International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
IJISAE
ISSN:2147-6799 www.ijisae.org Original Research Paper

Sentiment Analysis for Prediction of Brand Value Using Albert Model

Pallavi Suryavanshi[1*], Dr. Sharad Gangele[1]

Submitted:12/03/2024 **Revised**: 27/04/2024 **Accepted**: 04/05/2024

Abstract: In today's dynamic and global business environment, organizations face the challenge of meeting customer expectations while effectively managing their supply chain. Understanding customer demands and accurately getting product sales is critical to achieve this. The effectiveness of the existing “BERT” in sentiment analysis is well established and its resource-intensive models might face challenges in deployment, especially in scenarios with constraints on computational resources. This study explores the use of sentiment analysis and the ALBERT model to predict brand value based on customer reviews. Both BERT and ALBERT models are more powerful, but ALBERT offers a more efficient alternative without compromising performance, making it particularly appealing for tasks where computational efficiency is a priority. The proposed approach combines various techniques, including tokenization, POS tagging, and dependency parsing, to improve the accuracy of SA models. This study not only establishes the effectiveness of transformer architectures in sentiment analysis but also helps the advancement of brand valuation approaches. The findings have impacts on marketers, as they provide a powerful tool for assessing customer sentiment and obtaining brand value with unprecedented accuracy. The results of the proposed model outperform with 95.98% of accuracy, 96.72 % of precision, 94.38% of recall, and 95.53% of F-measure.

Keywords: *Sentiment Analysis (SA), BERT (Bidirectional Encoder Representations from Transformers), ALBERT (A Lite BERT), Customer Reviews, Supply Chain Management, Brand Value, Dependency Parsing, Part of Speech (POS).*

1. Introduction

With increasing demands from consumers for price and quality, manufacturers can no longer rely solely on their cost advantage over competitors. [1]. In contrast, an important tactic for businesses currently is to efficiently manage their vendor network and better understand what clients want. [2]. Over the years, information technologies have significantly aided manufacturers in improving supply chain management. A better estimate for sales is essential for successfully managing the supply chain so that a manufacturer doesn't excessively purchase items that are produced. Big data and user-generated information are two developing areas in product forecasting [3]. The latest marketing has shown the impact of content created by users on product sales. This type of user-generated content has an impact on online shopping that has become one of the most common business models in the industry. Amazon.com and Alibaba's Taobao.com are two successful e-commerce sites that enable prospective consumers to view suggestions from other individuals before buying choices. [4].

In an e-commerce environment, consumers are exposed to readily available facts which may greatly impact their purchasing decisions.

[1]Department of computer application Research scholar, RKDF University Airport by pass road,Gandhi Nagar,Bhopal-462033, Madhya Pradesh,India*
https://orcid.org/0009-0001-1322-3030
[2]Department of computer application Professor, RKDF University Airport by pass road,Gandhi Nagar,Bhopal-462033, Madhya Pradesh,India
https://orcid.org/0009-0005-0084-2392
Corresponding author mail id:Pallavisurijbp@gmail.com

In Amazon platform includes the item details such as costs, advertising offers such as price reductions, and free shipping, and online review confirmation like sentiment, quality, and content of the reviews.

Predicting a potential customer's purchasing decision and sales performance has become critical to a company's supply chain management as Digital commerce storefronts and marketplaces have become a single primary ordering network for consumers.

Since consumers are making immediate choices in the e-commerce environment utilizing different websites, has become possible to forecast product sales and client needs using data collected online. [5]. Classifying analysis into customer thought into positive and negative comments offers sentiment orientation of the review, resulting in a good assessment. Sentiment is a noun that refers to sensitivity or emotional sentiments. Sentiment is a true and sophisticated sense that is influenced by emotion rather than reason or truth [6]. Sentiment research is becoming an increasingly significant tool for firms trying to understand their customers better and increase the value of their brands [7]. Companies may get significant insights into their customer's needs and preferences by examining the emotions and views expressed in customer feedback, and utilize this knowledge to enhance their goods and services.

For example, if a company finds that consumers regularly express negative sentiments about a certain product feature or component of their service, they may utilize this knowledge to make adjustments to address these concerns.

They can improve customer satisfaction and, as a result, brand value. Businesses may enhance customer happiness, find new development possibilities, and ultimately boost their brand value by assessing consumer sentiment across many channels and leveraging this information to make data-driven choices. According to [8] SA detects trends in people's ideas by mining data and applying NLP, linguistic computation, and textual evaluation to personal emotions obtained from the web, social media, and other sources. In general, sentiment analysis is performed in two ways: lexicon-based and ML-based [9]. The lexicon-based technique makes use of pre-built lexicon resources that contain sentiment polarity terms [10]. Lexicon-based approaches are scalable and computationally efficient [11]. For classification, data is separated into training and testing sets in ML algorithms, which require a large quantity of training data and are semantically weak. While lexicon-based methods can be useful in some situations, they may be less accurate or flexible than ML-based approaches, particularly when applied to more formal or domain-specific text. In this study, we examined the feelings indicated by customers in their Amazon.com evaluations for various items. These attitudes were then associated with the general polarity of texts as positive, negative, and neutral. To analyze these attitudes, we employed a fine-tuned pre-trained model that performed admirably on huge sentence and token level tasks. Our study's primary goal is to look at user reviews and categorize these into negative, neutral, and positive.

The main findings of our research:

- To emphasize the significance of sentiment analysis and the ALBERT model in predicting brand value through the analysis of customer reviews, the research paper showcases the potential of user-generated content and big data in understanding consumer preferences and emotions.

