Hindawi
Scientific Programming
Volume 2020, Article ID 5812715, 14 pages
https://doi.org/10.1155/2020/5812715

Research Article

Summarizing Online Movie Reviews: A Machine Learning Approach to Big Data Analytics

Atif Khan,[1] Muhammad Adnan Gul,[1] M. Irfan Uddin,[2] Syed Atif Ali Shah ,[3] Shafiq Ahmad ,[4] Muhammad Dzulqarnain Al Firdausi,[4] and Mazen Zaindin[5]

[1]Department of Computer Science, Islamia College Peshawar, Peshawar, Pakistan
[2]Institute of Computing, Kohat University of Science and Technology, Kohat, Pakistan
[3]Faculty of Engineering and Information Technology, Northern University, Nowshehra, Pakistan
[4]King Saud University, College of Engineering, Department of Industrial Engineering, Riyadh, Saudi Arabia
[5]King Saud University, College of Science, Department of Statistics and Operations Research, Riyadh, Saudi Arabia

Correspondence should be addressed to Shafiq Ahmad; ashafiq@ksu.edu.sa

Received 23 February 2020; Accepted 7 May 2020; Published 1 August 2020

Academic Editor: Shaukat Ali

Information is exploding on the web at exponential pace, so online movie review is becoming a substantial information resource for online users. However, users post millions of movie reviews on regular basis, and it is not possible for users to summarize the reviews. Movie review classification and summarization is one of the challenging tasks in natural language processing. Therefore, an automatic approach is demanded to summarize the vast amount of movie reviews, and it will allow the users to speedily distinguish the positive and negative aspects of a movie. This study has proposed an approach for movie review classification and summarization. For movie review classification, bag-of-words feature extraction technique is used to extract unigrams, bigrams, and trigrams as a feature set from given review documents, and represent the review documents as a vector space model. Next, the Naïve Bayes algorithm is employed to classify the movie reviews (represented as a feature vector) into positive and negative reviews. For the task of movie review summarization, Word2vec feature extraction technique is used to extract features from classified movie review sentences, and then semantic clustering technique is used to cluster semantically related review sentences. Different text features are used to calculate the salience score of each review sentence in clusters. Finally, the top-ranked sentences are chosen based on highest salience scores to produce the extractive summary of movie reviews. Experimental results reveal that the proposed machine learning approach is superior than other state-of-the-art approaches.

1. Introduction

With the expansion of Web 2.0 that emphasizes the involvement of users, many websites such as a movie review website, such as Internet Movie Database (IMDB) and Amazon, encourage its users to write reviews for the products they liked or purchased, in order to enhance the shopping experience and satisfaction of customers. Online sellers often ask their customers to provide opinions or reviews over products or services they purchased online. The amount of reviews received by a product increases quickly as millions of customers post reviews about a product, which results in information overload. This information overload makes it a challenging task for a potential customer to scan each review of a product for making a quick decision whether to purchase a product or not. At the same time, it is also hard for service providers or online merchants/product manufacturers to keep track of a huge amount of reviews posted by customers related to the services or products [1]. In order to overcome the challenge of information overload, an automatic review classification and summarization system is needed [2].

In this study, we will focus on the movie review domain. Considering the movies, summarizing thousands of reviews received by a movie can help the viewers (customers) to swiftly scan the summary of it and promptly make a decision whether to watch a movie or not. On the other hand, the summary of movie reviews can assist the movie service provider such as Netflix to swiftly understand the watching patterns or the interests of their customers.

This study introduced an automatic approach that classifies and summarizes the movie reviews. Such approach will assist the new user to swiftly understand the positive and negative aspects of a specific movie and hence can quickly take a decision whether to watch a movie or not. The task of review classification/mining and summarization (RCS) comprises of two steps: the first step is review mining which mines the reviews received by a movie by classifying them into positive and negative. The second step is review summarization, which generates a concise summary from the classified reviews.

Now a days, RCS gained significant attention in several areas [3], such as, from the reviews/opinions given by people online about a political announcements or news, the government can observe the effect of current policies (or events) on the common public and take timely and proper actions on the basis of available information. On the other hand, product reviews gather feedback from customers, and summarizing such customer feedback assists the online manufacturer/retailer to know about their products perceived by the customers.

Review mining [4] classifies the review text into negative or positive. There are various approaches that classify review text into negative and positive reviews such as dictionary-based approaches and machine learning (ML) approaches. Various ML-based approaches such as support vector machine (SVM) [5], decision trees [6], and neural networks [7] have been presented for text classification and revealed their abilities in various domains. NB is a state-of-the-art ML algorithm and has been proved to be very effective in traditional text classification. The classification decisions of NB are remarkably good. NB is frequently used as a baseline in text classification and sentiment analysis research, since it combines good accuracy with efficiency [8]. Therefore, this study employs NB for movie review classification. On the other hand, dictionary-based approaches use word lexicons for semantic orientation of a document [9]. However, dictionary-based approaches are incapable to deal with domain-specific orientations.

Review summarization is a procedure in which a summary is generated from a gigantic amount of review sentences [10]. Numerous techniques such as supervised ML based [5, 6] and unsupervised/lexicon based [10, 11] have been applied for review summarization. However, the unsupervised/lexicon-based approaches heavily rely on linguistic resources and are limited to words present in the lexicon. On the other hand, mostly supervised ML approaches showed superior performance than unsupervised based approaches but they are limited to specific domains. Previous research shows that text summarization has been successfully applied in numerous domains [12–16]. The text summarization technique is employed to mine the salient information from source document and produce a short version of the document for different users [17–20].

Several users regularly post bulk reviews on movie review websites such as IMDB, which describe user attitude concerning a specific movie. Thus, automatically mining and summarizing these bulk reviews is desirable. Therefore, this study suggests an approach that automatically classifies and summarizes movie reviews by integrating the supervised ML algorithm with a semantic clustering approach. The proposed approach operates in the following manner: at first, it employs a simple feature extraction technique called bag of words (BoW) to extract a feature set (unigrams, bigrams, and trigrams) from movie reviews and represents them as a vector space model or feature vector. The next phase uses the Naïve Bayes machine learning algorithm to classify the movie reviews (represented as a feature vector) into positive and negative sentences. For the task of movie review summarization, Word2vec feature extraction technique is used to extract features from classified movie review sentences, and then semantic clustering technique is used to cluster semantically related review sentences. Different text features are used to calculate the salience score of each review sentence in clusters. Finally, the top-ranked sentences are chosen based on highest salience scores to produce an extractive summary of movie reviews. Our contributions are summarized as follows:

(a) To classify the movie reviews by using the Naïve Bayes machine learning algorithm and unigrams, bigrams, and trigrams as a feature set

(b) To propose sentence embedding based semantic clustering technique for extractive summary generation from classified movie reviews

(c) Evaluation of the proposed method with other benchmark methods in context of ROUGE-1 and ROUGE-2 evaluation metrics

The rest of paper is organized as follows: Section 2 demonstrates the previous work done in the area of opinion mining and summarization. Section 3 outlines the proposed method. Evaluation results and discussion are presented in Section 4. Finally, Section 5 presents the conclusion along with the future work.

2. Related Work

The task of review mining and summarization contains two main steps: review mining and review summarization. First, we discuss the relevant literature to review mining followed by review summarization. Review mining or opinion mining is a procedure in which we extract, analyse, and classify the subjective information and determination of sentiment related to a specific target. Different approaches are suggested by many researchers for the task of review mining [4], such as, considering text of a review document $A = [a_1, a_2, a_3, \ldots, a_n]$ with a set of classes $B =$ (positive, negative), the review mining job is to classify every single review sentence a_i within document A, with the help of the predefined class label (positive or negative) in set B [21].

