*Computers, Materials & Continua*
DOI:10.32604/cmc.2021.016727
***Article***

Tech Science Press

# Suggestion Mining from Opinionated Text of Big Social Media Data

**Youseef Alotaibi[1,*], Muhammad Noman Malik[2], Huma Hayat Khan[3], Anab Batool[2], Saif ul Islam[4], Abdulmajeed Alsufyani[5] and Saleh Alghamdi[6]**

[1]Department of Computer Science, College of Computer and Information Systems, Umm Al-Qura University, Saudi Arabia
[2]Department of Computer Science, Faculty of Engineering and Computer Sciences, National University of Modern Languages, Islamabad, Pakistan
[3]Department of Software Engineering, Faculty of Engineering and Computer Sciences, National University of Modern Languages, Islamabad, Pakistan
[4]Department of Computer Sciences, Institute of Space Technology, Islamabad, Pakistan
[5]Department of Computer Science, College of Computers and Information Technology, Taif University, Taif, 21944, Saudi Arabia
[6]Department of Information Technology, College of Computers and Information Technology, Taif University, Taif, Saudi Arabia
[*]Corresponding Author: Youseef Alotaibi. Email: yaotaibi@uqu.edu.sa
Received: 09 January 2021; Accepted: 15 March 2021

**Abstract:** Social media data are rapidly increasing and constitute a source of user opinions and tips on a wide range of products and services. The increasing availability of such big data on biased reviews and blogs creates challenges for customers and businesses in reviewing all content in their decision-making process. To overcome this challenge, extracting suggestions from opinionated text is a possible solution. In this study, the characteristics of suggestions are analyzed and a suggestion mining extraction process is presented for classifying suggestive sentences from online customers' reviews. A classification using a word-embedding approach is used via the XGBoost classifier. The two datasets used in this experiment relate to online hotel reviews and Microsoft Windows App Studio discussion reviews. F1, precision, recall, and accuracy scores are calculated. The results demonstrated that the XGBoost classifier outperforms—with an accuracy of more than 80%. Moreover, the results revealed that suggestion keywords and phrases are the predominant features for suggestion extraction. Thus, this study contributes to knowledge and practice by comparing feature extraction classifiers and identifying XGBoost as a better suggestion mining process for identifying online reviews.

**Keywords:** Suggestion mining; word embedding; Naïve Bayes; random forest; XGBoost; dataset

## 1 Introduction

Online texts of reviews and blogs are continuously increasing and constitute public opinions regarding products, services, individuals, organizations, or events. The expression of sentences in available online text can be related to sentiments and emotions [1], and generally referred to as opinions, recommendations, instructions, advice, and tips for others regarding any entity. Such opinions can be collectively termed as suggestions [2].

Studies have described suggestion mining as sentence classification, which is based on predicting opinionated text into the binary forms of suggestions and non-suggestions [3–5]. The literature has generally defined suggestion mining as the "extraction of suggestions from the opinionated text, where suggestions keyword denotes the recommendation, advice, and tips" [3]. These suggestions are valuable to customers and business organizations [6] if extracted comprehensively from opinionated text [7]. Suggestions must be extracted using computers because online reviews, blogs, and forums that contain suggestions are continuously increasing, resulting in large datasets [6]. The high data volume makes it challenging to extract suggestions [8]; therefore, automatic suggestion mining has emerged as a new research area [1].

Suggestion mining is an approach that largely emphasizes analyzing and identifying sentences to explore explicitly the suggestions they contain [2]. Identifying opinions about products and services that are discussed on social media is useful to organizations' management and to consumers. These opinions offer suggestions that assist management in deciding on improvements to products and services [6]. In addition, consumers can benefit from these suggestions by using them to decide whether to buy a particular product or service. Such increased opinionated text has constituted the major dataset in the majority of recent research [9–11]. Some studies have focused on product reviews [4,5,12] related to tourism (e.g., hotel service) [10,11] and on social media data (e.g., Twitter) [13].

Moreover, several challenges in suggestion mining approaches relate to analyzing the sentiments of the sentence, identifying the relationship between suggestions, and selecting annotators for supervised and unsupervised learning [14]. Suggestion mining is a recent research area, and thus, studies on extracting suggestions involving different classifiers and algorithms are relatively limited [15]. Studies related to support vector machines (SVMs) [16], long short-term memory (LSTM) [8], hidden Markov [17], Random Forest [18,19], Naïve Bayes [20,21], and other areas [22] have also contributed to improvements in suggestion mining.

Thus, the present study is among the few such studies that are aimed at improving suggestion mining results by experimenting with the word-embedding approach and the XGBoost classifier. This study is aimed to capture context and similarity with other words. Furthermore this study contributes by improving the classifier performance through the XGBoost classifier, as compared with Naïve Bayes and Random Forest. Moreover, variations in the proposed suggestion mining extraction process casting improved suggestion mining results. The remainder of the paper is structured as follows. Section 2 describes related work regarding suggestion mining and Section 3 explains the proposed suggestion mining extraction process. Section 4 describes the detailed experiment results and Section 5 presents a results analysis and discussion. Last, Section 6 describes the conclusion and future work.

## 2 Related Works

Prior approaches to suggestion mining focused on rules for linguistic and supervised machine learning through features that are manually identified. The key supervised learning algorithms

used in these studies were the hidden Markov model, the conditional random field (CRF) [9], factorization machines [4], and SVM [2]. Further, these studies used training datasets that had less than 8,000 sentences and an exceedingly imbalanced distribution of classes. Importantly, only a few of these datasets are publicly available. All these datasets contain suggestion class in the minority, and the ratio ranges from 8% to 27% of the entire dataset's sentences.

