

A novel dataset and part-of-speech tagging approach for enhancing sentiment analysis in Kannada

Sunil Mugalihalli Eshwarappa^{1,3}, Vinay Shivasubramanyan²

¹Department of Computer Science and Engineering, PES College of Engineering, Visvesvaraya Technological University, Belagavi, India

²Department of Information Science and Engineering, PES College of Engineering, Visvesvaraya Technological University, Belagavi, India

³Software Engineer, Wipro Limited, Bangalore, India

Article Info

Article history:

Received Apr 16, 2024

Revised Sep 11, 2024

Accepted Oct 7, 2024

Keywords:

Kannada

K-BERT model

Natural language processing

SemEval 2014 task 4

Sentiment analysis

ABSTRACT

The problem addressed in this research is the limited availability of labelled datasets and effective sentiment analysis tools for the Kannada language. Existing challenges include linguistic variations, cultural diversities, and the absence of comprehensive datasets designed specifically for sentiment analysis in Kannada. This research aims to enhance sentiment analysis capabilities for the Kannada language, addressing challenges posed by linguistic variations and limited labelled datasets. A novel Kannada dataset derived from SemEval 2014 task 4 was created using a conversion process. The dataset was processed using part-of-speech tagging, and a specialized model called K-BERT (Kannada bidirectional encoder representations from transformers) was introduced and implemented using Python within the Anaconda environment. Performance evaluation results showcased K-BERT's superiority over traditional machine learning (ML) algorithms and the BERT model, achieving an accuracy of 0.98, precision of 0.97, recall of 0.97, and F-score of 0.98 in sentiment classification for Kannada text data. This work contributes a unique Kannada dataset, introduces the K-BERT model specifically designed for Kannada sentiment analysis, and emphasizes the importance of collaborative efforts in advancing natural language processing (NLP) research for multilingual environments.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



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

India is known for its rich cultural diversity, and this diversity extends to the languages spoken across the country. There are more than 1,600 different dialects and languages spoken in India, each contributing to its unique heritage. Every state in India has its own language, adding to the country's linguistic diversity. This diversity is a reflection of India's cultural, historical, and geographical richness. When traveling through states in India, it can be noticed that each state predominantly speaks a different language. For example, Hindi is widely spoken in states like Uttar Pradesh and Bihar, while Tamil is the primary language in Tamil Nadu, Telugu in Andhra Pradesh and Kannada in Karnataka. This linguistic variation poses a challenge when analysing comments, reviews, and sentiments expressed by people in their native languages [1]. People in India express their opinions, give feedback, and write reviews in their

preferred languages in various social media platforms, and video sharing websites [2]. This diversity makes it complex to analyse sentiments accurately. For example, a positive comment in Kannada may have different cultural meanings compared to a positive comment in Hindi or Tamil. This diversity adds complexity to sentiment analysis, especially in a multilingual environment like India.

Sentiment analysis, also known as opinion mining, is a method used to extract subjective information from text [3]. It helps understand sentiments, emotions, attitudes, and opinions expressed by individuals. Natural language processing (NLP) algorithms [4], [5] and machine learning (ML) techniques [6], [7] in recent years are widely used for sentiment analysis to categorize text data into positive, negative, or neutral sentiments. Sentiment analysis is crucial in today's digital world where vast amounts of text data are generated daily on social media, e-commerce platforms, news websites, and customer feedback forums [8]. Businesses, organizations, and governments use sentiment analysis to understand public opinion, assess customer satisfaction, monitor brand perception, and make data-driven decisions. However, analysing sentiments in different languages is challenging due to linguistic variations, cultural diversities, and complex sentence structures [9]. ML and deep learning (DL) algorithms play a significant role in addressing these challenges by automating sentiment analysis across multiple languages [10]. These techniques learn linguistic patterns and semantic structures to improve the accuracy and efficiency of sentiment analysis.

Despite advancements, sentiment analysis for languages like Kannada faces specific challenges. The lack of labelled datasets containing aspects and sentiments in Kannada hinders the development of accurate sentiment analysis tools [11]. Additionally, linguistic importance, sentiment expressions, and cultural references unique to Kannada pose difficulties during sentiment analysis [12]. To address the aforementioned challenges, this study takes a proactive approach by introducing a novel Kannada dataset derived from the SemEval 2014 Task 4 dataset. The dataset is prepared by conversion process which involves translating the SemEval dataset from English to Kannada. This strategic step is taken because the SemEval 2014 Task 4 dataset offers a substantial number of aspects and sentiments, making it well-suited for evaluation purposes. Additionally, this conversion significantly reduces the time and effort required for labeling data, sentiments, and aspects in Kannada. Subsequently, the features extracted from this newly created Kannada dataset are processed using the part-of-speech (PoS) tagging method. PoS tagging is utilized to identify and categorize the grammatical components of the text, which is crucial for accurate feature extraction in Kannada. Moreover, a novel model specifically designed for Kannada, termed K-BERT (KannadaBERT), is introduced in this study. K-BERT is designed to effectively classify the extracted features from the Kannada dataset. This model leverages the advancements in BERT (bidirectional encoder representations from transformers), a state-of-the-art language model in NLP, to enhance the accuracy and performance of sentiment analysis in Kannada.

