Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Sentimental Data Isolation through Advancing Classification with K-BERT and Polarity Scoring Model

B. Shanthini, Dr. N. Subalakshmi

Department of Computer and Information Science, Annamalai University, India.
E-mail: shanthinib.66@gmail.com

Sentiment classification, also known as sentiment analysis, plays a crucial role in understanding and analyzing public opinion and customer feedback. Cross-domain sentiment classification aims to predict sentiment polarity in text across different domains. This research proposes a cross-platform sentiment classification approach using K-BERT (Knowledge enabled Bidirectional Encoder Representations from Transformers) and a scoring model. K-BERT, a powerful pre-trained language model, captures contextual relationships and generates high-quality word representations. The proposed approach leverages K-BERT to extract contextualized features from text data across different domains. K-BERT is utilized to encode the preprocessed text and generate contextualized word embeddings. These embeddings capture the sentiment-related text patterns from two popular sources of textual data for sentiment classification are Amazon reviews and Twitter. While sentiment classification for both Amazon reviews and Twitter data follows a similar approach, each dataset has specific considerations. The combination of sentiment analysis on these diverse datasets provides a comprehensive understanding of customer sentiment in different contexts and platforms.

Keywords: Amazon, K-BERT, Sentiment Classification, Twitter, Polarity Scoring Model.

1. Introduction

Sentiment analysis, frequently denoted to as opinion mining, is a branch of natural language processing (NLP) concerned with comprehending and retrieving sentiment or personal information from textual information. It entails mechanically analysing text to determine if it expresses positive, negative, or neutral mood [1]. Sentiment analysis is important in many fields; include research on markets, consumer survey analysis, online conversation tracking, and online reputation managing [2]. The rapid growth of digital communication and the abundance of online textual data have made sentiment analysis increasingly important. It enables enterprises and organisations to learn about the public opinion, understand customer satisfaction, track trends, and make data-driven decisions [3]. Sentiment analysis helps identify positive sentiment to leverage for marketing purposes, identify negative sentiment to address customer concerns, and monitor sentiment over time to detect emerging patterns. Sentiment analysis techniques have enhanced accuracy and scalability of sentiment analysis, enabling businesses to analyze large volumes of text efficiently [4]. The ability to understand and harness sentiment provides valuable insights into customer sentiment, market trends, and public opinion, allowing businesses to tailor their strategies, products, and services accordingly [5].

Cross-domain sentiment analysis is a specialized branch of sentiment analysis that focuses on classifying sentiment across different domains or topics. It involves developing models and techniques that can effectively handle sentiment classification tasks when the training and testing data come from different domains or contexts [6]. Cross-domain sentiment analysis plays a vital role in understanding sentiment variations across diverse domains and enables businesses to gain comprehensive client opinion and sentiment patterns are shown. In many real-world circumstances, sentiment analysis algorithms based to utilise with data from additional domain. This is due to variations in language use, vocabulary, sentiment expressions, and cultural factors that differ across domains. Cross-domain sentiment analysis addresses this challenge by developing approaches that can generalize sentiment analysis models across multiple domains [7].

K-BERT is an advanced language representation model that combines knowledge graph information with the popular BERT architecture. It enhances the capabilities of BERT by incorporating external knowledge and enabling a deeper understanding of textual data [8]. K-BERT leverages knowledge graph embeddings to improve the contextualized representation of words and their relationships, leading to more accurate and context-aware language understanding [9]. In traditional language models like BERT, the contextualized word representations are learned solely from the surrounding words in the input text. However, K-BERT takes a step further by integrating knowledge graph embeddings into the learning process. Knowledge graphs are structured databases that Entities and their connections are used to store facts. By leveraging this knowledge, K-BERT enriches the word representations with semantic and relational information from the knowledge graph, enabling a better understanding of the text [10].

The primary goal of cross-domain sentiment analysis is to build models that can effectively transfer knowledge learned from one domain to another. By leveraging existing sentiment knowledge from a source domain, these models can adapt and perform well in target domains with limited labeled data or even no labeled data. This ability to transfer sentiment understanding between domains allows businesses to analyze sentiment across different products, services, industries, or user communities.

The objective of Cross-Domain Sentiment Classification using K-BERT and a scoring model is to develop a robust and accurate sentiment analysis framework that can effectively classify sentiment polarity across diverse domains or topics. The combined approach aims to address the challenge of sentiment analysis models failing to generalize well across different domains due to domain-specific language use, sentiment expressions, and contextual variations.

2. Related Works

The Text Classification query has been extensively researched in relation to data retrieval issues and data mining jobs. It is useful in a variety of activities, such as medical diagnosis, the healthiness and care department, embattled marketing, the performing business, and group filtering procedures. The determination of this study [11] was to evaluate the effectiveness of transfer learning classification models by applying them to both of these datasets and analysing the results. Reviews left on social media platforms might be valuable for a variety of reasons within smart cities. In this article [12], a BERT-DCNN model is presented.

While the previous Bangla research has depend on on models of DL that place a substantial emphasis on context-independent word embeddings, these models have recently been used to revolutionise the state of the Bangla language [13]. The process of analysing people's feelings and thoughts, also known as sentiment analysis (SA), is an essential activity for many reasons. In this work [14], a new multi-class Urdu dataset was developed for the purpose of conducting sentiment analysis. The dataset was based on user evaluations.

The pre-processed texts are organised as a victimization of words, with each word being represented by a unique integer derived from pre-trained word embedding models. This work [15] also gives a surprisingly extensive examination of selected DL models along with some fine adjustment of the parameters. After that, the polarities, together with the text that has been preprocessed, are input into the neural network and used as training examples.

The purpose of the study [16] is to identify emerging patterns of research about issues that are relevant to relic tourism. To be more specific, this article searches the Web of Science database for all previously published research that include the terms "relic tourism" and finds those that have unresolved problems. The approach of features analysis is very important in the development of developing and improving a sentiment analysis method. The purpose of this study [17] is to put out a model for generalised sentiment analysis that is capable of dealing with noisy input.

