Enhancing sentiment analysis in Kannada texts by feature selection

Sunil Mugalihalli Eshwarappa[1,3], Vinay Shivasubramanyan[2]

[1]Department of Computer Science and Engineering, PES College of Engineering, Visvesvaraya Technological University, Belagavi, India
[2]Department of Information Science and Engineering, PES College of Engineering, Visvesvaraya Technological University, Belagavi, India
[3]Software Engineer, Wipro Limited, Bangalore, India

Article Info

Article history:

Received Apr 18, 2024
Revised Jul 30, 2024
Accepted Aug 6, 2024

Keywords:

BERT
Kannada
Probability clustering
SemEval 2014
Sentiment analysis

ABSTRACT

In recent years, there has been a noticeable surge in research activities focused on sentiment analysis within the Kannada language domain. The existing research highlights a lack of labelled datasets and limited exploration in feature selection for Kannada sentiment analysis, hindering accurate sentiment classification. To address this gap, the study aims to introduce a novel Kannada dataset and develop an effective classifier for improved sentiment analysis in Kannada texts. The study presents a new Kannada dataset from SemEval 2014 Task4 using Google Translate. It then introduces a modified bidirectional encoder representation from transformers BERT for Kannada dataset called as Kannada-BERT (K-BERT). Further, a probability-clustering (PC) approach is presented to extract the topics and its related aspects. Both the K-BERT classifier and PC approach were merged to attain a K-BERT-PC classifier, integrating a modified BERT model and probability clustering approach for achieving better results. Experimental results demonstrate that K-BERT-PC achieves superior performance in polarity and sentiment analysis accuracy, with an impressive accuracy rate of 91%, surpassing existing classifiers. This work contributes by providing a solution to the scarcity of labelled datasets for Kannada sentiment analysis and introduces an effective classifier, K-BERT-PC, for enhanced sentiment analysis outcomes in Kannada texts.

Corresponding Author:

Sunil Mugalihalli Eshwarappa
Department of Computer Science and Engineering, PES College of Engineering, Visvesvaraya Technological University
Belagavi-590018, India
Email: suni.mghalli@gmail.com

1. INTRODUCTION

Language diversity is a fundamental aspect of human communication, reflecting the rich cultural diversities that defines societies worldwide. From Asia to Africa, Europe to the Americas, people speak a plethora of languages, each carrying its unique cultures, expressions, and sentiments [1]. This linguistic diversity not only enriches their global heritage but also presents intriguing challenges, particularly in the realm of sentiment analysis. In a country like India, known for its cultural mosaic, linguistic diversity takes on an even more pronounced form. With 28 states and 8 union territories, India has a staggering variety of languages, with each state often having its distinct linguistic identity. From Hindi in the north to Tamil in the south, from Bengali in the east to Gujarati in the west, the linguistic landscape of India is a testament to its vibrant heritage and regional diversity. In today's digital age, where individuals express their opinions and emotions on a myriad of online platforms, sentiment analysis plays a crucial role [2]. People share their sentiments on e-commerce platforms about products they purchase [3], voice their concerns on social media platforms regarding political issues [4], and engage in discussions on video entertainment sites about various content, be it movies, music, or vlogs [5]. These online interactions provide a treasure trove of data for understanding public sentiments and opinions [6].

However, a significant challenge arises when these sentiments are expressed in native languages [7]. Many sentiment analysis classifiers struggle to comprehend and analyze text written in languages other than English, which can limit their effectiveness in capturing the true sentiment of the users [8]. This language barrier poses an issue in accurately collecting public sentiments, especially in multilingual and multicultural societies like India. To address this challenge, researchers and data scientists have turned to machine-learning (ML) [9], [10] and deep-learning (DL) [11], [12] approaches. By collecting data from newspapers, websites, blogs, and other online sources, they train models to classify sentiments expressed in native languages such as Hindi, Malayalam, Tamil, and Kannada (part of the Dravidian language family) [13]. However, despite these efforts, the availability of standard datasets for sentiment analysis in Kannada, for example, remains limited [14]. Furthermore, existing datasets often lack labelled sentiment polarity, which is essential for training sentiment analysis models. Labelling datasets with positive and negative sentiment requires significant time and effort, further complicating the process of building accurate classifiers for native languages [15]. Additionally, while some classifiers may show promising results, they may focus on specific domains like e-commerce sites and overlook other platforms like social media or video content sites, leading to a lack of diversity in sentiment analysis approaches [16]. As a result, there is a pressing need for more comprehensive research that considers diverse linguistic contexts, selects relevant features across various domains, and addresses the challenges posed by native language sentiment analysis. By overcoming these issues, more deeper insights into public sentiments across different cultural and linguistic landscapes, facilitating more accurate sentiment analysis in diverse digital environments can be acquired. To address the aforementioned challenges, this work makes the following contributions: i) Providing a standard Kannada dataset that includes sentiment aspects, polarity labels, and overall sentiment analysis of reviews; ii) Considering various features and selecting the most relevant ones to achieve optimal results within the given domain; iii) Proposing a novel classifier specifically designed for Kannada language sentiment analysis; and iv) Conducting a comparative analysis of the proposed classifier with existing ML approaches, evaluating performance metrics such as accuracy, precision, recall, and F1-Score.

By combining these contributions, this work aims to advance the field of sentiment analysis in Kannada and address the unique challenges posed by native language sentiment classification. The manuscript is structured as follows: section 2 presents the literature survey, providing insights into recent works and advancements in Kannada sentiment analysis, section 3 delves into the methodology adopted in this work, outlining the overall sentiment classification process by considering relevant features. In section 4, the results of the proposed classifier are presented and discussed. Finally, section 5 provides the conclusion of the work, summarizing the key findings, contributions, limitations, and future directions of the research.