The manuscript is structured as follows to provide a clear and systematic presentation of the research findings. Section 2 delves into the literature survey, where existing studies and methodologies related to sentiment analysis and opinion mining in Kannada are discussed. Moving on to section 3, the process of dataset preparation, preprocessing techniques, feature extraction methods, and the development of the K-BERT classifier are elaborated upon. Section 4 is dedicated to analysing and comparing the features extracted from the PoS tagging method and the K-BERT classifier. Various classifiers are employed and their performance is evaluated based on the extracted features, providing insights into the effectiveness of different classification techniques for sentiment analysis in Kannada. Lastly, section 5 encapsulates the conclusion of the work, summarizing the key findings, contributions, and implications of the study. This section also discusses potential areas for future research and highlights the significance of the proposed approach in advancing sentiment analysis capabilities for Kannada language processing.

2. LITERATURE SURVEY

The literature survey encapsulates a multifaceted exploration into sentiment analysis and opinion mining, with a specific focus on languages such as Kannada, Tamil, English, and their code-mixed variations. Each of the referenced studies adds a unique perspective by introducing distinct methodologies, techniques, and findings, thereby enriching the broader discourse of computational linguistics and paving the way for advancements in sentiment analysis research. Beginning with [13], researchers introduced an innovative hybrid approach called SAEKCS, which utilizes state-of-the-art DL approaches which include bi-directional long-short term-memory (LSTM) and convolutional-neural-network (CNN) for the purpose of analyzing sentiments on English-Kannada coded-switched text dataset. The experiments presented in the study demonstrated a notable accuracy score of 77.6% along with an overall coverage-rate of 69.6%. These results highlight the effectiveness of DL techniques in effectively processing code-switched linguistic information. Moving on to [14], researchers set out to discover and classify different points of view conveyed in Kannada text. The researchers utilized a range of methods, namely decision tree (DT), Naive Bayes (NB), and negator

approach. These methodologies yielded significant accuracy levels of 85%, 65%, and 53% respectively. This research highlights the complex procedure of analysing opinions in settings with a wide variety of languages. A thorough evaluation of Kannada-language IMDB reviews obtained from reliable sources was carried out in [15]. The researchers achieved 89% rate of accuracy by suggesting an ensemble classifier method that uses various vectorization algorithms. According to this work, for handling tasks of sentiment analysis across numerous fields, robust categorization approaches are crucial. In a comparable manner to [15], the study conducted by [16] explored the field of sentiment evaluation across various languages, such as Kannada, Hindi and English languages. By employing a CNN combined with LSTM framework, the investigation successfully obtained outcomes that outperformed established approaches. This highlights the promising capabilities of sophisticated neural-network structures in effectively capturing intricate sentiment relationships.

Shanmugavadivel *et al.* [17] tackled the challenging task of identifying offensive words and performing sentiment evaluation on code-mixed information that included both English and Tamil language. By utilizing advanced DL and ML techniques, specifically employing models like RoBERTa and BERT, researchers were able to showcase significant achievements in the field. Notably, they achieved accuracy levels of 65% for sentiment evaluation and 79% for inappropriate language recognition. Among the various approaches tested, the adapter-BERT approach proved to be efficient in achieving these results. Chundi *et al.* [18], presented an innovative lexicon-based approach called NBLex for accurately predicting sentiments in code-switched text written in Kannada and English languages. The approach utilized lexicons, which is a collection of words and their associated sentiments, to analyse the text and determine the sentiments expressed within it. This method demonstrated the importance of sentiment evaluation approaches utilizing lexicons in multilingual contexts by outperforming standard methods such as Bi-LSTM and NB with respect to true positive (TP) rate and accuracy. Roy [19] have focused on the difficulties of analysing sentiment in languages with limited resources such as Malayalam and Kannada. To tackle these difficulties, they have proposed an ensemble approach. The aforementioned approach demonstrated outstanding F1-scores when applied to code-mixed languages, thereby emphasizing the efficacy of ensemble methods for addressing the limitations posed by insufficient data availability. Chundi *et al.* [20], reconsidered the task of analyzing sentiment in Kannada-English code-switched text, which was presented in [18]. They employed the NBLex approach [18] and demonstrated that their approach achieved higher accuracy and F1-score in comparison with prior approaches. The aforementioned statement highlights the ongoing development and improvement of sentiment evaluation methods within code-switched language contexts. Chundi *et al.* [21], utilized a character-level n-grams method to effectively detect code-switched and monolingual content in English-Kannada online social networking data. The results obtained from this method have shown a notable improvement in F1-score and accuracy when compared to conventional ML approaches. This work highlighted the significance of employing context-aware feature-extraction approach in order to achieve better performance. Finally, the study conducted by [22] focused on the application of sentiment analysis techniques to analyse COVID-19 information containing the Kannada language. The researchers utilized various ML and ensemble approach to achieve their objectives. The findings of the research revealed accuracy scores that varied between 66% and 69%, thereby highlighting the versatility of methods for sentiment analysis in effectively analysing various fields and datasets.

