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 Zaindin⁵

¹Department of Computer Science, Islamia College Peshawar, Peshawar, Pakistan

²Institute of Computing, Kohat University of Science and Technology, Kohat, Pakistan

³Faculty of Engineering and Information Technology, Northern University, Nowshera, Pakistan

⁴Industrial Engineering Department, College of Engineering, King Saud University, P.O. Box 800, Riyadh 11421, Saudi Arabia

⁵Department of Statistics and Operations Research, College of Science, King Saud University, P.O. Box 2455, Riyadh 11451, 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

Copyright © 2020 Atif Khan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Information is exploding on the web at exponential pace, and online movie review over the web is a substantial source of information for online users. However, users write millions of movie reviews on regular basis, and it is not possible for users to condense the reviews. Classification and summarization of reviews is a difficult task in computational linguistics. Hence, an automatic method is demanded to summarize the vast amount of movie reviews, and this method will permit the users to speedily distinguish between positive and negative features of a movie. This work has proposed a classification and summarization method for movie reviews. 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. Next, the Naïve Bayes algorithm is employed to categorize the movie reviews (signified as a feature vector) into negative and positive reviews. For the task of movie review summarization, word2vec model 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 employed to compute the salience score of all review sentences in clusters. Finally, the best-ranked review sentences are picked based on top salience scores to form a summary of movie reviews. Empirical results indicate that the suggested machine learning approach performed better than benchmark summarization approaches.

1. Introduction

With the expansion of Web 2.0 that highlights the involvement of users, many websites such as a movie review website, such as Internet Movie Database (IMDB) and Amazon, persuade its users to write review/feedback for the products they liked or purchased in order to improve the satisfaction and spending experience of customers. Online sellers usually demand their customers to provide reviews/feedback on services or products they have obtained online. The reviews obtained by a product (movie) are rapidly increasing as thousands of customers write reviews of the product, resulting in an overload of information.

This overloaded information makes it a challenging task for a potential customer to glance over each product review to make a quick decision whether or not to buy a product. Meanwhile, it is also appalling for service providers or online suppliers/product makers to record a large number of customer reviews posted for different products/services [1]. In order to overcome the challenge of information overload, an automatic review classification and summarization system is needed [2]. In this research, we are going to focus on the domain of movie review. Assuming the movies, condensing bulk of reviews obtained by a movie can assist the viewers (customers) to quickly glance at the summary and make a prompt decision whether or not to watch a movie.

Furthermore, a movie review summary can support the movie access providers, such as Netflix, to speedily recognize the viewing patterns of their customers (users). This study introduced an automatic method that classifies and summarizes the movie reviews. The method will help a novice and inexperienced user to swiftly recognize the positive and negative features of a certain movie and therefore can promptly decide whether or not select a movie to watch. The job of review mining/classification and summarization (RCS) involves two steps: the former step is related to review classification, which categorizes the movie reviews as negative or positive. The latter step is related to review summarization, which produces a condensed summary from the movie reviews.

Now a days, RCS has received significant consideration in number of areas [3], such as from the feedback/opinions given by people online about a political announcements or news, the government can observe the effect of current events (or policies) on the common public and take timely and proper actions on the basis of available information. On the other side, product reviews gather feedback/opinions from users, and condensation of such user opinions encourages the online suppliers to be knowledgeable about their products.

Review mining/classification [4] categorizes the reviews as negative or positive. There are numerous approaches that classify review document as negative and positive such as approaches using dictionary and machine learning (ML) approaches. Various ML approaches such as support vector machine (SVM) [5], decision trees [6], and neural networks (NNs) [7] are investigated for the problem of text classification and have shown their efficiency in number of domains. NB is a state-of-the-art ML algorithm and has been verified to be very successful in classification problems related to textual data. The classification outcomes taken by NB are extraordinarily good. NB is commonly selected as a baseline classifier for the problems of sentiment analysis and text classification since it gives good accuracy along with efficiency [8]. Consequently, this work decided to use NB for categorization of movie reviews.

Other approaches, on the other hand, rely on the word lexicon/dictionary to find the polarity of the review documents [9] and are therefore not capable of addressing domain-specific orientation.

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. The lexicon-based/unsupervised approaches are restricted to the lexicon words and rely greatly on linguistic resources. On the other hand, supervised ML methods have shown superior performance over unsupervised approaches but are restricted to certain domains. Prior studies show that text summarization methods have utilized in a number of domains and produced successful results [12–16]. The summarization methods create a short version of the original document, present it to various users of the document, and keep the most significant content of the document [17–20].

