

# Automatic customer review summarization using deep learning-based hybrid sentiment analysis

Gagandeep Kaur<sup>1,2</sup>, Amit Sharma<sup>3</sup>

<sup>1</sup>Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune, India

<sup>2</sup>Department of Computer Science and Engineering, Lovely Professional University, Jalandhar, India

<sup>3</sup>School of Computer Applications, Lovely Professional University, Jalandhar, India

## Article Info

### Article history:

Received Feb 27, 2023

Revised Aug 26, 2023

Accepted Nov 12, 2023

### Keywords:

Aspect features

Customer review summarization

Deep learning

Hybrid features

Review features

Sentiment analysis

## ABSTRACT

Customer review summarization (CRS) offers business owners summarized customer feedback. The functionality of CRS mainly depends on the sentiment analysis (SA) model; hence it needs an efficient SA technique. The aim of this study is to construct an SA model employing deep learning for CRS (SADL-CRS) to present summarized data and assist businesses in understanding the behavior of their customers. The SA model employing deep learning (SADL) and CRS phases make up the proposed automatic SADL-CRS model. The SADL consists of review preprocessing, feature extraction, and sentiment classification. The preprocessing stage removes irrelevant text from the reviews using natural language processing (NLP) methods. The proposed hybrid approach combines review-related features and aspect-related features to efficiently extract the features and create a unique hybrid feature vector (HF) for each review. The classification of input reviews is performed using a deep learning (DL) classifier long short-term memory (LSTM). The CRS phase performs the automatic summarization employing the outcome of SADL. The experimental evaluation of the proposed model is done using diverse research data sets. The SADL-CRS model attains the average recall, precision, and F1-score of 95.53%, 95.76%, and 95.06%, respectively. The review summarization efficiency of the suggested model is improved by 6.12% compared to underlying CRS methods.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Corresponding Author:

Gagandeep Kaur

Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University)

Pune, India

Email: gagandeep.kaur@sitnagpur.siu.edu.in

## 1. INTRODUCTION

The advent of the internet of things (IoT) [1], Web 2.0 standards [2], and coronavirus disease 2019 (COVID-19) have resulted in a significant increase in the online shopping of food, electronic items, and the subsequent posting of reviews. This exponential surge of online reviews may assist in making educated choices regarding a service, brand, product [3]. The customer review summarization (CRS) tool is increasingly being used by business owners to improve their products and services. It incorporates the examination of the huge number of online reviews posted by customers to gain insight into their contentment and requirements.

The analysis of the emotions expressed in text for a specific entity or subject is called sentiment analysis (SA) [4]. SA can be categorized as: word level, phrase level, sentence, and document level. SA at the word level includes the determination of the individual's perception of the products, services, or their

aspects [5]. In phrase-level SA, multiple words are analyzed to determine their sentiment. The SA at the sentence level includes determining the sentence's overall sentiment [6]. Finally, SA at the document level uses average approaches to calculate the overall sentiment of a sentence [7].

Sentiment analysis is performed using methods based on machine learning (ML) or deep learning (DL), lexicons, and hybrid techniques [8]. In lexicon-based methodology, a dictionary of words labeled by sentiments is used to determine a given sentence's overall opinion [9]. The combination of sentiment ratings and additional rules ensures fewer instances of sarcasm, dependent clauses, and negations appear in sentences. Natural language processing (NLP) techniques such as lexicons, stemming, tokenization, and part-of-speech (PoS) tagging are included within the rules [10]. The lexicon-based SA systems are considered modest because they do not consider the ensuing integration of words. The combination of advanced processing techniques and the newest rules can be used to set up new expressions and vocabulary. When new rules are added, however, existing findings can be altered, thereby complicating the entire process. A lexicon-based system requires constant tweaking and maintenance, which makes implementation more difficult [9].

The dataset is divided into testing and training datasets for ML/DL-based techniques [8]. To learn the documents, the model must be trained to associate input text with conforming yields using training data sets. As part of the prediction process, the testing dataset is used to transform hidden textual input into a feature vector. These vectors are provided as input to the model, which generates prediction tags for the respective vectors. In a hybrid approach to SA, lexicon-based approach is combined with ML techniques.

Both ML and lexicon-based strategies are successful in conventional text sources, formal language, and well-stated domains when pre-labeled data is obtainable for training, or the lexicon coverage comprises those words that express certain emotions within a corpus [7], [8], [10], [11]. These technologies, however, cannot capture the volume, pace, and diversity of unstructured and informal data that is constantly being uploaded to the internet. The performance of ML-based SA approaches has recently been enhanced by integrating several types of feature extraction algorithms. However, this critical aspect of feature extraction encounters several obstacles, including ambiguous and unreliable features for accurate categorization [12], [13]. An appropriate hybrid model for extracting features is needed to overcome such obstacles. In addition to the features specific to a review, its aspects, and emoticons, should also be considered for clear and precise extraction of features.

The review "although this cell phone is too hefty, it is a bit inexpensive", includes implicit aspects: weight (indicated by the word "hefty"), price (indicated by the word "inexpensive"), and sentiment-bearing word relations. Although the overall feelings seem neutral, aspect-based sentiments exhibit both negative and positive polarities. Furthermore, the implicit qualities of the features derived from real-world data are poorly defined and are not articulated as general synonyms or conventional forms. One approach to this problem is to combine highly similar features to create attributes and then use attribute-based sentiment analysis (ABSA) [14]–[16]. Several ABSA algorithms have been described recently, but CRS requires a more effective mechanism that can successfully obtain implicit word relations, find related aspects, and cope with unusual words and ambiguities with automatic classification. The use of a DL classifier can further optimize the SA functionality [17], [18]. As compared to ML classifiers, the DL classifiers improve review classification performance, which improves the efficiency of the review summarization phase further.

The purpose of summarizing is to convey the key ideas from the text in a condensed form while removing redundant data and maintaining the original text's meaning. As social media has become a hub of abundant information, it is becoming increasingly important to analyze this text to find information and utilize it to the advantage of various applications and individuals. There are two categories of summarization: extraction and abstraction. The extractive summarization strategy involves concatenating extracts from a corpus into a summary. When using an abstraction technique, sentences are combined to create something new that is not present in the source and are replaced in the summary with the new concept.

In this research, we propose a novel framework called SA model employing deep learning for CRS (SADL-CRS) based on the effective approach of SA and review summarization (RS). The SADL phase is built using the hybrid technique for extracting features from pre-processed input reviews and an effective DL classifier-long short-term memory (LSTM). The hybrid feature extraction approach aims to eliminate the challenges of unclear and unreliable features for sentiment classification by combining review related features (RRF) and aspect related features (ARF). Furthermore, the sequential DL classifier LSTM has been trained with hybrid features and the appropriate amount of hyperparameters to boost classification accuracy. Finally, in the RS phase, the summary text is obtained using the results of the SADL phase, pre-processed reviews, and ARF features vector. The CRS algorithm is designed to produce the summarized text for the classified reviews in this paper. Section 2 of this paper covers the study of the related work. Section 3 presents the design and methodology of the proposed work. Section 4 presents the experimental results and analysis. The conclusion and suggestions for future work are described in section 5.

## 2. RELATED WORK

This section reviews the SA and RS methods using various approaches. Such methods are reviewed according to the methodology used i.e., machine learning, deep learning, and rule-based approaches. The research gaps and contributions are also discussed at the end of this section.

### 2.1. Sentiment analysis techniques

The SentiDiff algorithm suggested by Wang *et al.* [19], initially employed the sentiment reversal approach to examine sentiment spread and subsequently uncovered intriguing traits within the Twitter dataset. A correlation between sentiment diffusion patterns and the recorded data of a Twitter tweet worked well in predicting sentiment polarity. Hao *et al.* [20] introduced the stochastic word embedding approach called CrossWord for cross-domain sentiment encoding. By mapping occurrence info to word polarity, the researchers performed encryption with minimal computing effort for accurate analysis. For word embedding in SA, the SentiVec method developed by Zhu *et al.* [21] utilized the kernel optimization procedure. The first level incorporated supervised learning models, while the second level integrated unsupervised learning models such as context-to-object-word reward models and object-word-to-surrounding-word reward models.

Naresh and Krishna [22] presented a three-step sequential minimal optimization-based ML method. In the first step, data collection and preprocessing are completed, followed by optimization being done by obtaining relevant features in the second step, and finally, in the third step, the revised training set is classified into multiple classes using various ML algorithms. Singh *et al.* [23] categorized public opinion about coronavirus using the bidirectional encoder representations from transformers (BERT) model. The authors used SA on two datasets: one containing tweets from users around the world, and the other containing tweets from users in India. Munuswamy *et al.* [24] proposed a sentiment dictionary-based methodology for user SA. The automatic recommendation for product purchase is given to end users in accordance with the sentiment forecast. Feature extraction using n-grams and prediction using support vector machines (SVM) were employed by the authors.

