Received: 18 September 2020 | Revised: 15 January 2021 | Accepted: 20 January 2021

DOI: 10.1002/eng2.12374

RESEARCH ARTICLE

WILEY

Lexicon-pointed hybrid N-gram Features Extraction Model (LeNFEM) for sentence level sentiment analysis

James Mutinda[1] | **Waweru Mwangi**[2] | **George Okeyo**[3]

[1]Department of Information and Communication Technology, Kenya School of Government Embu Campus, Embu, Kenya

[2]School of Computing and Information Technology, Jomo Kenyatta University of Agriculture & Technology, Nairobi, Kenya

[3]School of Computer Science and Informatics, De Montfort University, Leicester, UK

Correspondence
James Mutinda, Department of Information and Communication Technology, Kenya School of Government Embu Campus, PO Box 402, 60100 Embu, Kenya.
Email: james.mutinda@ksg.ac.ke

Abstract

Sentiment analysis of social media textual posts can provide information and knowledge that is applicable in social settings, business intelligence, evaluation of citizens' opinions in governance, and in mood triggered devices in the Internet of Things. Feature extraction and selection is a key determinant of accuracy and computational cost of machine learning models for such analysis. Most feature extraction and selection techniques utilize bag of words, N-grams, and frequency-based algorithms especially Term Frequency-Inverse Document Frequency. However, these approaches do not consider relationships between words, they ignore words' characteristics and they suffer high feature dimensionality. In this paper we propose and evaluate a feature extraction and selection approach that utilizes a fixed hybrid N-gram window for feature extraction and minimum redundancy maximum relevance feature selection algorithm for sentence level sentiment analysis. The approach improves the existing features extraction techniques, specifically the N-gram by generating a hybrid vector from words, Part of Speech (POS) tags, and word semantic orientation. The vector is extracted by using a static trigram window identified by a lexicon where a sentiment word appears in a sentence. A blend of the words, POS tags, and the sentiment orientations of the static trigram are used to build the feature vector. The optimal features from the vector are then selected using minimum redundancy maximum relevance (MRMR) algorithm. Experiments were carried out using the public Yelp dataset to compare the performance of the proposed model and existing feature extraction models (BOW, normal N-grams and lexicon-based bag of words semantic orientations). Using supervised machine learning classifiers the experimental results showed that the proposed model had the highest F-measure (88.64%) compared to the highest (83.55%) from baseline approaches. Wilcoxon test carried out ascertained that the proposed approach performed significantly better than the baseline approaches. Comparative performance analysis with other datasets further affirmed that the proposed approach is generalizable.

KEYWORDS

feature selection, lexicon, minimum redundancy maximum relevance, N-gram2vec model, sentence level SA, sentiment classification, TF-IDF

1 | INTRODUCTION

User generated content has been on the increase with the growth of Internet and associated web 3.0 technologies. According to Reference 1 such data are increasing daily and are availed mostly through social media and worldwide web in the form of user comments and reviews. Sentiment analysis of these comments and reviews can enable businesses optimize their decisions on customer's opinions about the products they offer.[2] People are also keen on making decisions based on other people's opinions, governments and political leaders use sentiment analysis to discover citizens' preferences and opinions on governance issues.[3,4] Despite the potential benefits, analyzing such texts for sentiment classification still poses challenges to data analytics. Such content is characterized by short sentences that are unstructured, semi-structured, high in volume and normally full of colloquial language, thus posing a challenge in the effort to analyze them.[1,3,5] Therefore, identification of features to represent the sentences for input in classification models has become an important research problem.[3]

Feature representation and selection is the process of determining the most robust attributes that represent a text and can be used to effectively and efficiently predict the sentiment class of the text. It is thus a determinant factor in accuracy of Sentiment Analysis.[6,7] There are several techniques used for feature representation and selection in Sentiment Analysis which include Bag of Words,[8] Term Frequency-Inverse Document Frequency (TF-IDF),[9,10] N-grams, word embedding, and NLP (Natural Language processing).[11] Most of these techniques suffer from high feature dimensionality and computational complexity.[12,13]

Most feature extraction techniques generate features from the entire sentence or document. Recently, attention on feature extraction techniques that focus on a segment or section of a document or sentence has increased. This has mainly been done on document level sentiment analysis where the topic, introduction or conclusions of the document are considered.[14] This implies that an opinion in a subjective sentence or document may not necessarily be in the entire sentence or document but in a specific part. However, existing works[14,15] that focus on specific parts of a sentence for feature extraction do not clearly show how the sentences are classified but concentrate on generation of word dictionaries, similarities and sentiments. Research experiments that consider specific words in a sentence mostly use Word2Vec models to determine word semantic orientations, meanings and synonyms. In such cases Continuous Bag of Words and Skip-gram models are applied. However, vectors obtained through such models do not consider words arrangements, context of the words and other words syntaxes such as POS tags.[16] For effective feature extraction and representation for sentence level sentiment classification, it is crucial to include other linguistic aspects such as POS tags, word semantic orientations and arrangement. In relation to this, we develop an approach, herein the proposed Lexicon-pointed hybrid N-gram Features Extraction Model (LeNFEM). The model utilizes sentiment lexicon, N-grams and minimum redundancy maximum relevance feature selection algorithm for sentence level sentiment analysis. The approach is based on two algorithms: Fixed hybrid N-grams algorithm and MRMR features selection algorithm. A fixed word trigram is identified in a subjective sentence using a sentiment lexicon. The trigram words are then used to generate a vector from words, POS tags and word semantic orientations to represent a subjective sentence. Lastly, MRMR feature selection algorithm is used to identify the optimal features that are used as input to a machine learning classifier.

The main purpose of this work is to develop a systematic approach of extracting and selecting optimal features to effectively and efficiently classify subjective social media posts (sentences). This work, therefore, advances the feature extraction and selection phase in sentence level sentiment classification process. The approach proposed is slightly similar to existing works such as References 14,15 but with a key distinction. The key difference is that, while they use N-grams, Word2Vec models and TF-IDF to build document representations they do so from *the entire document*. However, our work builds the document representation in form of a hybrid vector identified from a *specific part of a sentence*. As a result, this paper makes the following contributions. First, we propose the use of hybrid features that are built from words, POS tags, and word semantic orientations identified using a fixed trigram. The novelty of the approach is in the utilization of sentiment lexicon in identification of a fixed trigram, instead of the conventional sliding window N-gram, from a subjective sentence. Furthermore, existing N-grams are based on words; however, this work investigates the use of hybrid N-grams of words, POS tags and sentiment polarities which is advancement in utilization of N-grams in sentiment analysis. The approach improves the traditional approaches since the semantic and syntactic information between words is considered through the utilization of the POS tags and sentiment polarities of words. Secondly, we apply minimum redundancy maximum relevance feature selection algorithm to optimize the features. This is an advancement of the norm where the algorithm is mostly used in gene data. We believe that the fixed trigram window provides an effective approach for identifying a specific part of a subjective sentence to simplify semantic classification of social media posts.

The rest of the paper is organized as follows. Section 2 presents related work. The proposed LeNFEM approach is described in Section 3. Section 4 describes the methodology and implementation details. The result and discussion are presented in Section 5. Finally, Section 6 concludes the paper, highlights the limitations, and recommends areas for future work.

2 | RELATED WORK

Feature extraction and selection in sentiment analysis is the process of identifying features that fully and optimally represent the text to be classified.[6,11] Two main challenges arise from feature extraction (i) the features that fully represent the text[17] and (ii) the optimal features that can be used to infer the sentiment class of the text. In the first challenge, there are many feature extraction techniques such as Bag of Words, TF-IDF, N-grams, word embedding and NLP techniques. These techniques mostly consider the entire document or sentence in generating the vector representation which leads to increase in feature dimensionality. According to Guo and Yao[18] these techniques do not consider semantic and syntactic information between words since each word is taken independently. In the second challenge, feature selection techniques, which are categorized as filters, wrapper and embedded techniques are used in feature dimensionality reduction. While feature selection techniques have proved to be efficient in optimal feature selection, there is need to ensure that the right features are extracted at the feature extraction stage.