2. LITERATURE SURVEY

The research conducted in [17] explored the domain of sentiment prediction by utilizing a dataset consisting of code-mixed Kannada-English text sourced extracted from Twitter. This dataset was analyzed with corresponding sentiments for every post. This work utilized different features such as repetitive characters, word and character n-grams. The efficiency of these approaches was subsequently assessed using support-vector-machines (SVM) and long short-term memory (LSTM). The results of the study indicated that SVM attained an average precision value that ranged from 0.22 to 0.29. Chandrika *et al.* [18] aimed to delve into the complex realm of sentiments conveyed through the biography or narratives published in the Kannada language. Their proposed method involved the use of a rule-based strategy to categorize sentiments in Kannada documents. The achieved accuracy score of 85% was quite noteworthy, especially when compared to the bag-of-words (BoW) method which only achieved an accuracy score of 67%. The need to conduct analysis of sentiment on Kannada internet pages, particularly news websites, was highlighted in [19]. The Decision-Tree (DT) approach was implemented by the researchers, incorporating an optimized data-dictionary. The precision rate achieved was 0.85, indicating the proportion of correctly predicted positive instances out of the total predicted positive instances. Sunil and Vinay [20] investigated sentiment evaluation on translated internet-movie database (IMDB) movie reviews in Kannada, in addition to reviews sourced from multiple reputable websites. The researchers put forward an ensemble classification approach, which involved combining multiple classification approaches, and reported a notable accuracy score of 89%. In [21], delved into the domain of sentiment evaluation encompassing the Kannada, Hindi and English languages. To accomplish this task, a neural network architecture based on convolutional-neural-network (CNN) and LSTM was employed. The approach developed in their research demonstrated remarkable levels of accuracy, with Hindi datasets achieving a remarkable accuracy level of 99%, English datasets achieving an accuracy level of 95%, and Kannada datasets achieving accuracy level of 99%.

Chakravarthi *et al.* [22] presented a novel dataset that is multilingual and manually classified into different categories of polarity. This dataset focuses on the Dravidian languages and was obtained through social-media reviews. The researchers employed conventional ML approaches along with transformer approaches such as cross-lingual language model, self-supervised cross-lingual model, bidirectional encoder representations from transformers (BERT), character BERT, distil BERT, and optimized BERT (RoBERTA) to conduct sentiment evaluation and inappropriate language identification. The effectiveness of these approaches was evaluated on datasets containing code-mixed text. Moreover, through the utilization of a constrained Kannada corpora, Shankar and Swamy [23] employed SVM to enhance sentiment categorization. Their approach yielded notable results, with a classification accuracy of 68%, precision of 80%, recall of 62%, along with an F-Score of 70%. Shankar *et al.* [24] investigated the application of ML and ensemble approaches for sentiment assessment on COVID-19 data in the Kannada language. Various approaches including logistic regression (LR), random forest (RF), XGBoost (XGB), AdaBoost (AB), voting classifier (VC), and gradient boosting (GB), were employed to analyze the data. The results indicated that the achieved accuracy, precision, recall, and F-Score ranged from 62% to 68%. Finally, a study conducted by Sreelakshmi *et al.* [25] utilized different transformer-based embedding approaches to analyze and identify such content. The researchers discovered that the MuRIL pre-trained embedding (a part of BERT) exhibited uniform and impressive performance throughout various datasets. It achieved remarkably high accuracy rates, varying from 66% to 96%, when applied to Kannada, Tamil and Malayalam datasets. From the above literature survey, it is seen that very limited labelled data is available for Kannada sentiment analysis. Also, it is seen that most of the works have utilized BERT approach to achieve better accuracy rates. Some of the works have used ML classifiers, but have achieved less accuracy which ranges from 66% to 80%. Hence, in this work, we present a novel Kannada dataset by converting the SemEval 2014 Task 4 dataset. Further, in this work we present a model based on BERT called as Kannada-BERT probability clustering classifier (K-BERT-PC). The methodology for the K-BERT-PC classifier is discussed in the next section.

3. METHOD

3.1. Architecture

The architecture for sentiment classification using K-BERT-PC classifier is presented in Figure 1. It begins with the acquisition of raw text data from the SemEval 2014 Task 4 dataset, renowned for its rich aspects, aspect terms, and polarity annotations. The next step involves the conversion of this English data to Kannada using Google Translate, expanding the linguistic diversity of the dataset. Following data, the raw data undergoes preprocessing to enhance its suitability for analysis. Further, K-BERT and probability-clustering (PC) model utilized to evaluate sentiment polarity (positive, negative, or neutral) for each text segment.

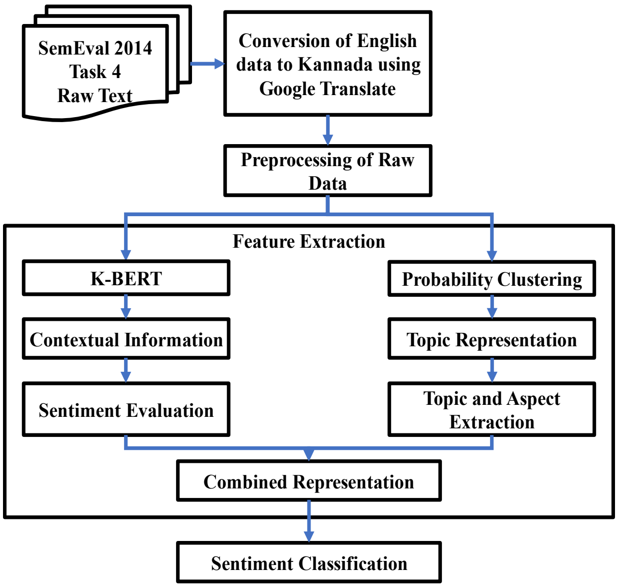

Figure 1. Architecture of K-BERT-PC classifier

3.2. Dataset collection

In this study, this work initially focused on the SemEval 2014 Task 4 dataset [26]. This dataset is renowned for its rich collection of aspects, aspect terms, and polarities, making it a valuable resource for sentiment analysis and related tasks. To further enhance the dataset's utility and diversity, this work translated the SemEval dataset from English to Kannada using Google Translate. This translation process allowed us to create both training and testing data in Kannada, thereby expanding the scope and applicability of the works analysis. The decision to utilize the SemEval 2014 dataset stems from its comprehensive coverage of various aspects and different polarities. By leveraging this dataset and translating it into Kannada, this work aimed to explore sentiment analysis within the Kannada language domain while benefiting from the well-structured and annotated SemEval dataset.