Numerous review mining techniques, such as ML based and sentiment lexicon based, have been proposed for mining reviews in different domains [1, 5, 22, 23]. The authors in [24] presented applications and difficulties in the area of opinion/review mining. ML-based algorithms [24–26] are also utilized for opinion classifications of documents. The ML algorithms are classified into two categories: supervised ML and unsupervised ML techniques. These techniques accomplish the aim of sentiment classification/opinion mining by extracting and selecting an appropriate set of features.

Supervised ML technique such as SVM [5] is applied for sentiment classification of movie review data. The authors in [6] used decision trees to classify high-/low-informative opinion phrases extracted from restaurant reviews. On the other hand, the authors in [27] used unsupervised ML techniques such as unsupervised feature clustering with topic modelling such as latent Dirichlet allocation (LDA) for obtaining labelled features. They trained the initial sentiment classifier with prior information (labelled features) from sentiment lexicon. The sentiment classifier learnt from unlabelled review documents extracted from different domains such as movies, books, and electronics. The labelled features were then used for the model's predictions on unlabelled instances using generalized expectation (GE) criteria. The authors in [28] introduced the OPINE system, which determined the semantic orientation of words by using relaxation labelling. A Pulse system [29] extracts topics along with sentiment orientation from customer feedback sampled from the car review database. The system trained a sentiment classifier by using a bootstrapping process. The features were determined by tagging sentence clusters based on their key terms.

Next, we discuss sentiment lexicon-based approach for review mining, which we can categorize into two classes: dictionary-based [30] and corpus-based [11] approaches. The authors in [30] proposed a dictionary-based approach integrated with the WordNet graph for polarity classification. The approach determined polarity scores from a thesaurus such as SentiWordNet [9] and combined it with random walk analysis of concepts found in the movie reviews. The dictionary-based approach has a drawback that it is unable to deal with context and domain-specific orientation, since the same term might have different meaning in different domains. The authors in [11] proposed a corpus-based technique by utilizing a movie review corpus that is annotated manually. The approach obtained linguistic features, e.g., nouns, adjectives, adverbs, and verbs, by performing POS tagging over the movie reviews. They also exploited a semantic resource such as SentiWordNet for computing the polarity score of the movie review document in the corpus. Both corpus- and dictionary-based approaches heavily rely on linguistic resources and are limited to words present in the lexicon.

Numerous techniques for review summarization have also been explored. Review summarization is an important step in the task of review mining and summarization [4], which extract salient information from the review document and present them in summary form. The final summary might be a feature-based summary, or it can be a generic summary covering the general information about a product (camera, cellular phone, and movie) in a concise manner [4]. The authors in [1] proposed an approach for a feature-based summary for customer product (camera and cellular phone) reviews. The approach used word attributes, including synset in WordNet, part-of-speech (POS), and occurrence frequency. The final summary was organized according to extracted features. The authors in [5] introduced a latent-semantic analysis (LSA)-based approach to detect product features from movie reviews. In order to generate a review summary, opinion words and product features were used to choose relevant sentences to form a review summary. However, this methodology was limited to Chinese movie reviews and has not been applied to English movie reviews. A multiknowledge approach was proposed in [3] for movie review summarization. The approach utilized WordNet, labelled movie training data and movie casts to produce a keyword list for determining features and opinions. Finally, summary sentences are reorganized according to the extracted features. However, this approach might not be able to find valid feature-opinion pairs as grammatical relations do not check the semantic relationship between opinion words and features. However, the previous approaches proposed for movie summarization are limited to generate a feature-based summary rather than a generic summary. Therefore, a text summarization approach is proposed that incorporates the supervised ML algorithm with a graph-based ranking algorithm to produce a generic summary of movie reviews. Moreover, a text summarization approach based on unsupervised ML [31] has also been proposed to generate a summary from online hotel reviews. However, this approach is limited to hotel reviews. On the other hand, we proposed a supervised ML approach for a different domain such as movie reviews.

Text summarization techniques have been employed in different application domains, such as summaries of news articles, webpages, and patents [32, 33]. The authors in [34] presented a text summarization technique to produce summaries from patents. The approach used several features such as sentence position and cue phrases while determining the sentence importance. The authors in [35] applied the term TF-IDF technique and ontology tree structure techniques for finding keywords and extracting the important content of a patent document. The important sentences were then grouped using a clustering technique to produce a summary. The authors in [36] generated summaries from webpages. The approach performed a query expansion by using WordNet, and then the expanded query is given to the Google search engine to find related documents. The final summary is produced based on sentences containing the relevant keywords. A statistical method was proposed by [37] for news article summarization. The sentences in the news documents are scored based on different features such as length of sentence, first sentence of news article, title of news article, proper nouns, and term frequency. Sentences with top score are selected to produce a summary. The authors in [38] presented a pattern-based method for news article summarization. The score of each sentence is found by adding the weights of covered patterns. Finally, the sentences are iteratively chosen to form a summary based on the low similarity with the previously selected sentences, and the maximum score among all candidate sentences.

In previous few years, various graph-based approaches [39] have attracted more attention and been effectively applied in area of text summarization. These methods utilize PageRank (PR) algorithm [40] and its other variants to give score/rank to graph nodes, which represent sentences or passages. The authors in [41] proposed a connectivity graph, which assumes that nodes only carry significant information if they are connected to many other nodes. The authors in [42] introduced a Lex-PageRank approach that is based on Eigen vector centrality, which creates a sentence connectivity matrix and utilizes the similar algorithm such as PR to find the significant sentences for a summary. A similar algorithm to PR is also proposed by [43] which finds salient sentences for summary generation. The authors in [23] presented a graph-based approach, which integrates surface features with text content and investigates subtopic features for multiple documents by incorporating these features into the graph-based ranking algorithm. An affinity graph-based multidocument summarization approach [44] exploits the algorithm similar to PR and computes sentence scores on the basis of information richness in the affinity graph. The authors in [45] proposed a multidocument generic summarization approach by employing a document-sensitive graph model and highlighted the effect of global document set information at sentence level. A generic multidocument summarization using the weighted graph model is introduced by [46] by combining sentence-clustering and sentence-ranking methods. The authors in [47] presented a graph-based summarization method for multiple documents for Vietnamese documents for ranking of important sentences in which they employed the traditional PR algorithm. The authors in [48] demonstrated an event graph-based approach for generating an extractive summary of a multidocument. However, it requires the creation of handcrafted extraction rules of argument, which is a tedious task and may possibly limit its application to a specific domain. All the previous graph-based summarization approaches were applied to a new article domain, and a simple PageRank algorithm is employed. However, we propose a graph-based summarization approach for the movie review domain and employ a ranking algorithm based on a weighted graph. The detail of the proposed approach is presented in the next section.

3. Proposed Methodology

In this section, the research framework of the proposed study is presented as depicted in Figure 1. The framework is divided into four phases: preprocessing, feature extraction, classification of reviews, and summarization of reviews.

3.1. Preprocessing. Data preprocessing in computational linguistics is a significant procedure, particularly in review mining and summarization (RMS). As the proposed work is associated with RMS, therefore the preprocessing of review documents is needed for the efficient use of the documents before performing any experiment or giving the document as an input to the system. The preprocessing phase includes four steps, i.e., sentence segmentation, tokenization, stop word removal, and word stemming.

3.1.1. Sentence Segmentation. It is a necessary step in NLP applications such as IR, machine translation, semantic role labelling, and summarization. Sentence segmentation is the process of detecting a boundary within a text and divides the text into sentences. Generally, an interrogation (?) sign, sign of exclamation (!), and full stop/period (.) are normally used to indicate the sentence boundary [49].

For example, we have a text document: "I like this movie. It is one of the best movies."

After segmenting text document, we get two sentences.
Input: Review text:
"I like this movie. It is one of the best movies."
Output: Segmented text:
Segment1: "I like this movie."
Segment2: "It is one of the best movies."

3.1.2. Tokenization. In this task, we use a simple program to split the sentences into distinct words by splitting them at whitespaces such as blanks, tabs, and punctuation marks such as period, semicolon, comma, and colon, which are the primary cues for splitting the text into tokens.