"Suggestion" can be defined in two ways. First, a generic definition [11,12] is that "a sentence made by a person, usually as a suggestion or an action guide and/or conduct relayed in a particular context." Second, an application-specific definition defines suggestion as "sentences where the commenter wishes for a change in an existing product or service" [15]. Although the generic definition is applied to all domains, the existing research has recorded evaluating suggestion mining on a solitary domain.

Various studies [23,24] have performed mining on weblogs and forums of what they denote as sentences that reveal advice. This mining is performed using learning methods by. Recently, neural networks and learning algorithms have been utilized for suggestion mining [13]. Tao et al. [13] used pretrained word insertion with a dataset that was related to gold-standard training. In addition, diverse classifiers were compared. These classifiers included manually expressed guidelines and SVM (with a diversity of manually reported features related to lexical, syntactic, and sentiment analysis), convolutional neural networks and LSTM networks.

Similarly, the authors in the study conducted in 2021 [4] engaged supervised learning and achieved suggestion detection on "tweets." These suggestions are regarded the phone that was launched by Microsoft. Zucco et al. [14] did not define the suggestions in their work; rather, they reported the objectives of the collection of suggestions, which was to progress and improve the quality and functionality of the product, organization, and service. The authors in [25] delivered an algorithm—"GloVE"—to train word embedding to the additional algorithms that highly perform on several benchmark tasks and datasets. The GloVE algorithm has outperformed various other algorithms, such as skip-grams and the continuous bag of words, which are variations of the "word2vec" model. Therefore, it is a strong base to use pretrained GloVE embeddings [25] to evaluate the performance of the embedding theory using the present study's dataset.

Training task-base embedding is verified as beneficial for tasks regarding short-text classification (e.g., sentiment analysis). In this regard, the authors in [26] reported the trained sentiment-related word embedding by using supervised learning on a large dataset regarding Twitter sentiments, which were characterized through the emotions displayed in the tweets. Recently, studies have focused on suggestion mining in regard to the problems involved in classifying the sentences and experimented with various statistical classifiers and their features [27]. However, improvement in classifiers in terms of their accuracy and datasets is a serious concern to achieve the desired complete results [28]. Thus, the existing algorithms need to be significantly improved to address this gap because it is an emerging and novel nature of classifying the text. Although existing studies have specified the feature extraction classifiers and their accuracies for suggestion mining, it is concluded that none have used the XGBoost classifier to identify suggestions from customer reviews.

Further, earlier studies have also not compared XGBoost with other classifiers to determine the better approach for identifying the suggestions from reviews. Therefore, this study defines suggestion classification and presents a better suggestion mining extraction process to identify suggestions from social media data regarding online customer reviews of the hotel industry.

The next section presents the proposed suggestion mining extraction process of the opinionated text of online customer reviews.

## 3 Methodology

This study presents a novel approach to the suggestion mining extraction process, which aims to extract useful features to train the classifier for improved results. Fig. 1 illustrates the suggestion mining steps used in this study and Algorithm 1 demonstrates the steps in training a model to predict a review as either a suggestion or non-suggestion.