Recently, vector representations of documents or sentences are used as input features into supervised machine learning classifiers. Techniques used to build the vector representations include, TF-IDF,[11,14] N-grams[11,19] and Word2Vec models[15,20] among others. Chug, Gupta and Ahuja[11] analyzed the impact of two feature extraction techniques; TF-IDF word level and N-Gram on SS-Tweet dataset of sentiment analysis. They found out that TF-IDF is better compared to N-gram (n = 2). Despite showing that N-grams are features that can be used to represent a text, they did not show clearly how the vector representation was built thus making the proposal difficult to replicate. Further, N-grams are converted into vectors using their occurrence, frequency or TF-IDF values. Therefore, arguing that TF-IDF performs better than N-grams simply portrays some disconnect between the techniques. For instance, when N = 1, the representation reduces to a mere bag of words that can be represented by TF-IDF values. However, several researchers have used TF-IDF for document vector representation.[14,15]

Barkha and Sangeet[14] used TF-IDF weighting scheme to find similar semantic words after which they used CBOW and Skip-gram models to classify online reviews. They however, proposed that concatenating word vectors for representing a document can be computationally expensive but can demonstrate more accurate results. This is supported by Fatma[19] that text has to be transformed into numeric representations suitable for learning algorithms. The weakness in their work is that they did not show how the vector representation were built since they used black box CBOW and Skip gram models used for classification by tuning the parameters. Nevertheless, they put forth two important recommendations which this research has exploited; that Lexicon based labeling can improve the Word2vec models and the need to test use of Word2Vec in short texts like tweets. These recommendations critically implied that any work that combines Word2Vec representations with lexicon labeling of words would improve feature extraction for sentiment analysis. Such a recommendation is also supported by Bagus et al[21] that semantic labeling of words has the potential of improving supervised sentiment classification since bag of words doesn't consider semantic of words.

Fatma[19] did text classification using bi-gram alphabet as features. The approach has two main contributions to text classification research. First, they demonstrated the possibility of using constant feature terms that are based on the standard alphabet without the need for the documents vocabularies which definitely helps in reducing the dimensions of the vector space for large corpus. The information content of an alphabet is difficult to infer. Sujata and Shinde[22] highlighted that when selecting features its crucial to consider information content of the features. Therefore, the use of a higher level linguistic gram like a word or phrase is better since a word or a phrase contains more information than an alphabet. Our work is similar to Fatma's work[19] because we use N-grams but differs since we use fixed words trigram instead of bigram alphabet. Although the use of word trigram increases dimensionality we control it by using a fixed trigram instead of the sliding window N-gram.

In their work Zhou and Liu[15] they proposed a new model named Latent Semantic Analysis and Word2Vec (LSA) + Word2Vec model to create document vectors in vector space using the Convolutional Neural Network (CNN) model. They generated a 2- dimensional document vector weighted with TF-IDF of the words to fully represent a document in vector space. Although the technique can map document to vector space under the premise of keeping document contents fully, they did not show how the same could be applied in sentence level sentiment analysis. Further the matrix

generated was a 3-dimensional matrix and the meaning of every word was considered thus leading to high computation cost and high features. Sujata and Shinde[22] used important words such as nouns, adjectives, intensifiers, verbs, and linguistic features of praise and complaint sentences and Affin dictionary to calculate extreme sentiment of each sentence. They further used Hybrid features like meta, Synthetic, content, and Semantic features of the sentences. The problem of high feature dimensionality again arises here since the features were selected from the entire sentence.

Chuhan et al[23] affirmed that sentiment words are important units in text and therefore can convey sentiment in a document or a sentence. The researchers also confirmed that sentiment lexicons have the potential to improve sentiment classification. In their work, they used neural networks to identify sentiment bearing words and further classify the sentiment words in various sentiment lexicons. Though their work can be used as a basis for using sentiment lexicons in sentiment classification they did not show how the sentiment lexicons and words could be applied in sentence level sentiment classification. Ren et al[24] agreed with Chuhan et al[23] that adding lexicons into machine learning classifiers can improve the performance of sentence level sentiment classification. They proposed lexicon-enhanced attention network (Lean) model based on bidirectional Long Short Term Memory (LSTM). In their work, they used lexicons and bidirectional LSTM to identify sentiment words in a sentence and the aspect that the sentiment is being directed to.

In their work,[20] word vectors were generated from pre-trained Word2Vec model and CNN layer was used to extract better features for short sentences categorization. Multiple channels pattern is applied for text processing in which one channel could be the sequence of words, another channel the sequence of corresponding POS tags, and the third one the shape of the word. The channels are then combined into a single vector that captures the most relevant features of the sentence. To solve the weakness of focusing on entire sentence for feature extraction, we considered three proposals; first was by Zhao et al[25] that attention based feature extraction in a sentence has the potential of improving sentiment classification task; and the second and third were by Li et al[26] and Chuhan et al[23] that cooperating prior sentiment knowledge into word representations has the potential to improve sentiment analysis. In both we use sentiment lexicon to identify a section of a sentence and to label the semantic orientation of the words from the identified section.

Although our work is in the area of feature extraction and selection for sentence level sentiment classification, it is slightly similar to the work in[14,15,19,22] since it uses word vector sentence representations using N-gram and Lexicon dictionary. The key difference with our work is that while they use N-grams, Word2Vec models and TF-IDF to build document representations they do so from the entire document (sentence), in our work, we use tri- grams words, their POS tags and Semantic orientations identified from a specific part of a sentence by utilizing the proposals by references.[25,26] As a result, our approach is able to utilize N-grams models, NLP techniques and sentiment lexicon to advance short sentences vector representation and consequently improve sentence level sentiment classification. Further the features are selected using the minimum redundancy maximum relevance feature selection algorithm. There are several feature selection algorithms categorized as filter, wrapper and embedded methods. The minimum redundancy maximum relevance was chosen since it's the most reliable approach due to its accuracy according to Gallego, et al.[27] It's a filter method thus faster since filter methods are faster than wrapper and embedded methods.[28]

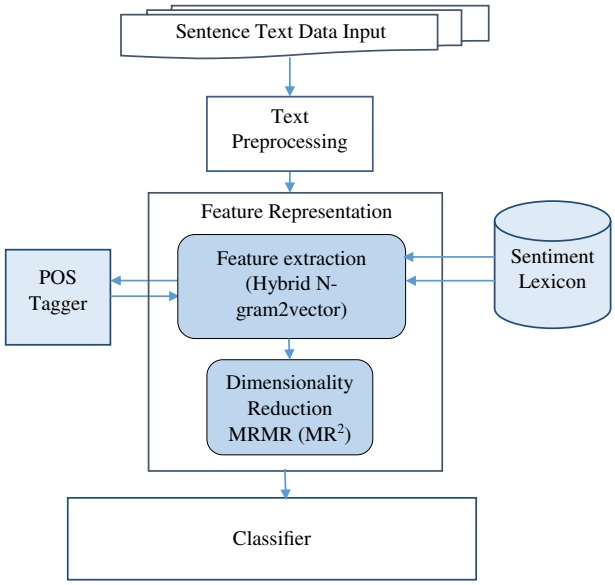
From the above discussion, we acknowledge that the use of vector representations as features for sentence level sentiment classification simplifies sentiment classification by use of N-grams, TF-IDF, and Word2Vec models. However, there is a need to improve the techniques to address high feature dimensionality and optimize performance of machine learning classifiers. We use a static trigram window based on the position of sentiment term in a subjective sentence. The sentiment term is identified from a sentiment lexicon. The features are further optimized using the minimum redundancy maximum relevance feature selection algorithm. The proposed approach provides an efficient and effective solution to the feature extraction problem that shows demonstrable improvements to classifier performance.