3.1.3. Stop Word Removal. Words with frequent occurrence in the document are known as stop words. It consists of conjunctions, prepositions, articles, and frequently occurring words such as "an," "the," "a," and "I". Stop words are those words that carry a little or no meaning in the text document, so removing them from a document set is a good idea. Removing stop words from review documents helps to improve system performance. A list of stop words proposed by Buckley et al. [50] is used in the proposed framework.

3.1.4. Word Stemming. It is an important task in the preprocessing phase. Word stemming transforms the derived words to its root word or stem for capturing the similar concept. In this study, a commonly used stemming algorithm known as Porter's stemming [51] is employed to perform word stemming by removing the word suffixes. For instance, the words "playing," "played," and "plays" will be transformed to its stem or root word "play" with the help of the stemming algorithm by removing suffixes -ing, -es, and -ers.

Playing, played, and plays $\Longrightarrow$ play.

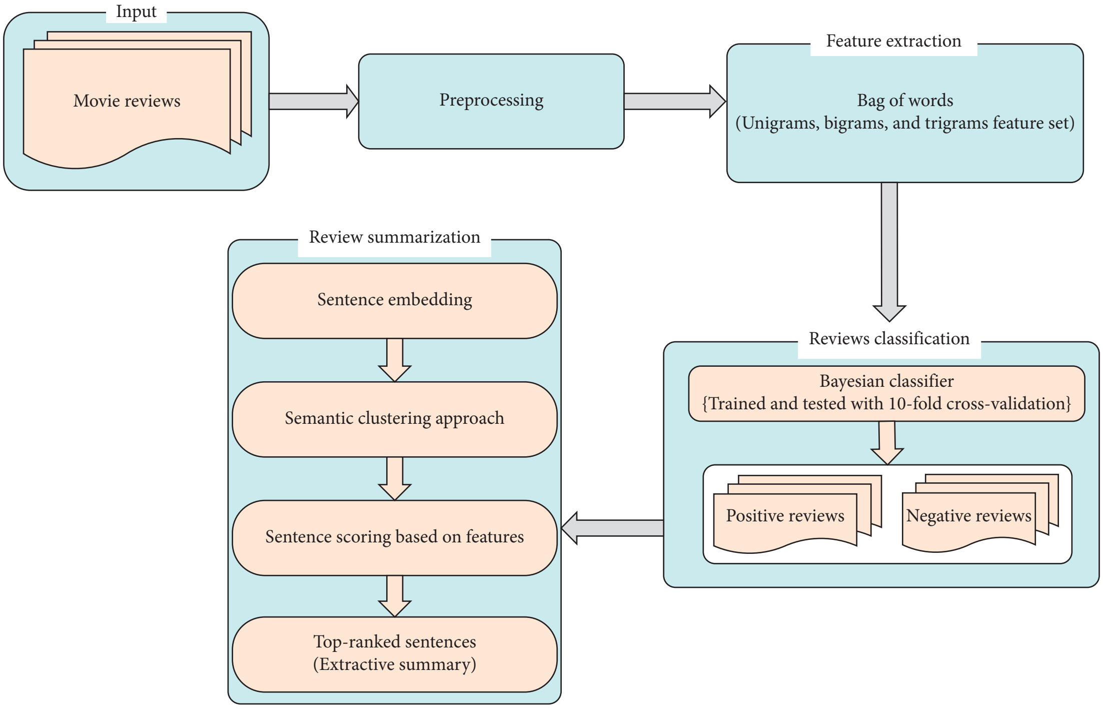

Figure 1: Proposed approach for movie review classification and summarization.

3.2. Feature Extraction. The aim of this phase is to extract features for review classification by employing a well-known feature extraction technique called bag of words (BoW). It is a simple technique used for feature extraction that represents the review text document as a vector space model. Each vector space dimension represents a feature. In this study, we use unigrams, bigrams, and trigrams as a feature set. The features in the vector space represent all the possible unigrams, bigrams (two-word sequence), and trigrams (three-word sequence) from the review text document, whereas the values of features refer to frequency or occurrence of unigrams, bigrams, and trigrams contained in the review document.

Example 1. Consider the following three review text documents, and for the sake of convenience, we have shown a single review sentence from each document.

Review document 1: “I loved this movie.”

Review document 2: “I hated this movie.”

Review document 3: “Great acting a good movie.”

There are 7 unique words (unigrams) extracted from the above review sentences. The extracted unigrams refer to the features which are “acting,” “good,” “great,” “hated,” “loved,” “movie,” and “this.” The collection of features representing the review text documents represents the vector space model. The values of features in Table 1 indicate the frequencies of unigrams.

In order to enhance the sentiment classification accuracy, this study combines unigrams, bigrams (a two-word pair), and trigrams (a three-word pair) vector space representation of a review. Bag of bigrams refers to a two-word pair in computational linguistics, for instance, “great movie,” “beautiful sky,” and “not yet”. Bigrams such as “good job,” “well done,” and “pretty good” have positive orientation. On the other hand, bigrams such as “quite expensive,” “no good,” and “bad luck” have negative orientation and bigram such as “to be” has neutral orientation. Bag of trigrams refers to a three-word pair in computational linguistics, for instance, trigram such as “acting good movie” has positive orientation while trigram such as “hated this movie” has negative orientation.

On the other hand, the BoW (unigram) approach splits a two-word pair such as “no good” into “no” and ”good” and hence the word “good” is considered as positive oriented. Bigrams also help to reduce vector space dimensions. Table 2 depicts bag-of-bigrams vector space model representation for the review documents. Referring to Example 1, the bag-of-bigrams vector space model for the review documents is shown below.

Table 3 shows the bag-of-unigrams and bigrams vector space model representation for the review documents given in Example 1.

Table 4 shows the bag-of-unigrams, bigrams, and trigrams vector space model representation for the review documents given in Example 1.

3.3. Classification of Reviews. The goal of this phase is to classify users’ review text using the supervised ML classification algorithm. The task of review classification categorizes the user’s reviews into positive and negative. In this study, we have used the Naïve Bayes (NB) classification algorithm since it is a robust classifier [52] and achieved higher accuracy on scalable datasets as compared to other state-of-the-art classification algorithms. Moreover, the NB classifier has several applications in text classification because of its simplicity and accuracy [52].

Table 1: BoW vector space model for unigrams.

Review docs	Acting	Good	Great	Hated	Loved	Movie	This	Class
Review doc1	0	0	0	0	1	1	1	+ve
Review doc2	0	0	0	1	0	1	1	−ve
Review doc3	1	1	1	0	0	1	0	+ve

Table 2: Bag-of-bigrams vector space model.

Review docs	Acting good	Good movie	Great acting	Hated this	Loved this	This movie	Class
Review doc1	0	0	0	0	1	1	+ve
Review doc2	0	0	0	1	0	1	−ve
Review doc3	1	1	1	0	0	0	+ve

Table 3: Bag-of-unigrams and bigrams vector space model.

Review docs	Acting	Acting good	Good	Good movie	Loved this	Movie	This	This movie	Class
Review doc1	0	0	0	0	1	1	1	1	+ve
Review doc2	0	0	0	0	0	1	1	1	−ve
Review doc3	1	1	1	1	0	1	0	0	+ve

Table 4: Bag-of-unigrams, bigrams, and trigrams vector space model.

Review docs	Acting	Acting good	Acting good movie	Loved	Loved this	Loved this movie	Movie	This	This movie	Class
Review doc1	0	0	0	1	1	1	1	1	1	+ve
Review doc2	0	0	0	0	0	0	1	1	1	−ve
Review doc3	1	1	1	0	0	0	1	0	0	+ve

In order to classify the reviews, the feature vectors along with their labels are given as input to the classifier. Probability of a term given certain category (positive or negative) is calculated based on the number of times a term occurs with that category in the review documents. Here, the term refers to either unigram, bigram or trigram since the features used in this study are unigrams, bigrams, and trigrams. In order to classify a new review document, the probability of each term (unigram, bigram, and trigram) in the document given class label (+ve) is determined, and then the probability of the review document given class label (+ve) is calculated by multiplying the probabilities of all terms with the probability of the target class (+ve). Similarly, the probability of the review document given class label (−ve) is calculated.