Ayyub *et al.* [25] examined a range of feature sets and classifiers to quantify sentiments. They conducted an experimental performance assessment of conventional ML-based approaches, ensemble-based techniques, and state-of-the-art DL techniques based on the feature set. The results demonstrated that DL techniques outperformed conventional ML algorithms. Oyeboode *et al.* [26] examined the sentiment classification of 88,125 user reviews from various mental health apps available on Google Play and App Store employing five supervised ML algorithms. Employing the most accurate classifier, the authors discovered themes that represent a variety of factors affecting the achievement of mental health. Iqbal *et al.* [27] proposed a genetic algorithm (GA) based feature reduction strategy for efficient SA. The proposed unified framework bridges the gap between ML and lexicon-based approaches to boost accuracy and scalability.

### 2.2. Aspect based sentiment analysis techniques

The above SA methods failed to address the challenges of sarcasm, feelings, emotions, and opinion-related features. Researchers have been focusing on aspect terms extraction for feature formation because it remarkably enhances SA accuracy. ABSA methods [28]–[37] were introduced to overcome such challenges to some extent.

Schouten *et al.* [28] proposed two methods: unsupervised and supervised for discovering the aspects. The unsupervised technique utilizes co-occurrence frequency data gathered from a corpus through association rule mining to extract aspect categories. The proposed unsupervised method performs better than several straightforward baselines, with an F1-measure of 67%. The supervised variation performs even better than existing baseline methods, with an F1-measure of 84%. Alqaryouti *et al.* [29] developed a hybrid ABSA approach that integrates rules and domain lexicons to evaluate entities in smart app reviews. This method employs lexicons, rules, and language processing techniques to overcome multiple sentiment analysis challenges and generate result summaries. According to the results, aspect extraction accuracy dramatically increases when implicit aspects are considered.

Nandal *et al.* [30] addressed one of the main issues with bipolar words in SA to enhance aspect-based sentiment analysis. Their study explores the impact of context on word polarity and how it affects overall product ratings and specific attributes, yielding impressive results. Prathi *et al.* [31] devised a dynamic aspect extraction approach based on automated sentiment assessment from input reviews to tackle the cold-start problem in ABSA approaches. Shams *et al.* [32] proposed an unsupervised learning approach called language independent aspect-based SA (LISA) to address the challenges of time and cost complexity. The approach consists of three coarse-grained steps that are further divided into several fine-grained processes. The initial polarity lexicon and aspect word sets serve as representations of aspects to extract domain knowledge from the dataset in the first stage. The plausibility of a word is then computed based on its aspect and sentiment, followed by establishing the polarity of each aspect in the third step.

Bie and Yang [33] introduced a unique multitask multiview network (MTMVN) model that explores end-to-end ABSA. The unified ABSA is the primary task, followed by two sub-tasks: aspect term mining and predicting aspect opinions. A multitasking strategy combines opinion polarity information and aspect boundary information to enhance task performance. Shim *et al.* [34] presented a label-efficient training system (LETS) to expedite development by eliminating the need for manual labeling tasks. The authors applied LETS to a novel use-case of ABSA, examining reviews of a health-related program aimed at improving sleep quality.

An augmented knowledge graph network (KGAN) proposed by Zhong *et al.* [35] aims to efficiently integrate external knowledge with explicitly syntactic and contextual information. KGAN captures sentiment features from a variety of perspectives, including context, syntax, and knowledge-based perspectives. To fully extract semantic features, KGAN learns both contextual and syntactic representations simultaneously. Following that, KGAN combines knowledge graphs into embedding spaces, which are then analyzed via an attention mechanism to identify aspect-specific knowledge representations. The final feature is a hierarchical fusion module that complements these multi-view representations on a local-to-global level.

### 2.3. Deep learning methods

To further increase efficiency, recent DL innovations [36]–[44] have also been applied to the SA domain. Kumar *et al.* [36] proposed an ABSA-based technique that uses three methods: creating ontologies for semantic feature extraction, using Word2vec to transform processed corpora, and developing convolutional neural networks (CNNs) for opinion mining. Particle swarm optimization (PSO) is utilized to tune CNN parameters to obtain non-dominant Pareto front optimum values. Li *et al.* [37] suggested a novel semi-supervised multi-task learning framework (SEML) to implement ABSA on user reviews. Both aspect mining and aspect sentiment classification are learned together in a joint session. The proposed approach uses cross-view training (CVT) to train auxiliary prediction modules on unlabeled reviews, which enhances representation learning.

Alamanda *et al.* [38] proposed sentiment extraction and polarity categorization from input reviews to enhance efficient ABSA. Polarity features were automatically extracted based on client preferences using both DL and ML algorithms. Lu *et al.* [39] introduced an aspect-gated graph convolutional network (AGGCN) that incorporates a specific aspect gate for encoding aspect-specific information. They utilize a graph convolution network based on sentence dependency trees to fully leverage sentiment dependencies. Datta and Chakrabarti [40] employed an enhanced DL algorithm for ABSA in the context of demonetization tweets. The retrieved aspect words are transformed into features with the aid of Word2vec and polarity measure computation. Sentiment classification is then conducted using a recurrent neural network (RNN) on the combined features.

Londhe *et al.* [41] presented a unique approach for ABSA using the DL classifier LSTM-RNN. The hybrid LSTM-RNN demonstrates high accuracy in predicting aspect polarity. Shanmugavadivel *et al.* [42] performed both sentiment analysis and identification of offensive language in low-resource code-mixed data, encompassing Tamil and English. It leverages machine learning, deep learning, and pre-trained models like BERT, robustly optimized BERT pre-training approach (RoBERTa), and adapter-BERT. The dataset employed for this research is derived from the shared task on multi-task learning at DravidianLangTech@ACL2022. Another focal point of this work involved addressing the challenge of extracting semantically meaningful information from code-mixed data through the application of word embedding techniques. Kaur *et al.* [43] investigated the impact of coronavirus on individuals' mental well-being using hashtag keywords like coronavirus, COVID-19, deaths, new cases, and recovered cases. They employed RNN and SVM to categorize sentiment scores as positive, negative, or neutral.

Balakrishnan *et al.* [44] compared several deep learning models, such as CNNs, RNNs, and Bi-directional LSTMs, using different word embedding techniques, including BERT and its variants, FastText, and Word2Vec. There were two steps in evaluating each model, namely a five-class evaluation and a three-class evaluation. The most accurate predictions were produced by models based on Word2Vec and NN. The authors found that DL detects text sentiment more accurately than supervised machine learning. Yu and Zhang [45] proposed a multiweight graph convolutional network (MWGCN) that aims to create a local context-weighted adjacency graph using two weighting methods, multigrain dot-product weighting (MGDW) and the local context graph (LCG). By emphasizing aspect-related features of the context, MGDW preserves the overall context semantics. Additionally, LCG's adjacency graph emphasizes local context words and reduces aspects' long-distance dependence. Contextual features are also extracted by using a multilayer graph convolutional network (GCN) that combines syntactic and aspect information.

The above methods [19]–[45] do not address RS except [31], which summarizes patients' reviews using ABSA. The automatic CRS has not yet been explored based on the SA outcome. A few recent attempts [46]–[49] were made for the CRS. Shuming *et al.* [46] suggested a model for collaborative learning of sentiment classification and text summarization, in which the sentiment classification label is viewed as an additional "summarization" of the text summarization output. Liu and Wan [47] investigated four models leveraging product information to aid in review summarization. In the first three models: *AttrEnc*, *AttrDec*, and

*AttrEncDec*, attribute information is directly injected into the pointer generation network. The last model: *MemAttr* combines text information and attribute information with a memory network for the generation of summary.

In the text summarization approach suggested by Marzijarani and Sajedi [48], sentences are first parsed, and their similarities are determined using the proposed similarity metric. Following the use of the gaussian mixture model (GMM) algorithm to cluster the sentences based on their similarity, a predetermined number of sentences are finally chosen from each cluster. The task of generating a summary in the form of terms from news articles and consumer reviews is undertaken by Sheela and Janet [49] by employing RNN-LSTM model in conjunction with recall vocabulary again (RVA) and copy procedure. Apart from ML-based methods [50]–[53] have recently been proposed for religious extremism detection on online user content, heart disease detection, and COVID-19 detection.