Numerous users frequently write large number of reviews on review platforms such as IMDB, describing user's attitude to a certain movie. Automatic mining/classification and summarizing these large number reviews is therefore desirable. This study suggests an approach that automatically categorizes and condenses the movie reviews by integrating the supervised ML algorithm with a semantic clustering approach. The working mechanism of the approach is as follows: at first, it employs a bag-of-words (BoW) technique to extract a feature set (unigrams, bigrams, and trigrams) from movie. The next step makes use of Naïve Bayes classifier, which takes feature vector representation of movie reviews as input and predicts the review label as positive or negative. For the task of movie review summarization, the word2vec feature extraction method is used to get features from classified movie review sentences, and then semantic clustering technique is used to cluster semantically related review sentences. Different text features are exploited to calculate the salience score of review sentences in clusters. The final step generates a movie review extractive summary by picking the best-ranked sentences based on high salience scores. This work contributes in the following way:

- (a) To use unigrams, bigrams, and trigrams as a feature set for NB classifier to categorize the movie reviews
- (b) To propose sentence embedding based semantic clustering technique for extractive summary generation from movie reviews
- (c) To evaluate the proposed method with other benchmark methods in perspective of ROUGE-N (1, 2) assessment metrics

The organization of different sections of paper is given as follows. Prior studies related to the area of mining and summarizing reviews are explained in Section 2. In Section 3, the suggested method is illustrated. Empirical results and discussion are given in Section 4. Lastly, the conclusion and future work are outlined in Section 5.

2. Related Work

First, we illustrate the relevant literature for mining of reviews. Review mining is a procedure in which we extract, analyze, and classify the subjective information and determine the sentiment/polarity related to a particular target. Different approaches are suggested by many researchers for the task of review mining [4], such as considering text of a review document $A = [a_1, a_2, a_3, \dots, a_n]$ with a set of classes $B = (\text{positive}, \text{negative})$, the task of review mining is to classify every single review sentence a_i in A [21].

Various review mining techniques such as sentiment lexicon and ML approaches are attempted for mining of reviews in diverse application domains [1, 5, 22, 23]. In [24], the authors presented the difficulties and applications in the field of review mining. Numerous ML algorithms [24–26] are employed to classify opinions in documents. There are two main categories of ML algorithms: (1) supervised ML and (2) unsupervised ML. These algorithms accomplish the polarity classification/opinion mining task by extracting and selecting an appropriate feature set.

A supervised ML technique such as SVM [5] is employed for sentiment/polarity classification task of movie reviews. Decision trees are used by authors in [6] to classify opinion phrases (extracted from restaurant reviews) as high or low informative. However, authors in [27] attempted unsupervised techniques such as unsupervised feature clustering with topic modelling to obtain labelled features. The authors in [28] presented OPINE system, which makes use of relaxation labelling to identify the semantic orientation/polarity of words. A pulse system presented by authors in [29] mines topics from customer feedback with their sentiment/polarity sampled from the car review database. Next, a sentiment lexicon-based approach is discussed for review mining; there are two major categories of lexicon-based approach: (1) corpus-based [11] and (2) dictionary-based [30] approaches. A dictionary-based approach combined with WordNet graph is introduced by authors in [30] for polarity classification of movie reviews. The polarity/sentiment scores for the reviews are calculated using a thesaurus such as Senti-WordNet [9]. The drawback of such approaches is that they cannot tackle domain and context-specific orientation as the same term may have varied sense in different domains. A corpus-based technique is proposed by [11] that utilizes a manually annotated corpus of movie review. Linguistic features, e.g., nouns, adjectives, adverbs, and verbs, are obtained by performing Part-of-Speech tagging on movie reviews. A Senti-WordNet resource is exploited for calculating the polarity/sentiment score of each document in the movie review corpus. The lexicon approaches greatly depends on linguistic resources and are restricted to lexicon words.

Several review summarization techniques are also proposed. Review summarization [4] picks the relevant text from the document and presents it in concise form. The final summary may be a feature-based or general summary covering relevant information of products (movie, camera, and cellular phone) [4]. A summarization approach proposed by the authors in [1] generates a feature-based summary for product (cellular phone and camera) reviews. Word attributes are used, along with synset in WordNet, part-of-speech (POS), and occurrence frequency. The final summary was rearranged on the basis of extracted features. An approach presented by authors in [5] used latent semantic analysis (LSA) for identification of product (movie) features from user reviews. To produce a review summary, product features and opinion words are exploited to select significant review sentences. However, the proposed approach was restricted to movie reviews in Chinese language. Author in [3] demonstrated a multiknowledge approach for summarization of movie review. In order to build a keyword list for recognizing features and opinions, the approach uses WordNet and labelled movie training data. Finally, on the basis of extracted features, summary sentences are reorganized. However, proposed approach may not be capable to detect effective feature-opinion pairs as the semantic relationship between features and opinion words cannot be checked using grammatical relations. However, earlier approaches proposed for summarization of movie reviews are restricted to the generation of product feature summary

while the general summary is not taken into account. Hence, a summarization approach is proposed that integrates the supervised ML algorithm with a clustering approach to constitute a generic movie review summary.