- To illustrate the importance of precise product sales predictions for effective supply chain management, the study utilizes techniques such as tokenization, POS tagging, and dependency parsing to improve the accuracy of SA models.

- To highlight SA, this research contributes leveraging information technologies for optimizing supply chain management and improving customer satisfaction.

- To demonstrate SA in ecommerce, our research highlights the potential for firms to use customer feedback for data-driven decisions, enhancing product offerings, customer satisfaction, and brand value.

2. Literature Review

Several researchers have proposed solutions to problems with SA, as well as the analysis and mining of consumer evaluations. This section contains a thorough examination of the previous study.

Based on the Amazon review dataset, Jagdale et al [12] developed a dictionary-based strategy using ML techniques such as NB and SVM. However, this method is not as precise. Guia et al. [13], Amazon Unlocked mobile phone reviews dataset compares four of the most popular text classification Algorithms. They are NB, SVM, DT, and RF. Dadhich et al. [14] proposed the Product Comment Summarizer and Analyser (PCSA), which summarised and categorized comments gathered from Amazon's data domains. Supervised learning methods such as NB, LR, SentiWordNet, RF, and KNN approaches, the network polarity was verified for positive, negative, and neutral remarks. These methods, however, may produce less accurate results and necessitate more manual intervention. Furthermore, Shrestha et al [15] propose a deep learning-based sentiment analysis approach to solving a common problem in e-commerce websites where the user's review does not match its rating. It can only consider the following terms in a sentence. As a result, it does poorly in terms of grasping the overall meaning of a statement and understanding the mood represented. To address this issue, Du et al [16] suggested a BERT-based model for cross-domain SA. Our goal is to incorporate domain knowledge into BERT and encourage BERT to be more domain-aware. The authors specifically carry out post-training and adversarial training. In a self-supervised environment, a unique domain-distinguishing pre-training task is devised to extract domain-specific features. However, this requires a larger number of parameters, reducing training efficiency.

Desai et al [17] used Deep Learning-BERT and LSTM- and ML -DT, LR, SGD, Multinomial NB, and SVM methods to categorize sentiments of Amazon reviews. However, BERT-LSTM can be slow to train and fine-tune, and it may struggle with sentiment analysis that requires a deeper understanding of long-term dependencies because it processes text in a fixed sequence and may not capture all relevant information. AlQahtani et al [18] analyze the Amazon Reviews Dataset and investigate sentiment classification using several ML approaches such as LR, RF, NB, Bidirectional LSTM (Long Short term memory), and BERT. It still has several issues, such as extra parameters for training, not collecting all necessary information, and human interaction, which reduces the accuracy of the outcome.

Geetha et al [19] introduced a BERT Base uncased model, a strong DL Model, to highlight the issue of SA, such as low accuracy and extensive training times. Mostafa et al. [20] investigated many ML algorithms, including KNN, DT, NB, RF, LR, SVM, Bidirectional LSTM, GRU, and Bert, and showed that BERT networks are well suited for sentiment categorization in Amazon product evaluations. However, we still need to improve BERT's efficiency and scalability.

For sentiment categorization and analysis, Iqbal et al [21] used DL methods such as LSTM, and RNN models. However, learning patterns and relationships in text data still necessitate a large amount of training data. Mutinda et al. [22] introduced LeBERT, a sentiment classification technique that combines sentiment lexicon, Ngrams, BERT, CNN. These models are used for word vectorization in NLP. CNN is utilized as a DNN classifier for feature mapping and sentiment classification. Despite producing the best results, this method is more complex and requires more training time.

However, due to the large volume of data collected from various sources, analyzing customer reviews to predict accurate sentiments has proven to be difficult and time-consuming. Several researchers have used algorithms, methods, and models to tackle this problem. ML and DL approaches, as well as pre-trained models and regression models, are examples of these. Studies and studies have found polarity incoherence, model overfitting, complexity and performance difficulties, as well as high data processing costs. This research was carried out to address these troubling concerns by developing a high-performance yet cost-effective algorithm for predicting accurate emotions from big datasets of customer evaluations.

3. Proposed Work

Sentiment analysis, a method for analyzing text data, automatically identifies and classifies ideas expressed in the text to determine overall emotion. The approach categorizes sentiments as positive, neutral, or negative using sentiment analysis tools. Due to the variety of linguistic styles, domain-specific terminology, and complex sentence structures in customer evaluations, pre-trained models tend to be more successful in sentiment analysis compared to lexicon-based techniques. The Amazon review dataset was utilized for sentiment analysis based on customer reviews to obtain brand value. The input was pre-processed using Tokenization, the first step to extract essential sentiment-bearing words from the text. This was followed by the utilization of POS tagging and dependency parsing, combining these techniques with ALBERT to further enhance the accuracy of sentiment analysis models. The combination of dependency parsing and POS tagging allowed for the extraction of both grammatical and semantic information from the text, leading to a more comprehensive understanding of the review sentence.

A Fine-tuned ALBERT model was employed for the sentiment analysis task, using the parse tree and POS tags for each sentence along with features such as the sentence length to generate a sentiment score for each sentence. During the training process, the weights of the ALBERT model were adjusted using the Adam optimizer to minimize the disparity between the predicted sentiment scores and the true labels. The learning rate and batch size were optimized to improve the training process. By leveraging these techniques, the research effectively utilized the strengths of each approach, resulting in improved accuracy of sentiment analysis models and providing more precise insights into the sentiment expressed in the review text.