3 | PROPOSED LENFEM

3.1 | Overview

Sentiment classification models consist of three modules; Text data preprocessing module, feature representation module and sentiment classification module.[10] Sentence level sentiment analysis is a sentiment classification problem in which a subjective sentence is categorized into an opinion class. In the proposed model, the feature representation module is adjusted. The conceptual architecture of the proposed approach is shown in Figure 1.

FIGURE 1 LeNFEM conceptual architecture



The conceptual model shows that the feature representation module uses a combination of N-grams, Sentiment Lexicon and POS tagger to build an N-gram2vector and further reduces the vector space by MRMR algorithm. Each component in Figure 1 is discussed in detail in Sections 3.2, 3.3, and 3.4.

3.2 | Sentence text data input and preprocessing

Social media texts are normally unstructured, noisy and inconsistent.[29] The tweets are cleaned of non-English words or sentences, abbreviations and emoticons and stop words after input. Tokenization and transformation of the texts into lower case is also done to split the sentences into separable words. Then the preprocessed data are used for feature vector construction and representation.

3.3 | Feature representation: The sentence vector

The novelty of our approach is in the modification of the feature representation module. We propose the lexicon hybrid N-gram2vector model to generate sentiment term aspects in form of a row vector to represent a subjective sentence. The vector attributes are further filtered using the minimum redundancy maximum relevance (MRMR) algorithm.

3.3.1 | Hybrid static N-grams2vec model

We first build a vector of hybrid static N-grams. To construct a vector using N-grams, each sentence is converted into overlapping N-grams by running a predefined window of size N. N-gram is a statistical language model (LM) where a document or a sentence is broken down into a sequence of words w_i (w_1, w_2, ..,w_n). N-gram is the most used LM[29] which makes a Markov assumption and defines the context Φ (W_{i-1}) as;

$$\Phi\ (W_{i-1}) = w_{i-n+1}, w_{i-n+2}, \ \dots\ , w_{i-1} = h. \tag{1}$$

For, N = 1 the context does not exist hence it's the normal bag of words.
N = 2 the context becomes $\Phi\ (W_{i-1}) = w_{i-1}, w_i$ Thus two words are considered.
N = 3 the context becomes $\Phi\ (W_{i-1}) = w_{i-2}, w_{i-1}, w_i$. Thus three words are considered.

From the formal definition, an N-gram is thus a textual sequence containing N adjacent 'textual units' from a particular sentence or document. A 'textual unit' can be identified as a character, word or a phrase depending on the context of interest from which a vector representation of the N-grams is formed. In this work we identify the N-gram at word level.

Each of the N-grams is a coordinate in a vector which represents the text under study. The frequency, occurrence or any other metric of this n-gram in the text becomes the value of this coordinate.[1] The simplest n-gram is the unigram, where n = 1, which is basically the normal "bag-of-words" (BOW) representation. To generate the vector from a huge text dataset can be challenging. N-grams models are widely used in NLP tasks since they are simple and effective.[30] However, in n-grams each sentence is converted into a bag of n-grams and represented as a vector of occurrence frequency without taking into consideration the information encapsulated in the n-grams of the original text. This leads to so many irrelevant and redundant attributes in the vector. The sliding window of the n-gram also makes the variables more and thus some become less relevant.

Here we proposed a novel way of utilizing N-gram model for feature extraction in sentence level sentiment analysis. After generating the N-grams (N = 3) representing a sentence, we identify one three words N-gram that contains a sentiment term. This is achieved by invoking a sentiment lexicon that points to an N-gram containing the sentiment term after generation of the three words N-grams of the sentence. The identified N-gram is then split to a bag of three words. The three words are then used to construct a hybrid vector for the sentence from the words, their POS tag and their sentiment orientation. From Equation (1) and using N = 3, the context becomes;

$\Phi\ (W_{i-1}) = w_{i-2}, w_{i-1}, w_i$ for words;

$\Phi\ (P_{i-1}) = P_{i-2}, P_{i-1}, P_i$ for POS tags and;

$\Phi\ (O_{i-1}) = O_{i-2}, O_{i-1}, O_i$ for semantic orientation.

We combine the three sets of contexts to get a hybrid of words, POS tags and Semantic orientations called the sentiment aspects (A) given as;

$$\Phi\ (A_{i-1}) = W_{i-2}, W_{i-1}, W_{i,} P_{i-2}, P_{i-1}, P_{i,} O_{i-2}, O_{i-1}, O_i. \tag{2}$$

To obtain the vector of the hybrid of words, POS tags and the sentiment orientations, binary occurrences or TF-IDF text vectorization schemes are used. In binary occurrence vectorization, if the word (w_i), the POS tag (P_i) or the sentiment orientation (O_i) is present in a sentence (S_i) a value one (1) is assigned otherwise a zero (0) is assigned.

TF-IDF is a text vectorization scheme which is calculated in two steps. First the term frequency (TF) is obtained as the ratio of number of times a term t appears in a sentence s to the total number of terms in the sentence given as;

$$\mathrm{TF}\,(t, s) = \log\,(1 + feq\,(t, s))\,. \tag{3}$$

Secondly, The Inverse Document Frequency (IDF) is given as the ratio of total number of sentences (S) in the corpus to the total number of sentences (S_t) that contain at least an occurrence of term (t) given as;

$$\mathrm{IDF}\ (t, S) = \log\ (S/S_t) + 1 \tag{4}$$

Then combining Equations (3) and (4) we have;

$$\mathrm{TF} - \mathrm{IDF} = \log\,(1 + feq\,(t, s)) * \log\ (S/S_t) + 1. \tag{5}$$

Figure 2 shows how a sentence is converted to N-grams where N = 3 and then vector representation build using the proposed approach.

From the sentence N-gram vector generator presented in Figure 2, the sentence is split into 3 N-grams from the sentence word tokens. An N-gram is then identified containing a sentiment term or a word from an existing or customized sentiment lexicon. For each word in the N-gram, its POS tag and semantic orientation is noted from a POS tagger and the sentiment lexicon, respectively. Using a suitable vectorization algorithm (TF-IDF or binary occurrence), the words, POS tags and semantic orientations are converted into a numeric vector. In the state of the art the vector is constructed from the entire sentence's BOW or N-grams. In our approach, we sought to identify a specific part of the sentence where the opinion is present and build a vector representation of the words, POS tags and semantic orientations hybrid features from it.

Assumptions

1. From the word trigram, three scenarios arise, first the sentiment word may be the first word, secondly the sentiment term may be the second word in between the other two words and thirdly, the sentiment word may be the last word among the three words. It is assumed that the order of the words has insignificant effect on the model.