Moreover, an unsupervised ML [31] based text summarization approach has also been introduced for hotel reviews. Different summarization methods have been actively operated in several domains, such as summarization of news articles, patents, and webpages [32, 33]. A text summarization technique is presented by the authors in [34] for generating patents summaries. Proposed approach uses various features such as cue marks and sentence position for finding the sentence importance. The approach in [35] joined ontology tree with term TF-IDF technique to identify keywords in order to retrieve the relevant text of a patent document. Clustering is then exploited to collect important sentences for summary generation. The authors in [36] used query expansion approach to produce a summary from the collection of webpages. A statistical approach was presented by [37] to constitute a summary from news articles. The articles sentences are given scores obtained from many features such as sentence length, news article first sentence, news article title, term frequency, and proper nouns. Finally, top score sentences are selected for summary generation. A pattern-based method is presented by the authors in [38] for summarization of news articles. Each sentence score is determined by addition of weights of the covered patterns. Finally, the summary is formed by choosing sentences iteratively based on the minimum resemblance with sentences that are selected previously and maximum score among all candidate sentences.

In previous few years, several graph-based approaches [39] have attracted more attention and have been successfully applied in area of text summarization. Such methods employ PageRank (PR) algorithm [40] and other variants of PR to determine rank score of graph vertices/nodes, which show passages or sentences.

A connectivity graph-based technique is proposed in [41], which assumes that vertices only hold important content if they are linked to many other vertices/nodes. A Lex-PageRank approach is introduced in [42] which is an Eigen vector centrality-based approach; it makes a connectivity matrix from sentences and exploits the similar algorithm as PR to determine the relevant sentences. A PR-like algorithm is also presented by the authors in [43], which discovers relevant sentences to compose a summary. Another graph-based approach is presented by the authors in [23], which explored subtopic features for multidocuments and incorporated these features into the ranking procedure. A summarization system based on affinity graph [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 graph-based method and discussed how the global information across the documents affects the selection of sentences. A generic multi-document summarization using the weighted graph model is introduced by the authors in [46], which fuses sentence-clustering with sentence-ranking methods. A graph-based method for summarizing several Vietnamese documents is

presented by the authors in [47]. An event graph method is demonstrated by the authors in [48], which constitute an extractive summary of multidocuments. However, it involves the creation of handcrafted extraction rules of argument, which is a tedious task. Our approach is discussed in detail in the following section.

3. Proposed Methodology

In this section, the framework of the suggested work is described as shown in Figure 1. The framework covers four core phases: preprocessing, feature extraction, reviews classification, and extractive summarization of reviews.

3.1. Preprocessing. Data preprocessing in computational linguistics is a vital procedure, which mainly reviews summarization. Preprocessing of the review documents is necessary for the efficient use of the documents before any experiment is carried out. The preprocessing phase includes the following four steps.

3.1.1. Sentence Segmentation. It is the procedure of detecting a boundary within a review document text and divides it into sentences. Generally, different symbols (?) and (!) are used, but we used period (.) to find the sentence boundary [49].

For instance, the following review document contains some text: "I like the characters of this movie. The movie is simply awesome."

After splitting the document, we obtain two sentences.

Segment 1: "I like the characters of this movie."

Segment 2: "The movie is simply awesome."

3.1.2. Tokenization. This step divides the review sentences into distinct tokens (words). Usually, the primary indications such as blank character, tab character, and punctuation symbols such as dot (.) and comma are utilized for tokenizing the review sentences into tokens.

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

3.1.4. Word Stemming. It is salient procedure in the preprocessing phase. It finds the derived words in the document and converts them to their stem in order to catch the similar notion. In this work, Porter's stemming [51] is employed to perform stemming by removing the word suffixes. For instance, the words "plan" "planned" and "planning" will be converted to its stem word "plan," when the stemming algorithm is applied.

Plan, planned, and planning $\Rightarrow$ plan. (1)

3.2. Feature Extraction. An important feature extraction method called bag-of-words (BoW) is performed in this phase, to get feature vectors from movie reviews. Using BoW technique, a vector space model is generated to represent the review document, while each vector space model dimension indicates a feature. This work utilizes unigrams, bigrams, and trigrams as a feature set. Unigram indicates a single word, while bigram denotes a sequence of two words and trigram denotes a sequence of three words. The features extracted from the movie reviews denote all possible unigrams, bigrams, and trigrams, while the feature values indicate occurrences/frequencies of single word, two-word, and three-word subsequences.

Example 1. Assuming the three movie review documents (Rev-doc), and for the reason of ease, we have provided one sentence from the review doc.

Rev-doc1: "He loved that film."

Rev-doc2: "He disliked that film."

Rev-doc3: "Great acting, a good film."

From these review sentences, 7 distinctive words (unigrams) are extracted. As said earlier, unigrams denote the features which in our case are "performance," "good," "great," "disliked," "loved," "film," and "that." The feature vector representation of documents is also called a vector space. The features values shown in Table 1 denote the unigram frequencies.

To achieve the improved accuracy of the sentiment/polarity classification, the proposed work merges a feature representation of unigrams, bigrams, and trigrams in order to represent a review document. Bag of bigrams indicates a two-word subsequence in NLP, for example, "great character," "Awesome movie," and "no excuse." Bigrams such as "good film," "nice work," and "great character" have positive polarity, while bigrams such as "poor performance" and "bad work" have negative polarity, and there are some bigrams such as "is being" that possess neutral polarity. Bag of trigrams indicates a three-word subsequence in NLP; for example, trigram such as "performance good film" has positive polarity but trigram such as "disliked that film" has negative polarity/orientation.