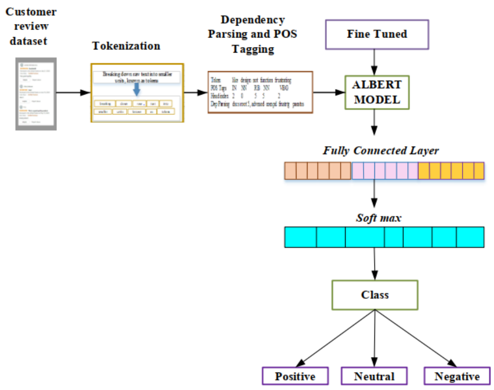

Fig. 1. Sentiment analysis using the ALBERT Model

3.1. Pre-Processing

Pre-processing is an initial stage in NLP tasks that involves cleaning and translating raw text data into an analysis-ready format. Tokenization and POS tagging are two important pre-processing approaches for understanding the structure and meaning of text. Tokenization is a crucial step in NLP that includes splitting the text into tokens. It helps in the creation of a structured representation of the text, which is required for further analysis. Dependency parsing examines a sentence's grammatical structure through the identification of word relationships. A dependency parser produces the output as a tree structure called a dependency tree, which describes syntactic links between words in a sentence. POS tags allow for a deeper understanding of the text's grammatical structure. It is useful for tasks like data extraction, sentiment analysis, and syntactic analysis. These pre-processing procedures set a foundation for advanced NLP activities, allowing machines to grasp and interpret textual data more meaningfully. This analysis, combined with the ALBERT model, enables accurate results about brand value.

3.2. Albert Model

Our research proposes the ALBERT model to enhance the accuracy. ALBERT employs a pre-training loss that predicts the ordering of two consecutive texts. The architecture of ALBERT is same as BERT. The non-linearity of the GELU (Gaussian Error Linear Unit) is widely used in NN models, most notably in BERT and ALBERT. The GELU activation function is well-known for its smoothness and effectiveness in a variety of NLP tasks. The majority of the issues with BERT are caused by the large number of parameters that are trained, making it slow and bulky. ALBERT attempted to reduce the total of 110 million parameters to 12 million, making it better for practical deployment and faster. The configuration of ALBERT model is shown in Table 1 and the structure is shown in Figure 2.

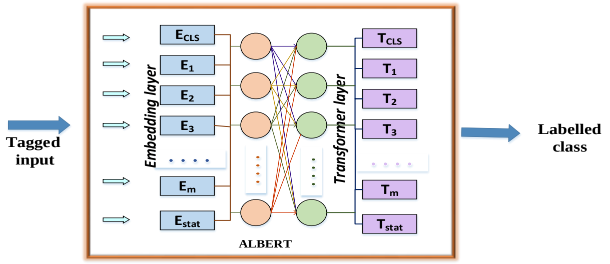

Fig. 2. ALBERT Architecture

The three main contributions that ALBERT provides to BERT's choices for design. Factorized embedding parameterization. By dividing the massive vocabulary embedding matrix into two small matrices, the hidden layer size (H) is decoupled from the vocabulary embedding size (E). This division facilitates increasing the H without considerably increasing the E. This parameter reduction is large when $H \gg E$. In practice, NLP requires a large vocabulary size (V). If $E = H$, then H expands the embedding matrix, which has dimension $V \times E$ resulting in a model with billions of parameters, the vast majority that will be sparingly updated while training..

By factorizing the embedding parameters for ALBERT, have to separate them into two fewer matrices. Then simply directing the one hot vectors into H, we present them first to a lower dimensional E and then into the hidden space. Reduce the embedding parameters from $O(V \times H)$ to $O(V \times E + E \times H)$ using decomposition. When $H \gg E$ is present, this parameter reduction is considerable. Choose the same E for all word fragments because it is considerably more equally dispersed throughout reports than whole word embedding, which requires a distinct embedding size for each word.

3.2.1. Cross Layer Parameter Sharing

It is another way for increasing parameter efficiency. The parameters can be shared through sharing only feed-forward network parameters across layers, sharing only attention parameters across levels, and sharing all parameters across layers. ALBERT uses a strategy of sharing all parameters across layers. Weight-sharing was discovered to affect network characteristics that stabilize. Despite a decrease in L_2 distance and cosine similarity to BERT, they still do not converge to zero after 24 layers.

3.2.2. Inter sentence coherence loss

ALBERT employed Masked Language model in training. ALBERT, on the other hand, employed a new loss called SOP (Sentence Order Prediction) instead of NSP (Next Sentence Prediction). NSP is a binary classification loss that determines the next sentence by checking for coherence and subject. The SOP, on the other hand, merely searches for sentence consistency and avoids subject.

The SOP loss employs both positive (two consecutive segments from the same document) and negative (the same two consecutive segments but with their order reversed) examples. Then the model detects minor differences in discourse level coherence features. During testing, it was discovered NSP cannot solve the SOP task at all i.e., it learns the easier topic-prediction signal and completes the SOP task at a random baseline level, whereas SOP can solve the NSP task to a significant extent. Finally, ALBERT models increase downstream task performance for multi sentence encoding tasks consistently.

Table 1. Configuration of ALBERT model

Model	***Parameters***	***Layers***	***Hidden***	***Embedding***	***Parameter***
Base	12M	12	768	128	True
Large	18M	24	1024	128	True
xlarge	60M	24	2048	128	True
xxlarge	235M	12	4096	128	True

The ALBERT model is trained on large corpora to learn broad language representations. However, fine-tuning is required to make it capable of more specialized aspects are text categorization and SA. To maximize the model effectiveness for targeted applications, initially, the ALBERT model has to be fine-tuned.