3.3. Preprocessing

In the preprocessing stage, the first step involved tokenization, which is the process of breaking down a sentence into individual words or tokens. This step aids in structuring the text for further analysis by separating it into its constituent elements. Following tokenization, normalization was performed to convert uppercase words to lowercase, ensuring uniformity in the text data. Additionally, symbols were removed from the raw data to eliminate noise and streamline the text. Subsequently, the data underwent stemming, lemmatization, and stopword removal processes. Stemming involved reducing words to their root or base form, lemmatization involved grouping inflected forms of words to their lemma or dictionary form, and stopword removal involved eliminating common words that do not contribute significantly to the meaning of the text.

3.4. K-BERT

The BERT approach is an ML approach that utilizes transformers for pre-training textual data in the field of natural-language-processing (NLP) [27]. This work aims to optimize the performance of BERT by incorporating additional layers such as dense and classification layers and presenting a novel BERT model called as K-BERT for the Kannada sentiment analysis as presented in Figure 2.

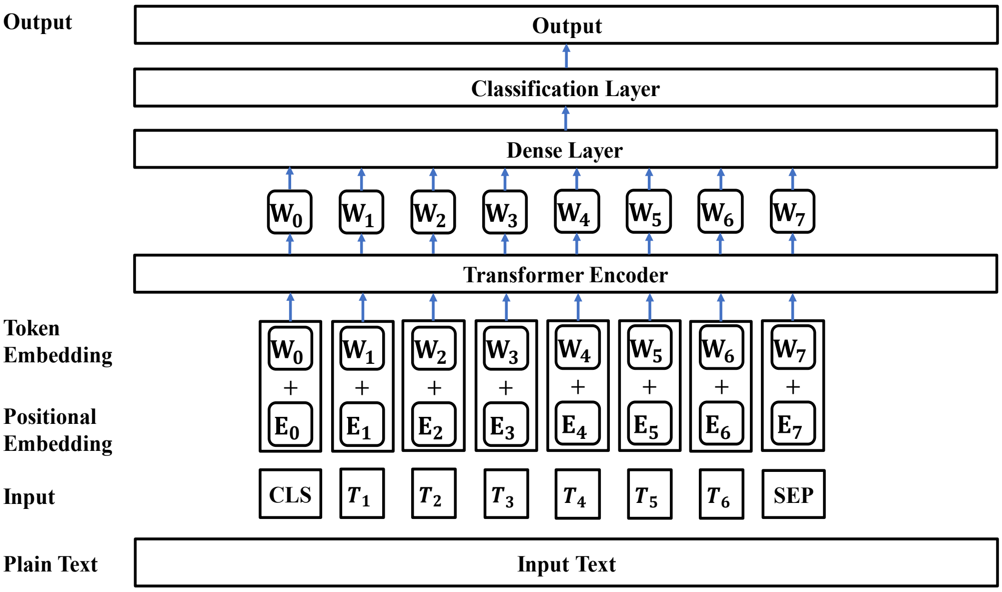

Figure 2. K-BERT classifier

Furthermore, the transformer-encoder utilized in K-BERT approach has been subjected to fine-tuning specifically for the Kannada dataset in this study. Initially, in this work, the pre-processed review was taken as input for the K-BERT. Instead of using the unigrams as input, the K-BERT approach uses the $Trigrams'n'Tags$ (TnT) part-of-speech (PoS) tagging method [28], [29]. As presented in Figure 2, there are two tokens present in input, i.e., at right a separator token (SEP) and at left a classification token (CLS). The function of SEP is to act as a separator between each trigram in the input text. The SEP token ensures that each trigram is distinct and identifiable within the input sequence. Also, the function of CLS is to classify or represent each trigram in such a way that it contributes to achieving a meaningful classification for the entire sentence or text. The CLS token is typically used in BERT models as the first token in the input sequence and is associated with tasks like sentiment analysis, where the model learns to classify the sentiment of the entire sentence based on the representation generated by the CLS token. Additionally, this work considers one-hot encoding approach for trigrams. This encoding method helped in creating a numerical representation of trigrams that were fed into K-BERT for sentiment analysis tasks. Further, the trigrams, were converted into positional and token embeddings. In positional embedding, the input sentence is defined as $E = \{E_1, E_2, E_3, \dots E_n\} \in R^{n\times dim_E}$, where E_n represents the position of word, R^n represents review sentence and dim_{E} represents dimension of position embedding. Moreover, the positional embedding is evaluated as (1), (2).

$$PE(pos, 2_i) = sin\,(\frac{pos}{1000^{\frac{2_i}{dh}}}) \quad (1)$$

$$PE(pos, 2_i + 1) = cos\,(\frac{pos}{1000^{\frac{2_i}{dh}}}) \quad (2)$$

where, pos represents the position, i represents embedding dimension, 2_i and $2_i + 1$ indicate alternating dimensions or indices within the positional embedding. The equation (1) and (2) generates a sinusoidal or cyclical pattern in the positional embedding based on the position of tokens in the input sequence. Further, the token embeddings are defined as $\mathrm{W} = \{\mathrm{W}_1, \mathrm{W}_2, \mathrm{W}_3, \dots \mathrm{W}_\mathrm{n}\} \in R^{n\times dim_\mathrm{W}}$, where W_n represents tokens for the words and dim_W represents dimension of token embedding. Both these embeddings, i.e., E and W are concatenated. Hence, the concatenated embeddings are achieved as dim_c=dim_E+dim_E. The dim_c is used as an input for the transformer encoder layer.

In the standard BERT transformer encoding layer, each component plays a crucial role in processing the input data and extracting meaningful representations. The input embedding layer converts tokens or words in the input sequence into high-dimensional vector representations, encoding semantic information based on pre-trained or learned embeddings. The positional encoding layer provides relative positional information, aiding in distinguishing token order and ensuring distinct embeddings for tokens with similar meanings but different positions. The multi-head attention layer computes attention scores between tokens, allowing focus on relevant parts of the input sequence and capturing complex dependencies with multiple attention heads. Finally, the fully connected layer, comprising the “add and norm” layer and the feed-forward neural network, integrates attention outputs with input embeddings, applies normalization, and utilizes non-linear transformations to capture intricate patterns, thus enabling the extraction of hierarchical representations vital for various natural language processing tasks. In this work, the multi-head attention layer has been fine-tuned for sentiment analysis. The multi-head attention layer has been shown in Figure 3.