However, the unigram-based BoW method divides a two-word sequence such as "great film" into "great" and "film," and therefore the word "great" is assumed as positive oriented. Bigrams and trigrams also assist to diminish the dimension of vector space. The bag-of-bigrams feature space representation for the reviews in Example 1 is shown in Table 2.

Table 3 shows the bag-of-unigrams and bigrams features representation for the review provided in Example 1. Table 4 shows the bag-of-unigrams, bigrams, and trigrams features representation for the review provided in Example 1.

FIGURE 1: Movie review sentiment classification and summarization.

TABLE 1: Unigrams based feature representation.

Review docs	Acting	Good	Great	Disliked	Loved	Film	That	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: Bigrams based feature representation.

Review docs	Acting good	Good film	Great acting	Disliked that	Loved that	That 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: Combination of unigrams and bigrams features representation.

Review docs	Acting	Acting good	Good	Good film	Loved that	Film	That	That film	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: Combination of unigrams, bigrams, and trigrams features representation.

Review-docs	Acting	Acting good	Acting good film	Loved	Loved that	Loved that film	Film	That	That film	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

3.3. Reviews Classification. This phase employs a ML-based classification algorithm for categorization of user's reviews. The job of review classification is to categorize the customer's reviews as negative and positive. In this work, we have employed the Naïve Bayes (NB) classifier as it is both robust and accurate classifier [52] and gave better results on scalable datasets in comparison to the other benchmark classification algorithms. Furthermore, the NB classifier is very easy to use and therefore has numerous applications in text classification problems [52].

At first, the NB classifier takes as input the feature vector representation of movie reviews along with the labels, to categorize the reviews. Probability/chance of a term given certain category (negative or positive) in the review document is computed based on occurrences a term happened with that category. Here, the term is unigram, bigram, or trigram. In our recently published work [53], we used NB with just unigrams and bigrams as features for review categorization, as shown in Table 2. However, the features employed in this research work are combination of unigrams, bigrams, and trigrams. In order to categorize a new instance of review, the chance/probability of each unigrams (single word), bigrams (2 words), and trigrams in the review given target label (+ve) is computed, and then overall probability/chance of the review given target label (+ve) is determined by taking product of all terms probabilities and the probability of the target class/label (+ve). Likewise, the chance of the review given target label (−ve) is estimated.

The review is assigned a positive class if its chance/probability given target label (+ve) is maximized; else a negative class is given. The mathematical statement of Bayes' theorem is given as follows:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \quad (2)$$

Assume that an unseen review instance document “I like this film” is fed to the trained NB, which will categorize it as either negative or positive. Here, the review instance is a single sentence. At first, a feature representation of unigrams, bigrams, and trigrams is generated from review instance document as given in Table 4. The probability/chance of the instance review given certain category (negative or positive) is determined using the following equation:

$$(\text{Doc}|\text{Class}) = \prod_{i=1}^{|\text{Doc}|} P(W|c_i), \quad (3)$$

whereas the Doc is the review instance document, |Doc| denotes the length of review, and $P(W|c_i)$ is the probability/chance of a term W in a review instance document given certain category (+ve or −ve).

For classification of new instance document “*I like this film*,” $P(w_k|c_i)$ denotes the probability/chance of all terms (unigrams, bigrams, and trigrams) given class c_i (−ve or +ve) in the instance document and is given as follows:

$$P(w_k|c_i) = \frac{n_k + 1}{n + |\text{VOC}|}, \quad (4)$$

where n_k is the frequency/occurrence of the term w_k appeared in positive instances and n is the count of words in positive instances. |VOC| shows the number of distinct unigrams, bigrams, and trigrams in the review instance documents. Equation (5) estimates the probability of the instance review, and review is assigned a category ($C_i = +ve$ or $-ve$) if its probability given that category is maximum:

$$C_{NB} = \arg \max_{C_i \in C} P(C_i) \prod_{w \in \text{words}} P\left(\frac{w}{C_i}\right). \quad (5)$$

3.4. Summarization of Reviews. The purpose of this phase is to condense the categorized reviews (both negative and positive) to form a summary. This phase includes five steps: sentence embedding, clustering of semantically similar sentences, text features extraction, selection of relevant sentences from clusters, and movie summary generation.

3.4.1. Sentence Embedding. This phase aims to split the categorized reviews into sentences and build sentence embeddings from sentences collection. Sentence embeddings are richer representations of text, which preserve information regarding contextual meaning and syntax for the given text and give superior performance in different NLP tasks.

In order to get sentence embedding, we utilized the word2vec model to get word embeddings/features for each word in review sentences. The pretrained word2vec model [54] introduced by Google is employed to learn word embeddings/features for all words in each review sentence. This pretrained model is based on implementation of neural network that exploits continuous BoW to learn distributed feature representations of words. About 100 billion words (taken from Google News dataset) are used to train this model. The length of word embedding/vectors is defined to be 300 features.