In order for the BERT algorithm to be able to identify crimes, a crime dataset that has been labelled using a lexicon-based method is required. In this study [18], the researchers employed a hybrid technique that blends lexicon-based learning with deep learning. The deep learning model they utilised was BERT. In natural language processing, the job of text categorization is considered to be the most basic and crucial. This research [19] addresses the issue by doing a complete evaluation of the methodologies, with a particular importance of extending from conventional models to deep learning.

Twitter, one of the most prominent social media sites, allows users to share their opinions on a variability of topics, including particular ideas, goods, and services. The huge volumes of data that are published in the form of tweets may be used to assist extract the sentiment of users and give useful input that can be used to enhance the quality of both goods and services. In this article [20], the authors offer a method for sentiment analysis that combines lexicon-based methods with other types of methods and deep learning models are utilised together.

On social media platforms, users are allowed to openly voice their thoughts about a wide variety of events. It is possible that doing an analysis of the data will be required in order to determine the perspective of the society about these occurrences. In this work [21], BERT models and machine learning algorithms were used to do sentiment analysis on tweets. In this study [22], the authors suggest a novel model for analysing the sentiment of Weibo content that is depends on BERT and deep learning. To be more specific, first representing the content is created with variable word vectors by the use of BERT.

For predicting the economic crises of institutions using historical data, a variety of data categorization algorithms have been developed. The selection of appropriate variables that are pertinent to the current problem is a crucial step in the creation of a precise financial crisis score FCS method. It is known as a feature selection issue, and it helps to improve classifier outcomes [23]. Sentiment analysis and sarcasm detection on social networking platforms have recently received a lot of interest. The presence of sarcasm in social networking data such as Twitter is a key cause of sentiment misclassification. It is still a difficult topic in natural language processing since it limits the ability to determine people's original attitude. To detect the existence of sarcasm in Twitter data, many feature engineering techniques are available in the literature [24]. Consumers have access to a more streamlined method of purchasing things, thanks to the digital reviews and ratings provided by e-commerce sites. Sentiment analysis is going to be performed on the customer review data in this study [25], and the results are going to be sorted into positive and negative sentiments.

The conventional word2vec approach is incapable of conveying the information that is present in words in their entirety. It has been suggested that the BERT model be used as the article feature extraction model, and that a deep CNN be used in order to abstract the local statistics contained within the article [26]. This work [27] presents a study of sentiment analysis by employing a pre-train model. After that, we investigate key entity recognition as a sentence matching or Machine Reading Comprehension job in various granularities.

3. Proposed Model

In our research work, we consider source of data collected from Amazon and Twitter for cross-domain analysis requires accessing and processing data from both platforms. The proposed model leverages the domain adaptation capabilities of K-BERT, integrating knowledge graph information to enhance sentiment understanding within each domain. The scoring model ensures the aggregation of sentiment scores, considering domain-specific factors, to generate an overall sentiment score that captures sentiment polarity across domains.

The sentiment output labels in sentiment analysis typically include positive, negative, and neutral sentiment categories. These labels are used to categorise the sentiment represented in the text data.

1. Positive: This label indicates that the text expresses a good sentiments or favorable. It implies that the author or speaker has a positive opinion or sentiment towards the discussed subject or topic.

2. Negative: The negative label signifies that the sentiment expressed in the text is negative or unfavorable. It indicates a negative opinion or sentiment towards the discussed subject or topic.

3. Neutral: The neutral label is used when the sentiment expressed in the text is neutral or lacks a clear positive or negative opinion. It indicates a lack of strong sentiment or an expression that is objective, factual, or impartial.

In this paper, the Sentiment Analysis of cross platform is examined using K-BERT model. It consists of three primary procedures; that is, data pre-processing, feature extraction, and feature extraction using Continuous Bag of Words (CBOW), and BERT for Classification with scoring model, as illustrated in Fig. 1.

For Amazon domain, we consider three-star rating value transformations: if the threshold is set at 3.5, a star rating of 4 or 5 would be classified as positive, a rating of 1 or 2 as negative, and a rating of 3 as either neutral or assigned to positive/negative based on additional criteria. For Twitter data involves categorizing tweets into sentiment polarities, typically positive, negative, or neutral. Due to the nature of Twitter, which often contains shorter and informal text, specific techniques are used for polarity classification.