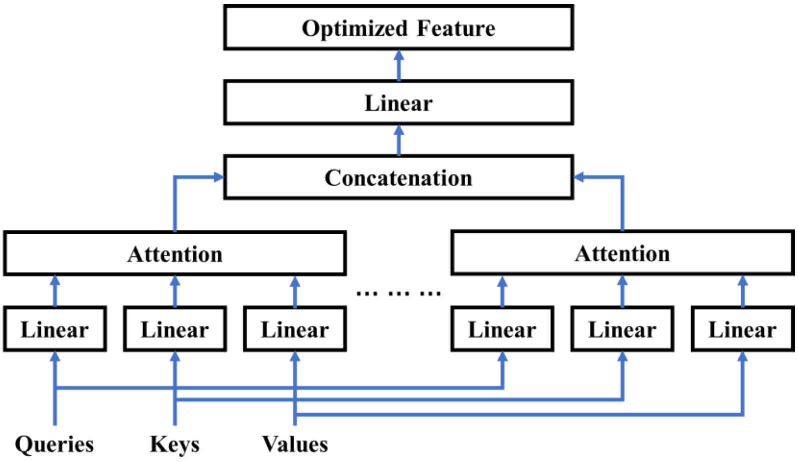

Figure 3. Multi-head attention framework

In the multi-head attention process, distinct query vectors are defined for different sequence tasks. Typically, the input to an attention function comprises a query q and a set of keys, where each key is denoted as $k = \{k_1, k_2, k_3, \ldots k_n\}$. It is important to note that in NLP tasks, $key = value$. The scaled dot product attention for multi-head attention is computed by calculating the weights through the dot product operation between the $keys$ and the query q. This process results in a distribution of attention scores, which is determined using (3). In (3), the variable s signifies the semantic gain, a parameter crucial for determining the semantic relevance between a word and its context. This calculation process is referred as the scoring process which plays a pivotal role in assessing the significance of the word within its surrounding context. The scoring process itself is computed using (4).

$$Attention(k, q) = Softmax(s(k, q)) \tag{3}$$

$$s = ktanh\,([k_i, q_j].W_a) \tag{4}$$

Here, s denotes the semantic gain or relevance score, k represents the key vector, $[k_i, q_j]$ represents the concatenation of the key vector k_i and query vector q_j, W_a represents the weight matrix associated with the attention process, and $tanh$ denotes the hyperbolic tangent function. The dot product $[k_i, q_j]$. W_a followed by the hyperbolic tangent activation function computes the final relevance score s based on the interaction between the key and query vectors weighted by the attention matrix W_a. In this work, eight heads were employed within the multi-head attention process, as individual parameters cannot effectively share a single head. This design choice stems from the utilization of linear projections to glean information from distinct subspaces, where the values of keys (k) and queries (q) tend to exhibit rapid fluctuations. The functioning of the multi-head attention process is further defined through (5) and (6), explaining its role in processing and contextualizing input data effectively.

$$head_i = Attention(k, q) \tag{5}$$

$$MHA = (head_1 \oplus head_2 \oplus head_3 \ldots \oplus head_h).W_o \tag{6}$$

This (5) represents the calculation of an attention head $(head_i)$ within the multi-head attention process. The attention function is applied to the key (k) and query (q) to compute the attention scores. The equation (6) represents the multi-head attention output. The outputs of all attention heads $(head_1, head_2, head_3, \ldots, head_n)$ are concatenated $\oplus$ and then multiplied by a weight matrix W_o to obtain the final output of the multi-head attention process. Also, $h \in [1,8]$ signifies the range of values for the weight W_o which belongs to $R^{(dim_{(hidden)})\times(dim_{(hidden)})}$. Here, $dim_{(hidden)}$ represents the hidden dimension. The purpose of introducing the attention function is to offer a more direct pathway for inputs, thus addressing the vanishing-gradient problem. This attention mechanism directs focus by enhancing the hidden and output states from preceding blocks with a context vector C_i as defined by (7). The equation (7) represents the computation of the context vector C_i, which is obtained by summing up the attention-weighted representations of the input tokens h_j based on their relevance scores a_{ij} for the i^{th} output token. This attention computation is done using (8) and (9).

$$C_i = \Sigma_{j=1}^{T_x} a_{ij} h_j \tag{7}$$

$$a_{ij} = softmax(e_{ij}) = \frac{exp\,(e_{ij})}{\Sigma_{j=1}^{T_x} exp\,(e_{ik})} \tag{8}$$

$$e_{ij} = f(S_{i-1}, h_i) \tag{9}$$

The (8) computes the attention weights a_{ij} for each token pair (i, j) in the input sequence. The softmax function ensures that the weights sum up to 1, representing the importance or relevance of each token in the sequence with respect to the current token being processed. Further, in (9), the function f represents the alignment model, which assesses the scores indicating how well the inputs surrounding the j output at the i position align or correlate. The hidden state S_{i-1} is derived from the previous time step and contributes to the computation of the alignment scores. Moreover, this work has expanded upon the established BERT framework, as illustrated in Figure 2, by integrating a classification layer and a dense layer. The main purpose of this enhancement is to empower the model to produce tag sequences for input sentences. To mitigate the potential of overfitting, a dropout normalization technique has been specifically applied to the dense layer. After the K-BERT process the sentiment polarity is achieved, i.e., whether it is positive, negative or neutral.