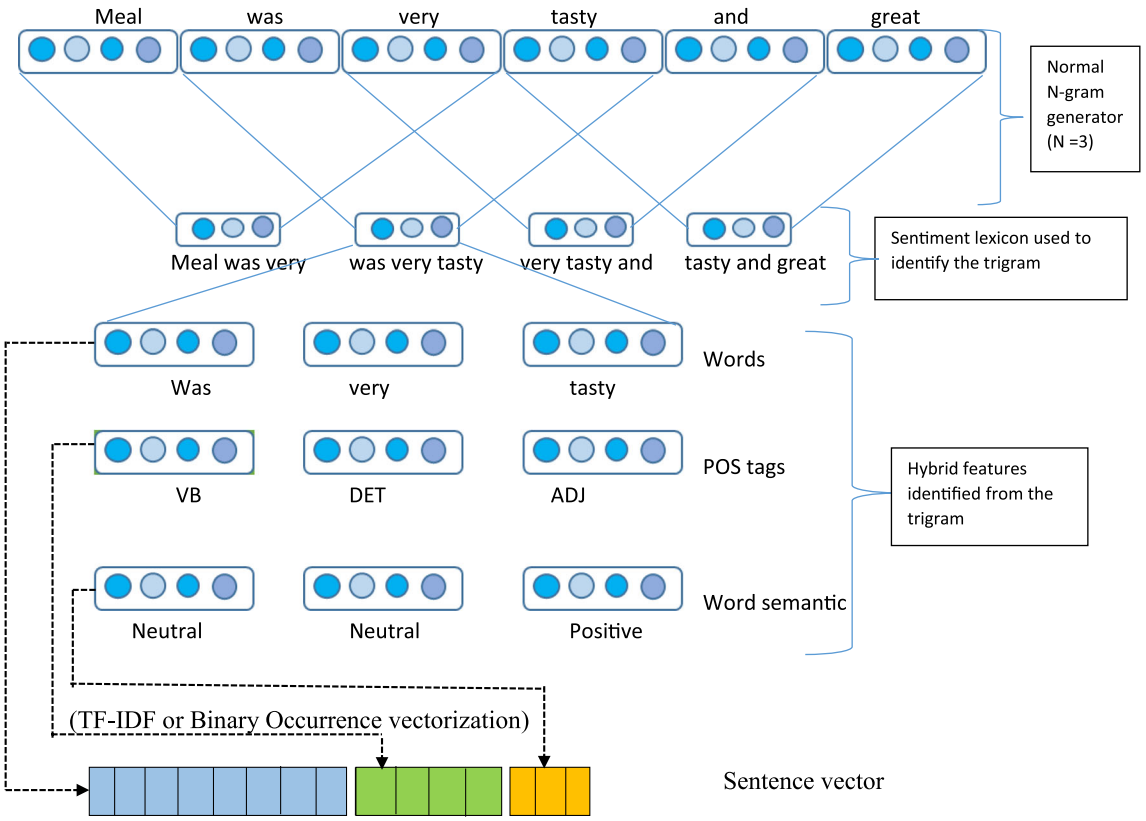


FIGURE 2 The sentence N-gram vector generator

2. It is assumed that the objective sentence has only one opinion thus one sentiment term in one part of the sentence. If two opinion terms appear close to one another they support the opinion or negate it thus giving the right opinion classification of the sentence. Sometimes a sentence may contain more than one opinion and thus more than one sentiment term.

In the next section, we present the algorithms that model and implement the static hybrid N-gram2vec representation.

3.3.2 | Static hybrid N-gram2vec algorithm

We propose a formal static hybrid N-gram2vec model whose characteristics and operation are described below. We define a number of parameters used to describe the operation of the model. Significant parameters include the size of the N-gram which is N = 3, which was chosen since it's a robust size of N-gram in terms of machine performance,[29] the sentiment term, the static N-gram words, their POS tags and their Semantic orientations. The algorithm aims at returning the vector representation of the subjective sentence.

Definitions

Let;

D: Sentiment Lexicon
P: Part of Speech tagger
S: Subjective sentence corpus
$\boldsymbol{v}$: Vector representation of a subjective sentence(S_i)
W_t: Sentiment term
W_1: the first word neighboring the sentiment term
W_2: the second word neighboring the sentiment term
P_t: POS tag of the Sentiment term
P_1: POS tag of the first word neighboring the sentiment term

P_2: POS tag of the second word neighboring the sentiment term
O_t: Semantic orientation of sentiment term
O_1: Semantic orientation of the first word neighboring the sentiment term
O_2: Semantic orientation of second word neighboring the sentiment term
$\boldsymbol{v}_w$: Vector of words
$\boldsymbol{v}_p$: Vector of POS tags
$\boldsymbol{v}_s$: Vector of Sentiment orientations.

We define the vector representation, $\boldsymbol{v}$, of a subjective sentence, S_i, as a concatenation of the three vectors; $\boldsymbol{v}_w$, $\boldsymbol{v}_p$ and $\boldsymbol{v}_s$. as shown below;

$\boldsymbol{v} : < \boldsymbol{v}_w \,\&\, v_p \,\&\, \boldsymbol{v}_s >$.

The three vectors $\boldsymbol{v}_w$, $\boldsymbol{v}_p$, and $\boldsymbol{v}_s$. are values obtained from the occurrences or TF-IDF of the words, POS tags, and sentiment orientations.

We also define the matrix $\mathbf{M}^1$ as the collection of row vectors B_i which are the bag of words, POS tags, and semantic orientations for each sentence given as;

$$M^1 = \begin{bmatrix} B_1 \\ \cdots \\ \cdots \\ B_n \end{bmatrix}$$

n is the number of sentences in the Corpus

The algorithm listing of the sentence vector representation generation is presented in Figure 3.

Input: Receives preprocessed Sentence corpus(S), the Sentiment Lexicon(D)and the POS tagger (P)
Output: A Vector representation ($\boldsymbol{v}_i$) representing the subjective Sentence

```
START
Set the N-gram value to N=3
FOR each sentence(Si∈S)word tokens
                PRINT the trigrams;
                Call the Sentiment Lexicon(D)
                   FOR each trigram check for a semantic word;
           IF a trigram contains a semantic word THEN
                PRINT the trigram (w)
                BREAK
              ELSE delete the trigram
           ENDIF
END
Generate sentence vector(wi,P, D)
FOR Each word(wi)in trigram(w)
    READ (wi)into Bag of words(Bwi)
    DETERMINE the POS tag (pi)of wi (using POS tagger P) into
              Bag of POS tags (Bpi)
    DETERMINE the sentiment orientation(Oi)of (wi)(from
              the sentiment Lexicon)into Bag of sentiment
              Orientation (Boi)
    END
  Update Bi:<Bwi& Bpi& Boi>

 RETURN Matrix M1=[ B1
                    …
                    …
                    Bn ]   n is the number of sentences in the
                             corpus
FOR each Wi, Pi and Oi in Bi
      COMPUTE its Binary Occurrence or TF-IDF* value into
              vector vi:<vwi & vpi & voi>
END
Return Vector Vi.
```

FIGURE 3 Sentence vector generation algorithm listing

3.3.3 | Dimensionality reduction using maximum relevance minimum redundancy (MRMR)

The N-gram vector generated contains many features hence the need for dimensionality reduction. In this research we apply and test Maximum Relevance Minimum Redundancy (MRMR) as proposed by the authors in References 31-33. In MRMR, Mutual Information is used since the features are discrete. First the Mutual Information (I*min*) for each feature is determined as the level of similarity between it and each of the other features calculated as shown in Equation (6).

$$Imin\left(f_i : f_y\right) = \sum_{i\in I}\sum_{y\in Y} P(i,y)\log\frac{P(i,y)}{P(i)\,P(y)}. \tag{6}$$

where f_i is the feature and f_y is the other feature being compared. This is calculated for all features in the vector. Secondly, the relevance is determined by calculating the Mutual Information between a feature and the class (I*max*) calculated as shown in Equation (7).

$$Imax(f_i : C_i) = \sum_{fi\in I}\sum_{Ci\in I} P(fi,Ci)\log\frac{P(fi,Ci)}{P(fi)\,P(Ci)}. \tag{7}$$

where f_i is the feature and C_i is the class. This is used to eliminate the irrelevant features since features with high mutual information with the class have high relationship with the class thus more relevant.

To select the most relevant features the Mutual Information Quotient between I*min* and I*max* for each feature is noted and the features sorted from the one with maximum quotient as suggested by Houda et al.[33] The feature selection algorithm listing is presented in Figure 4.

From Figure 4, the output is a sorted matrix of features and sentence representations, the optimal features (that provides the best classification performance) can then be selected from the entire set of features. Combining algorithms 1 and 2 gives the algorithm of the proposed approached as listed in Figure 5.

Input: Receives A matrix **M = (N*F + c)** where; **F** is the features of a collection of subjective Sentence corpus with **N** sentences and **c** is the class column of the sentences.