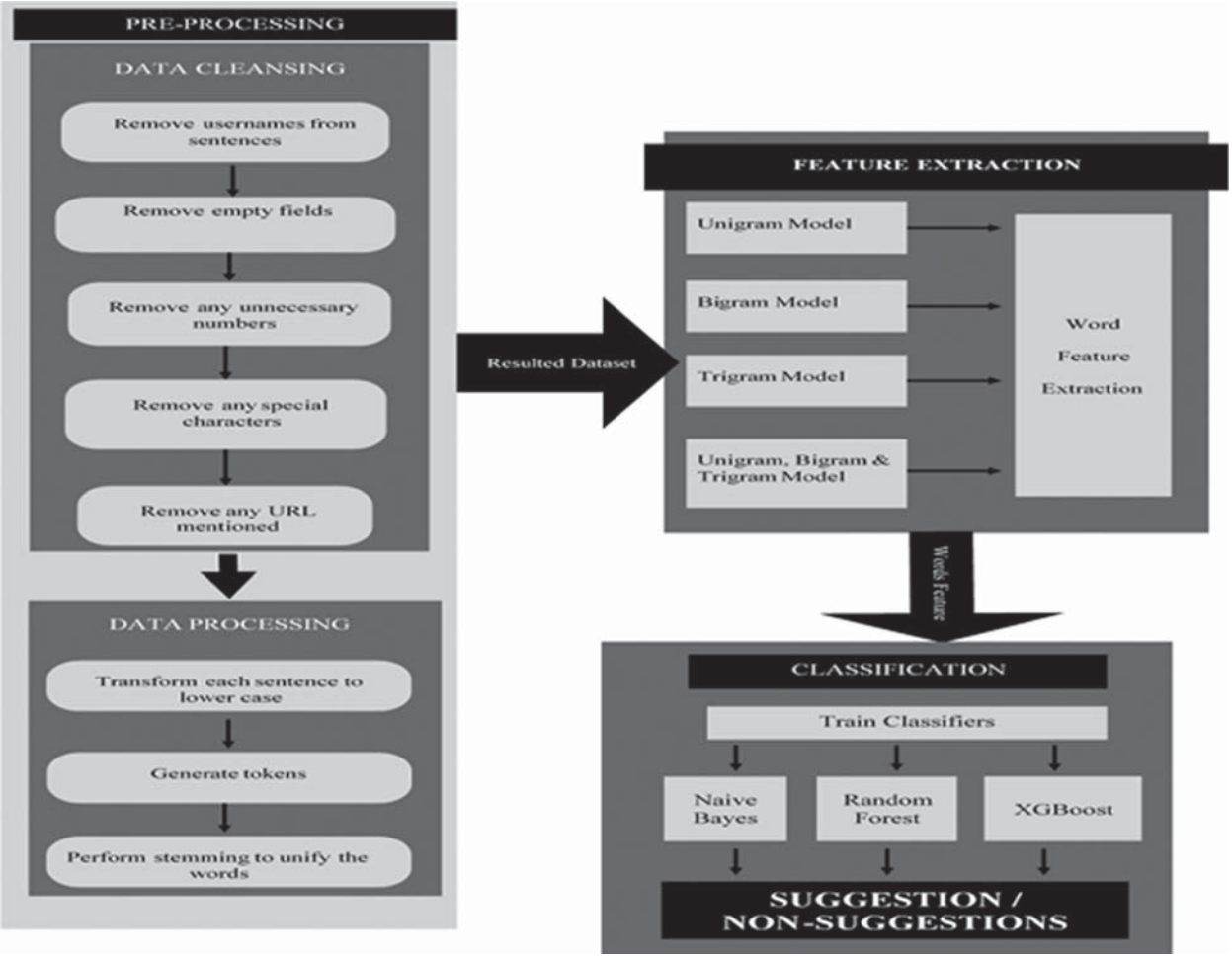



**Figure 1:** Suggestion mining extraction steps

### *3.1 Preprocessing*

First, this study preprocesses the text, which involves two sub-steps—data cleansing and data processing—to clean the data for further processing. Algorithm 2 describes the details of the preprocessing component.

*3.1.1 Data Cleansing*

The primary reason for using the data cleansing approach is to clean unusable data [23]. Generally, online reviews consist of rich information, such as usernames, blank spaces, special characters, and URLs. Removing such unnecessary information can assist in extracting suggestions from the cleaned opinionated text [1]. Therefore, this study performs data cleansing by removing unusable text in suggestion mining. The following information is removed from the dataset, using regular expressions, to ensure a clean dataset ready for further processing.

**Algorithm 1:** Training a model

***Input:*** *Review dataset (reviews, labels) where label = 1 for suggestion and label = 0 for non-suggestion*
***Output:*** *trained model that predicts a review as either a suggestion or non-suggestion*
***for each*** *review* ***in*** *dataset* ***do***
*tokenizedReviews[ ] ← preprocessing(review)*
***end for***
***for each*** *tokenizedReview* ***in*** *dataset* ***do***
*//word features in form of unigram, bigram, trigram, or all*
*wordFeatures[ ] ← featureExtraction(tokenizedReview)*
***end for***
***while*** *accuracy is not improved* ***do***
*trainClassifier*(*wordFeatures*)
***end while***

**Algorithm 2:** Data preprocessing

***Input:*** *Review dataset*
***Output:*** *Tokenized arrays of words*
***for each*** *review in dataset* ***do***
    *dataCleansing (**review**)*
    *split review into array of words*
    ***for each*** *word in review* ***do***
        *lowercase (**word**)*
        *stemming (**word**)*
    ***end for***
***end for***

- usernames in the sentences (e.g., @xyz)
- empty fields
- unnecessary numbers
- special characters used by customers and users in their reviews
- URLs

*3.1.2 Data Processing*

After data cleansing, the following data processing steps are undertaken. First, the tokenization process is applied, which helps decompose the whole sentence stream into portions of words or meaningful elements [23]. These elements are referred to as tokens; for example, words such as "suggest," "recommend," and "please" are usually used to express an opinion. Meaningful features

lead to classification success. In this study, all words in the review were tokenized using a pretrained version of the Punkt Sentence Tokenizer, from the Natural Language Toolkit (NLTK) library. Tab. 1 presents some of the tokens used in this study, which were useful for further data processing. Second, each token is transformed into lower case, to eliminate the repetition of words and terms and to place the entire text in a unique structure. Third, the stemming process is used to unify the words across the entire document and to highlight the uniqueness of words through their stems; for example, "computational," "compute,' and "computing" stem from "compute." During the feature extraction phase, this process helps to avoid duplications. This study used the Porter stemming algorithm to create stems for tokens that were included in the Python NLTK library.

**Table 1:** Sample of preprocessed tokens from two datasets (hotel reviews [HR], Microsoft windows app studio reviews [MSWASR])

| ID | Review | Dataset | Class |
|---|---|---|---|
| 0 | [without, doubt, on, of, the, favorite, hotel…] | HR | 0 |
| 1 | [mistakenly selected, ever, currently…] | MSWASR | 1 |
| 2 | [a, great, place, to, stay, staff, were, friendly…] | HR | 0 |
| 3 | [we, only, stay, here, on, night, but, the, ho…] | HR | 0 |
| 4 | [try other, shadow to distinguish, from content…] | MSWASR | 1 |

### *3.2 Feature Extraction*

Almost all supervised machine learning algorithms can classify data in the form of integer or floating-point vectors [29]. Feature extraction is the process of converting input data into the vector form for use in training classifiers. Machine learning classifiers do not work on data because they attempt to understand and extract data patterns for classification [27,30]. Feature extraction and selection play a primary role in classification accuracy. Using irrelevant features limits the classifiers' performance. The proposed suggestion mining extraction process experimented with four different features.

Reviews are converted into vectors containing Boolean values (i.e., 0 or 1) that correspond to unigrams, bigrams, trigrams, and the uni/bi/trigram combination. The translated review is given to classifiers to extract suggestions and non-suggestions. Tab. 2 depicts the vector size for each review using these feature extraction techniques. Algorithm 3 describes the review vectorization against unigram features. In the unigram feature extraction process, all words from the preprocessed dataset are removed and a bag of unique words is created. Next, a vector is created for each review by assigning 1 if the word exists in the review, and 0 otherwise. It is common for words such as "suggest," "recommend," and "please" to occur in suggestive text.