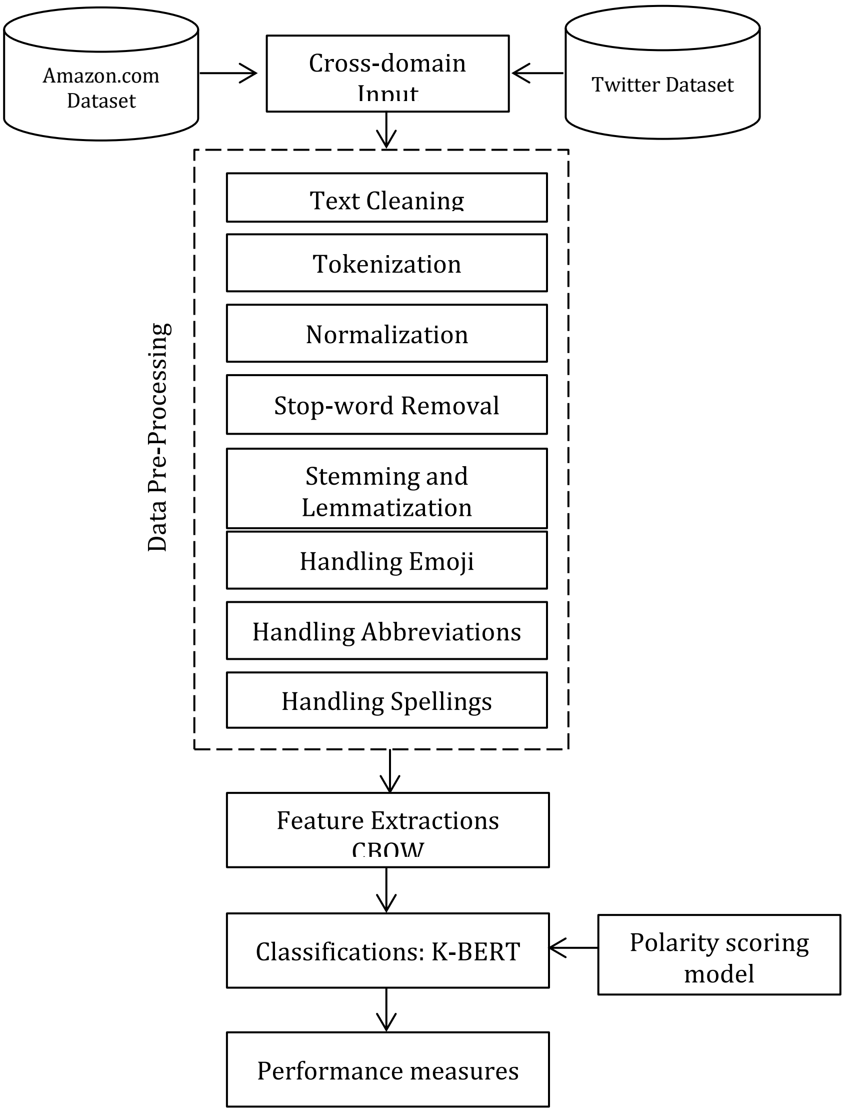

Figure 1: Workflow of Proposed Model

3.1 Data Pre-processing

Data preprocessing in sentiment analysis for both Amazon and Twitter data follows similar steps with slight variations based on the characteristics of the datasets. Here's an outline of the common preprocessing steps for sentiment analysis on Amazon reviews and Twitter data:

Text Cleaning: Remove irrelevant elements such as special characters, URLs, HTML tags, and formatting specific to each dataset. For Amazon reviews, you may need to handle product-specific elements like product names, ratings, or review metadata. For Twitter data, consider removing Twitter handles, hashtags, and retweet indicators.

Tokenization: Split the text into individual words or tokens. Utilize appropriate tokenization techniques like word tokenization or n-gram tokenization. Consider the specific language and any domain-specific considerations.

Normalization: Convert text to lowercase to ensure consistency and reduce the vocabulary size. However, be cautious as capitalization can carry sentiment information in some cases, especially in Twitter data where emphasis is often indicated through uppercase letters.

Stop Word Removal: Remove common stop words that do not contribute significant sentiment information, such as articles, prepositions, or pronouns. Consider domain-specific stop words if necessary.

Stemming and Lemmatization: Apply stemming or lemmatization to simplify words to their basic or root forms. This stage aids in the consolidation of words with comparable meanings. and improves generalization. However, note that stemming may result in the loss of sentiment-bearing information in some cases.

Handling Emoticons and Emoji: Emoticons and emoji can convey sentiment in both Amazon and Twitter data. Handle them appropriately by either removing them or converting them to textual representations that capture their sentiment meaning [28].

Handling Abbreviations and Acronyms: To guarantee accuracy, enlarge the acronyms and abbreviations into their full forms accurate sentiment interpretation. Both Amazon and Twitter data can contain a variety of abbreviations and acronyms specific to their domains.

Handling Spellings and Typos: Correct common misspellings or typos to improve the accuracy of sentiment analysis. Employ spell-checking algorithms or leverage external resources like dictionaries or language models to address this issue.

3.2 Feature Extraction

CBOW is a natural language processing (NLP) paradigm alternative to the traditional Bag of Words model. While BoW focuses on capturing the frequency or presence of individual words in a document, CBOW attempts to anticipate an intended word according to its context words within a fixed window size.

In CBOW, the model learns by considering the context words (input) and predicting the target word (output). The context words are usually represented as one-hot encoded vectors or embeddings. The model then maps the context words to a hidden layer and, finally, predicts the target word based on this hidden representation as shown in fig 2. The training objective is to minimize the prediction error using techniques such as softmax or negative sampling.

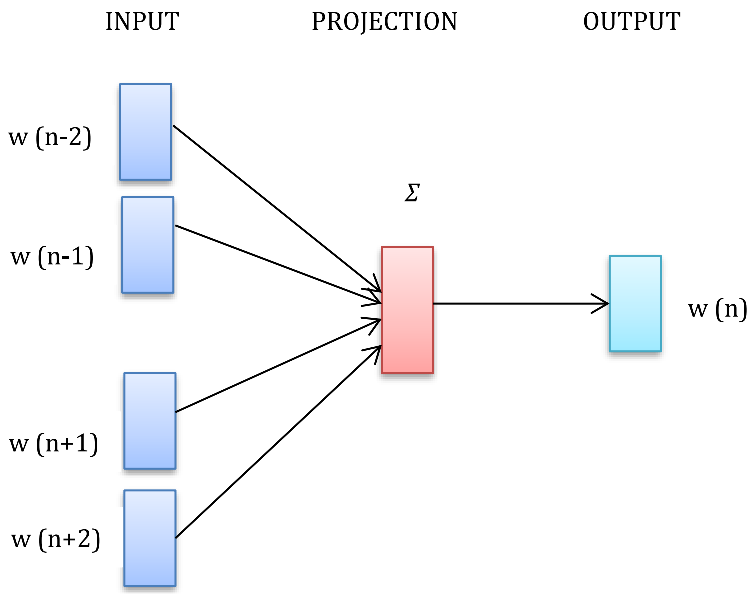

Figure 2: Structure of CBOW

Input Representation:

- Let V be the size of vocabulary, representing the total numeral of unique words in the corpus.
- Let C be the context window size, determining the number of words earlier and after the objective word to consider.
- Each word in the context window is represented by a one-hot encoded vector of length V, where only the corresponding index for that word is 1 and the rest are 0.
- Let x_i be the one-hot encoded i-th context word.

Projection Layer:

The input vectors are projected onto a shared embedding matrix W of dimensions V x D, D denotes one of the dimensions of the word embedding. The i-th context word's projection is indicated as

$$h\_i = W * x\_i \qquad (1)$$

resulting in a dense representation vector of the input.

Aggregation Layer:

The dense vector representations of the context words are aggregated using a summation operation. The aggregated context vector is denoted as

$$h\_agg = (1/CW) * (h\_i) \qquad (2)$$

Here, CW denotes the entire set of context words.