From the above analysis of the various studies in the field of sentiment analysis reveals a common challenge: the absence of a comprehensive and standardized dataset specifically designed for Kannada sentiment analysis. Despite the advancements in sentiment analysis techniques and the emergence of sophisticated models and methodologies, researchers consistently encounter limitations due to the lack of a robust dataset that accurately represents the nuances of sentiment in the Kannada language. The studies discussed earlier highlight the innovative approaches and techniques researchers have employed to overcome this issue. For instance, some studies resort to creating their own datasets, often by translating existing datasets from other languages to Kannada using tools like Google Translate. However, this approach may introduce challenges related to the accuracy and authenticity of sentiment labels, as machine translation may not always capture the subtleties of sentiment expressions in Kannada. Other studies leverage ensemble techniques, deep learning models, and lexicon-based approaches to enhance sentiment analysis accuracy despite the data scarcity. These approaches often involve a combination of feature extraction, PoS tagging, and emotion prediction methodologies to infer sentiment from limited datasets. Despite these innovative strategies, the lack of a standardized and widely accepted dataset for Kannada sentiment analysis remains a significant bottleneck in the field. A reliable dataset would not only facilitate more accurate sentiment analysis but also enable researchers to benchmark and compare different models and techniques effectively. In conclusion, while advancements in sentiment analysis methodologies are promising, the field would greatly benefit from the development and adoption of a standardized Kannada sentiment analysis dataset. Collaborative efforts towards dataset creation, validation, and sharing are essential to drive further progress and innovation in Kannada sentiment analysis research.

3. METHOD

Based on the literature survey conducted, it is evident that a majority of the existing works in sentiment analysis for Kannada language have mainly focussed on creating their own datasets for evaluation purposes. There is a notable scarcity of standardized datasets specifically designed for Kannada language sentiment analysis, especially those that come equipped with sentiment labels. Consequently, this study undertakes the task of preparing a dataset that already incorporates sentiment labels for analytical purposes. Hence, drawing inspiration from [15], wherein they translated the IMBD dataset from English to Kannada using Google Translate, this work adopts a similar methodology. Specifically, it translates the SemEval 2014 task 4 dataset [23] from English to Kannada. The choice of SemEval 2014 task 4 dataset is motivated by its inclusion of labelled aspect words, aspect categories (sentiments), and polarity, which prove instrumental in evaluating the effectiveness of this work when compared with standard datasets. The overall architecture of this study is presented in Figure 1. Initially, the SemEval 2014 task 4 dataset serves as the foundation. Subsequently, English data undergoes translation to Kannada through Google Translate. The resultant Kannada raw data then undergoes preprocessing to attain clean data. From this clean data, features are extracted using PoS tagging. These extracted features are subsequently employed in training the classifier model and the evaluation is done.

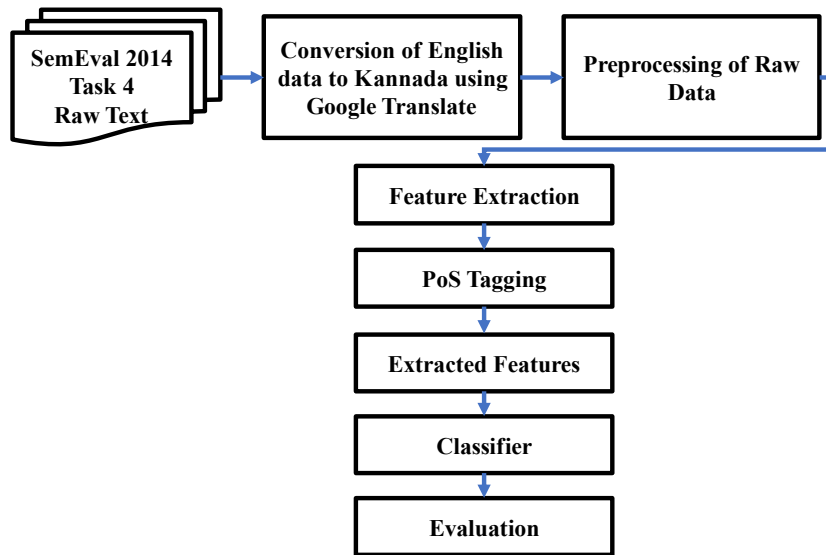


Figure 1. Proposed architecture

3.1. Preparation of dataset

In the initial phase, the focus is on the dataset, particularly the SemEval 2014 task 4 dataset, which comprises raw text encompassing diverse reviews pertaining to laptops and restaurants. This dataset is particularly valuable as it includes aspect terms and polarities corresponding to each review, thereby facilitating aspect-based sentiment classification. Subsequently, the English raw text undergoes translation into Kannada raw text using Google Translate. An example of some text is given in Table 1. Following the translation process, the raw data proceeds to preprocessing, as elaborated in the subsequent section.

Table 1. English to Kannada text

SL. No	English review	Kannada review
1	Other than not being a fan of click pads (industry standard these days) and the lousy internal speakers, it's hard for me to find things about this notebook I don't like, especially considering the \$350 price tag.	<i>Klik pyāḍgala (ī dinagaḷalli udyamada pramāṇita) mattu asahyavāda āntarika spīkargaḷa abhimāniyāgirade bēre, nānu iṣṭapaḍada ī nōḷbuk baḡge viṣayagaḷannu huḍukalu nanage kaṣṭavāḡuttade, viṣēṣavāḡi \$350 beleyannu pariḡaṇisi.</i>
2	No installation disk (DVD) is included.	<i>Yāvudē anusthāpanā ḍisk (ḍiviḍi) oḷaḡoṇḍilla.</i>
3	Works well, and I am extremely happy to be back to an apple OS.	<i>Uttamavāḡi kāryanirvahisuttade mattu āpal o'seḡe hintiruḡalu nanage tumbā santōṣavāḡide.</i>