3.5. Probability clustering

Probability-clustering (PC) has emerged as a powerful tool for extracting aspects and discovering hidden patterns [30], [31] through different reviews R. The PC possesses the capability to perform data analysis and generate topic-vectors by utilizing a sequence of probability distributions. The PC approach has demonstrated remarkable efficacy in the domain of text-clustering and topic-modeling. In PC, the η and α represents the Dirichlet parameter (probability distribution), β_K represents topics present in the dataset cluster, n denotes the n^{th} positioned word in the corpus review R, $W_{R,n}$ denotes the observed words in the cluster of reviews, $Z_{R,n}$ denotes per-word topic and θ_R represents per-document topic. This work assumes the prior-distribution of reviews as Dirichlet-distribution [32]. Hence, the overall distribution for PC model is evaluated using Gibbs-Sampling [33] as presented in (10). Also, the coherence score is evaluated using (11) [32].

$$p(\mathcal{W}_R, z_R, \theta_R, \beta_k | \alpha, \beta) = \prod_{n=1}^{N} p(\theta_R | \alpha) p(z_{Rn} | \theta_R) p(\beta_k | \beta) p(\mathcal{W}_{Rn} | \beta_k) \tag{10}$$

$$\phi S_i(\vec{u}, \vec{w}) = \frac{\sum_{i=1}^{|W|} u_i \cdot w_i}{||\vec{u}||_2 \cdot ||\vec{w}||_2} \tag{11}$$

where ϕ represents the confirmation measure, S_i represents the pair of vectors u and w. $\vec{u}$ and $\vec{w}$ represent cosine vector similarity. After the evaluation of the coherence score, the cluster of topics having highest coherence score is selected and then the topics are selected and its respective aspects are extracted.

3.6. K-BERT-PC

To enhance sentiment classification and topic mining extraction, this work has integrated the K-BERT classifier with PC using a combined representation approach. The combined representation approach merges the strengths of K-BERT and PC to achieve comprehensive text analysis. K-BERT, a customized BERT model for Kannada sentiment analysis, excels in understanding contextual information and sentiment expressions within text data. On the other hand, PC is better for topic modeling and aspect extraction, uncovering hidden patterns and thematic structures within textual content. By integrating these two models, the combined representation approach leverages K-BERT’s deep contextual understanding for sentiment classification and PC’s capability to extract meaningful topics and aspects from text data. This enables a more holistic analysis, providing valuable insights into sentiment trends, thematic content, and underlying sentiments associated with specific topics or aspects in textual content. Overall, the combined representation approach maximizes the analytical capabilities and utility of both K-BERT and PC, enhancing the overall text analysis process. In the next section, the results for the K-BERT-PC classifier are discussed and compared with existing works.

4. RESULTS AND DISCUSSION

The K-BERT-PC classifier was evaluated on a Windows 11 operating system consisting of 16 GB of RAM. The Anaconda platform was considered for building the classifier. The Anaconda built in Python framework was used. Most of the Natural Language Toolkit (NLTK) libraries were used for evaluation. Further, for evaluation of K-BERT-PC, the following performance metrics were used.

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \tag{12}$$

$$Precision = \frac{TP}{TP+FP} \tag{13}$$

$$Recall = \frac{TP}{TP+FN} \tag{14}$$

$$F-Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{15}$$

where TP is true positive, FP is false positive, TN is true negative, and FN is false negative. The results were compared with different classifiers presented in [24]. Additionally, Table 1 presents a subset of testing examples extracted from the dataset. This table offers a glimpse into the actual data used for evaluating the classifier's performance, showcasing a representative sample of instances with their corresponding sentiment labels.

Table 1. Test dataset

SL. No.	English	Kannada
1	Strong build though which really adds to its durability.	ಬಲವಾದ ನಿರ್ಮಾಣ ಆದರೂ ಇದು ನಿಜವಾಗಿಯೂ ಅದರ ಬಾಳಿಕೆಗೆ ಸೇರಿಸುತ್ತದೆ.
2	I can say that I am fully satisfied with the performance that the computer has supplied.	ಕಂಪ್ಯೂಟರ್ ಒದಗಿಸಿದ ಕಾರ್ಯಕ್ಷಮತೆಯೊಂದಿಗೆ ನಾನು ಸಂಪೂರ್ಣವಾಗಿ ತೃಪ್ತನಾಗಿದ್ದೇನೆ ಎಂದು ನಾನು ಹೇಳಬಲ್ಲೆ.
3	The battery life is excellent- 6-7 hours without charging.	ಬ್ಯಾಟರಿ ಬಾಳಿಕೆ ಅತ್ಯುತ್ತಮವಾಗಿದೆ - ಚಾರ್ಜ್ ಮಾಡದೆಯೇ 6-7 ಗಂಟೆಗಳು.
4	The Apple engineers have not yet discovered the delete key.	ಆಪಲ್ ಇಂಜಿನಿಯರ್‌ಗಳು ಇನ್ನೂ ಡಿಲೀಟ್ ಕೀಯನ್ನು ಕಂಡುಹಿಡಿದಿಲ್ಲ.
5	Only problem that I had was that the track pad was not very good for me, I only had a problem once or twice with it, but probably my computer was a bit defective.	ನಾನು ಹೊಂದಿದ್ದ ಒಂದೇ ಸಮಸ್ಯೆಯೆಂದರೆ ಟ್ರ್ಯಾ ಕ್ ಪ್ಯಾಡ್ ನನಗೆ ತುಂಬಾ ಒಳ್ಳೆಯದಲ್ಲ, ನನಗೆ ಅದರೊಂದಿಗೆ ಒಂದು ಅಥವಾ ಎರಡು ಬಾರಿ ಮಾತ್ರ ಸಮಸ್ಯೆ ಇತ್ತು, ಆದರೆ ಬಹುಶಃ ನನ್ನ ಕಂಪ್ಯೂಟರ್ ಸ್ವಲ್ಪ ದೋಷಯುಕ್ತವಾಗಿತ್ತು.
6	Lastly, Windows 8 is annoying.	ಕೊನೆಯದಾಗಿ, ವಿಂಡೋಸ್ 8 ಕಿರಿಕಿರಿ.

Table 2 consists of six examples in both English and Kannada languages, along with their corresponding polarity scores and sentiments. Upon analysis, it is observed that examples 1, 3, and 4 convey positive sentiments with high polarity scores (0.9564, 0.8954, and 0.9875, respectively), indicating satisfaction or approval. Conversely, examples 2, 5, and 6 express negative sentiments with lower polarity scores (0.2358, 0.1285, and 0.025, respectively), highlighting dissatisfaction or criticism. Additionally, example 5 exhibits different sentiment, mentioning a specific problem while acknowledging the possibility of a defective product. Overall, the dataset showcases a range of sentiments expressed by users, providing valuable insights into their experiences and perceptions.