**Table 2:** Feature extraction techniques and size

| Feature techniques | HR (size) | MSWASR (size) |
|---|---|---|
| Unigram | 2,266 | 2,782 |
| Bigram | 4,146 | 4,144 |
| Trigram | 10,015 | 728 |
| Uni/bi/trigram combination | 7,658 | 7,500 |

Algorithm 4 describes the bigrams feature model. In the bigram feature extraction process, all pairs of words are extracted from the dataset and a bag of bigram is created. For each review, (1, 0) vectors are created, depending on whether the bigram exists. Bigram features are used to cater to suggestive phrases, such as "would like," "would love," and "instead of." Similarly, trigrams phrase examples are "should come with" and "would be nice." Last, a set of unigrams, bigrams, and trigrams are combined and the vector is created. The more meaningful and relevant are the input features, the more will be the classifier's learning and prediction accuracy.

**Algorithm 3:** Unigram modelling algorithm

***Input:*** *Preprocessed reviews, bag of unigrams*
***Output:*** *Unigram features vector*
***for each*** *review* ***in*** *preprocessed reviews* ***do***
  ***for each*** *word* ***in*** *bag of unigrams* ***do***
    ***if*** *word exists* ***in*** *review* ***then***
      *vector[review, word]* = 1
    ***else***
      *vector[review, word]* = 0
    ***end if***
  ***end for***
***end for***

**Algorithm 4:** Bigram modelling algorithm

***Input:*** *Preprocessed reviews, bag of unigrams*
***Output:*** *Unigram features vector*
***for each*** *review* ***in*** *preprocessed reviews* ***do***
  ***for each*** *word* ***in*** *bag of unigrams* ***do***
    ***if*** *word exists* ***in*** *review* ***then***
      *vector[review, word]* = 1
    ***else***
      *vector[review, word]* = 0
    ***end if***
  ***end for***
***end for***

Tab. 3 shows the example association of words using the unigram word feature. The "class label" column shows whether the review is a suggestion (i.e., 1) or non-suggestion (i.e., 0). Further, in this table, 1 refers to the found association whereas 0 denotes that there is no association with the word in the given sentence.

### *3.2.1 Classification*

After the feature extraction process, the reviews are ready for classification. The proposed suggestion mining system used XGBoost classifier and compared the results with the Naïve Bayes and Random Forest algorithms. The XGBoost classifier is a relatively new machine learning algorithm that is based on decision trees and boosting. Nevertheless, it was used in this study because it is highly scalable and provides improved statistics and better results.

**Table 3:** Example association of words using the unigram word feature

| Review ID | Class label | Allow | Add | Suggest | Recommend | Please | Support | Visit | New | Help |
|---|---|---|---|---|---|---|---|---|---|---|
| HR | | | | | | | | | | |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| MSWASR | | | | | | | | | | |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 3 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |

*3.2.2 Experiment*

This study used two datasets of the hotel industry as well as the MSWASR dataset in relation to customer reviews (see Tab. 4). These reviews contain opinionated text with sentences that explicitly express suggestions and non-suggestions. To perform the experiments, a random data subset was created to foresee the overall performance of the algorithms.

**Table 4:** Datasets used in the experiment

| Dataset | Data source | N | S | Purpose |
|---|---|---|---|---|
| Hotel industry | Datafiniti | 34,000 | 10,500 | Extract suggestion |
| | Code source competition | 8,500 | 2,200 | Extract suggestion |
| MSWASR | Github | 9,000 | 2,700 | Extract suggestion |

Tab. 4 consists of five columns. First, "dataset" refers to the nature of the dataset. Second, "data source" describes the source of data in which the dataset was retrieved. Third, "N" refers to the total number of data collection instances from the data source. Fourth, "S" denotes the subset volume of the dataset that was randomly selected for the experiment. Last, "purpose" describes the tasks that need to be executed in this experiment.

This experiment used 42,000 online reviews from the hotel industry datasets and 9,000 reviews from the MSWASR dataset. All datasets comprised opinionated text (e.g., opinion, advice, suggestion, or tips), from which the experiment aimed to extract suggestions. In this experiment, the hotel industry Datafiniti dataset contained 34,000 data instances for training purposes, in which a subset of 10,500 instances was used to test the dataset. Similarly, the hotel industry Code Source Competition dataset contained 8,500 data instances for training purposes, in which a subset of 2,200 instances was used for evaluation. Further, the MSWASR Github dataset contained 9,000 data instances for training purposes, in which a subset of 2,700 instances was used to test the dataset.

As previously specified, the XGBoost classifier was used to classify suggestions. Initially, data cleansing was performed, which was followed by the tokenization process. The word2vec approach was used to generate word vectors, which continuously improve each time the classifier is executed. Therefore, training the classifier with a training set is important because it can assist in building vocabulary for the test set. This study used an online hotel review dataset to train the classifier. Next, the hotel industry testing datasets and MSWASR's total dataset were used to determine the performance of three classifiers—XGBoost, Naïve Bayes, and Random Forest. To obtain the

best performance, the semantic inclusion approach was utilized through a bag of words technique. Therefore, unique words were listed through a bag of words, which generated vectors.

## 4 Results

The performance measurement results were identified based on precision, recall, F1 score, and accuracy. Precision is generally used to measure the proportion of identification as a result of precision; for example, a precision score of 0.80% indicates that its predictions of suggestive reviews are correct 80% of the time. Next, recall generally refers to the completeness of the classifier used in a given dataset. It describes the proportion of actual positives, which means how many suggestions are identified correctly. Further, the F1 score refers to the average precision and recall; it reveals the highest best and worst values towards 0. Last, accuracy demonstrates the ratio of correctly predicted observations and explains the classifiers' ability to predict accurately. Moreover, the average accuracy is calculated to cross-validate the results.