3.2. Pre-processing

In NLP, preprocessing is a crucial step for any raw text. Therefore, in the second stage of this study, the Kannada raw text undergoes preprocessing. Initially, tokenization is performed to segment each word within the review sentence. Subsequently, brackets, symbols, hyphens, inverted commas, and other symbols were removed post-tokenization except full stop, exclamation mark and commas. Following the punctuation removal, the text undergoes stemming and lemmatization processes to get the intended meaning of words. Subsequently, a stopwords library is constructed to filter out commonly used words that contribute minimal meaningful information. Upon completion of this comprehensive preprocessing pipeline, a clean text is obtained, containing meaningful words extracted from the original review sentences. Subsequently, the feature extraction process is initiated, which is elaborated upon in detail in the subsequent section of this work.

3.3. Feature extraction

In this study, the feature extraction process from the clean reviews, comprising words in each sentence, employs the PoS tagging approach. The purpose of PoS tagging in this work is twofold: to comprehend the grammatical structure of review sentences and to disambiguate words with multiple meanings. The utilization of PoS tagging aids in gaining insights into the syntactic composition of review sentences and aids in resolving ambiguity within words. In this study, the Trigrams'n'Tags (TnT) model is employed as PoS taggers. The TnT method was initially introduced by [24], in which the researcher attempted to designate a suitable label or tag by computing the likelihoods of potential tags for every phrase. The TnT method serves as a variant of second-order Markov approach that integrates multiple n-gram models, including trigram, bigram and unigram, with the goal to determine the most appropriate tag for a given word. According to the research conducted by [25], the process of generating a sequence of PoS tags represented as t_1, \dots, t_n from the specific sequence of phrases/words represented as w_1, \dots, w_n can be achieved by utilizing (1). Moreover, the unigrams, bigrams and trigrams for a given sentence are converted using (2) to (4).

$$\arg \max_{t_1, \dots, t_T} [\prod_{i=1}^T P(t_i | t_{i-1}, t_{i-2}) P(w_i | t_i)] P(t_{T+1} | t_T) \tag{1}$$

$$Unigrams = P(w_i) = \frac{C(w_i)}{N} \tag{2}$$

$$Bigram = P(w_i | w_{i-1}) = \frac{C(w_{i-1}w_i)}{C(w_{i-1})} \tag{3}$$

$$Trigram = P(w_i | w_{i-2}w_{i-1}) = \frac{C(w_{i-2}w_{i-1}w_i)}{C(w_{i-2}w_{i-1})} \tag{4}$$

Where, $C(w_i)$ denotes the frequency of the occurrence for the word w_i , while N denotes the overall words present in training data. Further, for determining the likelihood of a particular PoS tag occurring after a specific PoS tag in a sequence, the (5) is utilized.

$$P(t_i | t_{i-1}) = \frac{C(t_i w_i)}{C(w_i)} \tag{5}$$

The (5) represents the principles of conditional probability, where the numerator ($C(t_i, w_i)$) represents the count of times the word (w_i) is associated with the PoS tag (t_i) in the dataset, and the denominator $C(w_i)$ represents the total count of occurrences of the word (w_i) in the dataset. This conditional probability calculation aids in the *TnT* approach by providing a mechanism to estimate the probability of PoS tag sequences, which is crucial for determining the most probable tag sequence for each word in a given sentence. Furthermore, it is worth noting that the computation of trigram probability employing the (1) utilizing the prepared dataset is not entirely helpful because of the issue of limited information. Consequently, the insufficient frequency of occurrences of every trigram prevents the reliable computation of its probability. In addition, assigning a probability of zero for a particular trigram can have unintended consequences, as it implies that the associated trigram was not previously observed in the collection of data. Thus, it is not feasible to categorize various sequences carrying a zero probability because the possibility of a whole sequence is determined to zero whenever its employment is required for an entirely novel sequence. Therefore, the utilization of a normalizing variable that incorporates the linear interpolation of trigrams, bigrams and unigrams has been found to yield the most favourable results in the *TnT* model for this work. Consequently, the evaluation of the trigram probability using the normalizing variable is conducted using (6).

$$P(w_i|w_{i-2}w_{i-1}) = \beta_1P(w_i) + \beta_2P(w_i|t_2) + \beta_3P(w_i|w_{i-2}w_{i-1}) \tag{6}$$

Where the sum of β_1 , β_2 , and β_3 is equal to 1, i.e., $\beta_1 + \beta_2 + \beta_3 = 1$. Within the scope of this study, it is important to note that the values of β s remain unaffected by the specific trigram being analyzed. This is due to the implementation of a context-independent linear-interpolation approach. The utilization of this approach facilitates the attainment of superior results compared to the prevailing context-dependent methodology. Because of the limited information issue, it is not feasible to determine an independent set of β s for every trigram. Therefore, the trigrams are organized based on their frequencies, and corresponding sets of β s are computed for every category. According to our current understanding, there has been no previous research that has explored the use of frequency variation classification for interpolation of linear frequencies in PoS tagging. Hence, the numerical values for the variables β_1 , β_2 , and β_3 are determined using the process of deleted-interpolation. This method helps to remove all trigrams from training-set in a sequential fashion and finds the best possible values for the β s through every single one of the remaining n-grams across all sets. Finding the count of frequency of unigrams, bigrams, and trigrams allows one to computationally efficiently construct the weights having a time complexity that is linear with the total number of distinct trigrams.