3.3. Fine-Tuning

Fine-tuning the ALBERT model involves training it on task-specific data to adjust the pre-trained model to specific data tasks. The process begins with the selection of a pre trained ALBERT architecture that is suitable for the downstream task. After absorbing a wide range of linguistic patterns during pre-training, this model provides a strong foundation for subsequent task-specific adaptations. The illustration of Fine-tuning ALBERT is in Figure 3.

To ensure consistency with the pre-training phase, the input data is tokenized using the same tokenizer as was used during ALBERT's pre-training. This stage ensures that the fine-tuned model analyses data in the same way that it did during its original training. Furthermore, the input data is prepared to meet the model's unique requirements, taking into consideration segment IDs and positional embeddings as needed.

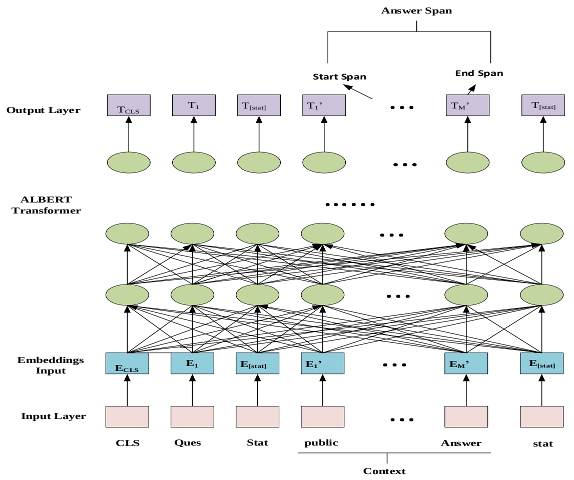

Fig. 3. Fine Tuning of ALBERT

The ALBERT model is fine-tuned by training by task-specific dataset, initializing with pre-trained weights, and changing the output layer to accept specific classes or labels. This process enables the model to learn task-specific patterns. Another important part of fine-tuning is hyper parameter tuning. Experimenting with parameters like learning rate, batch size, and regularization approaches improves performance. Regular evaluations on a separate validation set hyper parameter adjustments to attain the best outcomes feasible. Once the ALBERT model has been fine-tuned to satisfy the appropriate performance criteria, it may be used to make predictions on new, previously unknown data. This fine-tuning process allows for a smooth transition from generic knowledge obtained during pre-training to task-specific proficiency, making ALBERT a versatile tool across a range of natural language processing applications. Post Process of ALBERT fine-tuned has steps such as applying a softmax function for classification tasks, which may be implemented depending on the nature of the downstream task.

3.4. Fully Connected Layer (FCL)

Before the input data reaches the FCL, it is often flattened or vectorised. This process converts the input data into a one dimensional array or vector, making it suitable for processing. This layer also known as a dense layer, is an important component of the neural network design in sentiment analysis. It is critical in understanding complicated patterns and relationships in input data, which leads to sentiment predictions. It serves as a feature representation layer, transforming the input data into a format that the model can easily interpret. The process of steps to calculate the results of update weight and bias is represented in the block diagram Figure 4. The input data for SA is frequently word embeddings helps in decision-making by combining data from several input features and making predictions based on learned patterns. It transforms the input data into a representation which assists in making a final decision regarding the text.

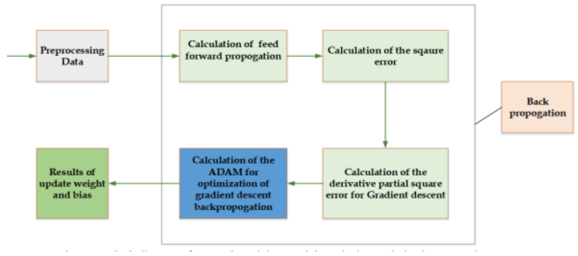

Fig. 4. Block diagram of Network Training: Weight and Bias Optimization Strategies

Each neuron in the FCL is connected to every neuron in layer before it which computes a weighted sum of the input values, where the weights are the model learning parameters. During the training phase, the FCL's weights are updated using backpropagation (BP) and Gradient Descent (GD) optimization algorithms. The goal is to reduce the differences between predicted and actual sentiments in the training data. The model learns the weight of different features, capturing the relevant patterns for sentiment analysis.

3.4.1. Back Propagation and Gradient Descent Algorithm

The neural network fed input data during the forward pass. Using the current set of weights, the input is transformed across each layer, including the FCL. The forward pass computes the network's predicted output for a given input. To calculate the loss, the predicted output is compared to the actual target values (ground truth).

BP is the process of computing loss gradients for model parameters (weights) layer by layer, beginning with the output layer and moving backward through the network. The chain rule of calculus is used to calculate the gradients, which represent how much the loss varies as the weights change. It attempts to reduce the cost function by adjusting network weights and biases.

