi.org/10.18653/v1/2020.emnlp-demos.2
71. Nikolenko, S. I., Koltcov, S., & Koltsova, O. (2017). Topic modelling for qualitative studies. *Journal of Information Science*. https://doi.org/10.1177/0165551515617393
72. Pandey, P. (2018). Simplifying Sentiment Analysis using VADER in Python (on Social Media Text). In *Analytics Vidhya*.
73. Pang, X., Wan, B., Li, H., & Lin, W. (2016). MR-LDA: An efficient topic model for classification of short text in big social data. *International Journal of Grid and High Performance Computing*, *8*(4), 100–113. https://doi.org/10.4018/IJGHPC.2016100106
74. Park, H., Reber, B. H., & Chon, M. G. (2016). Tweeting as health communication: Health organizations use of twitter for health promotion and public engagement. *Journal of Health Communication*. https://doi.org/10.1080/10810730.2015.1058435
75. Penghua, Z., & Dingyi, Z. (2019). Bidirectional-GRU based on attention mechanism for aspect-level Sentiment Analysis. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3318299.3318368
76. Qaiser, S., & Ali, R. (2018). Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. *International Journal of Computer Applications*. https://doi.org/10.5120/ijca2018917395
77. Ravi, K., & Ravi, V. (2015). A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. *Knowledge-Based Systems*, *89*, 14–46. https://doi.org/10.1016/j.knosys.2015.06.015

78. Rehman, A. U., Malik, A. K., Raza, B., & Ali, W. (2019). A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-01907788-7
79. Rhanoui, M., Mikram, M., Yousfi, S., & Barzali, S. (2019). A CNN-BiLSTM Model for Document-Level Sentiment Analysis. *Machine Learning and Knowledge Extraction*. https://doi.org/10.3390/make1030048
80. Ribeiro, F. N., Araújo, M., Gonçalves, P., André Gonçalves, M., & Benevenuto, F. (2016). SentiBench - a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, *5*(1), 1–29. https://doi.org/10.1140/epjds/s13688-016-0085-1
81. Rozental, A., & Fleischer, D. (2018). *Amobee at SemEval-2018 Task 1: GRU Neural Network with a CNN Attention Mechanism for Sentiment Classification*. https://doi.org/10.18653/v1/s18-1033
82. Schouten, K., & Frasincar, F. (2016). Survey on Aspect-Level Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering*, *28*(3), 813–830. https://doi.org/10.1109/TKDE.2015.2485209
83. Shin, B., Lee, T., & Choi, J. D. (2018). *Lexicon Integrated CNN Models with Attention for Sentiment Analysis*. https://doi.org/10.18653/v1/w17-5220
84. Shuang, K., Ren, X., Yang, Q., Li, R., & Loo, J. (2019). AELA-DLSTMs: Attention-Enabled and Location-Aware Double LSTMs for aspect-level sentiment classification. *Neurocomputing*, *334*, 25–34. https://doi.org/10.1016/j.neucom.2018.11.084
85. Sun, S., Luo, C., & Chen, J. (2017). A review of natural language processing techniques for opinion mining systems. *Information Fusion*. https://doi.org/10.1016/j.inffus.2016.10.004
86. Tao, J., & Fang, X. (2020). Toward multi-label sentiment analysis: a transfer learning based approach. *Journal of Big Data*, *7*(1), 1–26. https://doi.org/10.1186/s40537-019-0278-0
87. TF-IDF. (2012). In *SpringerReference*. https://doi.org/10.1007/springerreference_179544
88. Thu, P. P., & Aung, T. N. (2018). Implementation of emotional features on satire detection. *International Journal of Networked and Distributed Computing*. https://doi.org/10.2991/ijndc.2018.6.2.3
89. Tran, T. U., Hoang, H. T. T., & Huynh, H. X. (2019). Aspect extraction with bidirectional GRU and CRF. *RIVF 2019 - Proceedings: 2019 IEEE-RIVF International Conference on Computing and Communication Technologies*. https://doi.org/10.1109/RIVF.2019.8713663
90. Varalakshmi, R., & Swetha, R. (2020). Covid-19 lock down: People psychology due to law enforcement. *Asian Journal of Psychiatry*, *51*, 102102. https://doi.org/10.1016/j.ajp.2020.102102
91. Vibha, Prabhu, A. N., Kamath, G. B., & Pai, D. V. (2020). Keeping the country positive during the COVID 19 pandemic: Evidence from India. *Asian Journal of Psychiatry*, *51*, 102118. https://doi.org/10.1016/j.ajp.2020.102118
92. Wallach, H. M. (2006). Topic modeling: Beyond bag-of-words. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/1143844.1143967