#### 2.4. Research gaps

In the above section, we have reviewed the SA methods [19]–[45] under different categories like review-specific SA [19]–[27], ABSA [28]–[35], and deep learning-based SA [36]–[45]. After that, we reviewed recent CRS methods [48]–[51]. The below-mentioned research gaps identified from the existing work motivate us to propose a novel model in this paper:

- The recently presented CRS methods [31], [46]–[49] have not fully explored the SA methods which limit their scalability and efficiency of the summarization. The underlying CRS methods utilized basic approaches for features extraction, clustering, deep learning-based SA, hierarchical model, and did not consider the problems related to sarcasm, emoticons, ambiguous aspects.
- SA approaches specified in [19]–[27] are insufficient to handle the issues associated with the accurate portrayal of emotions, opinions, and sarcasm.
- ABSA methods provided in [28]–[35] addressed the challenge of sarcasm and ambiguity to some extent but failed to address the challenges of opinions, negations, and emotions for accurate classification.
- Some ABSA methods have not been tested on large review datasets [28], [31] and some relied primarily on unsupervised procedures [28], [32], requiring manual-annotated data. Due to some SA/ABSA [19], [21], [24]–[27], [34], [35] algorithms' reliance on symbolic feature extraction, their accuracy is limited.
- DL-based SA methods [36]–[45] demonstrated the impact of using DL methods for feature extraction or classifications. However, utilizing DL models for feature extraction leads to significant computational overhead. The CRS cannot benefit solely from the DL feature extraction results.

### 3. PROPOSED WORK

As shown in Figure 1, the suggested SADL-CRS model consists of four phases: pre-processing, hybrid feature engineering, DL classification, and summarization. The raw input reviews are first pre-processed with the help of NLP techniques. The pre-processing algorithm performs operations like stop word removal, stemming, uniform resource locator (URL) removal. The hybrid engineering phase performs the encoding of the preprocessed reviews into a unique numerical feature vector using aspect-related features and review-related features. The DL classification phase employs the sequential LSTM classifier for classifying the input review into negative, positive, or neutral classes. Finally, the classified sentiments, along with ARF and pre-processed reviews are supplied as input to the summarization phase.

#### 3.1. Data pre-processing

This initial step of SA cleans up the raw reviews by eliminating and fixing the complicated and unwanted content. In the proposed work, data pre-processing functionality begins with tokenization and concludes with the removal of meaningless words, digits, and words with fewer than three characters. As a part of tokenization, the input review sentences are divided into tokens. Then, using stemming, we reduced all the tokens to their singular forms (e.g., confirming or confirmed gets reduced to confirm). The various stop-words like 'I', 'an', 'a'. are removed to minimize the count of tokens further. Special letters (@, #, and so on), dates, trivial words (o+, A-), and any URLs are identified and eliminated. Furthermore, terms with fewer than three characters are also detected and eliminated. This phase of the proposed work ensures an effective decrease in the raw reviews' dimensional space. Table 1 contains samples of some test reviews that show the outcome of the preprocessing algorithm.

#### 3.2. Hybrid features engineering

A number of approaches have been proposed for representing input reviews as numerical features for training review mining (RM) systems. However, robust, accurate, and effective feature extraction remains a challenging research area for RM. To improve classification accuracy, it is important to develop efficient

and steadfast features in a sentiment analysis system. In this study, we use a hybrid method of feature engineering to address the issue of effective and resilient SA. The RRF feature extraction is performed first by using different techniques to attain the polarity of every word in the preprocessed text, including emoticons, and negations. The ARF approach is then used for extracting the aspect words and their polarity. ARF represents and addresses reviews containing sarcasm and ambiguity as well. Eventually, the combined results of the ARF and RRF are expressed as an HF vector-a hybrid feature vector.

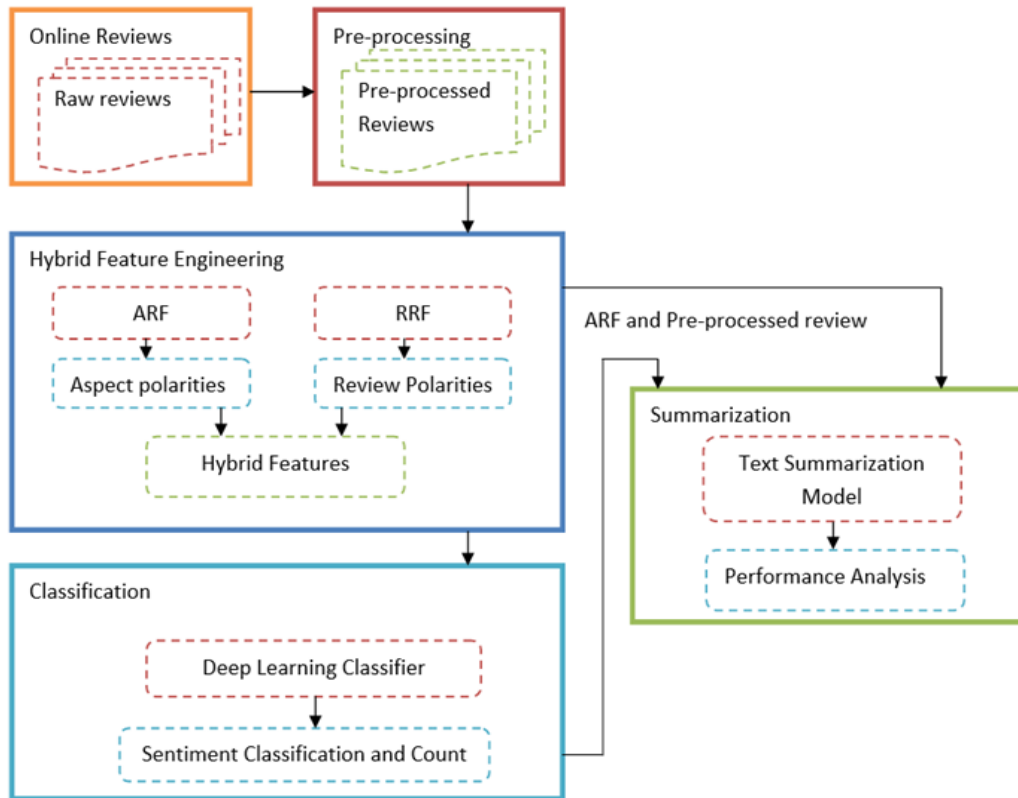


Figure 1. Architecture of the proposed SA and RS model

Table 1. Sample reviews before and after employing pre-processing algorithm

Before pre-processing	After pre-processing
“Food is always fresh and hot- ready to eat!”	“Food fresh hot ready eat”
“I was very disappointed with this restaurant”	“Disappoint restaurant”
“I had to ask cart attendant three times before she finally came back with the dish lotus leaf wrapped rice that I’ve requested.”	“Ask cart attendant three time come back lotus leaf wrap rice request”
“This is such a great deal! Already thinking about my second trip”	“Think second trip”

**3.2.1. RRF**

RRF is a feature representation method that includes emoticons along with the text to represent feelings, opinions, and negations in the input. To build RRF, conventional features like n-gram, term frequency-inverse document frequency (TF-IDF) and emoticon-specific polarity are extracted. TF-IDF is a technique that uses bag-of-words (BoWs) and relies on word embeddings. Using solitary words for feature extraction limits SA in several ways. Single-word features do not address negation issues, resulting in the misclassification of sentiments.

To address such issues, we first generated a word list by extracting n-gram features from the pre-processed reviews and then applied TF-IDF to the n-gram output. In addition to minimizing the dimensional space, the combination of n-gram and TF-IDF techniques also effectively depicts all the reviews. Further improvements to sentiment analysis accuracy are then achieved by retrieving emoticons-specific features. Hence, RRF is achieved through the combination of n-grams, TF-IDF, and emoticon-specific features. The RRF procedure is described in detail below and is outlined in algorithm 1.

**Algorithm 1. RRF extraction**

Inputs  
P: pre-processed training set  
n: number of grams  
Output  
NTE: extracted review related features set

1. Initialize  $Ngram, TFIDF, NT, NTE \leftarrow \emptyset$
2. For  $i = 1$  to  $length(P)$
3.     Compute N-gram features
4.      $Ngram(i) \leftarrow getNgram(P(i), n)$
5. End For
6. For  $i = 1$  to  $length(P)$
7.     Compute TF-IDF features
8.      $TFIDF(i) \leftarrow TF(Ngram(i)) + IDF(Ngram)$
9.     Compute Emoticons Related Features
10.     Initialize  $EF \leftarrow zeros(1, 2)$
11.      $E \leftarrow getEmoticons(P(i))$
12.     If ( $E \neq Null$ )
13.         For  $j = 1: length(E)$
14.             If ( $EF(j) == positive$ )
15.                  $EF(j, :) + 1$
16.             Else
17.                  $EF(:, j) - 1$
18.             End If
19.         End For
20.     End If
21.      $NTE(i) \leftarrow \{TFIDF(i), EF(i)\}$
22. End For
23. Stop

Step 1: Compute N-gram-Let P be the pre-processed document of online reviews and  $P(i)$  be the document that contains the  $i^{th}$  pre-processed review. We begin with the application of n-gram approach on pre-processed text for combined n-gram and TF-IDF. In n-grams,  $n$  words are grouped in a proximal order based on the given dataset. N-grams are referred to as unigram, bigram, trigram, and so on when the value of  $n$  is 1, 2, 3, respectively. For instance, "beautiful" and "very beautiful" are unigrams and bigrams, respectively.

$$Ngram = getNgram(P(i), n) \quad (1)$$

where  $Ngram$  is the collection of n-grams generated from the preprocessed input document. The parameters  $P(i)$  and  $n$  are passed to the method  $getNgram(.)$ . In this technique, we set  $n$  to 2 to strike a balance between efficiency and reliability when dealing with negations.