Lastly, sentence embeddings are acquired by averaging all word vectors appearing in the word2vec vocabulary; otherwise, the words are ignored. Each review sentence is now represented as numeric vector. Review sentences and its numeric representations are saved and will be used in later phase for choosing relevant sentences for summary.

3.4.2. Semantic K-Means Clustering. The goal of this step is to group semantically similar sentences using the semantic K-means (KM) algorithm. This algorithm is both effective and simple in comparison to other clustering algorithms. It is commonly used in industrial and academic applications and is applicable to data with large number of dimensions. However, the agglomerative hierarchical clustering algorithm is not efficient with scalable data. KM is an unsupervised iterative algorithm that strives to divide the data into K predefined distinctive nonoverlapping sets called clusters where each data point (Pt) goes to only one cluster. The Pts in our work indicate the review sentences (characterized 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 sentence) of corresponding sentences. Hence, it groups semantically similar sentences, which will decrease overlapping review sentences and eventually produce distinct review sentences in the summary. The basic objective of the SKM algorithm is to reduce the sum of semantic distances (errors) between data points (Pts) and their corresponding cluster centroids.

$$j = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j^2\|, \quad (6)$$

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 refers to centroid of j th cluster.

The semantic K-means clustering algorithm is illustrated as follows: (Algorithm 1)

We used the “elbow method” for calculating the optimum number of cluster centroids. For varied values of k (i.e., no. of clusters), the SSE value (sum of squared error) is calculated between Pts and their corresponding cluster’s centroids. In this study, the optimal clusters determined based on “elbow method” are $k=10$.

The summary length depends on the number of clusters that are formed. Thus, the final extractive summary will be produced from high scored 10 representative sentences taken from ten (10) clusters based on diverse text features, as explained in next step.

3.4.3. Extraction of Text Feature. Different text features play a substantial role in the selection of relevant content for summary. Different text features that have been chosen for this study include length of sentence [55], proper noun feature [56], and some other features like TF-IDF and 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 summarization task of reviews 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 summary [56]. Score of each text feature is normalized in the range 0 to 1.

(1) Length of Sentence. This feature is determined as the ratio of number of words in review sentence to the maximum length sentence.

$$\text{Sent}_{f1} = \frac{\text{No. of words in review sentence}}{\text{Max length of Sentence}}. \quad (7)$$

(2) Sentence to Sentence Average Semantic Similarity. This feature computes the similarity of each review sentence Sent_i with other review sentences in the cluster, then summarizes the obtained similarity scores of the review sentences, and divides it by maximum similarity score. We used the word2vec model for generating sentence vectors from sentences as discussed in Section 3.4.1, and cosine metric is

exploited to calculate the semantic similarity (sim) between any given review sentence vectors. Thus, a sentence with maximum average semantic similarity with all other sentences is a good candidate for summary.

$$\text{Sent}_{f2} = \sum_{i=1, j=2}^n \frac{\text{sim}(\text{sent}_i, \text{sent}_j)}{\max(\text{sim}(\text{sent}_i, \text{sent}_j))}. \quad (8)$$

(3) Proper Nouns. A sentence that includes large number proper nouns (PNs) is assumed as salient and needs to be part of summary.

$$\text{Sent}_{f3} = \frac{\text{No. of PNs in review sentence}}{\text{Review sentence length}}. \quad (9)$$

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

$$\text{Sent}_{f3} = \frac{\text{No. of nouns and verbs in review sentence}}{\text{Review sentence length}}. \quad (10)$$

(5) Sentence Similarity to Centroid of Cluster. A sentence that is more similar (closer) to the cluster centroid is considered to be important for summary generation.

$$\text{Sent}_{f4} = \text{Sim}(\text{review sentence}, \text{cluster centroid}). \quad (11)$$

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

$$\text{Sent}_{f6} = \text{TF} \times \text{IDF} = \text{TF} \times \log \frac{N}{n_i}. \quad (12)$$

3.4.4. Selection of Top-Ranked Sentences from Clusters. In this last step, we select sentences with the highest rank score from all cluster on the basis of textual features demonstrated 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 review sentence is obtained, then the sum of features score is taken to obtain the score of each review sentence as defined in the following equation:

$$\text{Score}(\text{Sent}) = \sum_{k=1}^6 \text{Sent}_{f_i}, \quad (13)$$

where $\text{Score}(\text{Sent})$ stands for the review sentence score and Sent_{f_i} shows the sentence features score. As equation (13) determines the sentence score, then the sentences in each cluster are arranged based on their scores, and the high

- (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 summation of squared distance between data points (Pts) and all the other cluster centroids
- (4) Allot each data point (Pt) to the nearby cluster centroid
- (5) Recompute centroids of the new clusters by getting mean of the all data points belonging to each cluster
- (6) Repeat steps 3 to 5 until centroids in new clusters do not change

ALGORITHM 1: Pseudocode of the semantic K-means clustering algorithm.

scored sentences are selected from each cluster. Since this work has chosen 10 optimal clusters, ten relevant sentences are taken from 10 different clusters to compose the extractive summary.