The review document is classified as positive if its probability given target class (+ve) is maximized, otherwise classified as negative.

Bayes' theorem is stated mathematically as given in the following equation:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}. \tag{1}$$

Consider a new review document "*I love this movie*" is given to the NB classifier which will classify it into either positive or negative. The review document here is a short sentence. At first, the review document is represented as bag-of-unigrams, bigrams, and trigrams feature vector representation as shown in Table 4. The probability of a review document given certain class (positive and negative) can be calculated using equation (2), given as follows:

$$P(\text{Doc} \mid \text{Class}) = \prod_{i=1}^{|\text{Doc}|} P(W \mid c_i), \tag{2}$$

where Doc is the review document, |Doc| is the length of document, and $P(W \mid c_i)$ is the probability of a term W in a review document given certain class (+ve or −ve). Table 3 shows unigrams and bigrams along with their vector representation for the corresponding review documents given in Example 1.

In order to classify a review document "*I loved this movie*," we need to determine the probabilities of all terms (unigrams and bigrams) in the review documents labelled as positive. The probability of each term given class c_i, $P(w_k \mid c_i)$, is computed as follows:

$$P\left(w_k \mid \text{positive}\right) = \frac{n_k + 1}{n + |\text{VOC}|}, \tag{3}$$

where n_k is the number of times the term w_k occurs in positive cases, and n is the total number of words in positive cases. |VOC| indicates the number of unique unigrams, bigrams, and trigrams in the review documents. Hence, the probability of the review document given positive/negative case is estimated based on probabilities of all unigrams, bigrams, and trigrams in the review document:

$$P(\text{positive}) = \frac{\text{number of positive review cases}}{\text{total number of review cases}}, \tag{4}$$

$$P\left({}''\text{loved}'' \mid \text{positive}\right) = \frac{\text{number of times “loved” occurs in positive cases} + 1}{\text{total number of words in positive cases} + |\text{VOC}|}, \tag{5}$$

$$P\left({}''\text{this movie}'' \mid \text{positive}\right) = \frac{\text{number of times “this movie” occurs in positive cases} + 1}{\text{total number of words in positive cases} + |\text{VOC}|}, \tag{6}$$

$$P\left({}''\text{loved this movie}'' \mid \text{positive}\right) = \frac{\text{number of times “loved this movie” occurs in positive cases} + 1}{\text{total number of words in positive cases} + |\text{VOC}|}. \tag{7}$$

Based on following equation (7), the review document is assigned to a class if the probability value of the review document given that class is maximized:

$$C_{\text{NB}} = \arg\max_{C_i \in C} P(C_i) \prod_{w \in \text{words}} P\left(\frac{w}{C_i}\right). \tag{8}$$

In other words, the review document is assigned to a positive class, if the probability value of the review document given that class is maximized and vice versa.

3.4. Summarization of Reviews. The goal of this phase is to summarize the classified reviews (both positive and negative reviews). This phase comprises of five steps: sentence embedding, clustering of semantically similar sentences, text feature extraction, selection of top rank sentences from clusters, and summary generation.

3.4.1. Sentence Embedding. The goal of this phase is to split the classified reviews into sentences and build sentence embeddings from collection of sentences. Sentence embeddings are richer representations of text which preserve both semantic and syntactic information in sentences, leading to enhanced performance in almost each NLP task.

In order to get sentence embedding, we extract word embeddings for each word in sentences using the Word2vec model. We employed Google's pretrained Word2vec model [53, 54] to learn word embeddings (word vectors) for each word in all sentences. The Word2vec model, released by Google, is a neural network-based implementation that learns distributed vector representations of words based on continuous bag of words. The model is trained approximately on 100 billion words from the Google News dataset. We leave the default word vector length to be 300 features.

Finally, sentence vectors/sentence embeddings are obtained by averaging all word embeddings present in the vocabulary of Word2vec and the words not present in the vocabulary are ignored. Each review sentence is now represented as a numeric vector. We also stored sentence text along with its numeric representation for later use in sentence selection for summary based on different text features.

3.4.2. Semantic K-Means Clustering. The goal of this step is grouping semantically similar sentences using the semantic K-means (KM) algorithm. This algorithm is extremely easy to implement and is also computationally very efficient compared to other clustering algorithms. It is widely used in academia, as well as in industry, and can be applied to data in higher dimensions. However, the agglomerative hierarchical clustering algorithm is not efficient with scalable data. KM is an unsupervised iterative algorithm that attempts to divide the data into K predefined distinctive nonoverlapping groups called clusters where each data point goes to only one cluster. The data points in our case refer to the sentences (represented as sentence embeddings/vectors). In this study, we used semantic K-means (SKM) clustering since it employs Euclidean distance between sentence embeddings (semantic representation of the sentence) of corresponding sentences. Hence, it groups semantically similar sentences, which will reduce overlapping review sentences and at the same time produce distinct review sentences in the summary. The key objective of the SKM algorithm is to reduce the sum of semantic distances between data points and their respective cluster centroids:

$$j = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j^2 \right\|, \tag{9}$$

where k is the number of cluster centroids, n is the number of sentences in the collection, $x_i^{(j)}$ refers to a sentence that belong to j^{th} cluster, and C_j refer to centroid of j^{th} cluster.

Semantic K-means clustering algorithm is illustrated as follows:

Pseudocode of semantic K-means clustering algorithm

```
(1) Pick the number of clusters, K
(2) Initialize cluster centroid by randomly choosing K random points from given data as cluster centroids
(3) Determine the sum of the squared distance between data points and all cluster centroids
(4) Assign each data point to the closest cluster centroid
(5) Recompute centroids of the new clusters by taking mean of the all data points belonging to each cluster
```

(6) Repeat steps 3 to 5 until centroids in new clusters do not change

We used the "elbow method" for finding the optimum number of cluster centroids. For different values of k (i.e., no. of clusters), the SSE value (sum of squared error) is calculated between data points and their respective clusters' centroids. In this study, we decided $k = 10$ to the optimum number of clusters based on the "elbow method."

The number of sentences that will be included in the final summary is dependent on the number of clusters that are formed. Here, the total number of clusters is ten and we will select high-scored information-rich sentences from ten clusters based on different text features to form an extractive summary of 10 sentences, as discussed in detail in the next step.

3.4.3. Text Feature Extraction. Text features play a significant role in the selection of relevant content for the summary. Different text features that have been chosen for this study include: length of sentence [55], proper noun feature [56], and some other features such as TF-IDF, semantic similarity among sentences. The rationale behind these features selection is that they have been widely applied in text summarization research [56]. Moreover, these features have been proven to be effective and relevant for the task of movie review summarization based on empirical observations. Therefore, this study takes into consideration the contribution of all text features rather than a few in the content selection for the summary [56]. The score of each text feature is normalized in the range of 0 to 1.