Step 2: Compute TF-IDF- We determined TF-IDF based on n-gram results resulting from the training/testing dataset for each word list. In the TF, the number of times a word appears in reviews is calculated, while in the IDF, the number of appearances of a word in reviews is divided by the absolute count of reviews.

$$TF - IDF = TF(Ngram(i)) \times IDF(Ngram) \quad (2)$$

where  $Ngram(i)$  denotes the word list for  $i^{th}$  review and  $Ngram$  denotes the word list for the entire dataset P. A vector NT containing all the features from the entire document is then created.

We also extracted the emoticons' specific features from the reviews as shown in algorithm I in steps 9-21. Because each review may or may not have emoticons, we set the emoticon feature vector (EF) of size  $1 \times 2$  to zero for each review. The emoticon-specific features are subsequently integrated with features as (3).

$$NTE(i) = [NT, EF] \quad (3)$$

**3.2.2. ARF**

After the extraction of RRF from each review, the ARF extraction procedure is employed to train and test datasets to improve the SA performance further. The goal is to count: lemmas' co-occurrences with sentence annotated categories, lemmas' co-occurrences with aspect types, and grammatical dependencies' co-occurrences with aspect types. The weight matrix from all the preprocessed reviews in the input data set is transformed into aspect features. As opposed to [28], this study does not use category estimation and instead extracts the aspect terms as well as their co-occurrence frequencies for each review. When compared to the work proposed in [28], it reduces the computation time of employing the supervised classification algorithm.

The suggested ARF mechanism with illustrations is presented in our recent publication [54] and is outlined in Algorithm 2 as well.

#### Algorithm 2. ARF extraction

```

Input
Q: Training dataset
Output
ARF: Set of aspect related features for each review
1. Initialize  $C, X, Y, W \leftarrow \emptyset$ 
2. For each review  $i = 1$  to length(Q)
3.    $[s, s_c] \leftarrow getLemmasDependencies(Q(i))$ 
4. For each set  $k = 1$  to length(s)
5.   For each term  $j = 1$  to length(s(k))
6.     Count lemma/dependency occurrence j
7.     If (s(k,j)  $\neq$  Y)
8.        $Y \leftarrow add(s(k,j))$ 
9.        $Y_j \leftarrow Y_j + 1$ 
10.    End
11. For each category  $c = 1$  to length(sc)
12.   Find and add unique categories c in C
13.   If (sc(c)  $\neq$  C)
14.      $C \leftarrow add(s_c(c))$ 
15.   End
16.   Count co-occurrence (sc(c), s(k,j)) in X
17.   If (sc(c), s(k,j))  $\neq$  X)
18.      $X \leftarrow add(s_c(c), s(k,j))$ 
19.      $X_{c,j} \leftarrow X_{c,j} + 1$ 
20.   End
21. End
22. End
23. End
24. Calculate weight matrix for aspect features
25. For each co-occurrence pair  $x = 1$  to length(X)
26.    $Y_j \leftarrow X(x, :)$ 
27.   If ( $Y_j > 0$ )
28.      $W_{x,j} \leftarrow (X_{x,j}/Y_j)$ 
29.      $A_i \leftarrow \max(W_{x,j})$ 
30.   End
31. End

```

Let  $Q$  be the training set, consisting of  $m$  raw online reviews. For each input review, we extracted categories and estimated their co-occurrence rates against the dependency forms and lemmas. The initial step of the proposed ARF algorithm includes the determination of lemmas, dependency forms, and categories for each review. The list of dependency forms and lemma is stored in the set  $s$ . For input review, the  $S_C$  contains a list of aspect categories. The Stanford CoreNLP framework [55] has been used to implement NLP processes such as dependency parsing, PoS tagging, and lemmatization on each review. Then, counting and addition of each unique occurrence of lemma or dependency form is performed in vector  $Y$ . A vector  $C$  is constructed by adding all input review aspect categories. Following the detection of the lemma/dependency form and distinctive categories, the co-occurrence frequency is recorded in vector  $X$ .

Additionally, vector  $Y$  is created to store the occurrence frequencies for all the dependency forms and lemmas of the analogous review sentences. Using vector  $X$  and  $Y$  for co-occurrence and occurrence frequency values, the weight matrix  $W$  is calculated for each pair of vector  $X$ . The weight vector ( $W_{x,j} \leftarrow (X_{x,j}/Y_j)$ ) is only calculated for each pair in  $X$  if the associated co-occurrence frequency goes beyond zero. It solves the issue of finding the best threshold for any dataset. As a final step, we take the largest co-occurrence value for each pair of  $W$  into vector  $A$  for estimating aspect-specific features.

Using this method, aspect-related information can be extracted without utilizing ML techniques requiring high-processing computations. Algorithm 2 is illustrated in [54] with a sample example. The hybrid feature vector is then constructed using the RRF and ARF vectors without losing generality. For each input review, the hybrid feature (HF) vector is formed by the concatenation function as (4).

$$HF(i) = [RRF(i), ARF(i)] \quad (4)$$

### 3.3. LSTM classifier

Based on the extracted HF vector, we classified the input review sentences into negative, positive, or neutral classes using the sequential DL classifier LSTM. It has already been demonstrated in section 2 that



DL classifiers are more efficient than ML classifiers. As a result, this paper uses LSTM to automate the SA process. The main concern with using conventional classifiers is that they are susceptible to vanishing gradients or exploding gradients. The vanishing gradient problem prevents the learning of long data sequence, causes the weights to oscillate, further deteriorating the quality of the network. Therefore, neural networks like CNNs/RNNs struggle to store information over extended periods due to a crumbling error backflow. On the other hand, LSTM is a kind of RNN that is capable of learning order dependence in sequence prediction problems. With LSTM, the problem of vanishing gradients is overcome by using a unique additive gradient structure that allows direct access to each forget gate's activations at each time step of the learning process. In this way, it defeats the error backflow problem with the minimum computational complexity of  $O(1)$ .

As part of the automatic classification process, the HF vector is fed into a sequential LSTM classifier. Suppose that the LSTM input layer receives the HF vector of a given input review at the current time interval  $t$ . The LSTM network consists of an input gate  $i$ , output gate  $o$ , forget gate  $f$ , and a memory cell  $c$  [56]. For every instance of time, LSTM computes its gate's activations  $\{i_t, f_t\}$ , updates its memory cell from  $c_{t-1}$  to  $c_t$ , computes the output gate activation  $o_t$ , and finally outputs a hidden representation  $h_t$ . The hidden representation from the previous time step is  $h_{t-1}$ . For updating functions, (5) and (9) are used in LSTM.

$$i_t = \sigma(W_i HF + U_i h_{t-1} + V_i c_{t-1} + b_i) \quad (5)$$

$$f_t = \sigma(W_f HF + U_f h_{t-1} + V_f c_{t-1} + b_f) \quad (6)$$

$$c_t = f_t \theta c_{t-1} + i_t \theta \tanh(W_c HF + U_c h_{t-1} + U_c h_{t-1}) \quad (7)$$

$$o_t = \sigma(W_o HF + U_o h_{t-1} + V_o c_{t-1} + b_o) \quad (8)$$

$$h_t = o_t \theta \tanh(c_t) \quad (9)$$

where  $\theta$  is an element-wise product of the output of the fully connected layers,  $\sigma$  is the logistic function, and  $\tanh$  activation function is applied element-wise to keep the value of new information between -1 and 1. The weight matrices ( $W_*$ ,  $V_*$ ,  $U_*$ ), and biases ( $b_*$ ) are the diagonal weight parameters for each gate such as input, output, forgot, and memory cell. The input and forget gates work together to refresh the memory cell. The forget gate examines the memory section to be forgotten, while the input gate estimates new values based on the view currently written in the memory cell. The hidden description is estimated by the output gate and the memory cell. Because LSTM cell activation includes summing over time and derivatives distributed across sums, the gradient in LSTM gets spread over a long time before vanishing. The fully connected layer, followed by the softmax layer, classifies the input features based on the training dataset into appropriate matching classes. The final categorization results are generated by the output layer. The list of hyperparameters that we used for the design of the LSTM classifier is shown in Table 2.