Gradient function $C(x_1, x_2, \dots, x_m)$ in point x is a vector C's partial derivatives in x.

$$\frac{\partial \mathrm{C}}{\partial \mathrm{x}} = \left[\frac{\partial \mathrm{C}}{\partial \mathrm{x}_1}, \frac{\partial \mathrm{C}}{\partial \mathrm{x}_2}, \dots, \frac{\partial \mathrm{C}}{\partial \mathrm{x}_\mathrm{m}}\right] \tag{1}$$

The derivative of function C quantifies the output value respect to changes in its parameter input value x. Gradient indicates how much the parameter x must be changed in either a positive or negative direction to minimize C. Using chain rule process, calculate the gradient for a single weight w_{jk}^{l}.

$$\frac{\partial \mathrm{C}}{\partial \mathrm{w}_{\mathrm{jk}}^{\mathrm{l}}} = \frac{\partial \mathrm{C}}{\partial \mathrm{z}_{\mathrm{j}}^{\mathrm{l}}} \frac{\partial \mathrm{z}_{\mathrm{j}}^{\mathrm{l}}}{\partial \mathrm{w}_{\mathrm{jk}}^{\mathrm{l}}} \text{ ; Chain rule} \tag{2}$$

Where

$$z_j^l = \sum_{k=1}^{m} w_{jk}^l a_k^{l-1} + b_j^l \tag{3}$$

Where m denotes neuron number in the l-1 layer, z denotes neuron value, a is the activation value

For differentiation,

$$\frac{\partial z_j^l}{\partial w_{jk}^l} = a_k^{l-1} \tag{4}$$

Final Value,

$$\frac{\partial C}{\partial w_{jk}^l} = \frac{\partial C}{\partial z_j^l} a_k^{l-1} \tag{5}$$

Similarly, the set of equations applies to b_j^l,

Chain rule,

$$\frac{\partial C}{\partial b_j^l} = \frac{\partial C}{\partial z_j^l}\frac{\partial z_j^l}{\partial b_j^l} \tag{6}$$

Calculating derivative,

$$\frac{\partial z_j^l}{\partial b_j^l} = 1 \tag{7}$$

Final Value,

$$\frac{\partial C}{\partial b_j^l} = \frac{\partial C}{\partial z_j^l} \tag{8}$$

The expression $\frac{\partial C}{\partial b_j^l}$ is called a gradient, which referred to as

$$\delta_j^l = \frac{\partial C}{\partial z_j^l} \tag{9}$$

Eq 9 is called the local gradient. We can optimize the model's parameters using gradients:

Steps for GD Algorithm

Step 1: Set the model parameters at random

Step 2: Determine the gradient cost function for each parameter. It involves making a partial differentiation of the cost function to the parameters.

Step 3: Update the model's parameters by moving in the opposite direction of the model. Here, select a hyperparameter learning rate. It helps in determining the gradient's step size.

Step 4: Iteratively repeat steps 2 and 3 to find the optimal parameter for the stated model. Each iteration is known as an epoch.

If the termination condition is not met

$$w := w - \varepsilon \frac{\partial C}{\partial w} \tag{10}$$

$$b := b - \varepsilon \frac{\partial C}{\partial b} \tag{11}$$

End

Where ε is the learning rate which defines the influence of the gradient, b and w are the matrix representations of biases and weights. The weight and bias optimization algorithm is also known as Gradient descent. Here initial values of w and b are chosen at random. The termination condition is satisfied once the cost function is minimized. Training procedure repeated until a predetermined stopping criterion is fulfilled. This criterion could be obtaining a certain level of performance, or observing a plateau in performance improvement.

3.4.2. Adam Optimizer

Adaptive Moment Estimation is an optimization algorithm for updating weights of a neural network during the training process. In sentiment analysis using the ALBERT model or any other neural network, Adam optimizer is employed to decrease the loss function and adjust model's parameters. In the output layer of weights and biases, ADAM computes partial derivatives of error. It then estimates the initial moment and corrects it in the output layer to acquire weight and bias. In the hidden layer, the second moment is calculated, and the weights and bias updates are obtained.

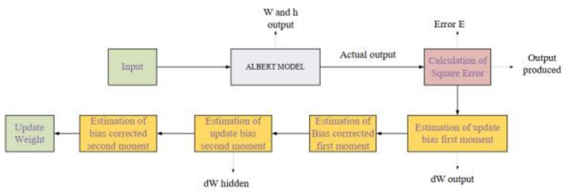

Fig. 5. Architecture of ADAM

In BP gradient descent, ADAM can change the parameters of the output torque, which is a torque distribution first and second moments. The main aim to reduce the squared error at neural network's output at each iteration (epoch). The results demonstrated that ADAM may minimize the squared error at the output of neural networks at each iteration (epoch). The gradient is determined on BP to optimize the network. The parameters of the Adam optimizer are updated based on these moving averages Here are the steps of ADAM as follows. There are several terms of formulas used to define

θ: Model parameters (weight and bias)

δ_j^l : Gradient loss function

β_1 *and* β_2: Exponential decay rates for the first and second moments are usually close to 1 i.e. 0.9 to 0.999

ϵ: a small constant to prevent division by zero usually10^{-8}.

t: current iteration(time step)

- Initialize the variables m_o=0 ;v_0 = 0 ie. $m_{weight} = m_{bias} = v_{weight} = v_{bias} = 0$ (If the initial iteration is t = 1, t = 1-1 = 0 (time step / early input iterations).

➢ Gradient calculation to estimate the first-time step moment $\delta_j^l = \frac{\partial C}{\partial z_j^l}$

➢ Update weight and biased first-moment estimation

$$\mathrm{m}_{\mathrm{weight(t)}} = \beta_1 * \mathrm{m}_{\mathrm{weight(t-1)}} + (1-\beta_1) * \delta_j^l \quad 12)$$

$$\mathrm{m}_{\mathrm{weight(t)}} = \beta_1 * \mathrm{m}_{\mathrm{weight(t-1)}} + (1-\beta_1) * \delta_j^l \quad (13)$$

➢ Calculate the bias and weight-corrected estimation for the first moment $\acute{m}_t$

$$\mathrm{m'}_{\mathrm{wight(t)}} = \frac{\mathrm{m}_{\mathrm{weight(t)}}}{1-\beta_1^t} \quad (14)$$

$$\mathrm{m'}_{\mathrm{bias(t)}} = \frac{\mathrm{m}_{\mathrm{bias(t)}}}{1-\beta_1^t} \quad (15)$$