4. Experimental Settings

4.1. Evaluation Data. Our approach consists of two modules. The first module utilizes NB classifier to categorize the review as positive or negative. The second module is semantic clustering approach, which aims to produce summary from movie reviews. In order to assess the first module (NB classifier), we set up two classification tasks (sentence level and document level) in the area of movie reviews.

We utilized two public datasets for movie reviews for classification task at document level. The first dataset is provided by Pang and Lee (<http://www.cs.cornell.edu/people/pabo/movie-review-data/>) [58], which consists of 2000 movie reviews (ver. 2). Both +ve and -ve reviews are same in number. The second review labelled dataset developed by Andrew [54] contains 50,000 reviews taken from the IMDB dataset. This dataset is evenly distributed, i.e., 25000 train sets and 25000 test sets. In order to assess the NB classifier for classification problem at sentence level, we used Pang and Lee dataset [58], which consists of same number (500) of subjective and objective sentences. We measured the effectiveness of NB classifier with feature set (unigrams, bigrams, and trigrams) against other variants including, unigrams and bigrams [53] and the benchmark model [54], with respect to classification accuracy on three different assessment tasks. The benchmark model fuses unsupervised and supervised learning approaches to learn word embeddings/vectors to capture semantic and sentiment information from document.

4.2. Experimental Steps. At first, we cleansed the movie reviews by applying the preprocessing techniques. As discussed earlier, our approach mainly consists of two modules; the former performs the classification job and the latter performs review summarization. In order to accomplish the classification task of the reviews, we used stratified 10-fold cross validation (SCV) technique to train and test NB classifier on three different review datasets. SVC decides the folds in such a manner that each fold approximately includes same proportion of target labels. For classification task at document level, we used two data collections, PL04 and Full IMDB as depicted in

Table 5. However, we used subjectivity data collection for subjectivity classification job at sentence level.

We assessed the accuracy of NB classifier with varied feature sets and the accuracy of benchmark model [54] for the job of sentiment classification as given in Table 5. The benchmark model fuses supervised and unsupervised learning approaches to learn word embeddings/vectors in order to catch both semantic and sentiment information.

Referring to Table 4, Line 1 describes that the NB algorithm with unigrams features on small scale review datasets (Subjectivity & PL04) performed better than NB with bigrams features in terms of accuracy. However, Line 2 demonstrates that NB with only bigrams features gave highest accuracy on huge IMDB dataset. Referring to our recently published work [53] in Line 3 indicates that fusing of unigrams and bigrams features has enhanced the classification accuracy of NB. Line 4 indicates that the accuracy of NB classifier is further boosted on IMDB dataset when unigrams, bigrams, and trigrams features are combined, but it slightly dropped down on other smaller datasets.

Line 5 illustrates that unigrams features counts when weighted with smoothed IDF and applied cosine normalization, it is observed that classifier accuracy slightly degraded on smaller datasets while the classifier accuracy boosted on IMDB dataset. Line 6 indicates that bigrams feature counts after weighting with IDF and cosine normalization improved the accuracy on all data collections. Referring to our recently published work [53], Line 7 illustrates that when unigrams and bigrams features are combined and their occurrences are weighted with smoothed IDF and applied cosine normalization, then NB with such feature set [53] exceeded the benchmark model and other models with varied features with respect to classification accuracy on subjectivity data collection.

Line 8 illustrates that when unigrams, bigrams, and trigrams features are combined and their occurrences are weighted with smoothed IDF and applied cosine normalization, then NB with this feature set exceeded the benchmark model and other models with varied features with respect to classification accuracy on large IMDB and subjectivity data collections; however, NB showed poor performance as compared with the benchmark model on PL04 data collection.

Once the NB classifier categorizes the reviews as positive or negative, the suggested approach uses a clustering technique to create a summary from the categorized reviews. At first, sentence embedding step uses the word2vec model to extract semantic representation from the categorized

TABLE 5: Movie reviews classification accuracy on three tasks: PL04 dataset contains 2,000 movie reviews and is used as standard data collection for the classification problems in sentiment analysis [58], Full IMDB data collection includes 50,000 reviews, and subjectivity data collection comprises of 1000 review sentences [58].

	NB with features	PL04	Full IMDB	Subjectivity
1	Unigrams features	81.5	86.66	90.75
2	Bigrams features	77.7	88.29	76.03
3	Unigrams + bigrams feature set [53]	82.4	88.91	91.22
4	Unigrams + bigrams + trigrams feature set	80.15	89.22	91.18
5	Unigrams occurrence + smooth IDF + cosine norm	82.1	87.36	90.7
6	Bigrams occurrence + smooth IDF + cosine norm	81.15	88.31	76.72
7	Unigrams + bigrams + smooth IDF + cosine norm [53]	83.7	89.28	90.91
8	Unigrams + bigrams + trigrams + smooth IDF + cosine norm	83.15	89.33	90.87
9	Benchmark model [54]	88.90	88.89	88.13

review sentences, and then the clustering approach is utilized to group semantically similar review sentences into clusters. Finally, the high ranked review sentences in each cluster are determined based on six text features and are combined to constitute the summary.