(1) Length of Sentence. The length of sentence is the ratio of the number of words in the review sentence to the maximum length of the sentence:

$$\text{Sent}_{f1} = \frac{\text{no. of words in review sentence}}{\text{max length of sentence}}. \tag{10}$$

(2) Sentence to Sentence Average Semantic Similarity. The score of this feature for a review sentence Sent_i is achieved by calculating the ratio of the sum of the similarity scores of sentences Sent_i with all other sentences in the cluster over the sentence with a maximum similarity score in the cluster of review sentences. We used the Word2vec model for generating sentence vectors from sentences as discussed in Section 3.4.1, and semantic similarity between any two sentence vectors is determined using cosine similarity. Thus, a sentence with maximum average semantic similarity with all other sentences is a good candidate for the summary:

$$\text{Sent}_{f2} = \sum_{i=1,j=2}^{n} \frac{\text{sim}\left(\text{sent}_i, \text{sent}_j\right)}{\max\left(\text{sim}\left(\text{sent}_i, \text{sent}_j\right)\right)}. \tag{11}$$

(3) Proper Nouns. A sentence that contains more proper nouns is assumed as salient for inclusion in the summary. The feature score is computed as the number of proper nouns in a review sentence over the length of the review sentence:

$$\text{Sent}_{f3} = \frac{\text{no. of proper nouns in review sentence}}{\text{review sentence length}}. \tag{12}$$

(4) Number of Nouns and Verbs. Another important text feature for finding sentence importance is checking the number of verbs and nouns a sentence contain:

$$\text{Sent}_{f4} = \frac{\text{no. of nouns and verbs in review sentence}}{\text{review sentence length}}. \tag{13}$$

(5) Sentence Similarity to Centroid of Cluster. Sentences that are most similar to cluster centroid are considered to be important for summary generation:

$$\text{Sent}_{f5} = \text{Sim}\,(\text{Review Sentence, Cluster Centroid}). \tag{14}$$

(6) Term Frequency-Inverse Document Frequency (TF-IDF). Sentences with high TF-IDF scores are deemed salient for summary generation [57]. The feature score is computed as the ratio of the sum of weights of all terms in a review sentence over the sentence with a max TF-IDF score in the collection of review sentences:

$$\text{Sent}_{f6} = \text{TF} \times \text{IDF} = \text{TF} \times \log\left(\frac{N}{n_i}\right). \tag{15}$$

3.4.4. Selection of Top-Ranked Sentences from Clusters. In this step, we select sentences with the highest rank score from all clusters on the basis of text features discussed in Section 3.4.3. The text features scores are computed for each sentence in clusters; hence, a vector is constructed using 6 text features to represent each sentence, i.e., $\text{Sent} = [\text{Sent}_{f1}, \text{Sent}_{f2}, \text{Sent}_{f3}, \text{Sent}_{f4}, \text{Sent}_{f5}, \text{Sent}_{f6}]$. Once the features score for each sentence is calculated, then the features score is summed up to obtain the ranking score of each sentence as given in the following equation:

$$\text{Score}\,(\text{Sent}) = \sum_{k=1}^{6} \text{Sent}_{f_i}, \tag{16}$$

where Score (Sent) represents the review sentences score and Sent_{f_i} indicates the sentence features score. Once the sentences score is attained using equation (16), the sentences in each cluster are ranked based on their score, and the top-scored sentences are selected from all clusters. In this study, the number of chosen clusters is 10, so we select ten representative sentences from 10 clusters for final extractive summary generation.

4. Experimental Settings

4.1. Evaluation Data. The proposed approach comprises of two components: the first component is the Naïve Bayes (NB) classifier, which classifies the review documents into positive and negative. The second component is the semantic clustering approach, which performs the task of movie review summarization. In order to evaluate the first component (the NB classifier), we considered document-level and sentence-level classification tasks in the domain of movie reviews.

For the document-level sentiment classification task, we used two publically available movie review datasets. The first one is introduced by Pang and Lee http://www.cs.cornell.edu/people/pabo/movie-review-data/ [53], which is a most widely used polarity dataset of 2000 movie reviews (version 2). It consists of 1000 positive movie reviews and 1000 negative reviews. Each review in the dataset is associated with the binary sentiment polarity label. The second benchmark dataset is constructed by Maas et al. [54], which consists of 50,000 reviews from the IMDB dataset, and each movie is restricted to have no more than 30 reviews. It comprises of movie reviews with their corresponding labels (sentiment polarity). The labelled dataset is evenly divided into 2.5 k training and 2.5 k train sets. Like previous work on polarity classification, this study also assumes high-polarized reviews. The negative reviews in the dataset are scored ≤4 out of 10, while the positive reviews are scored ≥7 out of 10.

We also evaluated the NB classifier on the sentence-level subjectivity classification task. For this task, we used the dataset introduced by Pang and Lee [53], which contains 5000 subjective and 5000 objective sentences taken from movie review summaries and movie plot summaries, respectively. We compared the NB classifier (with variations on bag-of-words features) with the benchmark model for sentiment analysis [54], in terms of classification accuracy on the three evaluation tasks discussed above. The benchmark model used a mix of unsupervised and supervised techniques to learn word vectors for capturing semantic and sentiment information.

The proposed semantic clustering approach for the task of movie review summarization is evaluated on 4 randomly chosen balanced subsets of classified reviews, where each subset roughly contains 100 positive and 100 negative reviews. We asked 2 Ph.D. students working in the area of natural language processing to manually produce summaries for each subset of classified reviews. The performance of the proposed method is compared with state-of-the-art summarization techniques using ROUGE-1 and ROUGE-2 evaluation metrics.

4.2. Experimental Steps. Given the dataset, at first, the preprocessing techniques are applied over the dataset to segment the dataset into sentences, tokenize the sentences into words, and remove the stop words. Word stemming is also performed on the remaining words to stem the words to their root form. Next, document features are extracted using the BoW technique. This study uses NB ML algorithms in order to classify the movie reviews. There are other commonly used supervised machine learning techniques for opinion mining such as SVM and neural network; however, Naïve Bayes is chosen for classification of movie reviews based on performance accuracy.

In order to perform the movie review classification task, the Naïve Bayes classifier is used to classify the movie reviews into positive and negative. For training and testing of NB, we applied the 10-fold cross-validation technique over the three balanced datasets. Two datasets, namely, PL04 and Full IMDB as shown in Table 5, were used for the document sentiment classification task, and the subjectivity dataset was used for the sentence-level subjectivity classification task. In this study, we used stratified 10-fold cross-validation (commonly used for classification problems), in which the folds are chosen in such a way so that each fold contains roughly the same proportion of class labels.

We evaluated the classification accuracy of the NB classifier with different variations on the bag-of-words feature sets and compared the results with the benchmark model [54] for sentiment classification as shown in Table 5. The benchmark model utilizes a mix of unsupervised and supervised techniques to learn word vectors that capture semantic term-document information as well as rich sentiment content. Line 1 in Table 4 shows that accuracy of the NB classifier with only unigrams as features on smaller datasets (PL04 and subjectivity) is superior than resulting accuracy with bigrams. However, Line 2 shows that on the large IMDB dataset, the accuracy of the classifier is boosted with only bigrams as features. Line 3 shows that the accuracy of the classifier is further improved when both unigrams and bigrams were used as a feature set. Line 4 shows that considering unigrams, bigrams, and trigrams as a feature set has improved the classifier accuracy on the large IMDB dataset while slightly fell down on smaller datasets.

Line 5 shows that unigrams frequency weighted with smoothed inverse document frequency (IDF) with cosine normalization has slightly degraded the classifier accuracy on smaller datasets and improved the accuracy on the large IMDB dataset. Line 6 shows that bigrams counts weighted with IDF with cosine normalization enhanced the accuracy on all datasets. Line 7 indicates that combination of unigrams and bigrams feature-count weighted with smoothed IDF with cosine normalization surpassed the benchmark model and all the variations of bag-of-words features in terms of classification accuracy on the subjectivity dataset.

Line 8 indicates that combination of unigrams, bigrams, and trigrams feature-count weighted with smoothed IDF and cosine normalization surpassed the benchmark model and almost all variations of bag of words, in terms of classification accuracy on the large IMDB dataset and subjectivity dataset, and performed inferior than the benchmark model on the smaller PL04 dataset.

Once the classifier classifies the reviews into positive and negative reviews, the proposed approach exploits semantic clustering technique to generate the summary from the classified reviews. The summarization technique represents the classified review sentences through a semantic representation called sentence embedding and then groups semantically similar review sentences into clusters. Finally, the top-ranked review sentences in each cluster are determined based on six text features and are combined to constitute the summary.

Table 5: Movie review classification accuracy on three tasks, PL04 refers to the collection of 2000 movie reviews often used as a benchmark dataset for sentiment classification [53], the Full IMDB dataset is a collection of 50,000 reviews, and the sentence subjectivity dataset is a collection of 1000 movie reviews [53].