Output Prediction:

For predicting the target word, the aggregates contextual vectors h_agg is input into a layer of softmax. The predicted output is calculated as

$$y = \mathrm{softmax}(U * h\_agg) \quad (3)$$

where U is a weight matrix of dimensions D x V. The function softmax computes the probability distribution across the vocabulary, showing how likely each word is to be the target word.

Training Objective:

The method is trained to reduce the cross-entropy loss among the expected and observed probabilities y and the true labels of the target words. A loss function has been defined as a negative log-likelihood for the target word t expressed as:

$$\mathrm{Loss} = -\log(y[t]) \qquad (4)$$

where y[t] is the predicted probability of the target word. During the training process, the model updates the parameters (embedding matrix W and weight matrix U) using backpropagation and gradient descent to minimize the loss. Where the input context words are projected onto embeddings, aggregated, and used to predict the target word. The model learns to capture the relationships between words within the local context window, enabling it to make accurate predictions based on context.

3.3 K-BERT Model

K-BERT, or Knowledge-enhanced BERT model that incorporates external knowledge to enhance its language understanding capabilities. It leverages knowledge graphs or structured knowledge sources to improve the representation learning of BERT. The K-BERT model aims to address the limitations of BERT in capturing explicit knowledge and making use of external knowledge sources. By incorporating knowledge graphs, K-BERT enriches the contextualized word representations learned by BERT with structured knowledge as shown in fig. 3.

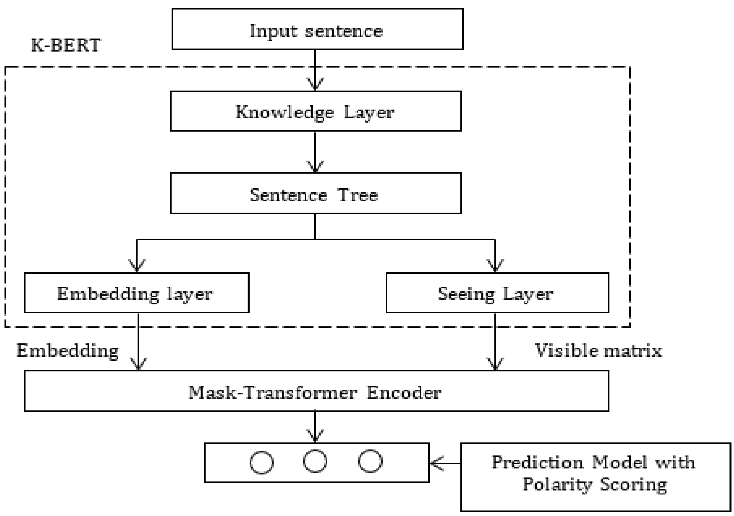

Figure 3: K-BERT Architecture

Transformer Encoder: K-BERT is based on the Transformer architecture, which consists of multiple stacked encoder layers. The forward pass of a single encoder layer can be described by the following equations:

Self-Attention: The self-attention mechanism computes the attention weights between all words in a sequence. Given an input sequence of length N, the self-attention weights A are computed as:

$$A = \mathrm{softmax}((Q * KY\text{^}T) / \sqrt{d}) \quad (5)$$

where Q, KY, and T are the query, key, and value matrices obtained from linear transformations of the input sequence, and d is the dimension of the attention space.

Attention output: By combining the attention weight by the value matrix, the attention output is calculated V:

Masked Language Modeling (MLM): K-BERT is pre-trained using the MLM objective, which involves randomly masking words in the input and predicting them based on the remaining context. The MLM objective is formulated as a maximum likelihood estimation problem. Given an input sequence X, the probability of predicting the masked word y_i is:

$$P(y\_i \mid X) = \mathrm{softmax}(W\_h * h\_i + b\_h) \quad (6)$$

where hidden state is represented as h_i to the masked position, and W_h and b_h are the parameters of the final softmax layer.

Next Sentence Prediction (NSP): K-BERT also incorporates NSP as a pre-training objective to capture sentence-level relationships. Given a pair of sentences A and B, the probability of predicting whether B follows A is computed using a binary classifier:

$$P(\mathrm{IsNext} \mid A, B) = \mathrm{sigmoid}(W\_n * h\_cls + b\_n) \quad (7)$$

where h_cls is the hidden state corresponding to the [CLS] token, and W_n and b_n are the parameters of the binary classifier.

Pseudocode for K-BERT Model

```
def preprocess_text(text): # Preprocessing
kbert_model = load_kbert_model()  # Load K-BERT model
# Load sentiment classifiers for Amazon reviews and tweets
amazon_classifier = load_amazon_classifier()
tweet_classifier = load_tweet_classifier()
# Sentiment classification function
def classify_sentiment(text, domain):
# Preprocess the input text
preprocessed_text = preprocess_text(text)
# Encode the preprocessed text using K-BERT
```

```
encoded_text = kbert_model.encode(preprocessed_text)
# Perform sentiment classification based on the domain
if domain == "amazon":
sentiment = amazon_classifier.predict(encoded_text)
elif domain == "twitter":
sentiment = tweet_classifier.predict(encoded_text)
else:
sentiment = None
return sentiment
# Main function
def main():
# Example input text
amazon_text = "This product is great!"
tweet_text = "I love the new movie!"
# Classify sentiment for Amazon review
amazon_sentiment = classify_sentiment(amazon_text, domain="amazon")
# Classify sentiment for tweet
tweet_sentiment = classify_sentiment(tweet_text, domain="twitter")
```

Polarity scoring model

A polarity scoring model for sentiment analysis is a system that assigns sentiment scores to pieces of text, indicating if the text's attitude is positive, negative, or neutral as shown in fig 4. The polarity scoring model is refined by making use of a dataset consisting of labeled examples, where each example is annotated with its equivalent sentiment polarity.

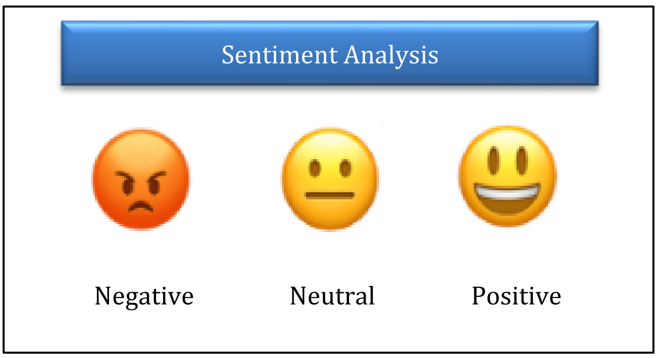

Figure 4: Labels of Sentiment Analysis

Scoring model is a widely used sentiment analysis tool that assigns sentiment scores to individual words or phrases and then combines them to calculate an overall sentiment score for a piece of text.