We briefly discuss about smoothed IDF and cosine normalization or Euclidean L2 norm. In a large movie review corpus, several words such as “a,” “this,” and “and” carry slight significant information about the actual text of the review document. If a classifier is given a direct data count, then frequent terms will overshadow the more

interesting terms having lesser frequencies. Thus, we used TF-IDF transform to reweight the count features that are appropriate for use by a Bayesian classifier.

TF indicates term-frequency, and TF-IDF shows term-frequency multiplied by inverse document-frequency:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t). \quad (14)$$

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

$$\text{TfidfTransformer}(\text{normalization} = \text{L2}, \text{used-idf} = \text{True}, \text{Smoothed-idf} = \text{True}). \quad (15)$$

In order prevent division by zero, smooth IDF is set to true, which means the constant 1 is added up to both numerator (num) and denominator (den) of IDF as if an additional document was appeared, which contains every term (token) in the corpus exactly once. Smoothed IDF is computed as follows:

$$\text{IDF}(t) = \log \frac{1 + n}{1 + \text{DF}(t)} + 1, \quad (16)$$

where n is the number of documents in the document collection and $\text{DF}(t)$ is the frequency of documents in the document collection that contain the term t .

The resulting TF-IDF matrix is obtained from the product of TF and smoothed IDF. The tf-idf matrix is 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 tfidf weight.

$$V_{\text{norm}} = \frac{v}{\|v\|_2} = \frac{v}{\sqrt{v_1^2 + v_2^2 + \dots + v_n^2}}, \quad (17)$$

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

In order to assess our approach, we establish two standard graph-based summarization approaches called LexRank [59] and TextRank [60]. The LexRank method makes a graph from review sentences and uses eigenvector centrality to compute the relevance score of each review

sentence. The method first makes a connectivity matrix, which indicates the intrareview sentence cosine similarity score and then constructs adjacency matrix based on connectivity matrix. The TexRank method, on the other hand, makes a graph representation of review sentences and decides the importance of review sentence by utilizing global information from the graph. The edge weight in both graph methods is computed from content similarity of the given review sentences. However, our semantic clustering approach makes use of semantic distance between review sentences to capture semantically related review sentences. In our recent work [53], we extract semantic representation of review sentences and then find pairwise semantic similarities between review sentences in to build a semantic matrix. An undirected weighted graph is constructed from the matrix in a manner that the nodes indicate review sentences, whereas the edges of graph designate semantic similarity weight. The modified ranking algorithm is then applied to find the relevance score for each sentence in the graph. Lastly, the highest scored review sentences (graph nodes) are picked to constitute the movie review summary.

In order to assess our approach and other standard graph methods for generic review summarization task, we used ROUGE-N (1, 2) assessment measures. The evaluation task is multidocument summarization since summaries are produced from many movie reviews.

There exists a few variations of ROGUE such as ROGUE-N ($N=1, 2, 3$, and 4), ROGUE-L, and ROGUE-SU.

TABLE 6: Summarization approaches assessed with ROUGE-1 measures.

Approaches	Average precision	Average recall	Average F -measure
Proposed approach	0.490	0.499	0.494
Recent work [53]	0.48485	0.47925	0.482
LexRank approach [59]	0.392	0.399	0.396
TextRank approach [60]	0.245	0.255	0.250

However, ROUGE-(1-2) are successfully tried for summarization task dealing with multidocuments [61]. ROUGE-N is illustrated [61] as an intersection of n -grams between

system (machine) summary and human summaries and is given as follows:

$$\text{ROUGE} - N = \frac{\sum_{S \in \{\text{Reference Summaries}\}} \sum_{\text{gram}_n \in S} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \{\text{Reference Summaries}\}} \sum_{\text{gram}_n \in S} \text{Count}(\text{gram}_n)}, \quad (18)$$

where $\text{count}_{\text{match}}(\text{gram}_n)$ is the total n -grams that appear at the same time in a system (machine) summary and human written summaries.

Other metrics such as precision, recall, and F -measure for machine (system) summary are defined as follows:

$$\begin{aligned} \text{precision} &= \frac{\text{system summary} \cap \text{human summary}}{\text{system summary}}, \\ \text{recall} &= \frac{\text{system summary} \cap \text{human summary}}{\text{human summary}}, \\ F - \text{measure} &= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \end{aligned} \quad (19)$$

Tables 6 and 7 demonstrate the assessment outcomes of our work and other summarization approaches based on ROUGE-N ($N=1, 2$) measures. The empirical results are obtained on randomly selected even subset of movie reviews as illustrated above. We hired 2 Ph.D. researchers specialized in NLP and requested them to write summaries of 10 sentences for the each subset of reviews.

Given the ROUGE-1 results shown in Table 6, our suggested clustering method surpassed the other summarization methods in terms of mean precision, F -measure, and recall. On the other side, the LexRank method gave superior summarization results as compared with the TextRank method.