Similarly, the calculation for the update weight and biased second-moment estimation

$$\mathrm{v}_{\mathrm{weight(t)}} = \beta_2 * \mathrm{v}_{\mathrm{weight(t-1)}} + (1-\beta_2) * (\delta_j^l)^2 \quad (16)$$

$$\mathrm{v}_{\mathrm{bias(t)}} = \beta_2 * \mathrm{v}_{\mathrm{bias(t-1)}} + (1-\beta_2) * (\delta_j^l)^2 \quad (17)$$

➢ The bias and weight-corrected estimation for the first moment $\acute{m}_t$

$$\mathrm{v'}_{\mathrm{wight(t)}} = \frac{\mathrm{v}_{\mathrm{weight(t)}}}{1-\beta_2^t} \quad (18)$$

$$\mathrm{v'}_{\mathrm{bias(t)}} = \frac{\mathrm{v}_{\mathrm{bias(t)}}}{1-\beta_2^t} \quad (19)$$

➢ Update the parameter for the weight and bias

$$\theta_{\mathrm{weight(t+1)}} = \theta_{\mathrm{weight(t)}} - \frac{\alpha}{\sqrt{\mathrm{v'}_{\mathrm{weight(t)}+\epsilon}}} * \mathrm{m'}_{\mathrm{weight(t)}} \quad (20)$$

$$\theta_{\mathrm{bias(t+1)}} = \theta_{\mathrm{bias(t)}} - \frac{\alpha}{\sqrt{\mathrm{v'}_{\mathrm{bias(t)}+\epsilon}}} * \mathrm{m'}_{\mathrm{bias(t)}} \quad (21)$$

α is the learning rate.

During each training iteration, these formulas are applied to each parameter. Based on the historical gradients, the Adam optimizer dynamically modifies the learning rates for every parameter, offering adaptive optimization and efficient convergence for an extensive variety of problems.

Thus, in SA models, the FCL is useful for capturing complicated patterns, learning relationships between features, and predicting the sentiment represented in a particular text. It contributes the NN to understand and interpret the intricacies of sentiment in natural language. The outcome of FCL serves as input to the output layer, which is commonly a softmax layer for sentiment analysis tasks. The softmax layer generates probabilities for each class, and highest probability class is considered to be the predicted sentiment.

3.5. Softmax

The softmax function is often used in the model's last layer to turn raw output scores into probability distributions across many classes. It is very effective for multi class classification problems, such as assigning one of several possible emotion labels to a given input. To convert raw output scores into probabilities, the softmax activation function is applied which is used to transform a vector of real numbers as input into a probability distribution. The softmax defined as follows for a vector z of length K (K is the number of classes).

$$\mathrm{softmax}(\mathrm{z})_{\mathrm{i}} = \frac{\mathrm{e}^{\mathrm{z_i}}}{\sum_{\mathrm{j=1}}^{\mathrm{K}} \mathrm{e}^{\mathrm{z_j}}} \quad (22)$$

Where e^{z_i} is the exponential function applied to the raw score z_i and the denominator is the sum of the exponentials for all of the vector's scores. This assures that the resulting numbers represent a proper probability distribution, with each value ranging from 0 to 1 and the sum equal to 1. The softmax function's output represents the probability of each class. Each entry in the output vector represents the probability that the input belongs to the corresponding class.

The softmax function will return a vector of three probabilities representing the chance of the input sentence falling into each sentiment category. The predicted sentiment for the input is the class with the highest probability. For example, if the softmax output is [0.1, 0.8, 0.1], the model outcomes that the input belongs to the second class (the one with the highest value representing a "positive" sentiment in this situation). Thus the softmax function is an important component in sentiment analysis utilizing the ALBERT model since it converts raw output scores into probabilities, allowing model predictions to be interpreted as class probabilities and allowing the model to learn and generalize from the training data.

4. Experiment and Results

This section explains the dataset and often describes model setup, methodology, and findings. Model formulation and evaluation methods and techniques are also discussed in this section.

4.1. Dataset

For our experimental study, we used the Amazon review dataset with 70,000 reviews. From the total number of samples, 50% for training and the remaining 50% for test samples. This dataset contains a few million Amazon customer reviews (input text) and star ratings (output labels) for SA. Amazon reviews provide useful information on the quality and performance of products. Potential purchasers frequently rely on these reviews to make informed purchasing decisions. The purpose of our approach to determine the polarity of the given review whether it is positive, negative, or neutral.

4.2. Model Architecture

Our study presents the ALBERT model with configuration in Table 1. The ALBERT (A Lite BERT) model architecture is intended to be an efficient and parameter-reduced variation of BERT model. ALBERT keeps BERT's core architecture, which is built on a transformer architecture, but adds various innovations to improve efficiency without sacrificing performance. The use of a factorized embedding parameterization, which divides the embedding size into two components and considerably reduces the overall number of parameters, is an important aspect. ALBERT also uses cross-layer parameter sharing, which allows parameters to be shared across layers, adding to parameter reduction and computational efficiency. During pre-training, ALBERT replaces the NSP loss with the SOP loss used in BERT. This modification is intended to improve the pre-training objectives.

4.3. Performance Metrics

The performance of a natural language processing (NLP) model must be evaluated to understand its effectiveness in handling certain tasks. The following are some common performance evaluation measures in NLP.