Pseudocode for Scoring Model

```
# Preprocessing
def preprocess_text(text):
# Perform text cleaning, tokenization, normalization, etc.
# Return preprocessed text
# Load K-BERT model
kbert_model = load_kbert_model()
# Load scoring model
scoring_model = load_scoring_model()
# Sentiment classification function
def classify_sentiment(text):
# Preprocess the input text
preprocessed_text = preprocess_text(text)
# Encode the preprocessed text using K-BERT
encoded_text = kbert_model.encode(preprocessed_text)
# Pass the encoded text through the scoring model
sentiment_score = scoring_model.predict(encoded_text)
# Classify sentiment based on the score
if sentiment_score > 0.5:
sentiment_label = "Positive"
elif sentiment_score < 0.5:
sentiment_label = "Negative"
else:
sentiment_label = "Neutral"
return sentiment_label
def main():
# Example input text
text = "This work is amazing!"
# Classify the sentiment using the cross-domain sentiment classification
```

sentiment = classify_sentiment(text)

The resulting sentiment score can be interpreted as follows:

- Positive sentiment: A score closer to 1 indicates a higher positive sentiment.
- Negative sentiment: A score closer to -1 indicates a higher negative sentiment.
- Neutral sentiment: A score around 0 indicates a relatively neutral sentiment.

4. Results and Discussions

4.1 Dataset Description

For amazon dataset, data collected from https://www.kaggle.com/code/benroshan/sentiment-analysis-amazon-reviews. It is made up of a few million Amazon client evaluations (input text) and ratings in stars (output labels) to teach Text how to use them for sentiment analysis. This file contains the following information as given in table 1:

Table 1: Dataset Description

Id_reviewer	Reviewer ID
P_ID	Product ID
Name_reviewer	Reviewer Name
Rating	Review Rating
Text_review	Review Text
Score	Overall Rating
Time	Review Time

For Twitter dataset, collected from https://www.kaggle.com. The dataset has three sentiments namely, negative(-1), neutral(0), and positive(+1) as shown in fig 5. It has two fields: one for the tweet, and another for the label.

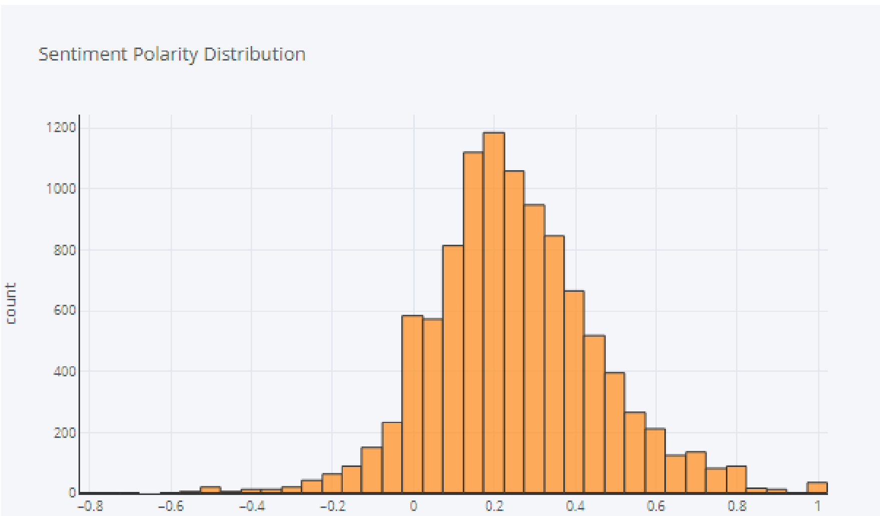

Figure 5: Polarity Distribution

4.2 Experimental Analysis

The suggested model is run on

Python Version: 3.6.5
Learning Rate: 0.01
Dropout Rate: 0.6
Batch Size: 6
Epoch Count: 60
Activation Function: ReLU

Dropout is a regularization technique that randomly sets a fraction of input units to zero at each update, preventing overfitting. Batch size specifies the number of training examples utilized in each iteration of gradient descent. Epoch count determines the number of times the entire dataset is passed forward and backward through the model during training. ReLU (Rectified Linear Unit) is an activation function commonly used in neural networks. Test the trained model's performance against a distinct assessment dataset.

By evaluating the performance of a sentiment analysis system involves assessing its accuracy in predicting sentiment polarity (positive, negative, neutral) for a given text. Here are some commonly used evaluation metrics:

Accuracy: The accuracy metric processes the percentage of correctly classified sentiments (positive, negative, neutral) out of the overall amount of items included in the assessment dataset. It gives an overall evaluation of the model's accuracy.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (TP + TN + FP + FN) \quad (8)$$

Precision: Precision is defined as the ratio of genuine positive predictions (properly recognised sentiment) to the over-all of true positives and false positives (incorrectly identified sentiment). Precision indicates the model's ability to avoid false positives.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (9)$$

Recall: Recall is the percentage of genuine positive expectations to total true positives and false negatives. Recall assesses the model's capability to properly identify favourable outcomes sentiment, for example, in cases where it could have been missed.

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (10)$$

F1 Score: The F1 score is an individual statistic that combines accuracy and recall. It accurately indicates the model's efficiency since it is the average of the harmonics of accuracy and recall. The F1 score considers both FP and FN.

$$\text{F1 Score} = 2 * (\text{Precision} * Recall) / (Precision + Recall) \quad (11)$$

Hamming Loss: Hamming Loss is a metric commonly used for evaluating multi-label classification tasks, including sentiment analysis when multiple sentiment labels are assigned to a text. It measures the fraction of incorrectly predicted labels compared to the total number of labels.

$$\text{Hamming Loss} = (\text{FP} + FN)/(TP + TN + FP + FN) \quad (12)$$

Log Loss: Log loss, frequently referred to as cross-entropy loss or logistic loss, is a popular statistic for evaluating binary classification problems. It processes the accuracy of the predicted probabilities compared to the true labels.