Further, positive and negative scores are categorized into true positive, false positive, true negative, and false negative. True positive means that the output class of review is a found suggestion and that it is correctly classed as a suggestion. Conversely, true negative describes that the output class of review is a non-suggestion and it is correctly classed a non-suggestion. Next, false positive describes that the output class of review is a non-suggestion but it is falsely classed as a suggestion. Conversely, false negative describes that the output class of review is a suggestion but it is falsely classed as a non-suggestion. In addition, the results and analysis are reported based on the unigram, bigram, and trigram models. Moreover, comparative statistics are also reported for all three models.

Tab. 5 reports statistics regarding the performance measurement of feature identification using the unigram model. Tab. 5 comprises two main columns, "hotel industry dataset" and "MSWASR" dataset," which are further split into three sub-columns of classifiers—Naïve Bayes, Random Forest, and XGBoost. Suggestions are reported against each classifier in regard to F1, precision, recall, and accuracy.

**Table 5:** Performance measurement of features using the unigram model

| | Hotel industry dataset | | | MSWASR dataset | | |
|---|---|---|---|---|---|---|
| | Naïve Bayes | Random forest | XGBoost | Naïve Bayes | Random forest | XGBoost |
| F1 | 0.49 | 0.45 | 0.53 | 0.79 | 0.80 | 0.89 |
| Precision | 0.66 | 0.60 | 0.78 | 0.71 | 0.80 | 0.80 |
| Recall | 0.44 | 0.30 | 0.43 | 0.71 | 0.80 | 0.82 |
| Accuracy | 0.70 | 0.64 | 0.84 | 0.81 | 0.82 | 0.87 |
| Average accuracy | 0.69 | 0.62 | 0.82 | 0.78 | 0.80 | 0.81 |

The results for the unigram model reveal the lowest scores for Naïve Bayes for F1, precision, recall, and accuracy. The highest scores are observed for Random Forest and XGBoost classifiers. However, the experimental results indicate that XGBoost scored higher than Random Forest.

Tab. 6 reports statistics regarding the performance measurement of feature identification using the bigram model. Tab. 6 comprises two main columns that represent both datasets, which are further split into sub-columns that represent the three classifiers. Again, suggestions are reported against each classifier in regard to F1, precision, recall, and accuracy.

The results indicate that all scores are higher for the XGBoost classifier. Random Forest outperformed Naïve Bayes in all categories except for precision.

**Table 6:** Performance measurement of features using the bigram model

| | Hotel industry dataset | | | MSWASR dataset | | |
|---|---|---|---|---|---|---|
| | Naïve Bayes | Random forest | XGBoost | Naïve Bayes | Random forest | XGBoost |
| F1 | 0.34 | 0.43 | 0.58 | 0.78 | 0.79 | 0.84 |
| Precision | 0.54 | 0.46 | 0.87 | 0.81 | 0.79 | 0.81 |
| Recall | 0.35 | 0.45 | 0.66 | 0.80 | 0.81 | 0.83 |
| Accuracy | 0.65 | 0.68 | 0.81 | 0.80 | 0.81 | 0.87 |
| Average accuracy | 0.63 | 0.65 | 0.80 | 0.79 | 0.80 | 0.86 |

Tab. 7 reports statistics regarding the performance measurement of feature identification using the trigram model. Tab. 7 comprises two main columns that represent both datasets, which are further split into sub-columns that represent the three classifiers. Suggestions are once again reported against each classifier in regard to F1, precision, recall, and accuracy.

The results demonstrate that Naïve Bayes has the lowest scores for F1, precision, recall, and accuracy. The highest scores are obtained by using the Random Forest and XGBoost classifiers. However, the results indicate that XGBoost scored higher than Random Forest.

**Table 7:** Performance measurement of features using the trigram model

| | Hotel industry dataset | | | MSWASR dataset | | |
|---|---|---|---|---|---|---|
| | Naïve Bayes | Random forest | XGBoost | Naïve Bayes | Random forest | XGBoost |
| F1 | 0.30 | 0.36 | 0.55 | 0.71 | 0.76 | 0.78 |
| Precision | 0.36 | 0.71 | 0.90 | 0.75 | 0.76 | 0.80 |
| Recall | 0.14 | 0.21 | 0.28 | 0.77 | 0.78 | 0.81 |
| Accuracy | 0.67 | 0.68 | 0.81 | 0.77 | 0.78 | 0.79 |
| Average accuracy | 0.65 | 0.65 | 0.80 | 0.77 | 0.78 | 0.83 |

**Table 8:** Performance measurement of features using the uni/bi/trigram combination model

| | Hotel industry dataset | | | MSWASR dataset | | |
|---|---|---|---|---|---|---|
| | Naïve Bayes | Random forest | XGBoost | Naïve Bayes | Random forest | XGBoost |
| F1 | 0.49 | 0.45 | 0.53 | 0.81 | 0.76 | 0.83 |
| Precision | 0.66 | 0.60 | 0.78 | 0.78 | 0.76 | 0.82 |
| Recall | 0.44 | 0.30 | 0.43 | 0.78 | 0.73 | 0.81 |
| Accuracy | 0.70 | 0.64 | 0.84 | 0.79 | 0.77 | 0.82 |
| Average accuracy | 0.69 | 0.62 | 0.82 | 0.79 | 0.73 | 0.87 |

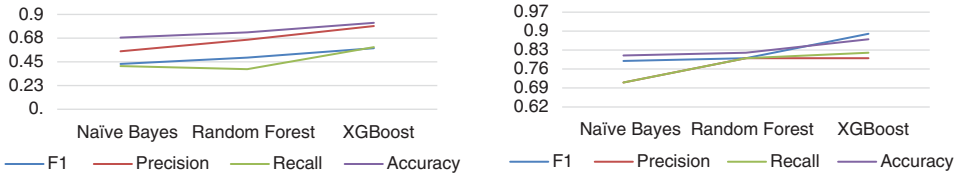
In addition, a combined performance evaluation is presented. Tab. 8 reports the comparative statistics of the unigram, bigram, and trigram models. Tab. 8 comprises two main columns that

represent both datasets, which are further split into sub-columns that represent the three classifiers. Suggestions are reported against each classifier in regard to F1, precision, recall, and accuracy.