3.4. Classifier

In this work, for classification, a classifier called KannadaBERT (K-BERT) is presented. The BERT framework consists of a multi-layer bidirectional-transformer-encoder [26]. The purpose of this framework is to pretrain deep bidirectional-representations using unlabelled phrases/text/words by conditioning both right and left background across every layer [26]. BERT is frequently utilized to find a vector representation for every word within a phrase. The standard BERT framework initially receives input in the form of sentences, which are broken down by a specific token known as separator (SEP). The initial input sequence token is commonly referred as classification (CLS) token. For tasks involving classification, every word of the sentence is represented by the last hidden state that corresponds with the CLS token. Notably, BERT already incorporates tokenization preprocessing by default. The BERT tokenizer employs a tokenization process that involves dividing the sentence into individual tokens. Additionally, it strategically places the unique tokens CLS and SEP in their respective positions within the tokenized sequence. By considering the standard BERT framework, this work presents the K-BERT similar to the BERT framework. Instead of the passing the complete sentence as input which goes for preprocessing and tokenization, the K-BERT model considers the trigrams as input. The proposed K-BERT model is shown in Figure 2. In this K-BERT model the function of SEP is to separate each trigram and the function of CLS is to classify each trigram so that a meaningful classification for a given sentence is achieved. Also, in this work one-hot encoding trigrams is used for converting each trigram into a high-dimensional vector where only the element corresponding to specific trigram is 1, and all others are 0.

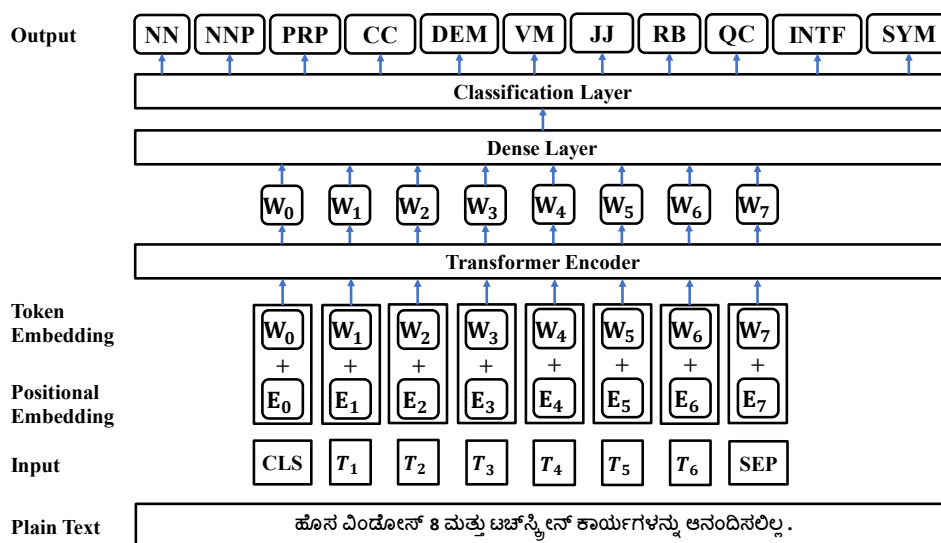


Figure 2. Proposed K-BERT model

In the standard BERT framework, for extracting PoS tags, the CLS is utilized to encode the input sentence as it does for classification-tasks, whereas in the K-BERT, every input token is sent through the same fully-connected classification layers for extracting PoS tags. Furthermore, it is important to note that the utilization of the word-piece tokenizer necessitates the establishment of a clear relationship among sub-words or word-pieces and their respective labels. Concerning the word-level tokenizer, there exists a direct mapping among the input tokens and their respective labels. Nevertheless, when employing a word-piece tokenizer like the BERT tokenizer, it is possible for every individual word to be divided into numerous tokens. It is imperative for establishing a “token-mapping” approach that maps word-pieces to corresponding labels. In standard BERT framework [1], the choice was made to use the depiction of the initial sub-token to be the input for the next layer. This decision was made with the intention of neglecting the depiction of the remaining sub-tokens. From a practical standpoint, the implementation of this approach involves allocating the word-label for the initial sub-word, while allocating an imaginary label “X” for the remaining sub-words. During the computation of the loss-function, the “X” labels associated with the sub-tokens are neglected. In addition, it is possible to allocate the label of a single word to determine the depiction of the final word-piece. Alternatively, the word-label could be extended across all sub-words, and subsequently, a mean depiction of the word-pieces can be determined. Hence, in K-BERT, this work has chosen to utilize the initial word-piece depiction. However, it is acknowledged that there are additional mapping methods that could be explored in future research. Upon completion of the K-BERT block, the resulting output is subsequently passed through a dense layer and then classification layer and then the output is achieved.