The equation for Log Loss is as follows:

$$\text{Log Loss} = -\left(\frac{1}{n}\right) * \Sigma\big(y * \log(p) + (1 - y) * \log(1 - p)\big) \quad (13)$$

Where:

n: Total no. of samples in the evaluation dataset.

y: The true label (0 or 1) for the sample.

p: The predicted possibility of the positive class (ranging from 0 to 1) for the sample.

The Log Loss equation calculates the negative logarithm of the predicted probability of the correct class (y=1) and the complementary probability of the incorrect class (y=0) for each sample. The predicted probabilities should be between 0 and 1.

A lower Log Loss value indicates better performance, with 0 being a perfect prediction. Higher Log Loss values indicate poor calibration or incorrect predictions.

Lift Value:

The lift value is the ratio of the predicted rate to the baseline rate. The baseline rate represents the probability of the targeted outcome occurring without any intervention or predictive model. This is often the natural occurrence rate in the absence of any targeting or strategy. The predicted rate is the probability of the targeted outcome occurring when using predictive model or strategy. This rate is calculated based on the model's predictions as shown in fig 6.

$$\text{lift} = (TP/(TP + FP)(TP + FN)/(TP + TN + FP + FN) \quad (14)$$

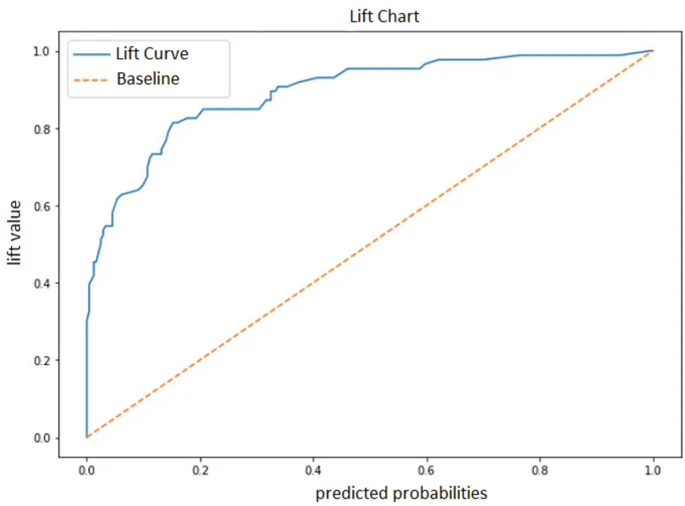

Figure 6: Lift Chart Analysis

The higher the lift value, the more effective your model is at targeting the desired outcome. The comparative analysis of proposed model with cross platform dataset is given in table 2 and 3.

Table 2: Comparative Analysis of Twitter Dataset

Twitter Dataset							
Methods	Accuracy	Precision	Recall	F-Score	Lift	Hamming Loss	Log Loss
K-BERT	98.31	95.8	95.2	96.01	93.76	0.03	1.25
BERT	96.26	94.43	93.52	94.57	91.74	0.2	2.56
Bi-LSTM	95.86	94.38	93.03	92.69	91.59	0.32	2.89
LSTM	94.36	94.08	92.39	92.03	92.11	0.41	3.21
CNN	94.02	93.89	91.68	91.09	91.55	0.46	3.69

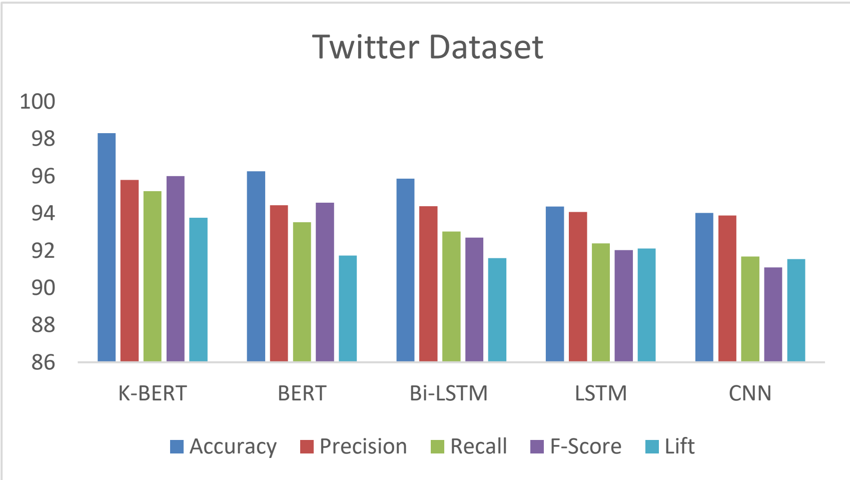

Figure 7: Comparison of Proposed Model using Twitter dataset

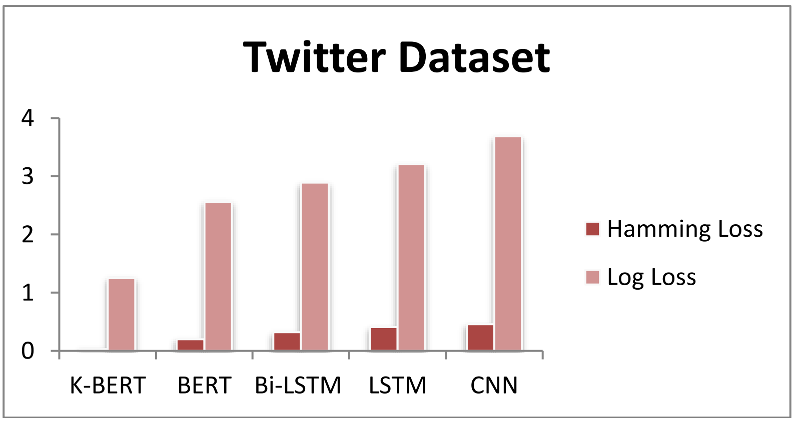

Figure 8: Comparison of Proposed Model using Twitter dataset

The proportionate analysis of the recommended model on the data set from Twitter is determined in Table 2 and Fig 7 and 8. The experimental findings showed that the LSTM as well as CNN models had relatively close accuracy of 94.36% and 94.02%, respectively, while the Bi-LSTM model had 95.86% accuracy. BERT model gives 96.26% accuracy. Finally, K-

BERT model has shown maximum conclusion with accuracy of 98.31%.