Table 2. Sentiment evaluation

SL. No.	English	Kannada	Polarity Score/Sentiment
1	Strong build though which really adds to its durability.	ಬಲವಾದ ನಿರ್ಮಾಣ ಆದರೂ ಇದು ನಿಜವಾಗಿಯೂ ಅದರ ಬಾಳಿಕೆಗೆ ಸೇರಿಸುತ್ತದೆ.	0.9564/Positive
2	The Apple engineers have not yet discovered the delete key.	ಆಪಲ್ ಇಂಜಿನಿಯರ್‌ಗಳು ಇನ್ನೂ ಡಿಲೀಟ್ ಕೀಯನ್ನು ಕಂಡುಹಿಡಿದಿಲ್ಲ.	0.2358/Negative
3	I can say that I am fully satisfied with the performance that the computer has supplied.	ಕಂಪ್ಯೂಟರ್ ಒದಗಿಸಿದ ಕಾರ್ಯಕ್ಷಮತೆಯೊಂದಿಗೆ ನಾನು ಸಂಪೂರ್ಣವಾಗಿ ತೃಪ್ತನಾಗಿದ್ದೇನೆ ಎಂದು ನಾನು ಹೇಳಬಲ್ಲೆ.	0.8954/Positive
4	The battery life is excellent- 6-7 hours without charging.	ಬ್ಯಾಟರಿ ಬಾಳಿಕೆ ಅತ್ಯುತ್ತಮವಾಗಿದೆ - ಚಾರ್ಜ್ ಮಾಡದೆಯೇ 6-7 ಗಂಟೆಗಳು.	0.9875/Positive
5	Only problem that I had was that the track pad was not very good for me, I only had a problem once or twice with it, but probably my computer was a bit defective.	ನಾನು ಹೊಂದಿದ್ದ ಒಂದೇ ಸಮಸ್ಯೆಯೆಂದರೆ ಟ್ರ್ಯಾ ಕ್ ಪ್ಯಾಡ್ ನನಗೆ ತುಂಬಾ ಒಳ್ಳೆಯದಲ್ಲ, ನನಗೆ ಅದರೊಂದಿಗೆ ಒಂದು ಅಥವಾ ಎರಡು ಬಾರಿ ಮಾತ್ರ ಸಮಸ್ಯೆ ಇತ್ತು, ಆದರೆ ಬಹುಶಃ ನನ್ನ ಕಂಪ್ಯೂಟರ್ ಸ್ವಲ್ಪ ದೋಷಯುಕ್ತವಾಗಿತ್ತು.	0.1285/Negative
6	Lastly, Windows 8 is annoying.	ಕೊನೆಯದಾಗಿ, ವಿಂಡೋಸ್ 8 ಕಿರಿಕಿರಿ.	0.025/Negative

The provided Figure 4 presents the accuracy scores of different classifiers used for sentiment analysis. From the figure, it is observed that the traditional ML classifiers such as RF, LR, XGB, AB, and GB achieve accuracy scores ranging from 0.6 to 0.67. Notably, the K-BERT-PC classifier outperforms all other classifiers with an impressive accuracy score of 0.91. Figure 5 presents the precision scores of different classifiers used for sentiment analysis. From figure, it is observed that the traditional ML classifiers such as RF, LR, XGB, AB, and GB achieve precision scores ranging from 0.6 to 0.67. The K-BERT-PC classifier stands out with the highest precision score of 0.92 among all classifiers. Figure 6 presents the recall scores of different classifiers used for sentiment analysis. Analyzing the figure reveals that traditional ML classifiers such as RF, LR, XGB, AB, and GB achieve recall scores ranging from 0.6 to 0.67. The K-BERT-PC classifier stands out with the highest recall score of 0.91 among all classifiers. Figure 7 shows the F1-Score. From figure, it is observed that traditional ML classifiers such as RF, LR, XGB, AB, and GB achieve F1-Scores ranging from 0.6 to 0.67. The K-BERT-PC classifier stands out with the highest F1-Score of 0.93 among all classifiers.