This work has opted to employ the freezing BERT approach, wherein the entire BERT architecture remains fixed, and only untrained layers and neurons at the end are added. Subsequently, a new model is trained in such a way that only the weights of the newly added layers are updated during training. This approach ensures that the core BERT layers remain unchanged while fine-tuning the model. Furthermore, this study has extended the existing BERT framework, as depicted in Figure 2, by incorporating a classification layer and a dense layer. The primary objective of this modification is to enable the model to generate tag sequences for input sentences. This is achieved through the utilization of the SoftMax activation function, which facilitates the generation of probability distributions over the output classes. To address the risk of overfitting, a dropout normalization technique has been applied specifically on the dense layer. Further, the results of the K-BERT model are evaluated and compared with other classifiers which are discussed in the next section.

4. RESULTS AND DISCUSSION

The K-BERT model was implemented on a system running the Windows 11 operating system, equipped with 16 GB of RAM and an NVIDIA GeForce GTX 1650 graphics card. The implementation was carried out using Python programming language within the Anaconda environment. Python provided a robust framework for ML and NLP tasks, making it well-suited for implementing complex models like K-BERT. For evaluating the performance of the classification model, various performance metrics were employed, including accuracy, precision, recall, and F-score, i.e., (7) to (10) respectively. These metrics provide a comprehensive assessment of the model's ability to correctly classify sentiment in the input text data. The performance metrics are evaluated as follows:

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \quad (7)$$

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$F - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (10)$$

Where, TP represents true-positive, FP represents false-positive, TN represents true-negative and FN represents false-negative. Further, from the dataset, a small part of the testing examples is presented in Table 2. The Table 3 presents the results achieved by the K-BERT model for the testing examples consisting of words or tokens along with their corresponding tags. Each token in the dataset is tagged with a specific POS tag, providing information about its grammatical function in a sentence. The tags include NN (noun), NNP (proper noun), PRP (pronoun), CC (conjunction), DEM (demonstrative), VM (verb finite), JJ (adjective), RB (adverb), QC (cardinal), INTF (intensifier), and SYM (symbol). For instance, in the first row, "Būṭ" is tagged

as NNP, "samayavu" as NN, "ati" as INTF, and so on. Similarly, other rows contain tokens along with their respective POS tags, providing a structured representation of the linguistic elements present in the dataset.

Table 2. Testing examples

SL. no	English	Kannada
1	"Boot time is super fast, around anywhere from 35 seconds to 1 minute."	<i>Būṭ samayavu ati vēgavāgiruttade, sumāru 35 sekeṇḍugaḷinda 1 nimiṣadavarege.</i>
2	"tech support would not fix the problem unless I bought your plan for \$150 plus."	<i>Nānu nim'ma yōjaneyannu \$150 plasge kharīdisada horatu tek bembalavu samasyeyannu pariharisuvudilla.</i>
3	"but in resume this computer rocks!"	<i>Ādare punarārambhadalli ī kampyūṭar rāk!</i>
4	"Set up was easy."	<i>Hondisuvudu sulabhavāyitu.</i>
5	"Did not enjoy the new Windows 8 and touchscreen functions."	<i>Hosa viṇḍōs 8 mattu ṭacskrīn kāryagaḷannu ānandisalilla.</i>

Table 3. Word-tags

SL. no	Word/-tag				
1	Būṭ-NNP sumāru-JJ	samayavu-NN 35-QC	ati-INTF sekeṇḍugaḷinda-NN	vēgavāgiruttade-VM 1-QC	,-SYM nimiṣadavarege- NN
2	Nānu-PRP kharīdisada-NN	nim'ma-PRP horatu-RB	yōjaneyannu-NN tek-NN	\$150-QC bembalavu-NN	plasge-NN samasyeyannu- NN
3	pariharisuvudilla.-NN Ādare-CC	punarārambhadalli- NN	ī-DEM	kampyūṭar-NN	rāk-NNP
4	!-SYM Hondisuvudu -VM	sulabhavāyitu-VM	. -SYM		
5	Hosa-JJ kāryagaḷannu-NN	viṇḍōs-NNP ānandisalilla-VM	8-QC .-SYM	mattu-CC	ṭacskrīn-NNP

The Table 4 presents a POS Tagset generated by the proposed K-BERT model, providing a structured representation of linguistic elements with their respective tags and descriptions. Each row in the table corresponds to a specific word or token along with its POS tag and description. The tags include NN, NNP, PRP, CC, DEM, VM, JJ, RB, QC, INTF, and SYM. For instance, the word "Samayavu" is tagged as NN, "Būṭ" as NNP, "Nānu" as PRP, "Ādare" as CC, "ī" as DEM, "Vēgavāgiruttade" as VM, "Sumāru" as JJ, "horatu" as RB, and so on. These tags and descriptions provide valuable insights into the grammatical roles and functions of the words or tokens within the dataset, facilitating linguistic analysis and NLP tasks.