Output: A matrix (**M'= N*F + c**) containing ordered features in importance representing subjective sentences in the Corpus

START

Sort features()

FOR each feature $\mathbf{f_i}$ in **M**

CALCULATE its mutual information in relation to other features as

$$Imin(f_i: f_y) = \sum_{i\in I}\sum_{y\in Y} P(i,y)\log\frac{P(i,y)}{P(i)P(y)}$$

CALCULATE its mutual information in relation to the target opinion class as;

$$Imax(f_i: C_i) = \sum\nolimits_{fi\in I}\sum\nolimits_{Ci\in I} P(fi,Ci)\log\frac{P(fi,Ci)}{P(fi)P(Ci)}$$

CALCUATE MRMR of feature $\mathbf{f_i}$ as

$$MRMR(f_i) = \frac{I\max(fi:Ci)}{Imin(fi:fy)}$$

END

SORT the features(f)in descending order of values of $MRMR(f_i)$ and update matrix(**M'**) with sorted features

END

Return (**M'**)

FIGURE 4 Feature selection (MRMR) algorithm listing

```
Input: Receives a preprocessed Sentence (Si), the Sentiment
Lexicon(D)and the POS tagger (P)
Output: A vector (Vi') containing ordered features in relevance
representing subjective sentence Si
START
FOR each tokenized sentence(Si)in Corpus(S)
      Call function Generate vector representation () from figure3.
Return Vi:{Vwi,Vpi,Voi}

Update matrix M2 = [Vi] = (n*F); where, F is the features of a
       collection of subjective Sentence corpus with n sentences.

   FOR each coordinate in M2
      Call function Sort features()in figure 4
    END

Return vector(Vi')the representation of sentence Si
```

FIGURE 5 The proposed approach algorithm listing

To identify the optimal features, cross validation is used for various values of k (number of features) and the F-measure performance of each k noted. The optimal k (number of features) is the maximum number of features that gives the highest model performance (Y). The optimal number of features is dependent on the classifier used. The optimal features are used with supervised learning machine classifier to classify the post.

3.4 | Classification

There are several methods of classification that can utilize the optimized features generated by the approach. They broadly categorized into unsupervised and supervised Machine learning classifiers. The supervised machine learning classifiers are used when there is a labeled data available. In this case there should be labeled sentences which can be used to train a machine learning classifier. In absence of a labeled data unsupervised machine learning classifiers are used. It is also worthy to note that the output of the approach can also be used in deep learning as input layer to the learner.

4 | EXPERIMENTAL SETUP AND EVALUATION DETAILS

This section describes the dataset used, the preprocessing techniques used and the experiments carried out to evaluate the performance of the proposed approach. The tools and techniques used in model formulation and evaluation are also discussed.

4.1 | Dataset description and preprocessing

A sentence text dataset compiled by Kotzias[34] was used to perform the experiments. The dataset contains 3000 sentences labeled with positive or negative sentiment. The sentences were extracted from reviews of products, movies, and restaurants obtained from amazon.com and imdb.com websites respectively. For each website, there exist 500 positive and 500 negative sentences which were selected randomly from larger datasets of reviews. The tweets were cleaned of non-English words or sentences, abbreviations, emoticons and stop words using Java Stanford Core NLP libraries. Stanford Core NLP tokenizer was used to tokenize the posts and POS tagging. The Sentiment lexicon[35] was used to identify sentiment terms. Thorough experiments were first done using the Yelp dataset and later we confirmed the performance of the proposed model using the movies and products dataset. The Yelp dataset's 1000 tweets were cleaned to 996 tweets. Of this, 498 were negative and 498 were positive.

4.2 | Preparation of the experimental matrices

Four Matrices were prepared from the Yelp dataset for the experiments. One using the proposed approach, the second matrix was prepared using the Bag of Words of the tweets, the third one compiled using the normal N-grams (n = 3) and

the fourth using a lexicon to convert the sentences into bag of words semantic orientations. Using the proposed approach, the hybrid vector was compiled as discussed in Section 3. The latter three matrices were used to generate results as baseline approaches for comparison with the proposed approach. After investigating the performance of the proposed approach using the Yelp dataset, an experiment was also performed to test the approach with the other two datasets (the Amazon's products dataset and the Imdb's movie dataset).

4.3 | Model formulation and performance evaluation

Cross validation method was used with 30 random splits of the data. The performance measure (F measure) was computed for each split and the average result and the standard deviation recorded for the proposed approach and baseline approaches using the four machine learning classifiers. Rapid Miner Studio 9.0, a data mining environment was used to formulate and evaluate the models using four supervised machine learning algorithms (Naïve Bayes, K-nearest Neighbor, Decision Tree and Support Vector Machines). To evaluate the performance of the proposed approach, the performance results of the supervised machine learning algorithms used to simulate the model were plotted in a confusion matrix.[36]

From the confusion matrix, weighted accuracy, precision, recall and F measure were used as performance metrics to evaluate the performance of the approach. The metrics are defined and presented in Equations (5)–(8).

Accuracy is the proportion of correctly predicted cases to the total cases.

$$\text{Accuracy} = \frac{A + D}{\text{total cases}} \tag{8}$$

Recall is the proportion of actual cases correctly classified. It is also referred to as sensitivity or true predicted (TP) rate.

$$\text{Weighted Recall} = \frac{\text{TP positive} + \text{TP negative}}{2} \tag{9}$$

where

$$\text{TP positive} = \frac{A}{A + B}$$
$$\text{TP negative} = \frac{D}{C + D}$$

Precision is the proportion of predictions that are correct.

$$\text{Weighted Precision} = \frac{\text{Precision positive} + \text{Precision negative}}{2} \tag{10}$$

where

$$\text{Precision positive} = \frac{A}{A + C}$$
$$\text{Precision negative} = \frac{D}{B + D}$$

$$\text{F measure} = 2 * \text{precision} * \textit{Recall}/\,(\text{precision} + \text{Recall}) \tag{11}$$

Lastly, we compare the performance of the proposed approach with baseline feature selection approaches using F Measure performance metric.

5 | EXPERIMENTAL RESULTS

This section describes the results obtained from the three experiments conducted. The first experiment was conducted using text vector obtained from the normal Bag of Words (BoW), normal 3 N-Grams, Bag of words' sentiment polarities

and the proposed approach (Static Hybrid 3 N-gram) using binary occurrence algorithm. The bag of words' sentiment polarities were obtained by using the lexicon to identify the polarities of the words in the sentences into positive, negative or neutral (for words not found in the lexicon). In the second experiment vectorization was done using TF-IDF algorithm. The third experiment was a comparison of the performance of the proposed approach using the two vectorization algorithms; the binary occurrence and the TF-IDF. The fourth experiment was an analysis of the effect of using MRMR feature selection on the proposed approach. The fifth experiment was a performance investigation of the proposed approach with other datasets (products and movies datasets). To evaluate the performance and the feasibility of the approach in each experiment we obtained the accuracy, recall and precision metrics from which F-measure was calculated as shown in Equation (11).

5.1 | Experiment results I: Using binary occurrences vector

The experiment was done using the normal Bag of Words (BoW), normal 3 N-Grams, lexicon-based bag of words' polarities and the proposed approach (LeNFEM). Using the proposed approach (LeNFEM) we first identify the optimal features. Since MRMR algorithm sorts the features in order of importance, Figure 6 presents how the performance varies as the number of selected features increased. We present the results for the first 200 features from the total 973 features since the performance seemed to stabilize for all classifiers used from 50 features and reduce at 200 features.

From Figure 6, it is clear that as the number of features selected increases, the classification performance increases to a point where it stabilizes. The highest performance for Decision Tree was achieved with 25 features, K – NN (K = 5) with 65 features, Naïve Bayes with 130 features and for Support vector machine was 150 features. Details of the F-measure performance for each approach and machine classifier used were as presented in Table 1.