Table 2. List of LSTM hyperparameters

Parameter	Value
Number of hidden layers	5
Activation function	tanh
Batch size	27
Number of epochs	70
Learning rate	0.1
Number of classes	3
Gradient threshold	1

### 3.4. RS model

The last phase of the proposed model involves the summarization of the input customer reviews according to SA classification outcome. The CRS model represents the customer's perception of the input reviews concerning the products/services in either a positive, negative, or neutral manner. As shown in Figure 2 and algorithm 3, the proposed novel lightweight CRS model takes pre-processed reviews, SA classification outcomes, and the ARF vector as input. The pre-processed review is further processed to get the tokens list in  $T1$ . The classification outcome is recorded into the SA variable which either contains the word 'positive', 'negative', or 'neutral'. Finally, in the  $T2$  variable, we encoded the list of aspect terms extracted in Algorithm 2. All three outcomes  $T1$ ,  $T2$ , and  $SA$  are further concatenated to build the vector  $V$  of

all relevant tokens for the input review. On vector  $V$ , we applied the function  $extractSummary(.)$  to get the initial summary which is further optimized by heading the title of SA. The  $concatenate(.)$  function aims to produce the fused list of all tokens in  $V$ . The second function  $strcat(.)$  aims to produce the SA outcome as a heading followed by the summary for each input review. Table 3 shows some examples of the proposed CRS model.

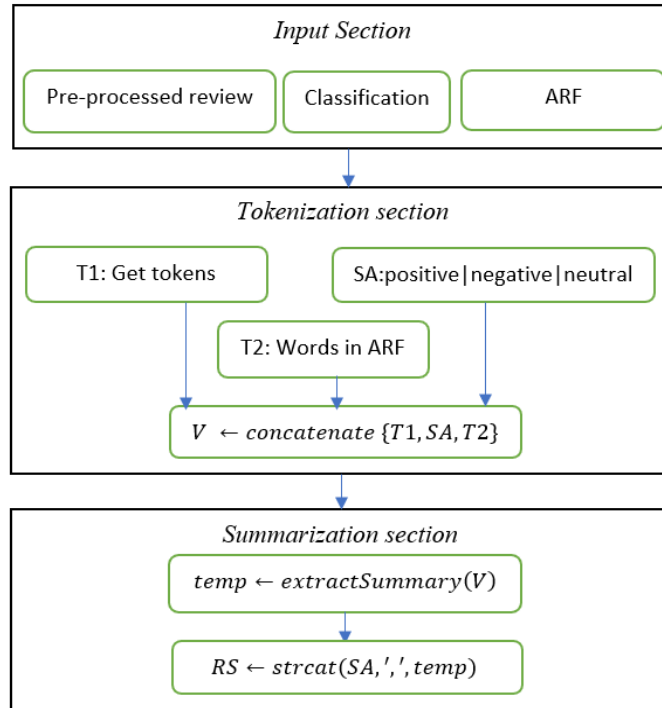


Figure 2. Architecture of the proposed CRS model

Algorithm 3. Proposed CRS

Inputs  
 $p \in P$ : pre – processed input review  
 $SA$ : Outcome of sentiment analysis  
 $T2$ : aspects in ARF  
 Output  
 $RS$ : Summarized review  
 1. Acquire inputs  $p, SA, \& T2$   
 2. Build topic modelling  
 3.  $T1 \leftarrow getTokens(p)$   
 4.  $c1 \leftarrow concatenate(T1, T2)$   
 5.  $V \leftarrow concatenate(c1, SA)$   
 6. End topic building  
 7. Summarization  
 8.  $temp \leftarrow extractSummary(V)$   
 9.  $RS \leftarrow strcat(SA, ', ', temp)$   
 10. End of Summarization  
 11. Return (RS)

Table 3. Sample examples of outcome of RS model

Raw reviews	SA outcome	RS outcome
“Food is always fresh and hot- ready to eat!”	positive	Positive “Food fresh hot”
“@VirginAmerica I didn’t today... Must mean I need to take another trip!”	neutral	Neutral “Need another trip”
“I was very disappointed with this restaurant”	negative	Negative “Disappoint restaurant”
“This is such a great deal! Already thinking about my second trip”	positive	Positive “Think second trip”

## 4. EXPERIMENTAL RESULTS

We used the MATLAB tool, Windows 11 operating system with an I5 processor and 8 GB RAM for the experimental study of the proposed model. We employed three publicly available research datasets: SemEval-2014 [57], Sentiment140 [58], and STS-Gold [59]. The data is used to implement and test the performance of the proposed model.

### 4.1. Datasets

SemEval-2014: Semantic evaluation (SemEval) is an ongoing series of assessments of computer semantic analysis systems coordinated by SIGLEX, the Association for Computational Linguistics Special Interest Group on the Lexicon. The SemEval-2014 includes 3,000 training reviews and 800 test reviews. This data set contains one or more annotated aspect terms for each review.

Sentiment140: Sentiment140 is a Twitter sentiment analysis tool that finds out the twitter sentiment of a brand or product. This dataset is comprised of 1.6 million annotated tweets. Within this dataset, sentiment labels are assigned, with 0 denoting a negative tweet, 2 representing a neutral tweet, and 4 indicating a positive tweet.

STS-Gold: The STS-Gold data set contains 2,026 tweets with their IDs and polarities. The datasets Sentiment140 and STS-Gold are divided into 70 percent training and 30 percent testing datasets. This division ensures a balanced distribution of data for effective model training and evaluation.

### 4.2. Performance parameters

First, we compared the proposed DL-based model with the ML-based approaches for SA to demonstrate its efficiency over ML. Then, the proposed DL-based SA approach is compared with existing methods. Finally, the proposed RS model's performance is compared with existing methods. The SA methods are evaluated using well-known parameters such as precision, recall, F1-score, accuracy and average SA time (ASAT). The CRS methods are evaluated using commonly used performance metrics called recall-oriented understudy for gisting evaluation (ROUGE). We measured the three ROUGE metrics as ROUGE-1, ROUGE-2, and ROUGE-L as per their definitions provided in [46], [60]. Equations (10)-(12) presents the formulas for computing F1-score, precision, and recall parameters.

$$F1 - score = \frac{2 \times P \times R}{P + R} \quad (10)$$

where  $P$  stands for precision and  $R$  stands for recall which are computed as (11) and (12):

$$P = \frac{TP}{FP + TP} \quad (11)$$

$$R = \frac{TP}{FN + TP} \quad (12)$$

where, FP represents false positive, TP represents true positive, and FN represents false negative of sentiment classification. The computational time, i.e., the average processing time for sentiment categorization is related to the parameter ASAT. To estimate the ASAT parameter, 50 instances of each method were executed for the classification of SA outcome.

### 4.3. SA analysis using ML and DL-based classifiers

Figure 3 demonstrates the outcomes for F1-score, precision, recall, accuracy, and ASAT parameters for different classifiers based on different datasets. We have implemented the proposed model using machine learning (ML) classifiers, support vector machine (SVM), random forest (RF), naïve Bayes (NB), and deep learning (DL) classifier long short-term memory (LSTM). Figure 3 shows that the proposed model SADL using LSTM performs better than the ML classifiers in terms of F1-score, precision, recall, and accuracy. The classifiers: SVM, RF, NB, and LSTM have been applied to the hybrid feature extraction outcomes. It can be observed that the ML classifiers' overall SA efficiency ranged from 0.85 to 0.94, however, the SADL classifiers' overall SA efficiency ranged from 0.94 to 0.96 for F1-score, precision, recall, and accuracy parameters. The LSTM network generates predictions based on how sequence data changes over time-based on sequence data input. The LSTM efficiently overcomes conventional classifier problems associated with vanishing gradients and misclassifications. It also includes a wide range of parameters such as input biases, output biases, and learning rates due to which no fine modifications are required. LSTM is advantageous over ML classifiers since the complexity of updating each weight gets lowered, like back propagation through time (BPTT).

We have also investigated the different datasets using the proposed SA model utilizing ML and DL classifiers. The dataset Sentiment140 delivered a higher F1-score, precision, recall, and accuracy performances compared to other datasets SemEval-2014 and STS-Gold. The reason behind this improvement is the presence of around 160,000 reviews and a large number of samples for each class in the Sentiment140 dataset, which is much larger than the other two datasets. The STS-Gold delivered lower efficiency using each classifier than SemEval-2014 and Sentiment140 datasets as it has fewer training samples. The proposed LSTM-based model outperformed the other ML-based SA model. It can also be observed from Figure 3 that the LSTM-based SA model (SADL) has a higher ASAT compared to other classifiers. The obvious reason is that DL-based methods take extra time for training and classification. Nevertheless, considering the improvements that it has shown in SA, it is acceptable.