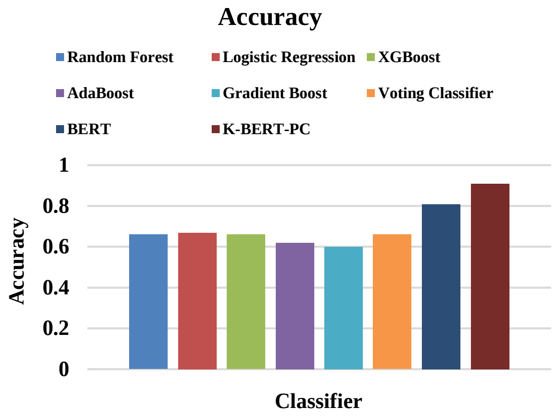

Figure 4. Accuracy

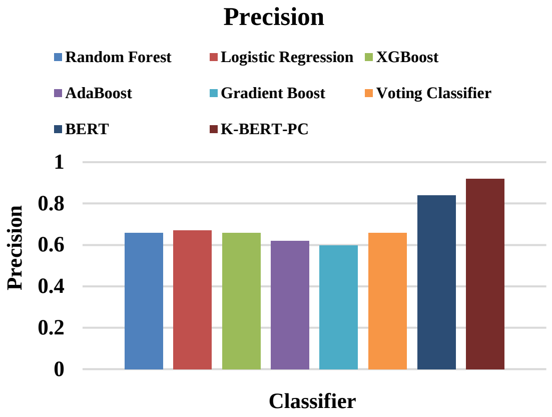

Figure 5. Precision

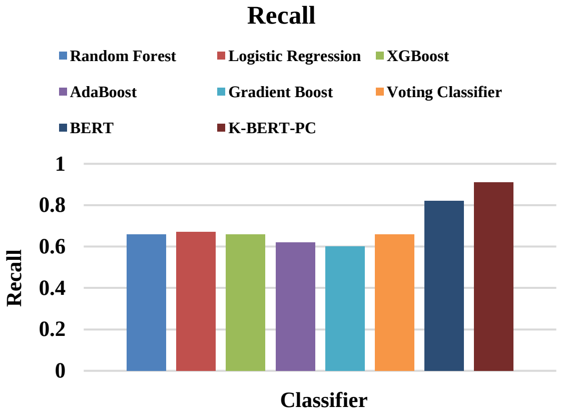

Figure 6. Recall

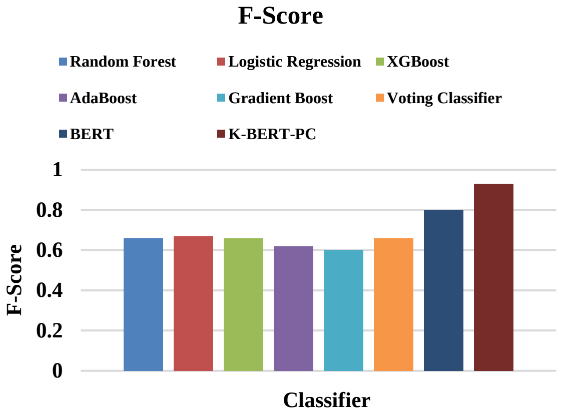

Figure 7. F-Score

5. CONCLUSION

In conclusion, it is evident from existing research that there exists a scarcity of labelled datasets specifically designed for Kannada sentiment analysis. Additionally, there has been limited exploration in terms of feature selection for sentiment analysis in Kannada. To address this issue, the present study introduces a novel Kannada dataset derived from the SemEval 2014 Task4 dataset through Google Translate. Furthermore, this work introduces the K-BERT-PC classifier, which incorporates a modified BERT model along with a probability clustering approach. The experimental results demonstrate that K-BERT-PC achieves superior performance in terms of polarity and sentiment analysis accuracy. Specifically, the K-BERT-PC classifier achieves an impressive accuracy rate of 91%, outperforming existing classifiers. For future research work, this work can be expanded to include testing on various other Kannada sources such as documents, newspapers, and reviews from different websites, thereby contributing to further advancements in Kannada sentiment analysis.

REFERENCES

[1] S. Kumawat, I. Yadav, N. Pahal, and D. Goel, “Sentiment analysis using language models: a study,” in *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Jan. 2021, pp. 984–988, doi: 10.1109/Confluence51648.2021.9377043.

[2] M. Rodríguez-Ibánez, A. Casánez-Ventura, F. Castejón-Mateos, and P.-M. Cuenca-Jiménez, “A review on sentiment analysis from social media platforms,” *Expert Systems with Applications*, vol. 223, Aug. 2023, doi: 10.1016/j.eswa.2023.119862.

[3] S. Elzeheiry, W. A. Gab-Allah, N. Mekky, and M. Elmogy, “Sentiment analysis for e-commerce product reviews: current trends and future directions,” *Preprints*, 2023, 2023051649, doi: 10.20944/preprints202305.1649.v1.

[4] F. Arias, M. Z. Nunez, A. Guerra-Adames, N. Tejedor-Flores, and M. Vargas-Lombardo, “Sentiment analysis of public social media as a tool for health-related topics,” *IEEE Access*, vol. 10, pp. 74850–74872, 2022, doi: 10.1109/ACCESS.2022.3187406.

[5] M. Wollmer *et al.*, “YouTube movie reviews: sentiment analysis in an audio-visual context,” *IEEE Intelligent Systems*, vol. 28,

no. 3, pp. 46–53, May 2013, doi: 10.1109/MIS.2013.34.

[6] J. Shaynn-Ly Kwan and K. Hui Lim, “Understanding public sentiments, opinions and topics about COVID-19 using Twitter,” in *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Dec. 2020, pp. 623–626, doi: 10.1109/ASONAM49781.2020.9381384.

[7] K. Rakshitha, R. H M, M. Pavithra, A. H D, and M. Hegde, “Sentimental analysis of Indian regional languages on social media,” *Global Transitions Proceedings*, vol. 2, no. 2, pp. 414–420, Nov. 2021, doi: 10.1016/j.gltp.2021.08.039.

[8] J. R. Jim, M. A. R. Talukder, P. Malakar, M. M. Kabir, K. Nur, and M. F. Mridha, “Recent advancements and challenges of NLP-based sentiment analysis: a state-of-the-art review,” *Natural Language Processing Journal*, vol. 6, Mar. 2024, doi: 10.1016/j.nlp.2024.100059.

[9] R. K. Das, M. Islam, M. M. Hasan, S. Razia, M. Hassan, and S. A. Khushbu, “Sentiment analysis in multilingual context: comparative analysis of machine learning and hybrid deep learning models,” *Heliyon*, vol. 9, no. 9, Sep. 2023, doi: 10.1016/j.heliyon.2023.e20281.

[10] D. Draskovic, D. Zecevic, and B. Nikolic, “Development of a multilingual model for machine sentiment analysis in the Serbian language,” *Mathematics*, vol. 10, no. 18, Sep. 2022, doi: 10.3390/math10183236.

[11] M. M. Agüero-Torales, J. I. Abreu Salas, and A. G. López-Herrera, “Deep learning and multilingual sentiment analysis on social media data: an overview,” *Applied Soft Computing*, vol. 107, Aug. 2021, doi: 10.1016/j.asoc.2021.107373.

[12] P. Dominic, N. Purushothaman, A. S. A. Kumar, A. Prabagaran, J. Angelin Blessy, and J. A, “Multilingual sentiment analysis using deep-learning architectures,” in *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Jan. 2023, pp. 1077–1083, doi: 10.1109/ICSSIT55814.2023.10060993.

[13] B. R. Chakravarthi *et al.*, “Findings of the sentiment analysis of Dravidian languages in code-mixed text,” *arXiv preprint arXiv:2111.09811*, 2021, doi: 10.48550/arxiv.2111.09811.

[14] A. Hande, R. Priyadharshini, and B. R. Chakravarthi, “KanCMD: Kannada CodeMixed dataset for sentiment analysis and offensive language detection,” in *Proceedings of the Third Workshop on Computational Modeling of People’s Opinions, Personality, and Emotion’s in Social Media*, 2020, pp. 54–63.