	Features	PL04	Full IMDB	Subjectivity
1	Unigrams with NB	81.5	86.66	90.75
2	Bigrams with NB	77.7	88.29	76.03
3	Unigrams + bigrams with NB	82.4	88.91	91.22
4	Unigrams + bigrams + trigrams with NB	80.15	89.22	91.18
5	Unigrams frequency + smoothed IDF + cosine normalization	82.1	87.36	90.7
6	Bigrams frequency + smoothed IDF + cosine normalization	81.15	88.31	76.72
7	Unigrams + bigrams + smoothed IDF + cosine normalization	83.7	89.28	**90.91**
8	Unigrams + bigrams + trigrams + smoothed IDF + cosine normalization	83.15	**89.33**	90.87
9	Benchmark model [54]	**88.90**	88.89	88.13

We briefly discuss about smoothed IDF and cosine normalization or Euclidean L2 norm. In a large movie review corpus, some words (e.g., "the," "a," and "is" in English) carry little meaningful information about the actual contents of the review document. If the direct count of data is given to a classifier, then frequent terms will overshadow the frequencies of fewer yet more interesting terms. Thus, we used TF-IDF transform to reweight the count features into floating point values, which are appropriate for usage by a Bayesian classifier.

TF means term frequency while TF-IDF means term frequency-inverse document frequency: $\mathrm{TF} - \mathrm{IDF}\,(t, d) = \mathrm{TF}\,(t, d) \times \mathrm{IDF}\,(t)$.

We used TfidfTransformer class with its default settings from the scikit-learn library in Python language for computing TF-IDF.

TfidfTransformer (norm = "L2," use_idf = True, Smooth_idf = True).

In order prevent division by zero, smooth IDF is set to true, which means the constant 1 is added to the numerator and denominator of the IDF as if an extra document was seen, which contains every term in the collection exactly once. Smoothed IDF is computed as follows:

$$\mathrm{IDF}\,(t) = \log \frac{1 + n}{1 + \mathrm{DF}\,(t)} + 1, \tag{17}$$

where n is the total number of documents in the document set and DF(t) is the number of documents in the document set that contains the term t.

The resulting TF-IDF matrix is obtained from the product of TF and smoothed IDF. The resulting TF-IDF vectors are then normalized by the Euclidean L2 norm (also called cosine normalization), which is the square root of the sum of the square of each term's TF-IDF weight:

$$V_{\text{norm}} = \frac{v}{\|v\|_2} = \frac{v}{\sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}}, \tag{18}$$

where v_{norm} represents the normalized form of the TF-IDF matrix.

For comparative evaluation, we set up two state-of-the-art graph-based summarization techniques, namely, LexRank [58] and TextRank [59]. The LexRank model represents sentences through a graph and determines their salience based on the notion of eigenvector centrality. The model builds the adjacency matrix (graph representation of sentences) from the connectivity matrix, which is based on intrasentence cosine similarity. The LexRank model is another graph-based ranking algorithm that creates graph representation of sentences and utilizes global information from the whole graph to decide the salience of a vertex (sentence) within a graph. The edge weight is determined from content similarity between sentences. However, our semantic clustering approach utilizes semantic distance between sentences to capture semantically related sentences. This study employs ROUGE-1 and ROUGE-2 evaluation metrics to compare our proposed semantic graph approach with the state-of-the-art graph-based approaches for summarization, in the context of the generic movie review extractive summarization task. Our proposed approach and other models perform the task of multidocument summarization, since they generate summaries from multiple movie reviews (or documents).

The ROUGE evaluation metric has several variations: *ROUGE-N* (N = 1, 2, 3, and 4), *ROUGE-S*, and *ROUGE-L*, etc. However, ROUGE-1 and ROUGE-2 are efficiently applied for the multidocument extractive summarization task [60]. $ROUGE - N$ can be defined [60] as an n-gram recall between a system summary and set of human (reference) summaries and is calculated as follows:

$$\mathrm{ROUGE} - N = \frac{\sum_{S\in\{\text{Reference Summaries}\}}\sum_{\text{gram}_n\in S}\text{Count}_{\text{match}}\,(\text{gram}_n)}{\sum_{S\in\{\text{Reference Summaries}\}}\sum_{\text{gram}_n\in S}\text{Count}\,(\text{gram}_n)}, \tag{19}$$

where n is the length of the n-gram, gram$_n$, and count$_{\text{match}}$ (gram$_n$) is the maximum number of n-grams that simultaneously occur in a system summary and a set of human summaries.

The precision, recall, and F-measure for the system summary (or the candidate summary) are computed as follows:

$$\text{Precision} = \frac{\text{System Summary}\cap\text{Human Summary}}{\text{System Summary}}, \tag{20}$$

$$\text{Recall} = \frac{\text{System Summary}\cap\text{Human Summary}}{\text{Human Summary}}, \tag{21}$$

$$F - \text{Measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{22}$$

TABLE 6: Comparison of the proposed summarization technique with other summarization models based on different measures obtained with **ROUGE-1**.

Techniques	Average precision	Average recall	Average F-measure
Proposed technique	**0.490**	**0.499**	**0.494**
LexRank [58]	0.392	0.399	0.396
TextRank [59]	0.245	0.255	0.250

TABLE 7: Comparison of the proposed summarization technique with other summarization models based on different measures obtained with **ROUGE-2**.

Techniques	Average precision	Average recall	Average F-measure
Proposed technique	**0.401**	**0.403**	**0.402**
LexRank [58]	0.302	0.301	0.305
TextRank [59]	0.135	0.142	0.139

Tables 6 and 7 illustrate the comparative evaluation results of the proposed approach and other summarization models based on ROUGE-1 and ROUGE-2 measures, respectively. These results are achieved on the randomly chosen balanced subset of classified movie reviews as discussed above. For the same subset of movie reviews, we asked 2 Ph.D. students working in area of natural language processing to manually create summaries of 10 sentences.

Referring to the ROUGE-1 results given in Table 6, our proposed clustering technique performs better than other summarization models based on average precision, recall, and F-measure. LexRank produces better summarization results as compared to TextRank.

Similarly, based on ROUGE-2 results given in Table 7, the proposed technique still outperforms other summarization models based on average precision, recall, and F-measure. LexRank also maintained to produce better summarization results than TextRank based on ROUGE-2.

Figures 2 and 3 visualize the summarization results of the proposed approach and other summarization models based on ROUGE-1 and ROUGE-2, respectively.

4.3. Discussion. This section discusses the evaluation results of classification and summarization approaches presented in previous section. First, we discuss the classification approaches for sentiment classification of movie reviews. In this study, we proposed to use the NB classifier with unigrams, bigrams, and trigrams as a feature set for sentiment classification of movie reviews. We evaluated the classification accuracy of the NB classifier with different variations on the bag-of-words feature sets in the context of three datasets that are PL04 (2000 reviews), IMDB dataset (50,000 reviews), and subjectivity dataset (1000 sentences). It can be observed from results given in Table 5 that the accuracy of the NB classifier surpassed the benchmark model on IMDB and subjectivity datasets, when unigrams, bigrams, and trigrams are used as a feature set. However, the accuracy of NB on the PL04 dataset was lower as compared to the benchmark model. Referring to Line 7 in Table 5, when the combination of unigrams, bigrams, and trigrams feature-count is weighted with smoothed IDF with cosine normalization, the classification accuracy of the NB classifier is further improved and surpassed the benchmark model and almost all variations of bag-of-words features on all benchmark datasets except for the subjectivity dataset where the accuracy marginally fell down by 0.31% as compared to

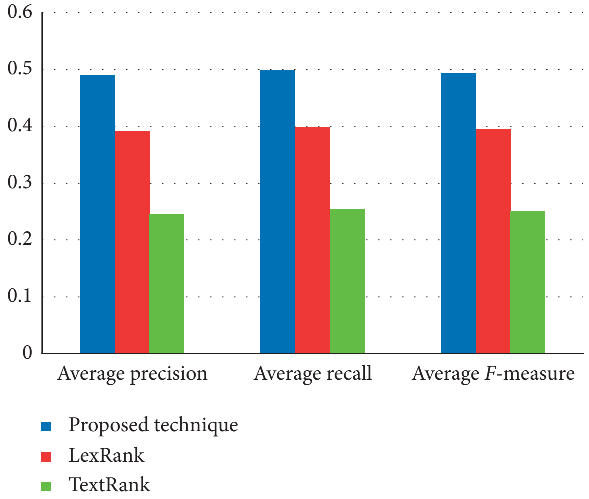

FIGURE 2: Comparison of summarization models in terms of ROUGE-1 measures.