Table 3: Comparative Analysis of Amazon Review Dataset

Amazon Review Products							
Methods	Accuracy	Precision	Recall	F-Score	Lift	Hamming Loss	Log Loss
K-BERT	98.56	95.35	94.36	93.89	92.28	0.04	1.29
BERT	96.55	94.26	93.24	92.15	91.02	0.21	2.59
Bi-LSTM	95.36	93.56	92.58	92.03	90.83	0.34	2.93
LSTM	95.04	92.58	91.59	91.89	89.21	0.46	3.63
CNN	94.56	92.05	90.58	91.36	88.19	0.59	3.95

Table 3 and Figure 9 and 10 demonstrate a comparative examination of the suggested model using the Amazon review dataset. The replication results showed that the CNN and LSTM algorithms had accuracy of 94.56% and 95.04%, respectively. Although the Bi-LSTM and BERT methods have a little higher accuracy of 98.56%. The accuracy of the model that was proposed is 7% higher than the performance of the model that is currently being used. The effectiveness of the system that was suggested is compared with that of the current system in the table that you can see above. The current framework improves the accuracy of the proposed model by 7%. The effectiveness of the proposed system is compared with that of the current system in the table that you can see above.

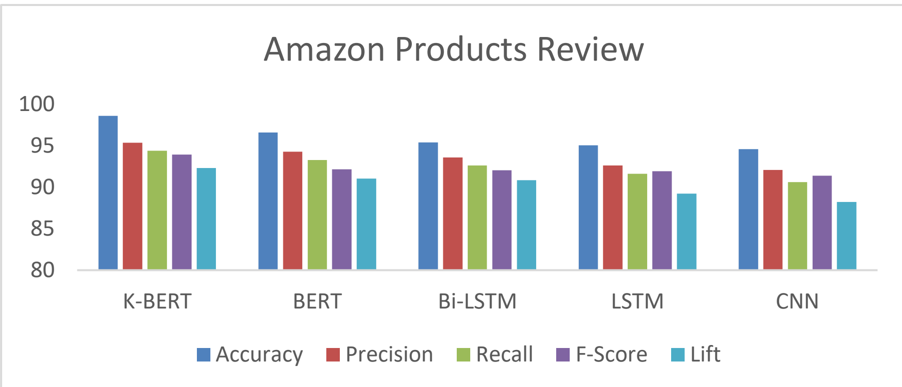

Figure 9: Comparison of Proposed Model using Amazon Product Reviews

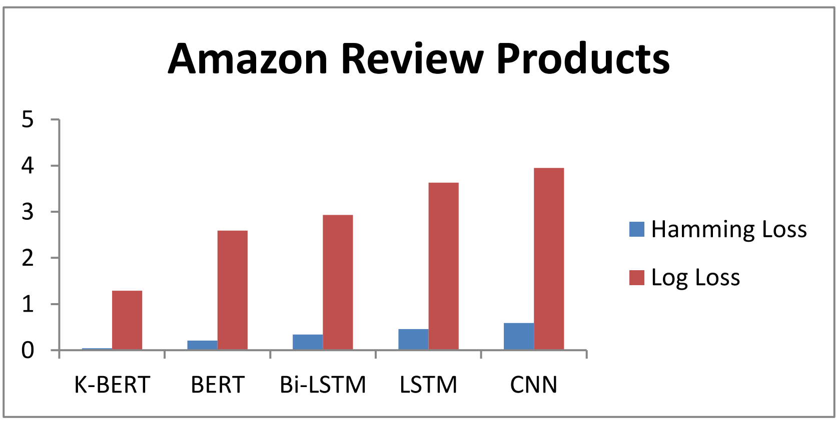

Figure 10: Comparison of Proposed Model using Amazon Product Reviews

5. Conclusion

The combination of K-BERT, a pre-trained language model specifically designed for knowledge-intensive tasks, with a scoring model for cross-domain sentiment classification shows promise in achieving accurate sentiment analysis across different domains. By leveraging the contextual understanding and knowledge embeddings captured by K-BERT, the scoring model can effectively capture the sentiment expressed in text across various domains. The K-BERT model provides a knowledge graphs and pre-training on a vast amount of textual data, enabling it to capture domain-specific information and contextual understanding. This enhances its ability to comprehend the sentiment nuances present in different domains. The scoring model complements K-BERT by utilizing the learned representations and further refining sentiment analysis predictions. The scoring model can be trained using a labeled dataset for sentiment polarity, to fine-tune the sentiment analysis performance. This allows the model to adapt to specific domain characteristics and achieve higher accuracy in sentiment classification.

6. Future Work

In future, extend the K-BERT and scoring model to perform context-aware sentiment classification. Sentiment can change based on the surrounding text, and considering the wider context can lead to more accurate sentiment predictions to minimize the loss level functions. Develop techniques to incorporate contextual information, such as the preceding or following sentences, into the models to improve their understanding of sentiment nuances. On the other hand, social media networking sites, sentiment analysis and sarcasm detection have recently attracted a lot of attention. Sarcasm is a prominent factor in the incorrect classification of sentiments in social networking data, such as that seen on Twitter. Since it limits the means of detecting the people's original attitude, it continues to be a difficult problem for natural language processing NLP. To find sarcasm in data from Twitter, Facebook, Instagram etc., there are a variety of feature engineering techniques concentrate on upcoming work.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author Contributions

Conceptualization, Shanthini and Subalakshmi; Methodology, Shanthini and Subalakshmi; Software, Shanthini; Validation, Shanthini; Formal analysis, Shanthini and Subalakshmi; Investigation, Shanthini and Subalakshmi; Resources, Shanthini; Data curation, Shanthini; Writing-original draft preparation, Shanthini; Writing-review and Editing, Shanthini; Visualization, Shanthini; Supervision, Shanthini; Project administration, Shanthini; Funding acquisition, Shanthini. All authors have read and approved the final manuscript.