Accuracy is the proportion of correctly classified predictions to the total number of predictions.

$$\text{Accuracy} = \frac{\text{TP+TN}}{(\text{TP+FN+FP+TN})} \tag{23}$$

Precision is the ratio of correctly classified data to total data classified in the class.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP+FP})} \tag{24}$$

Recall is the ratio of true positives to the sum of true positives and false negatives, which is especially important in activities where minimizing false negatives is critical.

$$\text{Recall} = \frac{\text{TP}}{(\text{TP+FN})} \tag{25}$$

F1 score is the mean of precision and recall

$$\text{F1} = \frac{2\times\text{precision}\times\text{recall}}{(\text{precision+recall})} \tag{26}$$

4.4. Evaluation Result

For our experimental evaluation, the ALBERT model performs better than baseline models. This demonstrates the ALBERT model's greater capacity to capture subtle sentiment patterns across various textual materials. The experimental results included a thorough comparison of the ALBERT model's performance values with those of many well-known word embedding and language representation models, including GLOVE, LeGLOVE, Word2Vec, Leword2VEC, BERT, and LeBERT. The purpose of this comparison was to assess the relative efficacy of these models across various NLP tasks.

The comparison was carried out across various performance parameters, depending on the task requirements among others. The performance evaluation of SA using the ALBERT model using the Amazon dataset is shown in Figure 6 which was compared [22] and the comparison table is shown in Table 2. In tasks requiring a nuanced knowledge of context and semantics, the ALBERT model regularly beats the classic word embedding models (GLOVE, LeGLOVE, Word2Vec, and Leword2VEC).

Table 2. Performance Evaluation Result

Model	*Accuracy %*	*Precision %*	*Recall %*	*F-measure %*
Glove[22]	79	79	79.65	79.32
Le-Glove[22]	79.60	80	80.45	80.22
Word2Vec[22]	79.50	79.50	80.25	79.87
Le-word2Vec[22]	81.50	81.50	82.05	81.77
BERT[22]	81.72	81.75	82.04	81.89
LeBERT[22]	82.40	82.40	82.64	82.52
ALBERT	95.98	96.72	94.38	95.53

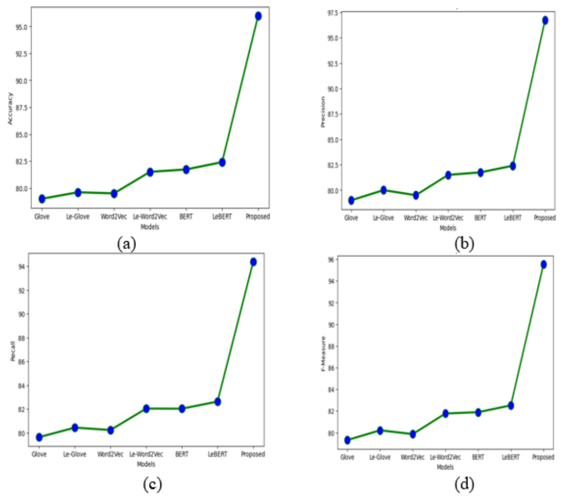

Fig. 6. Sentiment classification prediction using the Amazon dataset (a) Accuracy, (b) Precision (c) recall (d) F1

The experimental results offer light on the developing landscape of natural language processing models, emphasizing the importance of transformer structures such as ALBERT in presenting innovative outcomes. The transformer-based models BERT and LeBERT performed well, but ALBERT outperformed them in terms of efficiency and computing resource utilization. This comparison gives important insights into the strengths and limits of various language representation models, assisting researchers and practitioners in selecting models that are adapted to specific task difficulties and contextual needs. The overall performance graph of our proposed model is in Figure 7.

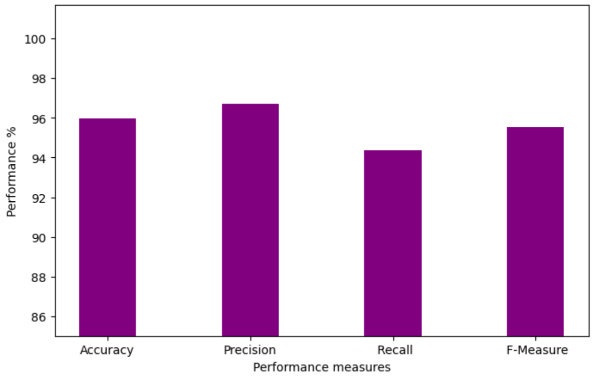

Fig. 7. Overall performance of SA using the ALBERT model

Thus the ALBERT model presented an impressive capacity to capture subtle subtleties in sentiment, demonstrating its ability to handle a wide range of verbal expressions. This model exhibited promising signs of generalization to real-world data, highlighting its potential application in sentiment analysis tasks across a wide range of areas. The model's robustness shows that it is suitable for use in real-world circumstances.

5. Conclusion

This study looks into the use of sentiment analysis and the ALBERT model to estimate brand value from customer reviews. Manufacturers can obtain vital insights into client preferences and emotions by integrating user-generated content and big data. Notably, the data demonstrate the ALBERT model's robustness, with an improved accuracy rate of 95.98%, as well as precision, recall, and F-measure values of 96.72%, 94.38%, and 95.53%, respectively. These performance indicators highlight the capability of advanced sentiment analysis techniques to provide meaningful insights for marketing strategies and decision-making processes. Future work should focus on improving the explainability and interpretability of the model's predictions which provide insights into the factors driving brand value judgments. It is critical to build trust in real-world applications by inventing strategies to make the model's decision-making process transparent.

Declarations

Ethics approval and consent to participate

This article does not contain any studies with human participants or animals performed by any of the authors

Conflict of interest

The authors declare that they have no conflict of interest.