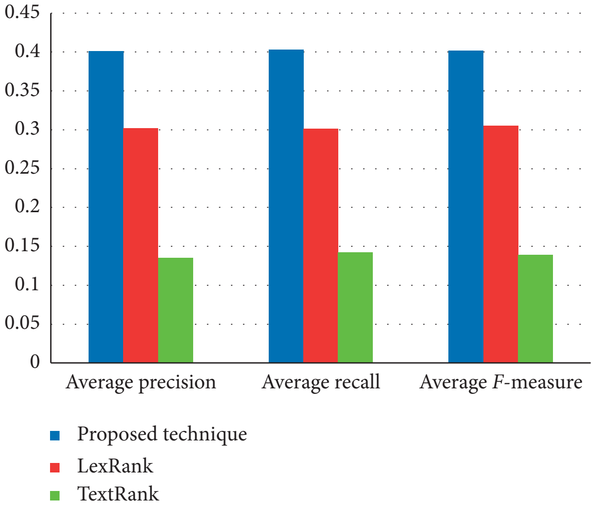

FIGURE 3: Comparison of summarization models in terms of ROUGE-2 measures.

the same feature set with no IDF and cosine normalization in Line 3 of Table 4. It is concluded from the empirical results that the combination of unique and bigrams as features is an effective feature set for the NB classifier as it significantly improved the classification accuracy.

Now, we discuss the summarization results of our proposed semantic clustering approach and other state-of-the-art summarization models in the context of the generic movie summarization task. The proposed approach is compared with other summarization models in terms of average precision, recall, and *F*-measure obtained with ROUGE-1 and ROUGE-2.

It can be examined from ROUGE-1 and ROUGE-2 results in Tables 6 and 7 that the proposed method outperformed the state-of-the-art summarization techniques and achieved improved performance in terms of precision, recall, and *F*-measure. LexRank stood second and TexRank stood third in terms of summarization results.

The proposed approach utilizes the Word2vec model to extract word vectors for all words in sentences. The feature vector for sentences is computed by averaging all the word vectors in each sentence. The approach then applies the semantic K-means (SKM) clustering algorithm to cluster semantically related sentences by making use of Euclidean distance measure on feature vector representation of sentences. The top-scored representative sentences in each cluster are picked based on six text features, and these sentences are combined to form the extractive summary of movie reviews. The experimental outcomes validate that the proposed semantic clustering algorithm embedded with semantic distance significantly enhanced the summarization results.

In order to validate the results, we also carried out a statistical significance tests (*T*-tests) to show the enhancement of our proposed approach with other state-of-the-art summarization models. The paired-sample *T*-test procedure is used to compare the means of two results that represent the same test group and obtained low significance values of 0.039, 0.030, and 0.029 for average precision, recall, and *F*-measure, respectively. The low significance values for the *T*-test (typically less than 0.05) show that there is a significant difference between the results of the proposed approach and other summarization models.

5. Conclusion and Future Work

Since movie review mining and summarization is a challenging task, this study sets a new direction in movie review summarization. We proposed an approach that classifies and summarizes the movie reviews using ML techniques. The proposed approach is general and is applicable to any domain by just providing the training data of that specific domain.

In the context of movie review sentiment classification, we found that Naïve Bayes classifier performed very well as compared to the benchmark method when unigrams, bigrams, and trigrams were used as features. The performance of the classifier was further improved when the frequency of features (unigrams, bigrams, and trigrams) was weighted with IDF.

Finally, we used the semantic clustering approach to summarize the classified movie reviews in order to give an essence of an enormous amount of movie reviews. From the empirical results, we concluded that the proposed approach performs superior than the state-of-the-art summarization models.

In future, we plan to apply deep learning models to generate the abstractive summary from movie reviews. Furthermore, we extend our technique to other domains and examine the effectiveness of the proposed technique.

Data Availability

The data used to support the findings of this study are available from the corresponding website: https://www.imdb.com/.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through the research group no. RG-1438-089.

References

[1] M. Hu and B. Liu, “Mining and summarizing customer reviews,” in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 168–177, Seattle, WA, USA, August 2004.

[2] A. F. Alsaqer and S. Sasi, “Movie review summarization and sentiment analysis using rapidminer,” in *Proceedings of 2017 International Conference on Networks & Advances in Computational Technologies (NetACT)*, pp. 329–335, Thiruvanthapuram, India, July 2017.

[3] L. Zhuang, F. Jing, and X.-Y. Zhu, “Movie review mining and summarization,” in *Proceedings of Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, pp. 43–50, Indianapolis, IN, USA, 2006.

[4] V. B. Raut and D. Londhe, “Survey on opinion mining and summarization of user reviews on web,” *International Journal of Computer Science and Information Technologies*, vol. 5, no. 2, pp. 1026–1030, 2014.

[5] C.-L. Liu, W.-H. Hsaio, 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, 2012.

[6] J. Liu, S. Seneff, and V. Zue, “Harvesting and summarizing user-generated content for advanced speech-based HCI,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 8, pp. 982–992, 2012.

[7] C. Manning, P. Raghavan, and H. Schütze, “Introduction to information retrieval,” *Natural Language Engineering*, vol. 16, pp. 100–103, 2010.

[8] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.

[9] A. Esuli and F. Sebastiani, "Sentiwordnet: a publicly available lexical resource for opinion mining," *Proceedings of LREC*, vol. 6, pp. 417–422, 2006.

[10] A. Nenkova and K. McKeown, "A survey of text summarization techniques," in *Mining Text Data*, pp. 43–76, Springer, Berlin, Germany, 2012.

[11] M.-T. Martín-Valdivia, E. Martínez-Cámara, J.-M. Perea-Ortega, and L. A. Ureña-López, "Sentiment polarity detection in Spanish reviews combining supervised and unsupervised approaches," *Expert Systems with Applications*, vol. 40, no. 10, pp. 3934–3942, 2013.

[12] A. Abdi, N. Idris, R. M. Alguliev, and R. M. Aliguliyev, "Automatic summarization assessment through a combination of semantic and syntactic information for intelligent educational systems," *Information Processing & Management*, vol. 51, no. 4, pp. 340–358, 2015.

[13] C.-Y. Liu, M.-S. Chen, and C.-Y. Tseng, "Incrests: towards real-time incremental short text summarization on comment streams from social network services," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 11, pp. 2986–3000, 2015.

[14] D. K. Ly, K. Sugiyama, Z. Lin, and M.-Y. Kan, "Product review summarization from a deeper perspective," in *Proceedings of Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries*, pp. 311–314, Ottawa, Canada, 2011 June.

[15] P. Mehta, "Survey on movie rating and review summarization in mobile environment," *International Journal of Engineering Research and Technology*, vol. 2, no. 3, 2013.

[16] Y. Sankarasubramaniam, K. Ramanathan, and S. Ghosh, "Text summarization using Wikipedia," *Information Processing & Management*, vol. 50, no. 3, pp. 443–461, 2014.

[17] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artificial Intelligence Review*, vol. 47, no. 1, pp. 1–66, 2017.