93. Wang, X., McCallum, A., & Wei, X. (2007). Topical N-grams: Phrase and topic discovery, with an application to information retrieval. *Proceedings - IEEE International Conference on Data Mining, ICDM*. https://doi.org/10.1109/ICDM.2007.86
94. Wu, C., Wu, F., Wu, S., Yuan, Z., & Huang, Y. (2018). A hybrid unsupervised method for aspect term and opinion target extraction. *Knowledge-Based Systems*. https://doi.org/10.1016/j.knosys.2018.01.019
95. Xiao, Z., Li, X., Wang, L., Yang, Q., Du, J., & Sangaiah, A. K. (2018). Using convolution control block for Chinese sentiment analysis. *Journal of Parallel and Distributed Computing*. https://doi.org/10.1016/j.jpdc.2017.10.018
96. Xie, X., Ge, S., Hu, F., Xie, M., & Jiang, N. (2019). An improved algorithm for sentiment analysis based on maximum entropy. *Soft Computing*, *23*(2), 599–611. https://doi.org/10.1007/s00500-017-2904-0
97. Xu, G., Meng, Y., Qiu, X., Yu, Z., & Wu, X. (2019). Sentiment analysis of comment texts based on BiLSTM. *IEEE Access*. https://doi.org/10.1109/ACCESS.2019.2909919
98. Yadollahi, A., Shahraki, A. G., & Zaiane, O. R. (2017). Current state of text sentiment analysis from opinion to emotion mining. *ACM Computing Surveys*. https://doi.org/10.1145/3057270
99. Yang, C., Zhang, H., Jiang, B., & Li, K. (2019). Aspect-based sentiment analysis with alternating coattention networks. *Information Processing and Management*, *56*(3), 463–478. https://doi.org/10.1016/j.ipm.2018.12.004
100. Yang, M., Zhao, W., Chen, L., Qu, Q., Zhao, Z., & Shen, Y. (2019). Investigating the transferring capability of capsule networks for text classification. *Neural Networks*, *118*, 247–261. https://doi.org/10.1016/j.neunet.2019.06.014
101. Yang, Q., Rao, Y., Xie, H., Wang, J., Wang, F. L., & Chan, W. H. (2019). Segment-Level Joint Topic-Sentiment Model for Online Review Analysis. *IEEE Intelligent Systems*, *34*(1), 43–50. https://doi.org/10.1109/MIS.2019.2899142
102. Yang, Yi, & Eisenstein, J. (2017). Overcoming Language Variation in Sentiment Analysis with Social Attention. *Transactions of the Association for Computational Linguistics*. https://doi.org/10.1162/tacl_a_00062

103. Yang, Yongshi, Peng, F., Wang, R., Guan, K., Jiang, T., Xu, G., Sun, J., & Chang, C. (2020). The deadly coronaviruses: The 2003 SARS pandemic and the 2020 novel coronavirus epidemic in China. In *Journal of Autoimmunity*. https://doi.org/10.1016/j.jaut.2020.102434
104. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*.
105. Yin, Z., Fabbri, D., Rosenbloom, S. T., & Malin, B. (2015). A scalable framework to detect personal health mentions on Twitter. *Journal of Medical Internet Research*. https://doi.org/10.2196/jmir.4305
106. Yoon, S., Elhadad, N., & Bakken, S. (2013). A practical approach for content mining of tweets. *American Journal of Preventive Medicine*. https://doi.org/10.1016/j.amepre.2013.02.025
107. Yu, J., Jiang, J., & Xia, R. (2019). Global inference for aspect and opinion terms co-extraction based on multi-task neural networks. *IEEE/ACM Transactions on Audio Speech and Language Processing*, *27*(1), 168–177. https://doi.org/10.1109/TASLP.2018.2875170
108. Zhang, Y., Jin, R., & Zhou, Z. H. (2010). Understanding bag-of-words model: A statistical framework. *International Journal of Machine Learning and Cybernetics*. https://doi.org/10.1007/s13042-010-0001-0
109. Zhang, Z., Zou, Y., & Gan, C. (2018). Textual sentiment analysis via three different attention convolutional neural networks and cross-modality consistent regression. *Neurocomputing*. https://doi.org/10.1016/j.neucom.2017.09.080
110. Zhao, R., & Mao, K. (2018). Fuzzy Bag-of-Words Model for Document Representation. *IEEE Transactions on Fuzzy Systems*. https://doi.org/10.1109/TFUZZ.2017.2690222
111. Zhao, Z., Liu, T., Li, S., Li, B., & Du, X. (2017). Ngram2vec: Learning improved word representations from ngram co-occurrence statistics. *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 244–253. https://doi.org/10.18653/v1/d17-1023
112. Zhou, J., Lu, Y., Dai, H. N., Wang, H., & Xiao, H. (2019). Sentiment analysis of Chinese microblog based on stacked bidirectional LSTM. *IEEE Access*.
113. Zou, Y., Gui, T., Zhang, Q., & Huang, X. (2018). A Lexicon-Based Supervised Attention Model for Neural Sentiment Analysis. *Coling-2018*. https://doi.org/10.1371/journal.pone.0078994.