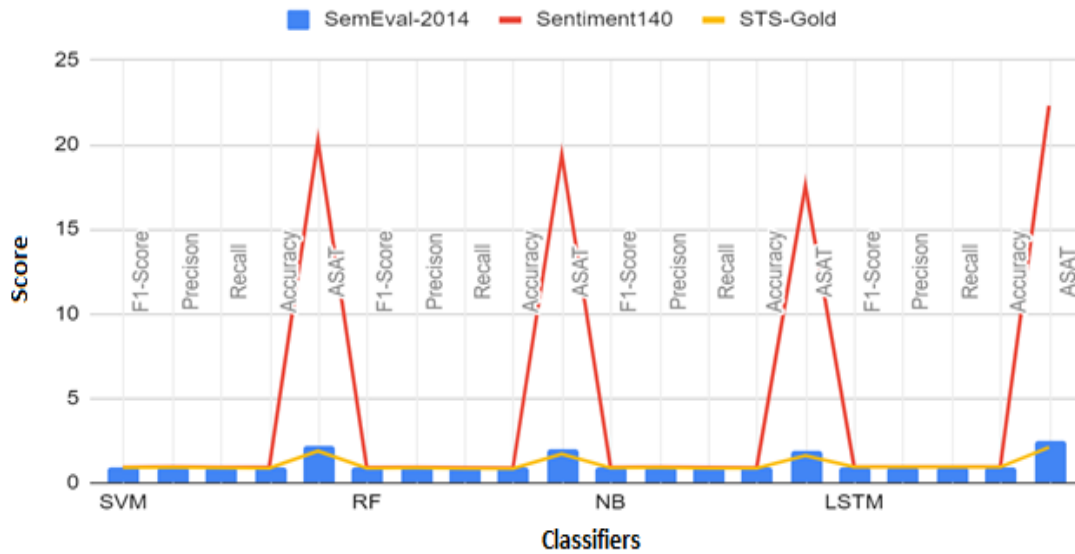


Figure 3. Analysis of F1-score, precision, recall, and ASAT for different classifiers based on different datasets

The performance measurements for each parameter using each classifier are shown in Table 4. Among the three ML classifiers: NB, SVM, and RF, the proposed feature engineering technique employing SVM outperforms the other two classifiers in terms of F1-score, recall, accuracy, and precision (refer to Table 4). This improvement is due to its ability to calculate the best boundary between various sentiment classes.

Table 4. Performance analysis of proposed features extraction techniques using LSTM

F1-Score	SVM	RF	NB	LSTM
SemEval-2014	0.922	0.891	0.916	0.945
Sentiment140	0.939	0.922	0.937	0.956
STS-Gold	0.913	0.881	0.895	0.951
<b>Precision</b>	<b>SVM</b>	<b>RF</b>	<b>NB</b>	<b>LSTM</b>
SemEval-2014	0.946	0.912	0.931	0.956
Sentiment140	0.961	0.933	0.950	0.963
STS-Gold	0.927	0.899	0.913	0.954
<b>Recall</b>	<b>SVM</b>	<b>RF</b>	<b>NB</b>	<b>LSTM</b>
SemEval-2014	0.912	0.891	0.919	0.954
Sentiment140	0.938	0.912	0.922	0.965
STS-Gold	0.899	0.862	0.877	0.947
<b>Accuracy</b>	<b>SVM</b>	<b>RF</b>	<b>NB</b>	<b>LSTM</b>
SemEval-2014	0.914	0.885	0.908	0.946
Sentiment140	0.935	0.903	0.918	0.967
STS-Gold	0.881	0.852	0.879	0.945
<b>ASAT</b>	<b>SVM</b>	<b>RF</b>	<b>NB</b>	<b>LSTM</b>
SemEval-2014	2.18	2.03	1.93	2.45
Sentiment140	20.23	19.39	17.56	22.31
STS-Gold	1.89	1.71	1.62	2.11

#### 4.4. Comparison of SADL with state-of-the-Art SA methods

Tables 5, 6, and 7 show the comparative analysis of the proposed SADL model with state-of-the-art methods utilizing datasets SemEval-2014, Sentiment140, and STS-Gold datasets, respectively. The various recent SA methods that have been used for investigation include supervised ABSA (SABSA) [28], SentiVec [21], TF-IDF+N-gram+SVM [25], SEML [37], MTMVN [33], and hybrid analysis of sentiments (HAS) [54]. We assessed the performance of proposed and existing methods based on precision, recall, and F1-score parameters.

The comparison of the proposed method with state-of-the-art SA methods employing all three datasets reveals that the proposed automated SADL model performs better than the existing methods and our former model HAS. The difference between the previous HAS model and the new SADL model lies in the use of LSTM for the SA classification and the removal of n-gram features in RRF. Mainly, the LSTM leads to performance improvements over the previous ML-based HAS model. The SADL improves the overall SA classification efficiency by approximately 3.5% compared to our previous HAS model. Apart from this, the SADL model also outperforms the recently proposed SA models in terms of precision, F1-score, and recall parameters. The hybrid feature engineering mechanism and DL classification delivered substantially enhanced results than existing methods for SA. The SADL approach's performance improvement is mostly attributable to the fact that it creates a feature vector that addresses the issues connected with ambiguity, negation, sarcasm, emotions, and feelings. In comparison with TF-IDF+N-gram+SVM and SentiVec, the ABSA methods SEML, MTMVN, and SABSA perform poorly because they lack negation handling and review-specific features.

Table 5. Comparison of SADL with state-of-the-art SA approaches utilizing SemEval-2014 dataset

SA/ABSA approach	Precision	Recall	F1-score
SABSA [28]	0.844	0.831	0.838
SentiVec [21]	0.862	0.842	0.854
TF-IDF+N-gram+SVM [25]	0.851	0.838	0.845
SEML [37]	0.841	0.824	0.833
MTMVN [33]	0.793	0.773	0.785
HAS [54]	0.946	0.912	0.922
SADL	0.956	0.954	0.945

Table 6. Comparison of SADL with state-of-the-art SA approaches utilizing Sentiment140 dataset

SA/ABSA approach	Precision	Recall	F1-score
SABSA [28]	0.858	0.839	0.8485
SentiVec [21]	0.877	0.858	0.8675
TF-IDF+N-gram+SVM [25]	0.866	0.846	0.856
SEML [37]	0.854	0.837	0.8455
MTMVN [33]	0.817	0.789	0.803
HAS [54]	0.961	0.938	0.9495
SADL	0.963	0.965	0.956

Table 7. Comparison of SADL with state-of-the-art SA approaches utilizing STS-Gold dataset

SA/ABSA approach	Precision	Recall	F1-score
SABSA [28]	0.791	0.784	0.7875
SentiVec [21]	0.845	0.837	0.841
TF-IDF+N-gram+SVM [25]	0.831	0.817	0.824
SEML [37]	0.826	0.809	0.8175
MTMVN [33]	0.776	0.758	0.767
HAS [54]	0.927	0.899	0.913
SADL	0.954	0.947	0.951

#### 4.5. CRS performance analysis

The comparison of the proposed model's CRS phase with recently proposed CRS models is shown in Table 8, based on performance metrics ROUGE-1, ROUGE-2, and ROUGE-L. Based on 10 samples from each dataset, we averaged the RS outcomes. We used 30 reviews to measure ROUGE-1, ROUGE-2, and ROUGE-L parameters for the proposed CRS model and three recently proposed models by Liu and Wan [47], Marzijarani and Sajedi [48], and Sheela and Janet [49]. As shown in Table 8, the proposed CRS model achieves higher RS performance than all three existing methods. This improvement is primarily due to the inclusion of SADL and ARF outcomes, as well as the abstract summarization function.

Table 8. Performance analysis of ROUGE review summarization parameter

Method	Rouge-1	Rouge-2	Rouge-L
Liu and Wan [47]	18.05	6.88	17.84
Marzizarani and Sajedi [48]	17.51	5.73	16.53
Sheela and Janet [49]	18.56	7.01	17.95
CRS	19.31	7.67	18.71

## 5. CONCLUSION AND FUTURE DIRECTIONS

The SADL-CRS model proposed in this paper automatically analyses the raw input reviews and generates a summary of the sentiments. The SADL-CRS has dealt with various problems, such as poor SA accuracy caused by ineffective feature extraction methods, lack of scalability, and insufficient experimental evaluations. The proposed model has been designed in such a way that it handles emotions, negations, sarcasm, ambiguity, and aspect-related feature extraction and represents each review uniquely and accurately for efficient classification. The experimental results demonstrate that the SADL model outperforms existing models using different datasets like SemEval-140, Sentiment140, and STS-Gold datasets. Additionally, the SADL-CRS model has a CRS phase that takes three inputs: pre-processed reviews, SADL outcomes, and ARF outcomes, and produces a more effective summarization for each input review. The experimental results prove the efficiency of the proposed CRS model compared to existing methods. The future recommendations include applying deep learning methods for automatic feature extraction, investigating other new data sets, and investigating other NLP methods to make the model linguistically independent.