Consent for publication

All contributors agreed and given consent to Publication.

Availability of data and material

Data that has been used is confidential

Competing interests

On behalf of all authors, the corresponding author states that they have no competing interest.

Funding

No fund was received for this work

Authors' contributions

The authors confirm contribution to the paper as follows and all authors reviewed the results and approved the final version of the manuscript.

First author (Corresponding author): Pallavi Suryanshi

Writing original draft, Methodology, study conception and design, analysis and interpretation of results, Reviewing and editing

Second author: Dr.Sharad Gangele

Conceptualization, data collection, Reviewing and editing

Acknowledgements

The authors would like to thank the RKDF University for supporting this work.

References

[1] X. Zhang, & J. S. L. Lam, "Measuring the Impact of E-Collaboration on Supply Chain Parties: A Value-Based Management Approach," *IEEE Access*, vol. 9, pp. 118181-118193, 2021.

[2] S. Ammirato, R. Linzalone, & A. M. Felicetti, "Business model innovation drivers as antecedents of performance," *Measuring business excellence*, vol. 26, no. 1, pp. 6-22, 2022.

[3] P. Sánchez-Núñez, M. J. Cobo, C. De las Heras-Pedrosa, J. I. Pelaez, & E. Herrera-Viedma, "Opinion mining, sentiment analysis, and emotion understanding in advertising: a bibliometric analysis," *IEEE Access*,vol. 8, pp. 134563-134576, 2020.

[4] C. S. Wu, A. Madotto, W. Liu, P. Fung, & C. Xiong, "Qaconv: Question answering on informative conversations," *arXiv preprint arXiv:2105.06912*, 2021.

[5] K. Kim, S. Yoon, & Y. K. Choi, "The effects of eWOM volume and valence on product sales–an empirical examination of the movie

industry," *International Journal of Advertising*, vol. 38, no. 3, pp. 471-488, 2019.

[6] M. Wankhade, A. C. S. Rao, & C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731-5780, 2022.

[7] S. Sachin, A. Tripathi, N. Mahajan, S. Aggarwal, & P. Nagrath, "Sentiment analysis using gated recurrent neural networks," *SN Computer Science*, vol.1, pp. 1-13, 2020.

[8] A. Amin, I. Hossain, A. Akther, & K. M. Alam, "Bengali vader: A sentiment analysis approach using modified vader," In *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pp. 1-6. IEEE, 2019, February.

[9] T. N. Prakash, & A. Aloysius, "A Comparative Study of Lexicon-based and Machine learning based classifications in Sentiment analysis," *International Journal of Data Mining Techniques and Applications*, vol. 8, no. 1, pp. 43-47, 2019.

[10] B. Liu, *Sentiment analysis and opinion mining*. Springer Nature, 2022.

[11] R. Catelli, S. Pelosi, & M. Esposito, "Lexicon-based vs. Bert-based sentiment analysis: A comparative study in Italian," *Electronics*, vol. 11, no. 3, pp. 374, 2022.

[12] R. S. Jagdale, V. S. Shirsat, & S. N. Deshmukh, "Sentiment analysis on product reviews using machine learning techniques," In *Cognitive Informatics and Soft Computing: Proceeding of CISC 2017*, pp. 639-647. Springer Singapore, 2019.

[13] M. Guia, R. R. Silva, & J. Bernardino, "Comparison of Naïve Bayes, Support Vector Machine, Decision Trees, and Random Forest on Sentiment Analysis," *KDIR*, vol. 1, pp. 525-531, 2019.

[14] A. Dadhich, & B. Thankachan, "Sentiment analysis of Amazon product reviews using a hybrid rule-based approach," In *Smart Systems: Innovations in Computing: Proceedings of SSIC 2021*, pp. 173-193. Springer Singapore, 2022.

[15] N. Shrestha, & F. Nasoz, "Deep learning sentiment analysis of Amazon.com reviews and ratings," *arXiv preprint arXiv:1904.04096*, 2019.

[16] C. Du, H. Sun, J. Wang, Q. Qi, & J. Liao, "Adversarial and domain-aware BERT for cross-domain sentiment analysis," In *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*, pp. 4019-4028, 2020, July.

[17] Z. Desai, K. Anklesaria, & H. Balasubramaniam, "Business Intelligence Visualization Using Deep Learning Based Sentiment Analysis on Amazon Review Data," In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1-7. IEEE, 2021, July.

[18] A. S. AlQahtani, "Product sentiment analysis for Amazon reviews," *International Journal of Computer Science & Information Technology (IJCSIT*) vol. 13, 2021.

[19] M. P. Geetha, & D. K. Renuka, (2021). "Improving the performance of aspect-based sentiment analysis using a fine-tuned Bert Base Uncased model," *International Journal of Intelligent Networks*, vol. 2, pp. 64-69.

[20] M. Mostafa, & A. AlSaeed, "Sentiment Analysis Based on Bert for Amazon Reviewer," *Journal of the ACS Advances in Computer Science*, vol. 13, no. 1, pp. 1-10, 2022.

[21] A. Iqbal, R. Amin, J. Iqbal, R. Alroobaea, A. Binmahfoudh, & M. Hussain, (2022). "Sentiment Analysis of Consumer Reviews Using Deep Learning," *Sustainability*, vol. 14, no. 17, pp. 10844.

[22] J. Mutinda, W. Mwangi, & G. Okeyo, "Sentiment analysis of text reviews using lexicon-enhanced Bert embedding (LeBERT) model with convolutional neural network," *Applied Sciences*, vol. 13, no. 3, pp. 1445, 2023.