When the unigram, bigram, and trigram models are executed together, the results varied regarding Naïve Bayes and Random Forest. Specifically, Random Forest had the lowest scores for F1, precision, recall, and accuracy. Interestingly, Naïve Bayes performed better in this scenario than in the previous scenarios, in which the models were not executed simultaneously. However, XGBoost once again displayed the highest results.

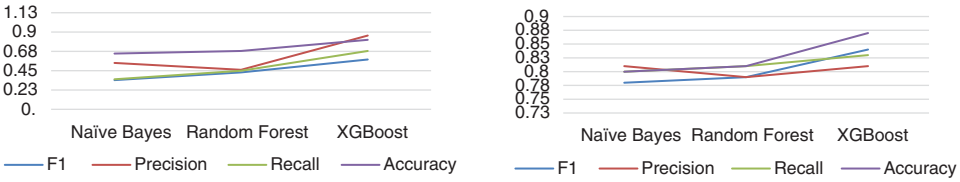
## 5 Discussion

Based on the experiments conducted in this study, it can be observed that the XGBoost classifier has outperformed the other two classifiers. The findings of the experiments are shown in Figs. 2–5, in which the results for the F1, precision, recall, and accuracy of the three classifiers are reported.



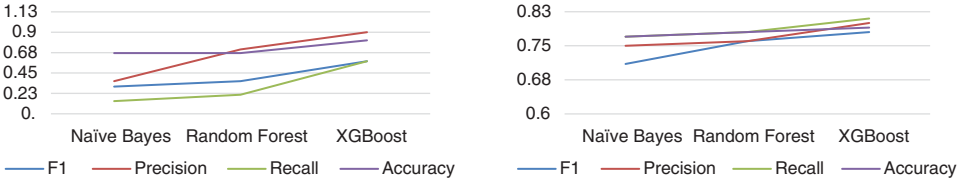

(a) (b)

**Figure 2:** (a) Unigram model scores for the hotel dataset. (b) Unigram model scores the for MSWASR dataset




(a) (b)

**Figure 3:** (a) Bigram model scores for the hotel dataset. (b) Bigram model scores for the MSWASR dataset




(a) (b)

**Figure 4:** (a) Trigram model scores for the hotel dataset. (b) Trigram model scores for the MSWASR dataset

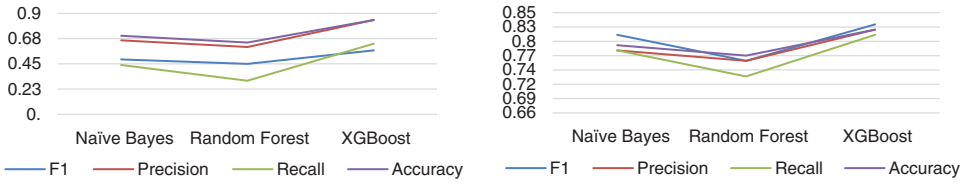



**Figure 5:** (a) Uni/Bi/Trigram model scores for the hotel dataset. (b) Uni/Bi/Trigram model scores for the MSWASR dataset

Further, an accuracy comparison among Naïve Bayes, Random Forest, and XGBoost classifiers was conducted for the hotel industry and MSWASR datasets. The detailed illustration of the accuracy comparison of the three classifiers is shown in Fig. 6

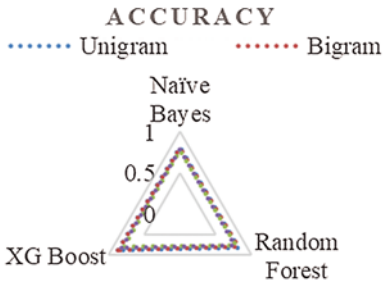



**Figure 6:** Accuracy comparison of Naïve Bayes, random forest, and XGBoost classifiers for the MSWASR dataset

As demonstrated in Fig. 6a, Random Forest performed better than Naïve Bayes in terms of the accuracy of results; however, its results varied among the unigram, bigram, trigram, and the combination of all three models (0.64, 0.68, 0.68, and 0.64, respectively). Interestingly, the results for XGBoost accuracy were better than those for Random Forest in all models (0.84, 0.81, 0.81, and 0.84, respectively). As shown in Fig. 6b, similar results were found for the MSWASR dataset, in which Random Forest outperformed Naïve Bayes in terms of accuracy, but again had varied results among the unigram, bigram, trigram, and the uni/bi/trigram combination (0.82, 0.81, 0.78, and 0.77, respectively). Once again, the results for XGBoost accuracy were better than those for Random Forest in all models (0.87, 0.89, 0.87, and 0.82, respectively). Based on these findings, the XGBoost classifier performed better than the others on the given online review dataset. The Random Forest method is unsustainable because its accuracy values were more distributed than other classifiers.