## ACKNOWLEDGEMENTS

We would like to thank Symbiosis International (Deemed University), Pune, India for providing research facilities.

## REFERENCES




- [1] H. B. Mahajan, A. Badarla, and A. A. Junnarkar, "CL-IoT: cross-layer internet of things protocol for intelligent manufacturing of smart farming," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 7, pp. 7777–7791, Sep. 2021, doi: 10.1007/s12652-020-02502-0.
- [2] G. J. Baxter and T. M. Connolly, "Using web 2.0 tools to support the theoretical constructs of organisational learning," in *Advances in Intelligent Systems and Computing*, vol. 239, Springer International Publishing, 2014, pp. 679–688.
- [3] J. Vila and D. Ribeiro-Soriano, "An overview of web 2.0 social capital: a cross-cultural approach," *Service Business*, vol. 8, no. 3, pp. 399–404, May 2014, doi: 10.1007/s11628-014-0245-y.
- [4] A. R. Hanni, M. M. Patil, and P. M. Patil, "Summarization of customer reviews for a product on a website using natural language processing," in *2016 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2016*, Sep. 2016, pp. 2280–2285, doi: 10.1109/ICACCI.2016.7732392.
- [5] A. Ligthart, C. Catal, and B. Tekinerdogan, "Systematic reviews in sentiment analysis: a tertiary study," *Artificial Intelligence Review*, vol. 54, no. 7, pp. 4997–5053, Mar. 2021, doi: 10.1007/s10462-021-09973-3.
- [6] D. M. E. D. M. Hussein, "A survey on sentiment analysis challenges," *Journal of King Saud University - Engineering Sciences*, vol. 30, no. 4, pp. 330–338, Oct. 2018, doi: 10.1016/j.jksues.2016.04.002.
- [7] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 813–830, Mar. 2016, doi: 10.1109/TKDE.2015.2485209.
- [8] M. E. Moussa, E. H. Mohamed, and M. H. Haggag, "A survey on opinion summarization techniques for social media," *Future Computing and Informatics Journal*, vol. 3, no. 1, pp. 82–109, Jun. 2018, doi: 10.1016/j.fcij.2017.12.002.
- [9] J. Singh, G. Singh, and R. Singh, "A review of sentiment analysis techniques for opinionated web text," *CSI Transactions on ICT*, vol. 4, no. 2–4, pp. 241–247, Dec. 2016, doi: 10.1007/s40012-016-0107-y.
- [10] N. U. Pannala, C. P. Nawarathna, J. T. K. Jayakody, L. Rupasinghe, and K. Krishnadeva, "Supervised learning based approach to aspect based sentiment analysis," in *Proceedings - 2016 16th IEEE International Conference on Computer and Information Technology, CIT 2016, 2016 6th International Symposium on Cloud and Service Computing, IEEE SC2 2016 and 2016 International Symposium on Security and Privacy in Social Netwo*, Dec. 2017, pp. 662–666, doi: 10.1109/CIT.2016.107.
- [11] C. L. Liu, W. H. Hsiao, C. H. Lee, G. C. Lu, and E. Jou, "Movie rating and review summarization in mobile environment," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 3, pp. 397–407, May 2012, doi: 10.1109/TSMCC.2011.2136334.
- [12] T. Parlar, S. A. Özel, and F. Song, "QER: a new feature selection method for sentiment analysis," *Human-centric Computing and Information Sciences*, vol. 8, no. 1, Dec. 2018, doi: 10.1186/s13673-018-0135-8.
- [13] P. H. Shahana and B. Omman, "Evaluation of features on sentiment analysis," *Procedia Computer Science*, vol. 46, pp. 1585–1592, 2015, doi: 10.1016/j.procs.2015.02.088.
- [14] S. de Kok, L. Punt, R. van den Puttelaar, K. Ranta, K. Schouten, and F. Frasincar, "Review-aggregated aspect-based sentiment analysis with ontology features," *Progress in Artificial Intelligence*, vol. 7, no. 4, pp. 295–306, Sep. 2018, doi: 10.1007/s13748-018-0163-7.
- [15] M. S. Hossain, M. R. Rahman, and M. S. Arefin, "Aspect based sentiment classification and contradiction analysis of product reviews," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 49, Springer International Publishing, 2020, pp. 631–644.
- [16] S. Chakraborty, P. Goyal, and A. Mukherjee, "Aspect-based sentiment analysis of scientific reviews," in *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, Aug. 2020, pp. 207–216, doi: 10.1145/3383583.3398541.
- [17] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: a review," *Artificial Intelligence*