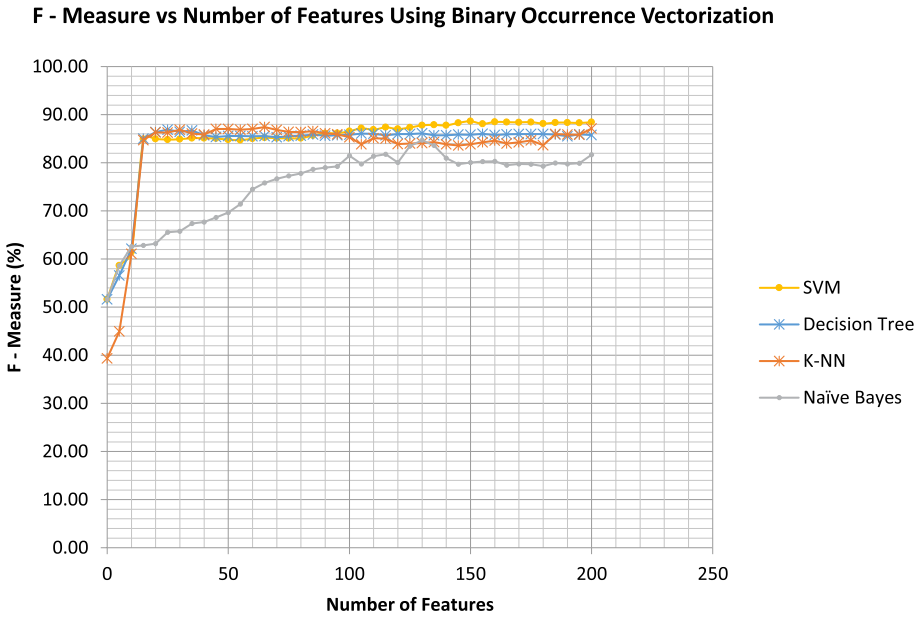


FIGURE 6 F-Measure performance trend against number of features

TABLE 1 F-Measure results for binary occurrence vector

Feature extraction approach	Machine learning classifier F-measure performance (%)							
	Naïve Bayes		K – NN		Decision tree		Support vector machines	
	n	F (%)	n	F (%)	n	F (%)	n	F (%)
Bag of words	1783	72.24 +/−0.520	1783	67.40 +/−0.572	1783	70.22 +/−0.531	1783	75. 93 +/−0.545
3N-gram	8626	70.72 +/−0.551	8626	69.88 +/−0.621	8626	70.08 +/−0.574	8626	65.81 +/−0.560
Bag of words' polarities	4	82.34 +/−0.651	4	83.55 +/−0.542	4	82.54 +/−0.341	4	83.44 +/−0.359
LeNFEM	130	84. 68 +/− 0.671	65	87.43 +/− 0.539	25	86.91 +/− 0.331	150	**88.64 +/− 0.362**

Note: n is the number of features used, F (%) is the F-measure +/− the standard deviation.

From Table 1, the proposed LeNFEM approach outperformed the baseline approaches. Using the bag of words' polarities only four features were identified; negative, positive, neutral or a missing value. It is also worth noting that of the four classifiers used, support vector machine classifier produced the highest F-measure score of 88.64%.

5.2 | Experiment results II: Using TF-IDF vector

The experiment was done using the normal Bag of Words (BoW), lexicon-based bag of words' polarities, normal 3 N-Grams and the proposed (LeNFEM) approach. Using the proposed approach, MRMR algorithm sorts the features in order of importance. Figure 7 shows how the performance varies as the number of selected features increased using the proposed approach. This comparison was used to identify the number of features that produced the highest classification prediction performance.

From Figure 7, it is clear that as the number of features selected increased, the classification performance increases to a point where it stabilizes. The highest performance for Decision Tree was achieved with 30 features, K – NN (K = 5) with 80 features, Naïve Bayes with 105 features and for Support vector machine was 95 features. Details of the performance results for each approach and machine classifier used were as presented in Table 2.

From Table 2, the proposed LeNFEM approach outperformed the baseline approaches considered. It is also worth noting that of the four classifiers used, support vector machine classifier produced the highest F-measure score of 87.45%.

We further used two tailed Wilcoxon Sign-Rank test at a significance level of 0.05 to ascertain whether the LeNFEM approach performed better than the baseline approaches. We paired all the F-measure results from the classifiers for the

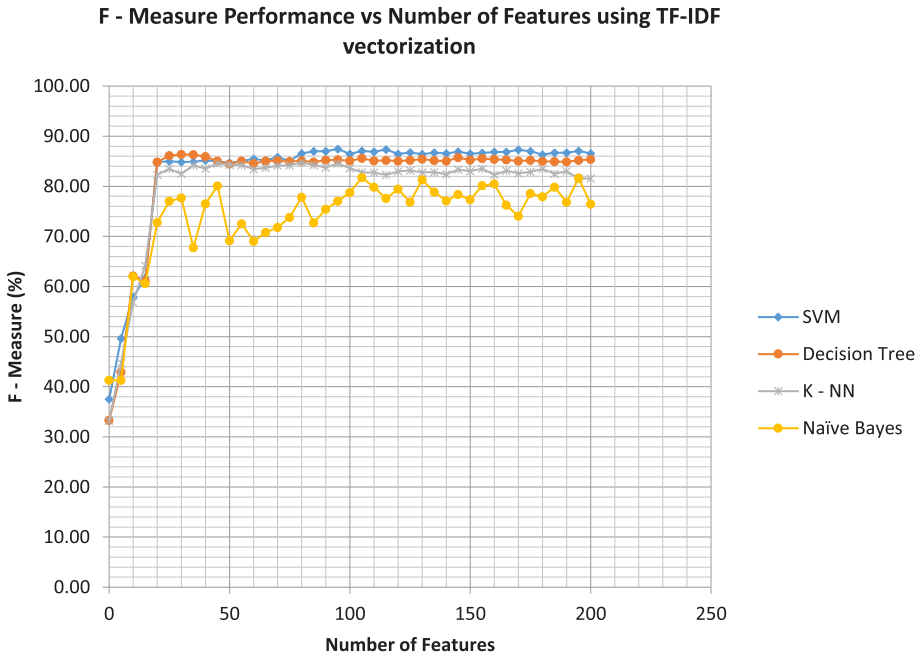


FIGURE 7 F-Measure performance vs number of features using TF-IDF vector

TABLE 2 F-Measure results using TF-IDF vector

Feature extraction approach	Machine learning classifier F-measure performance (%)							
	Naïve Bayes		K – NN		Decision tree		Support vector machines	
	n	F (%)	n	F (%)	n	F (%)	n	F (%)
Bag of words	1783	65.64 +/− 0.532	1783	59.38 +/−0.602	1783	68.85 +/− 0.576	1783	66.62 +/−0.556
3N-gram	8626	67.25 +/−0.645	8626	64.24 +/−0.590	8626	65.92 +/− 0.625	8626	62.48 +/−0.634
Bag of words' polarities	4	78.56 +/−0.632	4	82.75 +/−0.607	4	82.36 +/−0.334	4	81.46 +/−0.536
LeNFEM	105	81.75 +/− 0.358	80	84.71 +/− 0.350	30	86.37 +/− 0.356	95	**87.45 +/− 0.330**

Note: n is the number of features used, F (%) is the F-measure +/− the standard deviation.

approaches. We first compared the F-measure results of LeNFEM with the Bag of Words approach which gave W-value: 0, Mean Difference: −19.15, Sum of positive ranks: 0, Sum of negative ranks: 36, Z-value: −2.5205, and Sample Size (N): 8. Therefore, since the value of W is 0 which is less than the critical value for W_{critical} at $N = 8$ ($p < 0.05$) which is 3, the proposed approach performed better than the Bag of Words approach. After comparing the F measure of the proposed LeNFEM approach with 3N-gram, the Wilcoxon Sign Rank test gave; W-value: 0, Mean Difference: −20.38, Sum of positive ranks: 0, Sum of negative ranks: 36, Z-value: −2.5205, Sample Size (N): 8. Therefore, we use W value since the sample of the results was less than ten(N = 8) and because the value of W is 0 which is less than the critical value for W_{critical} at $N = 8$ ($p < 0.05$) which is 3, the proposed approach produced better performance than the 3N-gram approach. Similarly, the F measure performance of the proposed LeNFEM approach and the bag of words' polarities approach were compared, the Wilcoxon Sign Rank test gave; W-value: 0, Mean Difference: −18.58, Sum of positive ranks: 0, Sum of negative ranks: 36, Z-value: −2.5205, Sample Size (N): 8. Therefore, we use W value since the sample of the results was less than ten(N = 8) and because the value of W is 0 which is less than the critical value for $W_{critical}$ at $N = 8$ ($p < 0.05$) which is 3, the proposed approach produced better performance than the bag of words' polarities approach.