Further, average accuracies were also analyzed on the given data for the three classifiers on unigram, bigram, trigram, and uni/bi/trigram modelling (see Figs. 7a and 7b). Fig. 7a demonstrates that the lowest average accuracy value (0.63) was found in the bigram of Naïve Bayes and the highest value (0.82) was found in the uni/bi/trigram combination for XGBoost. Likewise, Fig. 7b shows that the lowest average accuracy value (0.77) was found in the trigram of Naïve Bayes and the highest value (0.87) was found in the uni/bi/trigram combination for XGBoost. Although Random Forest achieved better average accuracy results than Naïve Bayes, there is no significant difference. Conversely, the average accuracy scores for XGBoost were stable and

demonstrated fewer distribution scores on the given data in the unigram, bigram, trigram, and uni/bi/trigram combination modelling.

The authors attempted to conduct this study in such a way that the results could be generalized. This became possible by selecting datasets from two different domains (hotel and software industry), in which the various classifiers were executed. The authors have noted that the results would be more generalizable and reliable if they were statistically evaluated through performing non-parametric tests. Because of a lack of any statistical proof, the scope of the analysis is limited.

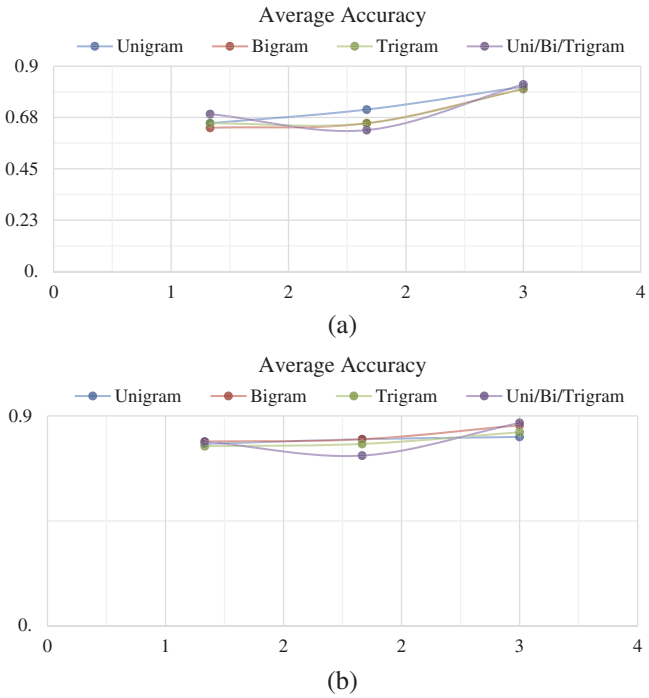



**Figure 7:** (a) Average accuracy comparison of Naïve Bayes, random forest, and XGBoost classifiers for the hotel industry dataset. (b) Average accuracy comparison of Naïve Bayes, random forest, and XGBoost classifiers for the MSWASR dataset

## 6 Conclusion and Future Work

The availability of opinionated text regarding social media data is increasing, which can assist in decision-making if extracted and analyzed carefully. The extracted suggestions, tips, and advice must be carefully analyzed to improve the business and subsequently benefit customers. Recent studies have explored suggestions from online reviews through different classifiers, such as Random Forest and Naïve Bayes. The results of these studies are not mature enough and require further improvements. Therefore, this study proposed a suggestion mining process to improve the results further.

To this end, the authors used various techniques, such as word embedding, bag of words, and word2vec. In addition, XGBoost classifiers were used to train the dataset. The results revealed that the XGBoost classifier outperformed and gave an accuracy of 0.8. Moreover, the results also indicated that suggestion keywords and phrases are the predominant features for suggestion

extraction. This study contributes to the methodological approach for suggestions mining through the XGBoost classifier that can be replicated in other datasets. It contributes toward the state of knowledge and practice by comparing feature extraction classifiers. In addition, it presents XGBoost as a better suggestion mining extraction process for social media data about online customer reviews of the hotel industry.

Nevertheless, the present study has some limitations. Although this study used more than 8,500 online hotel reviews, it is suggested that further results can be found by using a larger dataset. Second, the test dataset was manually analyzed for its suggestions class, which could impart biasness. However, this limitation was overcome by involving other researchers to perform this task. Future research is needed to improve the suggested suggestion mining extraction process using the XGBoost classifier on larger review datasets. These datasets could be related to products, shopping sites, or services. Another promising research area could be extending the results of the XGBoost classifier by providing beyond domain-based training for its versatility.

**Acknowledgement:** We deeply acknowledge Taif University for supporting this study through Taif University Researchers Supporting Project Number (TURSP-2020/115), Taif University, Taif, Saudi Arabia.

**Funding Statement:** This research is funded by Taif University, TURSP-2020/115.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] M. V. Mäntylä, D. Graziotin and M. Kuutila, "The evolution of sentiment analysis—A review of research topics, venues, and top cited papers," *Computer Science Review*, vol. 27, no. 1, pp. 16–32, 2018.

[2] P. Buitelaar, I. D. Wood, S. Negi, M. Arcan, J. P. McCrae *et al.,* "Mixed emotions: An open-source toolbox for multimodal emotion analysis," *IEEE Transactions on Multimedia*, vol. 20, no. 9, pp. 2454–2465, 2018.

[3] V. Grover, R. H. Chiang, T. P. Liang and D. Zhang, "Creating strategic business value from big data analytics: A research framework," *Journal of Management Information Systems*, vol. 35, no. 2, pp. 388–423, 2018.

[4] J. A. García-Díaz, M. Cánovas-García, R. Colomo-Palacios and R. Valencia-García, "Detecting misogyny in Spanish tweets. An approach based on linguistics features and word embeddings," *Future Generation Computer Systems*, vol. 114, no. 2, pp. 506–518, 2021.