- Review*, vol. 53, no. 6, pp. 4335–4385, Dec. 2020, doi: 10.1007/s10462-019-09794-5.
- [18] L. Zhang, S. Wang, and B. Liu, “Deep learning for sentiment analysis: A survey,” *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, Jul. 2018, doi: 10.1002/widm.1253.
  - [19] L. Wang, J. Niu, and S. Yu, “SentiDiff: combining textual information and sentiment diffusion patterns for Twitter sentiment analysis,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 10, pp. 2026–2039, Oct. 2020, doi: 10.1109/TKDE.2019.2913641.
  - [20] Y. Hao, T. Mu, R. Hong, M. Wang, X. Liu, and J. Y. Goulermas, “Cross-domain sentiment encoding through stochastic word embedding,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 10, pp. 1909–1922, Oct. 2020, doi: 10.1109/TKDE.2019.2913379.
  - [21] L. Zhu, W. Li, Y. Shi, and K. Guo, “SentiVec: Learning sentiment-context vector via kernel optimization function for sentiment analysis,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 6, pp. 2561–2572, Jun. 2021, doi: 10.1109/TNNLS.2020.3006531.
  - [22] A. Naresh and P. V. Krishna, “An efficient approach for sentiment analysis using machine learning algorithm,” *Evolutionary Intelligence*, vol. 14, no. 2, pp. 725–731, Jun. 2021, doi: 10.1007/s12065-020-00429-1.
  - [23] M. Singh, A. K. Jakhar, and S. Pandey, “Sentiment analysis on the impact of coronavirus in social life using the BERT model,” *Social Network Analysis and Mining*, vol. 11, no. 1, Dec. 2021, doi: 10.1007/s13278-021-00737-z.
  - [24] S. Munuswamy, M. S. Saranya, S. Ganapathy, S. Muthurajkumar, and A. Kannan, “Sentiment analysis techniques for social media-based recommendation systems,” *National Academy Science Letters*, vol. 44, no. 3, pp. 281–287, Jul. 2021, doi: 10.1007/s40009-020-01007-w.
  - [25] K. Ayyub, S. Iqbal, E. U. Munir, M. W. Nisar, and M. Abbasi, “Exploring diverse features for sentiment quantification using machine learning algorithms,” *IEEE Access*, vol. 8, pp. 142819–142831, 2020, doi: 10.1109/ACCESS.2020.3011202.
  - [26] O. Oyeboode, F. Alqahtani, and R. Otji, “Using machine learning and thematic analysis methods to evaluate mental health apps based on user reviews,” *IEEE Access*, vol. 8, pp. 111141–111158, 2020, doi: 10.1109/ACCESS.2020.3002176.
  - [27] F. Iqbal *et al.*, “A hybrid framework for sentiment analysis using genetic algorithm based feature reduction,” *IEEE Access*, vol. 7, pp. 14637–14652, 2019, doi: 10.1109/ACCESS.2019.2892852.
  - [28] K. Schouten, O. van der Weijde, F. Frasinca, and R. Dekker, “Supervised and unsupervised aspect category detection for sentiment analysis with co-occurrence data,” *IEEE Transactions on Cybernetics*, vol. 48, no. 4, pp. 1263–1275, Apr. 2018, doi: 10.1109/TCYB.2017.2688801.
  - [29] O. Alqaryouti, N. Siyam, A. A. Monem, and K. Shaalan, “Aspect-based sentiment analysis using smart government review data,” *Applied Computing and Informatics*, Jul. 2019, doi: 10.1016/j.aci.2019.11.003.
  - [30] N. Nandal, R. Tanwar, and J. Pruthi, “Machine learning based aspect level sentiment analysis for Amazon products,” *Spatial Information Research*, vol. 28, no. 5, pp. 601–607, Feb. 2020, doi: 10.1007/s41324-020-00320-2.
  - [31] J. K. Prathi, P. K. Raparathi, and M. V. Gopalachari, “Real-time aspect-based sentiment analysis on consumer reviews,” in *Advances in Intelligent Systems and Computing*, vol. 1079, Springer Singapore, 2020, pp. 801–810.
  - [32] M. Shams, N. Khoshavi, and A. Baraani-Dastjerdi, “LISA: Language-independent method for aspect-based sentiment analysis,” *IEEE Access*, vol. 8, pp. 31034–31044, 2020, doi: 10.1109/ACCESS.2020.2973587.
  - [33] Y. Bie and Y. Yang, “A multitask multiview neural network for end-to-end aspect-based sentiment analysis,” *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 195–207, Sep. 2021, doi: 10.26599/BDMA.2021.9020003.
  - [34] H. Shim, D. Lowet, S. Luca, and B. Vanrumste, “LETS: A label-efficient training scheme for aspect-based sentiment analysis by using a pre-trained language model,” *IEEE Access*, vol. 9, pp. 115563–115578, 2021, doi: 10.1109/ACCESS.2021.3101867.
  - [35] Q. Zhong, L. Ding, J. Liu, B. Du, H. Jin, and D. Tao, “Knowledge graph augmented network towards multiview representation learning for aspect-based sentiment analysis,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 10, pp. 10098–10111, Oct. 2023, doi: 10.1109/TKDE.2023.3250499.
  - [36] R. Kumar, H. S. Pannu, and A. K. Malhi, “Aspect-based sentiment analysis using deep networks and stochastic optimization,” *Neural Computing and Applications*, vol. 32, no. 8, pp. 3221–3235, Mar. 2020, doi: 10.1007/s00521-019-04105-z.
  - [37] N. Li, C. Y. Chow, and J. D. Zhang, “SEML: A semi-supervised multi-task learning framework for aspect-based sentiment analysis,” *IEEE Access*, vol. 8, pp. 189287–189297, 2020, doi: 10.1109/ACCESS.2020.3031665.
  - [38] M. S. Alamanda, “Aspect-based sentiment analysis search engine for social media data,” *CSI Transactions on ICT*, vol. 8, no. 2, pp. 193–197, Jun. 2020, doi: 10.1007/s40012-020-00295-3.
  - [39] Q. Lu, Z. Zhu, G. Zhang, S. Kang, and P. Liu, “Aspect-gated graph convolutional networks for aspect-based sentiment analysis,” *Applied Intelligence*, vol. 51, no. 7, pp. 4408–4419, Jul. 2021, doi: 10.1007/s10489-020-02095-3.
  - [40] S. Datta and S. Chakrabarti, “Aspect based sentiment analysis for demonetization tweets by optimized recurrent neural network using fire fly-oriented multi-verse optimizer,” *Sadhana - Academy Proceedings in Engineering Sciences*, vol. 46, no. 2, Apr. 2021, doi: 10.1007/s12046-021-01608-1.
  - [41] A. Londhe and P. V. R. D. P. Rao, “Aspect based sentiment analysis – an incremental model learning approach using LSTM-RNN,” in *Communications in Computer and Information Science*, vol. 1440, Springer International Publishing, 2021, pp. 677–689.
  - [42] K. Shanmugavadivel, V. E. Sathishkumar, S. Raja, T. B. Lingaiah, S. Neelakandan, and M. Subramanian, “Deep learning based sentiment analysis and offensive language identification on multilingual code-mixed data,” *Scientific Reports*, vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-26092-3.
  - [43] H. Kaur, S. U. Ahsaan, B. Alankar, and V. Chang, “A proposed sentiment analysis deep learning algorithm for analyzing COVID-19 tweets,” *Information Systems Frontiers*, vol. 23, no. 6, pp. 1417–1429, Dec. 2021, doi: 10.1007/s10796-021-10135-7.
  - [44] V. Balakrishnan, Z. Shi, C. L. Law, R. Lim, L. L. Teh, and Y. Fan, “A deep learning approach in predicting products’ sentiment ratings: a comparative analysis,” *Journal of Supercomputing*, vol. 78, no. 5, pp. 7206–7226, Nov. 2022, doi: 10.1007/s11227-021-04169-6.
  - [45] B. Yu and S. Zhang, “A novel weight-oriented graph convolutional network for aspect-based sentiment analysis,” *Journal of Supercomputing*, vol. 79, no. 1, pp. 947–972, Jul. 2023, doi: 10.1007/s11227-022-04689-9.
  - [46] S. Ma, X. Sun, J. Lin, and X. Ren, “A hierarchical end-to-end model for jointly improving text summarization and sentiment classification,” in *IJCAI International Joint Conference on Artificial Intelligence*, Jul. 2018, vol. 2018-July, pp. 4251–4257, doi: 10.24963/ijcai.2018/591.
  - [47] H. Liu and X. Wan, “Neural review summarization leveraging user and product information,” in *International Conference on Information and Knowledge Management, Proceedings*, Nov. 2019, pp. 2389–2392, doi: 10.1145/3357384.3358161.
  - [48] S. B. Marzijarani and H. Sajedi, “Opinion mining with reviews summarization based on clustering,” *International Journal of Information Technology (Singapore)*, vol. 12, no. 4, pp. 1299–1310, Sep. 2020, doi: 10.1007/s41870-020-00511-y.




- [49] J. Sheela and B. Janet, "An abstractive summary generation system for customer reviews and news article using deep learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 7, pp. 7363–7373, Aug. 2021, doi: 10.1007/s12652-020-02412-1.
- [50] S. Mussiraliyeva, B. Omarov, P. Yoo, and M. Bolatbek, "Applying machine learning techniques for religious extremism detection on online user contents," *Computers, Materials and Continua*, vol. 70, no. 1, pp. 915–934, 2021, doi: 10.32604/cmc.2022.019189.
- [51] B. Omarov et al., "Artificial intelligence in medicine: real time electronic stethoscope for heart diseases detection," *Computers, Materials and Continua*, vol. 70, no. 2, pp. 2815–2833, 2022, doi: 10.32604/cmc.2022.019246.
- [52] A. H. Barshooi and A. Amirkhani, "A novel data augmentation based on Gabor filter and convolutional deep learning for improving the classification of COVID-19 chest X-Ray images," *Biomedical Signal Processing and Control*, vol. 72, Art. no. 103326, Feb. 2022, doi: 10.1016/j.bspc.2021.103326.
- [53] A. Amirkhani and A. H. Barshooi, "Consensus in multi-agent systems: a review," *Artificial Intelligence Review*, vol. 55, no. 5, pp. 3897–3935, Nov. 2022, doi: 10.1007/s10462-021-10097-x.
- [54] G. Kaur and A. Sharma, "HAS: Hybrid analysis of sentiments for the perspective of customer review summarization," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 9, pp. 11971–11984, Feb. 2023, doi: 10.1007/s12652-022-03748-6.
- [55] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The stanford CoreNLP natural language processing toolkit," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2014, vol. 2014-June, pp. 55–60, doi: 10.3115/v1/p14-5010.
- [56] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, "Sentiment analysis using Word2vec and long short-term memory (LSTM) for Indonesian hotel reviews," *Procedia Computer Science*, vol. 179, pp. 728–735, 2021, doi: 10.1016/j.procs.2021.01.061.
- [57] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutopoulos, and S. Manandhar, "SemEval-2014 Task 4: Aspect based sentiment analysis," in *8th International Workshop on Semantic Evaluation, SemEval 2014 - co-located with the 25th International Conference on Computational Linguistics, COLING 2014, Proceedings*, 2014, pp. 27–35, doi: 10.3115/v1/s14-2004.
- [58] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," Stanford University, Jan. 2019.
- [59] H. Saif, M. Fernandez, and H. Alani, *Evaluation datasets for Twitter sentiment analysis. a survey and a new dataset, the STS-Gold*, vol. 1096. 2013.
- [60] C. Y. Lin and E. Hovy, "Automatic evaluation of summaries using N-gram co-occurrence statistics," *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003*, 2003, doi: 10.3115/1073445.1073465.

## BIOGRAPHIES OF AUTHORS



**Gagandeep Kaur**    is pursuing Ph. D degree in computer science and engineering from LPU, Punjab, India, focusing on aspect-based sentiment analysis and how to shift towards a more semantics-oriented kind of sentiment analysis. She is currently working as assistant professor in Department of Computer Science at Symbiosis Institute of Technology, Nagpur. She has more than 11 years of experience in teaching. Her area of interests includes NLP, AI/ML, data science, and Image processing. She can be contacted at email: gagandeep.kaur@sitnagpur.siu.edu.in.



**Amit Sharma**    holds a Ph.D degree in CSE from I.K Gujral Punjab Technical University, Kapurthala. His core area of research is network security, IoT and big data. He has more than 17 years of experience in Teaching and Research. He is currently serving Lovely Professional University, Jalandhar as a Professor. He is a reviewer of various National and International Journals (SCI and Scopus). He can be contacted at email: amit.25076@lpu.co.in.