[18] V. Gupta and G. S. Lehal, "A survey of text summarization extractive techniques," *Journal of Emerging Technologies in Web Intelligence*, vol. 2, no. 3, pp. 258–268, 2010.

[19] M. Kar, S. Nunes, and C. Ribeiro, "Summarization of changes in dynamic text collections using Latent Dirichlet Allocation model," *Information Processing & Management*, vol. 51, no. 6, pp. 809–833, 2015.

[20] I. Mani, *Advances in Automatic Text Summarization*, MIT Press, Cambridge, MA, USA, 1999.

[21] M. Tsytsarau and T. Palpanas, "Survey on mining subjective data on the web," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 478–514, 2012.

[22] A. Ortigosa, J. M. Martín, and R. M. Carro, "Sentiment analysis in Facebook and its application to e-learning," *Computers in Human Behavior*, vol. 31, pp. 527–541, 2014.

[23] A. Joshi, A. Balamurali, P. Bhattacharyya, and R. Mohanty, "C-Feel-It: a sentiment analyzer for micro-blogs," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations*, pp. 127–132, Portland, OR, USA, June 2011.

[24] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[25] A. Trilla and F. Alias, "Sentence-based sentiment analysis for expressive text-to-speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 2, pp. 223–233, 2013.

[26] A. Dahou, M. A. Elaziz, J. Zhou, and S. Xiong, "Arabic sentiment classification using convolutional neural network and differential evolution algorithm," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 2537689, 16 pages, 2019.

[27] Y. He and D. Zhou, "Self-training from labeled features for sentiment analysis," *Information Processing & Management*, vol. 47, no. 4, pp. 606–616, 2011.

[28] A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," in *Natural Language Processing and Text Mining*, pp. 9–28, Springer, Berlin, Germany, 2007.

[29] M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger, "Pulse: mining customer opinions from free text," in *Lecture Notes in Computer Science*, pp. 121–132, Springer, Berlin, Germany, 2005.

[30] A. Montejo-Ráez, E. Martínez-Cámara, M. T. Martín-Valdivia, and L. A. Ureña-López, "Ranked wordnet graph for sentiment polarity classification in twitter," *Computer Speech & Language*, vol. 28, no. 1, pp. 93–107, 2014.

[31] Y.-H. Hu, Y.-L. Chen, and H.-L. Chou, "Opinion mining from online hotel reviews–a text summarization approach," *Information Processing & Management*, vol. 53, no. 2, pp. 436–449, 2017.

[32] H. Jeong, Y. Ko, and J. Seo, "How to improve text summarization and classification by mutual cooperation on an integrated framework," *Expert Systems with Applications*, vol. 60, pp. 222–233, 2016.

[33] C. N. Silla Jr, C. A. A. Kaestner, and A. A. Freitas, "A nonlinear topic detection method for text summarization using wordnet," in *Proceedings of Workshop of Technology Information Language Human (TIL'2003)*, São Carlos, Brazil, October 2003.

[34] Y.-H. Tseng, Y.-M. Wang, Y.-I. Lin, C.-J. Lin, and D.-W. Juang, "Patent surrogate extraction and evaluation in the context of patent mapping," *Journal of Information Science*, vol. 33, no. 6, pp. 718–736, 2007.

[35] A. J. Trappey, C. V. Trappey, and C.-Y. Wu, "Automatic patent document summarization for collaborative knowledge systems and services," *Journal of Systems Science and Systems Engineering*, vol. 18, no. 1, pp. 71–94, 2009.

[36] D. Vazhenin, S. Ishikawa, and V. Klyuev, "A user-oriented web retrieval summarization tool," in *Proceedings of 2009 Second International Conference on Advances in Human-Oriented and Personalized Mechanisms, Technologies, and Services*, pp. 73–78, Porto, Portugal, September 2009.

[37] J. S. Kallimani, K. G. Srinivasa, and B. E. Reddy, "Summarizing news paper articles: experiments with ontology-based, customized, extractive text summary and word scoring," *Cybernetics and Information Technologies*, vol. 12, no. 2, pp. 34–50, 2012.

[38] J.-P. Qiang, P. Chen, W. Ding, F. Xie, and X. Wu, "Multi-document summarization using closed patterns," *Knowledge-Based Systems*, vol. 99, pp. 28–38, 2016.

[39] X. Han, T. Lv, Z. Hu, X. Wang, and C. Wang, "Text summarization using FrameNet-based semantic graph model," *Scientific Programming*, vol. 2016, Article ID 5130603, 10 pages, 2016.

[40] L. Page, S. Brin, R. Motwani, and T. Winograd, *The Pagerank Citation Ranking: Bringing Order to the Web*, Stanford University, Stanford InfoLab, Stanford, CA, USA, 1999.

[41] I. Mani and E. Bloedorn, "Summarizing similarities and differences among related documents," *Information Retrieval*, vol. 1, pp. 35–67, 1999.

[42] G. Erkan and D. R. Radev, “LexPageRank: prestige in multi-document text summarization,” in *Proceedings of EMNLP 2004*, pp. 365–371, Barcelona, Spain, 2004 June.

[43] R. Mihalcea and P. Tarau, “A language independent algorithm for single and multiple document summarization,” 2005.

[44] X. Wan and J. Yang, “Improved affinity graph based multi-document summarization,” in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pp. 181–184, New York, NY, USA, June 2006.

[45] F. Wei, W. Li, Q. Lu, and Y. He, “A document-sensitive graph model for multi-document summarization,” *Knowledge and Information Systems*, vol. 22, no. 2, pp. 245–259, 2010.

[46] S. S. Ge, Z. Zhang, and H. He, “Weighted graph model based sentence clustering and ranking for document summarization,” in *Proceedings of the 4th International Conference on Interaction Sciences*, pp. 90–95, Busan, Republic of Korea, August 2011.

[47] T.-A. Nguyen-Hoang, K. Nguyen, and Q.-V. Tran, “TSGVi: a graph-based summarization system for Vietnamese documents,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 3, no. 4, pp. 305–313, 2012.

[48] G. Glavaš and J. Šnajder, “Event graphs for information retrieval and multi-document summarization,” *Expert Systems with Applications*, vol. 41, no. 15, pp. 6904–6916, 2014.

[49] A. Mikheev, “Document centered approach to text normalization,” in *Proceedings of Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 136–143, Athens, Greece, July 2000.

[50] C. Buckley, G. Salton, J. Allan, and A. Singhal, *Automatic Query Expansion Using SMART: TREC 3*, in Proceedings of the Third Text Retrieval Conference (TREC-3), NIST Special Publication SP, Gaithersburg, MA, USA, 1995.

[51] M. F. Porter, “An algorithm for suffix stripping,” *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[52] X. Wu, V. Kumar, J. R. Quinlan et al., “Top 10 algorithms in data mining,” *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008.

[53] B. Pang and L. Lee, “A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts,” in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, p. 271, Barcelona, Spain, July 2004.

[54] A. L. Maas, R. E. Daly, P. T. Pham, D. M. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pp. 142–150, Portland, OR, USA, January 2011.

[55] Y. J. Kumar, N. Salim, and B. Raza, “Cross-document structural relationship identification using supervised machine learning,” *Applied Soft Computing*, vol. 12, no. 10, pp. 3124–3131, 2012.

[56] M. A. Fattah and F. Ren, “GA, MR, FFNN, PNN and GMM based models for automatic text summarization,” *Computer Speech & Language*, vol. 23, no. 1, pp. 126–144, 2009.

[57] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Reading*, Addison-Wesley, Boston, MA, USA, 1989.

[58] G. Erkan and D. R. Radev, “Lexrank: graph-based lexical centrality as salience in text summarization,” *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–479, 2004.

[59] R. Mihalcea and P. Tarau, “Textrank: bringing order into text,” in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 404–411, Barcelona, Spain, July 2004.

[60] C.-Y. Lin, “Rouge: a package for automatic evaluation of summaries,” in *Proceedings of Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pp. 74–81, Barcelona, Spain, July 2004.