Table 4. PoS Tagset generated by proposed K-BERT

SL. no	Tag	Description	Word
1	NN	Noun	<i>Samayavu, sekeṇḍugaḷinda, nimiṣadavarege, yōjaneyannu, plasge, kharīdisada, tek, bembalavu, samasyeyannu, pariharisuvudilla, punarārambhadalli, kampyūṭar, kāryagaḷannu</i>
2	NNP	Proper Noun	<i>Būṭ, rāk, viṇḍōs, ṭacskrīn</i>
3	PRP	Pronoun	<i>Nānu, nim'ma</i>
4	CC	Conjunction	<i>Ādare, mattu</i>
5	DEM	Demonstrative	<i>ī,</i>
6	VM	Verb Finite	<i>Vēgavāgiruttade, Hondisuvudu, sulabhavāyitu, ānandisalilla</i>
7	JJ	Adjective	<i>Sumāru, Hosa</i>
8	RB	Adverb	<i>horatu</i>
9	QC	Cardinal	<i>35, 1, \$150, 8</i>
10	INTF	Intensifier	<i>Ati</i>
11	SYM	Symbol	<i>, ! ,</i>

The results presented in Table 5 show the performance evaluation metrics, including accuracy, precision, recall, and F-score, for various ML models used in sentiment analysis. Extreme gradient boosting (XGBoost) achieved an accuracy of 0.68, precision of 0.67, recall of 0.67, and F-score of 0.69. Logistic regression (LR) demonstrated similar performance with an accuracy of 0.67, precision of 0.68, recall of 0.64, and F-score of 0.68. Random forest (RF) had an accuracy of 0.66, precision of 0.65, recall of 0.68, and F-score of 0.62. AdaBoost and gradient boosting exhibited comparable results with accuracy scores of 0.67 and 0.69, respectively, along with precision, recall, and F-score values around 0.65-0.68. The voting

ensemble model showed lower performance with an accuracy of 0.58 and precision of 0.62, but higher recall and F-score values at 0.68 and 0.64, respectively. In contrast, the BERT model achieved significantly higher performance with an accuracy of 0.81, precision of 0.79, recall of 0.8, and F-score of 0.82. Notably, the proposed K-BERT model outperformed all other models, showcasing exceptional results with an accuracy of 0.98, precision of 0.97, recall of 0.97, and F-score of 0.98. These findings highlight the superior performance of K-BERT in sentiment analysis tasks, emphasizing its effectiveness in accurately classifying sentiment in text data compared to traditional ML algorithms and even the BERT model. The results are graphically shown in Figure 3.

Table 5. Performance evaluation

Models	Accuracy	Precision	Recall	F-Score
XGBoost	0.68	0.67	0.67	0.69
LR	0.67	0.68	0.64	0.68
RF	0.66	0.65	0.68	0.62
AdaBoost	0.67	0.68	0.65	0.64
Gradient	0.69	0.66	0.64	0.63
Voting	0.58	0.62	0.68	0.64
BERT	0.81	0.79	0.8	0.82
K-BERT	0.98	0.97	0.97	0.98

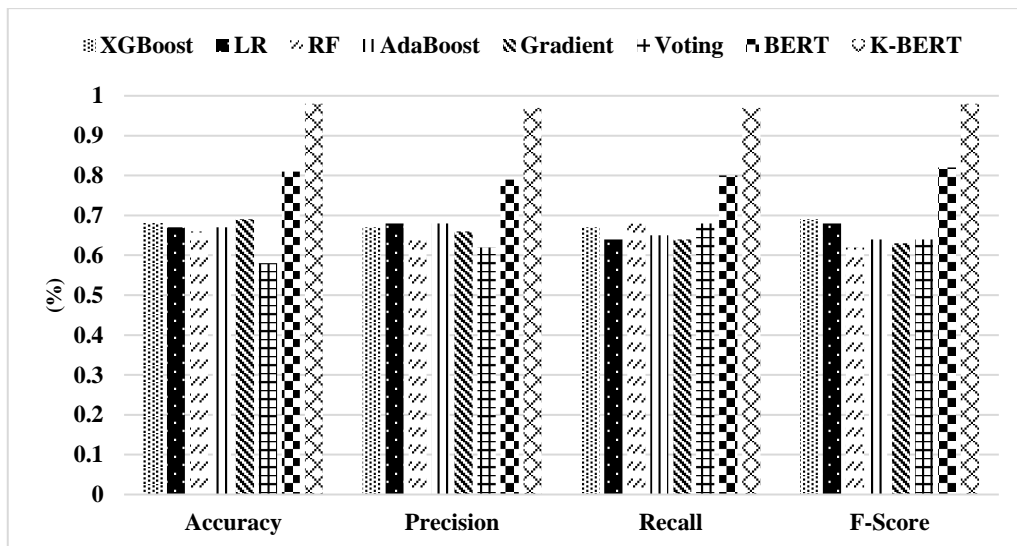


Figure 3. Performance evaluation





5. CONCLUSION

In conclusion, this work has made significant improvements in advancing sentiment analysis capabilities for the Kannada language. By introducing the K-BERT model and leveraging advanced ML and NLP techniques, we have addressed the challenges posed by linguistic variations, cultural nuances, and limited labelled datasets. The performance evaluation results demonstrate that the K-BERT model outperforms traditional ML algorithms, including XGBoost, LR, RF, AdaBoost, and gradient boosting, as well as the BERT model. With an exceptional accuracy of 0.98, precision of 0.97, recall of 0.97, and F-score of 0.98, the K-BERT model showcases its effectiveness in accurately classifying sentiment in Kannada text data. This work not only contributes a novel dataset derived from SemEval 2014 task 4 for Kannada sentiment analysis but also introduces a robust model specifically designed for Kannada, paving the way for further advancements in NLP research and applications designed for the linguistic diversity of India. Collaborative efforts towards dataset creation, model development, and evaluation methodologies are essential for enhancing sentiment analysis capabilities in multilingual environments and driving innovation in computational linguistics. For future work, the BERT model can be further enhanced for achieving better results and compared with other datasets.