[5] Y. Alotaibi, "Automated business process modelling for analyzing sustainable system requirements engineering," in *2020 6th IEEE Int. Conf. on Information Management*, London, UK, pp. 157–161, 2020.

[6] A. K. Nassirtoussi, S. Aghabozorgi, T. Y. Wah and D. C. L. Ngo, "Text mining for market prediction: A systematic review," *Expert Systems with Applications*, vol. 41, no. 16, pp. 7653–7670, 2014.

[7] S. K. Lakshmanaprabu, K. Shankar, D. Gupta, A. Khanna, J. J. Rodrigues *et al.,* "Ranking analysis for online customer reviews of products using opinion mining with clustering," *Complexity*, vol. 2018, pp. 1–9, 2018.

[8] S. Negi and P. Buitelaar, "Suggestion mining from opinionated text," in *Sentiment Analysis in Social Networks*, Elsevier, pp. 129–139, 2017.

[9] K. Lee, S. Han and S. H. Myaeng, "A discourse-aware neural network-based text model for document-level text classification," *Journal of Information Science*, vol. 44, no. 6, pp. 715–735, 2018.

[10] K. Liang and J. He, "Analyzing credit risk among Chinese P2P-lending businesses by integrating text-related soft information," *Electronic Commerce Research and Applications*, vol. 40, pp. 100947, 2020.

[11] E. Haris and K. H. Gan, "Mining graphs from travel blogs: A review in the context of tour planning," *Information Technology & Tourism*, vol. 17, no. 4, pp. 429–453, 2017.

[12] B. Bansal and S. Srivastava, "Hybrid attribute based sentiment classification of online reviews for consumer intelligence," *Applied Intelligence*, vol. 49, no. 1, pp. 137–149, 2019.

[13] J. Tao and L. Zhou, "A weakly supervised WordNet-Guided deep learning approach to extracting aspect terms from online reviews," *ACM Transactions on Management Information Systems*, vol. 11, no. 3, pp. 1–22, 2020.

[14] C. Zucco, B. Calabrese, G. Agapito, P. H. Guzzi and M. Cannataro, "Sentiment analysis for mining texts and social networks data: Methods and tools," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 1, pp. e1333, 2020.

[15] R. Piryani, D. Madhavi and V. K. Singh, "Analytical mapping of opinion mining and sentiment analysis research during 2000–2015," *Information Processing & Management*, vol. 53, no. 1, pp. 122–150, 2017.

[16] L. Tao, J. Cao and F. Liu, "Quantifying textual terms of items for similarity measurement," *Information Sciences*, vol. 415, no. 13, pp. 269–282, 2017.

[17] M. Kang, J. Ahn and K. Lee, "Opinion mining using ensemble text hidden Markov models for text classification," *Expert Systems with Applications*, vol. 94, no. 6, pp. 218–227, 2018.

[18] L. Liu, R. C. Chen, Q. Zhao and S. Zhu, "Applying a multistage of input feature combination to random forest for improving MRT passenger flow prediction," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 11, pp. 4515–4532, 2019.

[19] A. F. Subahi, Y. Alotaibi, O. I. Khalaf and F. Ajesh, "Packet drop battling mechanism for energy aware detection in wireless networks," *Computers, Materials & Continua*, vol. 66, no. 2, pp. 2077–2086, 2021.

[20] H. Zhang, H. Zhang, S. Pirbhulal, W. Wu and V. H. C. D. Albuquerque, "Active balancing mechanism for imbalanced medical data in deep learning-based classification models," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 1s, pp. 1–15, 2020.

[21] D. Liciotti, M. Bernardini, L. Romeo and E. Frontoni, "A sequential deep learning application for recognising human activities in smart homes," *Neurocomputing*, vol. 396, no. 6, pp. 501–513, 2020.

[22] R. Arulmurugan, K. R. Sabarmathi and H. Anandakumar, "Classification of sentence level sentiment analysis using cloud machine learning techniques," *Cluster Computing*, vol. 22, no. 1, pp. 1199–1209, 2019.

[23] U. Naseem, I. Razzak, K. Musial and M. Imran, "Transformer based deep intelligent contextual embedding for Twitter sentiment analysis," *Future Generation Computer Systems*, vol. 113, no. 2, pp. 58–69, 2020.

[24] F. Smarandache, M. Colhon, Ş. Vlăduţescu and X. Negrea, "Word-level neutrosophic sentiment similarity," *Applied Soft Computing*, vol. 80, no. 1, pp. 167–176, 2019.

[25] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes *et al.,* "Text classification algorithms: A survey," *Information-an International Interdisciplinary Journal*, vol. 10, no. 4, pp. 150, 2019.

[26] A. Chatterjee, U. Gupta, M. K. Chinnakotla, R. Srikanth, M. Galley *et al.,* "Understanding emotions in text using deep learning and big data," *Computers in Human Behavior*, vol. 93, no. 3, pp. 309–317, 2019.

[27] H. Wang, K. Tian, Z. Wu and L. Wang, "A Short Text Classification Method Based on Convolutional Neural Network and Semantic Extension," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 367–375, 2020.

[28] W. Liu, P. Liu, Y. Yang, J. Yi and Z. Zhu, "A <word, part of speech> embedding model for text classification," *Expert Systems*, vol. 36, no. 6, pp. e12460, 2019.

[29] M. A. Khan, M. Rashid, M. Sharif, K. Javed and T. Akram, "Classification of gastrointestinal diseases of stomach from WCE using improved saliency-based method and discriminant features selection," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 27743–27770, 2019.

[30] Y. Alotaibi, "A new database intrusion detection approach based on hybrid meta-heuristics," *Computers, Materials & Continua*, vol. 66, no. 2, pp. 1879–1895, 2021.