[15] J. Hartmann, M. Heitmann, C. Siebert, and C. Schamp, “More than a feeling: accuracy and application of sentiment analysis,” *International Journal of Research in Marketing*, vol. 40, no. 1, pp. 75–87, Mar. 2023, doi: 10.1016/j.ijresmar.2022.05.005.

[16] I. Dias, R. Rei, P. Pereira, and L. Coheur, “Towards a sentiment-aware conversational agent,” in *Proceedings of the 22nd ACM International Conference on Intelligent Virtual Agents*, Sep. 2022, pp. 1–3, doi: 10.1145/3514197.3549692.

[17] A. R. Appidi, V. K. Srirangam, D. Suhas, and M. Shrivastava, “Creation of corpus and analysis in code-mixed Kannada-English Twitter data for emotion prediction,” in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 6703–6709, doi: 10.18653/v1/2020.coling-main.587.

[18] C. P. Chandrika, J. S. Kallimani, H. P. Adarsha, A. Nagabhushan, and A. Chavan, “Rule-based approach for emotion detection for Kannada text,” 2021, pp. 463–472.

[19] P. Ranjitha and K. N. Bhanu, “Improved sentiment analysis for Dravidian language-Kannada using decision tree algorithm with efficient data dictionary,” *IOP Conference Series: Materials Science and Engineering*, vol. 1123, no. 1, Apr. 2021, doi: 10.1088/1757-899X/1123/1/012039.

[20] M. E. Sunil and S. Vinay, “Kannada sentiment analysis using vectorization and machine learning,” in *Sentimental Analysis and Deep Learning: Proceedings of ICSADL 2021*, 2022, pp. 677–689.

[21] S. Shetty *et al.*, “Sentiment analysis of Twitter posts in English, Kannada and Hindi languages,” in *Recent Advances in Artificial Intelligence and Data Engineering: Select Proceedings of AIDE 2020*, 2022, pp. 361–375.

[22] B. R. Chakravarthi *et al.*, “DravidianCodeMix: sentiment analysis and offensive language identification dataset for Dravidian languages in code-mixed text,” *Language Resources and Evaluation*, vol. 56, no. 3, pp. 765–806, Sep. 2022, doi: 10.1007/s10579-022-09583-7.

[23] Shankar R and S. Swamy, “Hybrid sentiment analysis: a novel integration of corpus-driven and machine learning approaches to perform sentiment analysis of Kannada Political Tweets,” *Tuijin Jishu/Journal of Propulsion Technology*, vol. 44, no. 5, pp. 3246–3259, 2023.

[24] Shankar R, S. Swamy, and S. Hegde, “Exploring sentiment analysis in Kannada language: a comprehensive study on COVID-19 data using machine learning and ensemble algorithms,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 11, pp. 21–29, 2024.

[25] K. Sreelakshmi, B. Premjith, B. R. Chakravarthi, and K. P. Soman, “Detection of hate speech and offensive language CodeMix text in Dravidian languages using cost-sensitive learning approach,” *IEEE Access*, vol. 12, pp. 20064–20090, 2024, doi: 10.1109/ACCESS.2024.3358811.

[26] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, “SemEval-2014 Task 4: aspect based sentiment analysis,” in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 27–35, doi: 10.3115/v1/S14-2004.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805.*, 2019, doi: 10.18653/v1/N19-1423.

[28] T. Brants, “TnT-a statistical part-of-speech tagger,” in *Proceedings of the sixth conference on Applied natural language processing -*, 2000, pp. 224–231, doi: 10.3115/974147.974178.

[29] T. Hariyanti, S. Aida, and H. Kameda, “Samawa language part of speech tagging with probabilistic approach: comparison of Unigram, HMM and TnT models,” *Journal of Physics: Conference Series*, vol. 1235, no. 1, Jun. 2019, doi: 10.1088/1742-6596/1235/1/012013.

[30] B. Ozyurt and M. A. Akcayol, “A new topic modeling based approach for aspect extraction in aspect based sentiment analysis: SS-LDA,” *Expert Systems with Applications*, vol. 168, Apr. 2021, doi: 10.1016/j.eswa.2020.114231.

[31] M. Venugopalan and D. Gupta, “An enhanced guided LDA model augmented with BERT based semantic strength for aspect term extraction in sentiment analysis,” *Knowledge-Based Systems*, vol. 246, Jun. 2022, doi: 10.1016/j.knosys.2022.108668.

[32] S. Syed and M. Spruit, “Full-text or abstract? examining topic coherence scores using latent Dirichlet allocation,” in *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct. 2017, pp. 165–174, doi: 10.1109/DSAA.2017.61.

[33] K. S. Arun and V. K. Govindan, “A hybrid deep learning architecture for latent topic-based image retrieval,” *Data Science and Engineering*, vol. 3, no. 2, pp. 166–195, Jun. 2018, doi: 10.1007/s41019-018-0063-7.

BIOGRAPHIES OF AUTHORS

Sunil Mugalihalli Eshwarappa currently working as a software engineer at Wipro Private Limited, Bangalore. completed his graduation from East West Institute of Technology in computer science and engineering and completed his M.Tech. from NMAMIT Nitte in computer science and engineering. He has authored and coauthored papers in various journals. His current research area includes machine learning, natural language processing, data modeling, and data mining. He can be contacted at email: suni.mghalli@gmail.com.

Vinay Shivasubramanyan is currently working as vice principal at P.E.S. College of Engineering, Mandya. He is also serving as a professor in the Department of Computer Science and Engineering at PESCE, Mandya. He has 21 years of experience. He has authored and co-authored 40 papers in various journals, IEEE and Springer conferences. He is an editorial board member of the International Journal on Software Engineering and Applications. He has received grants from various agencies such as the Karnataka state government, AICTE MODROBS, and NOKIA to the tune of 1.3 crore. His current research area includes machine learning. He received CMI level 5 certificate in management and leadership from Chartered Management Institute, United Kingdom in 2019. He can be contacted at email: vinay@pesce.ac.in.