REFERENCES

- [1] K. R. Mabokela, T. Celik, and M. Raborife, "Multilingual sentiment analysis for under-resourced languages: a systematic review of the landscape," *IEEE Access*, vol. 11, pp. 15996–16020, 2023, doi: 10.1109/ACCESS.2022.3224136.
- [2] M. Z. Ansari, M. B. Aziz, M. O. Siddiqui, H. Mehra, and K. P. Singh, "Analysis of political sentiment orientations on twitter," *Procedia Computer Science*, vol. 167, pp. 1821–1828, 2020, doi: 10.1016/j.procs.2020.03.201.
- [3] M. Kaur, K. Joshi, B. Goyal, and A. Dogra, "An approach to perform sentiment analysis using data mining algorithms," in *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*, Jul. 2023, pp. 803–808, doi: 10.1109/ICECAA58104.2023.10212404.
- [4] 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, p. 100059, Mar. 2024, doi: 10.1016/j.nlp.2024.100059.
- [5] P. Nandwani and R. Verma, "A review on sentiment analysis and emotion detection from text," *Social Network Analysis and Mining*, vol. 11, no. 1, p. 81, Dec. 2021, doi: 10.1007/s13278-021-00776-6.
- [6] V. Joshi, S. Patel, R. Agarwal, and H. Arora, "Sentiments analysis using machine learning algorithms," in *2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*, Mar. 2023, pp. 1425–1429, doi: 10.1109/ICEARS56392.2023.10085432.
- [7] M. Arumugam, S. S R, and C. Jayanthi, "Machine learning for sentiment analysis utilizing social media," in *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*, Jul. 2023, pp. 523–530, doi: 10.1109/ICECAA58104.2023.10212135.
- [8] M. Rodríguez-Ibáñez, A. Casáñez-Ventura, F. Castejón-Mateos, and P.-M. Cuenca-Jiménez, "A review on sentiment analysis from social media platforms," *Expert Systems with Applications*, vol. 223, p. 119862, Aug. 2023, doi: 10.1016/j.eswa.2023.119862.
- [9] G. Manias, A. Mavrogiorgou, A. Kiourtis, C. Symvoulidis, and D. Kyriazis, "Multilingual text categorization and sentiment analysis: a comparative analysis of the utilization of multilingual approaches for classifying twitter data," *Neural Computing and Applications*, vol. 35, no. 29, pp. 21415–21431, Oct. 2023, doi: 10.1007/s00521-023-08629-3.
- [10] V. Umarani, A. Julian, and J. Deepa, "Sentiment analysis using various machine learning and deep learning techniques," *Journal of the Nigerian Society of Physical Sciences*, pp. 385–394, Nov. 2021, doi: 10.46481/jnsps.2021.308.
- [11] Y. Garani, S. Joshi, and S. Kulkarni, "Offensive sentiment detection with Chat GPT and other transformers in Kannada," in *2023 IEEE 2nd International Conference on Data, Decision and Systems (ICDDS)*, Dec. 2023, pp. 1–6, doi: 10.1109/ICDDS59137.2023.10434684.
- [12] 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.
- [13] R. Chundi, V. R. Hulipalled, and J. . Simha, "SAEKCS: sentiment analysis for English – Kannada code switchtext using deep learning techniques," in *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, Oct. 2020, pp. 327–331, doi: 10.1109/ICSTCEE49637.2020.9277030.
- [14] 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, p. 012039, Apr. 2021, doi: 10.1088/1757-899X/1123/1/012039.
- [15] 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, doi: 10.1007/978-981-16-5157-1_53.
- [16] 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, doi: 10.1007/978-981-16-3342-3_29.
- [17] K. Shanmugavadeivel, V. E. Sathishkumar, S. Raja, T. B. Lingaiah, S. Neelakandan, and M. Subramanian, "Deep learning based sentiment analysis and offensive language identification on multilingual code-mixed data," *Scientific Reports*, vol. 12, no. 1, p. 21557, Dec. 2022, doi: 10.1038/s41598-022-26092-3.
- [18] R. Chundi, V. R. Hulipalled, and J. B. Simha, "NBLeX: emotion prediction in Kannada-English code-switch text using naïve bayes lexicon approach," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 2, pp. 2068–2077, Apr. 2023, doi: 10.11591/ijece.v13i2.pp2068-2077.
- [19] P. K. Roy, "Deep ensemble network for sentiment analysis in bi-lingual low-resource languages," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 23, no. 1, pp. 1–16, Jan. 2024, doi: 10.1145/3600229.
- [20] R. Chundi, V. R. Hulipalled, and J. Bharthish Simha, "Lexicon-based sentiment analysis for Kannada-English code-switch text," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 12, no. 3, pp. 1500–1507, Sep. 2023, doi: 10.11591/ijai.v12.i3.pp1500-1507.
- [21] R. Chundi, V. R. Hulipalled, and J. B. Simha, "Identification of monolingual and code-switch information from English-Kannada code-switch data," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 5, pp. 5632–5640, Oct. 2023, doi: 10.11591/ijece.v13i5.pp5632-5640.
- [22] R. Shankar, 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.
- [23] 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.
- [24] 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.
- [25] 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, p. 012013, Jun. 2019, doi: 10.1088/1742-6596/1235/1/012013.
- [26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North*, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

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.



Dr. 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.