Similarly, given the ROUGE-2 results shown in Table 7, the suggested clustering approach still gave excellent results in comparison to other summarization approaches based on mean precision, F -measure, and recall. The LexRank method also remained consistent to give better results than TextRank in terms ROUGE-2 measures.

The summarization results for the proposed approach and other summarization approaches obtained using ROUGE-N ($N=1, 2$) are visualized in Figures 2 and 3, respectively.

4.3. Discussion. This section demonstrates the assessment results of review classification and summarization approaches given in the previous section. In this work, we

suggested to employ NB classifier with feature set (combinations of unigrams, bigrams, and trigrams) for polarity classification of reviews. We assessed accuracy of NB classifier with varied feature sets in the context of three standard movie review datasets (PL04, IMDB dataset, and subjectivity dataset). Based on the empirical results shown in Table 5, we concluded that the most efficient feature set for NB classifier is the combination of unigrams, bigrams, and trigrams since such feature set drastically boosted the classification accuracy. For the task of review summarization, the suggested approach and other standard graph methods are assessed in terms of mean precision, recall, and F -measure attained with ROUGE-N (1, 2) metrics.

It can be examined from ROUGE-1 and ROUGE-2 results in Tables 6 and 7 that the proposed method outperformed the standard graph-based summarization methods and got superior results in terms of precision, recall, and F -measure. In terms of summarization results, the LexRank method came on second position while the TextRank method came on third position. On the other hand, the proposed method has outperformed our recent work for review summarization in terms of ROUGE-1 measures, but it has slightly degraded in terms of ROUGE-2 measures compared with our recent work. The proposed approach makes use of the pretrained word2vec model to get word embeddings/vectors for all words in review sentences. The feature vector space for review sentences is determined by taking mean of all word embeddings/vectors in each review 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 high scored representative sentences in each cluster are picked based on six text features, and these sentences are combined to make a summary of movie reviews. The experimental outcomes validate that the proposed semantic clustering algorithm embedded with semantic distance significantly enhanced the summarization results.

The paired-samples T -test procedure is applied to validate the summarization results. It is used to compare the average of two results obtained for the same test set and attained low significance values of 0.037, 0.029, and 0.027 for

TABLE 7: Summarization approaches assessed with ROUGE-2 measures.

Approaches	Average precision	Average recall	Average F -measure
Proposed approach	0.401	0.403	0.402
Recent work [53]	0.444	0.439	0.441
LexRank approach [59]	0.302	0.301	0.305
TextRank approach [60]	0.135	0.142	0.139

FIGURE 2: Summarization approaches assessed based on ROUGE-1 measures.

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

average precision, recall, and F -measure, respectively. The results of the proposed approach and other graph approaches are statistically significant as we obtained low significance values of 0.04 for the T -test.

5. Conclusion and Future Work

This study attempted to propose an approach that combines both supervised and unsupervised learning to categorize the

reviews as positive or negative and then summarize (condense) the categorized reviews in the domain of movie review. The suggested approach is quite generic and can apply to any certain domain provided that the training data for that domain are available.

For task of review polarity classification, we discovered that Naïve Bayes classifier with feature set of unigrams, bigrams and, trigrams performed extraordinary well as compared with the benchmark method. We further noticed that the NB classifier accuracy was further enhanced when count of features (unigrams, bigrams, and trigrams) was weighted with IDF.

Finally, we applied the semantic clustering approach to summarize (condense) the categorized reviews in order to provide a gist of large number of movie reviews. It has been verified from the experimental findings that the proposed approach outperformed the standard graph-based summarization methods.

We are planning to use deep learning models such as long short-term memory (LSTM) to produce abstracts from movie reviews. In addition, we spread our approach to other fields/domains and study the efficacy 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 that they have 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, Thiruvananthapuram, India, July 2017.
- [3] L. Zhuang, F. Jing, and X.-Y. Zhu, "Movie review mining and summarization," in *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*, 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 the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries*, pp. 311–314, Ottawa, Canada, June 2011.
 - [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 non-linear 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. C. 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 the 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. Eswara 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, CA, USA, 1999.
 - [41] I. Mani and E. Bloedorn, “Summarizing similarities and differences among related documents,” *Information Retrieval*, vol. 1, no. 1-2, 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, June 2004.
 - [43] R. Mihalcea and P. Tarau, *A Language Independent Algorithm for Single and Multiple Document Summarization*, Springer, Berlin, Germany, 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*, 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 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, and J. R. Quinlan, “Top 10 algorithms in data mining,” *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008.
 - [53] A. Khan, M. A. Gul, and M. Zareei, “Movie review summarization using supervised learning and graph-based ranking algorithm,” *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 7526580, 14 pages, 2020.
 - [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*, 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 using supervised machine learning,” *Applied Soft Computing*, vol. 12, no. 10, pp. 3124–3131, 2012.
 - [56] M. A. Fattah and F. Ren, “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] 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.
 - [59] 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.
 - [60] 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.
 - [61] C.-Y. Lin, “Rouge: a package for automatic evaluation of summaries,” in *Proceedings of the Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pp. 74–81, Barcelona, Spain, July 2004.