References

1. Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. Artificial Intelligence Review, 55(7), 5731-5780.
2. Yuan, Y., & Lam, W. (2022, February). Sentiment Analysis of Fashion Related Posts in Social Media. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (pp. 1310-1318).
3. Wang, W., Guo, L., & Wu, Y. J. (2022). The merits of a sentiment analysis of antecedent comments for the prediction of online fundraising outcomes. Technological Forecasting and Social Change, 174, 121070.
4. Christ, L., Amiriparian, S., Baird, A., Tzirakis, P., Kathan, A., Müller, N., ... & Schuller, B. W. (2022, October). The muse 2022 multimodal sentiment analysis challenge: humor, emotional reactions, and stress. In Proceedings of the 3rd International on Multimodal Sentiment Analysis Workshop and Challenge (pp. 5-14).
5. Rodrigues, A. P., Fernandes, R., Shetty, A., Lakshmanna, K., & Shafi, R. M. (2022). Real-time twitter spam detection and sentiment analysis using machine learning and deep learning techniques. Computational Intelligence and Neuroscience, 2022.
6. Wu, H., & Shi, X. (2022, May). Adversarial soft prompt tuning for cross-domain sentiment analysis. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 2438-2447).
7. Luo, Y., Guo, F., Liu, Z., & Zhang, Y. (2022). Mere contrastive learning for cross-domain sentiment analysis. arXiv preprint arXiv:2208.08678.
8. Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., & Wang, P. (2020, April). K-bert: Enabling language representation with knowledge graph. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 03, pp. 2901-2908).
9. Wu, Z., Jiang, D., Wang, J., Zhang, X., Du, H., Pan, L., ... & Hou, T. (2022). Knowledge-based BERT: a method to extract molecular features like computational chemists. Briefings in Bioinformatics, 23(3), bbac131.
10. Bai, H., Wang, D., Feng, S., & Zhang, Y. (2022, August). EK-BERT: An Enhanced K-BERT Model for Chinese Sentiment Analysis. In China Conference on Knowledge Graph and Semantic Computing (pp. 136-147). Singapore: Springer Nature Singapore.
11. Qasim, R., Bangyal, W. H., Alqarni, M. A., & Ali Almazroi, A. (2022). A fine-tuned BERT-based transfer learning approach for text classification. Journal of healthcare engineering, 2022.
12. Jain, P. K., Quamer, W., Saravanan, V., & Pamula, R. (2022). Employing BERT-DCNN with sentic knowledge base for social media sentiment analysis. Journal of Ambient Intelligence and Humanized Computing, 1-13.
13. Prottasha, N. J., Sami, A. A., Kowsher, M., Murad, S. A., Bairagi, A. K., Masud, M., & Baz, M. (2022). Transfer learning for sentiment analysis using BERT based supervised fine-tuning. Sensors, 22(11), 4157.
14. Khan, L., Amjad, A., Ashraf, N., & Chang, H. T. (2022). Multi-class sentiment analysis of urdu text using multilingual BERT. Scientific Reports, 12(1), 5436.
15. Bhowmik, N. R., Arifuzzaman, M., & Mondal, M. R. H. (2022). Sentiment analysis on Bangla text using extended lexicon dictionary and deep learning algorithms. Array, 13, 100123.
16. Das, S., Mondal, S., Puri, V., & Vrana, V. (2022). Structural review of relics tourism by text mining and machine learning. Journal of Tourism, Heritage & Services Marketing (JTHSM), 8(2), 25-34.
17. Kokab, S. T., Asghar, S., & Naz, S. (2022). Transformer-based deep learning models for the sentiment analysis of social media data. Array, 14, 100157.
18. Boukabous, M., & Azizi, M. (2022). Crime prediction using a hybrid sentiment analysis

approach based on the bidirectional encoder representations from transformers. Indones. J. Electr. Eng. Comput. Sci, 25(2), 1131-1139.

19. Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., ... & He, L. (2022). A survey on text classification: From traditional to deep learning. ACM Transactions on Intelligent Systems and Technology (TIST), 13(2), 1-41.
20. Aljedaani, W., Rustam, F., Mkaouer, M. W., Ghallab, A., Rupapara, V., Washington, P. B., ... & Ashraf, I. (2022). Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry. Knowledge-Based Systems, 255, 109780.
21. Guven, Z. A. (2021, September). Comparison of BERT models and machine learning methods for sentiment analysis on Turkish tweets. In 2021 6th International Conference on Computer Science and Engineering (UBMK) (pp. 98-101). IEEE.
22. Li, H., Ma, Y., Ma, Z., & Zhu, H. (2021). Weibo text sentiment analysis based on bert and deep learning. Applied Sciences, 11(22), 10774.
23. Veeramanikandan V & Jeyakarthic M (2019), A Futuristic Framework for Financial Credit Score Prediction System using PSO based Feature Selection with Random Tree Data Classification Model, International Conference on Smart Systems and Inventive Technology (ICSSIT), (pp. 826-831). IEEE.
24. Jeyakarthic M, & Senthilkumar J, (2022). Optimal Bidirectional Long Short Term Memory based Sentiment Analysis with Sarcasm Detection and Classification on Twitter Data, 2nd Mysore Sub Section International Conference (MysuruCon), (pp. 1-6). IEEE.
25. Geetha, M. P., & Renuka, D. K. (2021). Improving the performance of aspect based sentiment analysis using fine-tuned Bert Base Uncased model. International Journal of Intelligent Networks, 2, 64-69.
26. Man, R., & Lin, K. (2021, April). Sentiment analysis algorithm based on bert and convolutional neural network. In 2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC) (pp. 769-772). IEEE.
27. Zhao, L., Li, L., Zheng, X., & Zhang, J. (2021, May). A BERT based sentiment analysis and key entity detection approach for online financial texts. In 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD) (pp. 1233-1238). IEEE.
28. Arifiyanti, A. A., & Wahyuni, E. D. (2020, October). Emoji and emoticon in tweet sentiment classification. In 2020 6th Information Technology International Seminar (ITIS) (pp. 145-150). IEEE.