5.3 | Experiment results III: Comparison of binary occurrence and TF-IDF vectorization

From Section 5.2, it was evident that the proposed approach performed better than the baseline approaches considered. We then sought to investigate the performance of the binary occurrence and TF-IDF vectorization algorithms. This was done by comparing their performance in the four machine learning classifiers using the proposed approach. The results were as shown in Table 3.

When comparing the performance of the two vectorization algorithms, the binary occurrences algorithm was found to be superior to the TF-IDF algorithm, as it is evident in Table 3, with the highest F-Measure (88.64%) using the support vector machine classifier. It is also worth noting that even though the TF-IDF algorithm presented lower performance results than that of binary occurrences algorithm, it also showed promising results of greater than 80%.

5.4 | Experiment results IV: Performance analysis of feature selection in the proposed approach

The performance of the MRMR feature selection algorithm in the proposed approach was investigated in this experiment. Since it was evident that binary occurrences algorithm performed better than TF-IDF algorithm, the performance of the feature selection algorithm was evaluated by comparing the F-measure performance of the four machine learning classifiers using the proposed approach with feature selection and without feature selection. The results obtained were as shown in Table 4.

TABLE 3 Comparison of F-measure for the text vectorization algorithms using the proposed approach

Text vectorization algorithm	Machine learning classifier			
	Naïve Bayes	K – NN	Decision tree	Support vector machines
Binary occurrences	84.68+/− 0.671	87.43+/− 0.539	86.91+/− 0.331	**88.64**+/− 0.362
TF-IDF	81.75+/− 0.358	84.71+/− 0.350	86.71+/− 0.356	87.45+/− 0.330

TABLE 4 F-Measure performance analysis of feature selection

Machine learning classifier F-measure performance (%)	Proposed LeNFEM (with MRMR feature selection)	LeNFEM (without MRMR feature selection)
Naïve Bayes	84.68+/− 0.671	82.52 +/− 0.364
K – NN	87.43+/− 0.539	82.65 +/− 0.336
Decision tree	86.91+/− 0.331	81.11 +/− 0.525
Support vector machines	**88.64**+/− 0.362	82.76 +/− 0.351

TABLE 5 F- measure performance analysis of the proposed approach with various datasets

Machine learning classifier F-measure performance (%)	Proposed LeNFEM (with MRMR feature selection)		
	Yelp restaurant dataset	Amazon's products dataset	Imdb's movies dataset
Naïve Bayes	84.68+/− 0.671	83.84+/− 0.604	85.04+/− 0.562
K – NN	87.43+/− 0.539	88.12+/− 0.341	87.82+/− 0.354
Decision tree	86.91+/− 0.331	86.72+/− 0.323	87.02+/− 0.507
Support vector machines	88.64+/− 0.362	**90.15**+/− 0.512	89.01+/− 0.366

In Table 4, a comparative analysis is made for feature selection performance in terms of F-measure. The LeNFEM approach achieves F-measure of 88.64% with feature selection process. On the other hand, LeNFEM approach without feature selection achieved only 82.76%. From the results, LeNFEM with feature selection achieved better performance.

5.5 | Experiment results V: Performance analysis of the proposed approach with other datasets

In this experiment, the performance of the proposed LeNFEM (with MRMR feature selection) approach and using the binary occurrences algorithm was investigated with the products and movies datasets.[34] The performance was evaluated by comparing the F-measure performance of the four machine learning classifiers using the proposed approach and the three datasets. The results are shown in Table 5.

In Table 5, a comparative analysis is presented for the F-measure performance of the machine learning classifiers using the proposed approach with various datasets. The LeNFEM approach achieves F-measure of 88.64% with Yelp Restaurant Dataset, 90.15% with Amazon's Products Dataset and 89.01% with Imdb's Movies Dataset. From the results, the proposed approach very good performance with all the datasets used with Amazon's Products dataset having the highest F-measure (90.15%) for the support Vector Machine classifier.

6 | CONCLUSION AND FUTURE WORK

This paper proposed an approach based on static hybrid N-gram model and minimum redundancy maximum relevance feature selection algorithm (LeNFEM). The aim of the approach is to improve feature extraction and selection in Sentence level Sentiment Analysis. LeNFEM is based on N-gram model where a lexicon is used to identify from a sentence a three-word N-gram containing a sentiment term. The N-gram is expanded to form sentence features using the words, POS tags of the words and sentiment orientation of the words in the N-gram. A text vectorization algorithm is then used to convert the features to numerical values that can be used as input to a machine learning classifier.

The paper has presented illustrations and operation algorithms of the approach in short sentences sentiment analysis. Further, three possible scenarios regarding the identification of the N-grams were discussed. The approach makes it possible to identify a specific part of a subjective sentence that contains sentiment information. Semantic and contextual information of words from that part are utilized in sentiment analysis of the sentence. Experiments carried out to evaluate the feasibility of the approach were described and the results obtained presented. F-measure performance metric from supervised machine learning classifiers; Naïve Bayes, K-Nearest Neighbor, Decision Tree, and Support Vector Machines was used as an evaluation metric. The results obtained attested the viability of the approach compared to the baseline approaches chosen. Further Wilcoxon test carried out proved superiority of the approach. The comparative experiments done on public datasets (Yelp, Amazons and Imdb reviews datasets) showed that the proposed approach provided a high F-measure of 90.15%. Therefore, the approach can be used to accurately classify social media texts into negative or positive opinion.

The binary occurrence vectorization algorithm produced the best results compared to the TF-IDF algorithm. It is therefore recommended that the binary occurrence algorithm used as the vectorization algorithm in the proposed approach due to its demonstration of high metric results in the experiments. TF-IDF can also be used with the approach as an alternative algorithm since it provided better results when used with the proposed approach than with the baselines used.

Potential future works include: (a) evaluating the performance of the proposed approach with other classifiers such as deep learning algorithms or testing the approach using other datasets such as Facebook text data; (b) applying the approach in sentences containing two or more opinions in different parts of the same sentence; (c) investigating the application of the proposed approach in other challenging text classification problems like misinformation and truth classification of posts rather than the subjectivity; and (d) developing a framework that utilizes the proposed approach in real life architecture and deploys it in social media hosts like Facebook, Twitter and Instagram for automated real time text classification decisions like flagging or blocking certain posts.

PEER REVIEW INFORMATION

Engineering Reports thanks the anonymous reviewers for their contribution to the peer review of this work.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1002/eng2.12374.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in UCI Machine Learning Repository, Centre for Machine Learning and Intelligent Systems at https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences which was compiled by Kotzias et al.[34]

CONFLICT OF INTEREST

The authors declare no potential conflict of interest.

AUTHOR CONTRIBUTIONS

Ronald Mwangi: Conceptualization; formal analysis; investigation; methodology; project administration; supervision; validation; visualization; writing-original draft; writing-review and editing. **George Okeyo:** Conceptualization; data curation; formal analysis; investigation; methodology; supervision; writing-original draft; writing-review and editing.

ORCID

James Mutinda https://orcid.org/0000-0001-9926-8688

REFERENCES

1. Brindha S, Prabha K, Sukumaran S. A survey on classification techniques for text mining. Paper presented at: 3rd International Conference on Advanced Computing and Communication Systems (ICACCS-2016); January 22–23, 2016; Coimbatore, India.
2. Li G, Lin Z, Wang H, Wei X. A discriminative approach to sentiment classification. *Neural Process Lett*. 2020;51:749-758 (2020. https://doi.org/10.1007/s11063-019-10108-7.
3. Mingyong L, Yang J. An improvement of TFIDF weighting in text categorization. Paper presented at: 2012 International Conference on Computer Technology and Science (ICCTS 2012) IPCSIT, vol. 47; 2012; Singapore: IACSIT Press.
4. Kalarani P, Brunda SS. An overview on research challenges in opinion mining and sentiment analysis. *Int J Innov Res Comput Commun Eng*. 2015;3(10):2320-9801.
5. Haddi E, Liu X, Shi Y. The role of text preprocessing in sentiment analysis. *Procedia Comput Sci*. 2013;17:26-32.
6. Onan A̋, Serdar Kl. A feature selection model based on genetic rank aggregation for text sentiment classification. *J Inform Sci*. 2017;43(1):25-38. https://doi.org/10.1177/0165551515613226.
7. Bhadane C, Dalal H, Doshi H. Sentiment analysis: measuring opinions. *Procedia Comput Sci*. 2015;45:808-814.
8. Mozetič I, Grčar M, Smailovič J. Multilingual twitter sentiment classification: the role of human annotators. *PLoS One*. 2016;11(5):e0155036. https://doi.org/10.1371/journal.pone.0155036.
9. Bin Li, Yuan G. Improvement of TF-IDF algorithm based on Hadoop framework. Paper presented at: The 2nd International Conference on Computer Application and System Modelling; 2012.
10. Ankit NS. An ensemble classification system for Twitter sentiment analysis. *Procedia Comput Sci*. 2018;132(2018):937-947.
11. Chug A, Kohli S, Gupta S, Ahu P, Ahuja R. The impact of features extraction on the sentiment analysis. *Procedia Comput Sci*. 2019;152:341-348.
12. Tian X &Yanmei C. An improvement to TF-IDF: term distribution-based term weight algorithm. *J Softw*. 2011:6(3):413-420.
13. Ramos J. *Using TF-IDF to Determine Word Relevance in Document Queries*. Piscataway, NJ: Department of Computer Science, Rutgers University; 2003.
14. Bansala B, Srivastava S. Sentiment classification of online consumer reviews using word vector representations. *Procedia Comput Sci*. 2018;132(2018):1147-1153.

15. Ronghui J, Pan Z, Li CH, Liu L. An efficient method for document categorization based on Word2Vec and latent semantic analysis. Paper presented at: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing; 2015.
16. Parwez MA, Abulaish M, Jahiruddin. Multi-label classification of microblogging texts using convolution neural network. *IEEE Access*. 2019;7:68678-68691. https://doi.org/10.1109/ACCESS.2019.2919494.
17. Çoban Ö, Özel SA. An evaluation of character level N-gram termsets in text categorization. Paper presented at: 2018 International Conference on Artificial Intelligence and Data Processing (IDAP); 2018; Malatya, Turkey:1–6. doi:10.1109/IDAP.2018.8620827.
18. Guo S, Yao N. Generating word and document matrix representations for document classification. *Neural Comput Appl*. 2020;32:10087-10108 (2020. https://doi.org/10.1007/s00521-019-04541-x.
19. Elghannam F. Text representation and classification based on bi-gram alphabet. *J King Saud Univ Comput Inform Sci*. 2020;https://doi.org/10.1016/j.jksuci.2019.01.005.
20. Sharmaa AK, Chaurasiaa S, Srivastava DK. Sentimental short sentences classification by using CNN deep learning model with fine tuned Word2Vec. *Procedia Comput Sci*. 2020;167(2020):1139-1147.
21. Rintyarna BS, Sarno R, Fatichah C. Evaluating the performance of sentence level features and domain sensitive features of product reviews on supervised sentiment analysis tasks. *J Big Data* (2019. 2019;6:84. https://doi.org/10.1186/s40537-019-0246-8.
22. Khedkar S, Shinde DS. Deep learning and ensemble approach for praise or complaint classification. *Procedia Comput Sci*. 2020;167(2020):449-458.
23. Wu C, Wu F, Liu J, Huang Y, Xie X. Sentiment lexicon enhanced neural sentiment classification. Paper presented at: Proceedings of the 28th ACM International Conference on Information and Knowledge Management; 2019; New York, NY: Association for Computing Machinery:1091–1100. doi:10.1145/3357384.3357973.
24. Ren Z, Zeng G, Chen L, Zhang Q, Zhang C, Pan D. A lexicon-enhanced attention network for aspect-level sentiment analysis. *IEEE Access*. 2020;8:93464-93471. https://doi.org/10.1109/ACCESS.2020.2995211.
25. Zhao Z, Liu T, Li S, Li B, Du X. Guiding the training of distributed text representation with supervised weighting scheme for sentiment analysis. *Data Sci Eng*. 2017;2:178-186. https://doi.org/10.1007/s41019-017-0040-6.
26. Li Y, Pan Q, Yang T, Wang S, Tang J, Cambria E. Learning word representations for sentiment analysis. *Cogn Comput*. 2017;9:843-851. https://doi.org/10.1007/s12559-017-9492-2.
27. Gallego SR, Lastra L, Martinez RD, et al. Fast-MRMR: fast minimum redudancy maximum relevance algorithm for high-dimensinality big Data. *Int J Intell Syst*. 2016;32(2):134-152.
28. Zhou H, Wang X, Zhang Y. Feature selection based on weighted conditional mutual information. *Appl Comput Informat*. 2020.
29. Noaman HM, Sarhan SS, Rashwan MAA. Enhancing recurrent neural network-based language models by word tokenization. *Hum Cent Comput Inf Sci*. 2018;8:12. https://doi.org/10.1186/s13673-018-0133-x.
30. Aisopos F, Tzannetos D, Violos J, Varvarigou T. Using n-grams graphs for sentiment analysis: an extended study on Twitter. Paper presented at: IEEE Second International Conference on Big Data Computing Services and Applications; 2016: IEEE.
31. Milos R, Mohamed G, Nenad F, Zoran O (2017). Minimum redundancy maximum relevance feature selection approach for temporal gene expression data. *BMC Bioinformat*. 18: 9. Published online 2017 Jan 3. https://doi.org/10.1186/s12859-016-1423-9 PMCID: PMC5209828
32. Nagamanjula R, Pethalakshmi A. A novel framework based on bi-objective optimization and LAN2FIS for Twitter sentiment analysis. *Soc Netw Anal Min*. 2020;10:34 (2020. https://doi.org/10.1007/s13278-020-00648-5.
33. Houda O, Omar N, Phillippe B. Minimum redundancy and maximum relevance for single and multi-document Arabic text summarization. *J King Saudi Univ Comput Inform Sci*. 2014;26:450-461.
34. Kotzias D, Denil M, Freitas ND, Smyth P. From group to individual labels using deep features. Paper presented at: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2015.
35. Github P. 2017. https://github.com/jeffreybrean/twitter-sentimentanalysistutorial-201107/data/opinion-lexicon-English.
36. Egejuru NC, Balogun JA, Mhambe PD, Asahiah FO, Idowu PA. Model for prediction of cataracts using supervised machine learning algorithms. *Comput Inform Syst Dev Informat Allied Res J*. 2017;8(3):47-62.

How to cite this article: Mutinda J, Mwangi W, Okeyo G. Lexicon-pointed hybrid N-gram Features Extraction Model (LeNFEM) for sentence level sentiment analysis. *Engineering Reports*. 2021;e12374. https://doi.org/10.1002/eng2.12374