

Sentiment Classification on Customer Reviews using Variants of Deep Learning Frameworks and Embedding Techniques

Ajeet Kumar¹, Bhawana Singh¹, Shambhu Shankar Bharti²

¹Assistant Professor, Bakhtiyarpur College of Engineering, Bakhtiyarpur, Bihar, India; azit.bce@gmail.com (A.K.);
bhawana.singh.25@gmail.com (B.S.)

²Assistant Professor, Lok Nayak Jai Prakash Institute of Technology, Chapra, Bihar, India; shambhu4u08@gmail.com
(S.S.B.)

* Correspondence author: Ajeet Kuma, email azit.bce@gmail.com,

Received date: 5 June 2024; Accepted date: 2 July 2024; Published online: 10 July 2024

Abstract: Sentiment analysis has become an essential task in natural language processing (NLP), particularly in understanding customer opinions and feedback. This paper presents a deep learning-based approach for sentiment analysis on review datasets, leveraging NLP techniques and word embeddings. We utilize Amazon review dataset to evaluate the performance of three deep learning models: Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and a Multichannel CNN model. The data preprocessing phase includes stop words removal, tokenization, stemming, and lemmatization, ensuring refined textual input. For word representation, we employ Word2Vec, GloVe, and FastText embeddings to capture contextual meaning effectively. Experimental results indicate that the Multichannel model with FastText embeddings achieves the highest accuracy, outperforming other models in sentiment classification. The findings highlight the significance of word embedding selection and model architecture in improving classification performance.

Keywords: Sentiment analysis; deep learning; NLP; word embedding; CNN; RNN; FastText; Amazon reviews

1. Introduction

In the digital era, online reviews have become a crucial factor influencing consumer decisions, shaping business strategies, and providing insights into public opinions [1]. Sentiment analysis, also known as opinion mining, is a branch of natural language processing (NLP) that focuses on determining the sentiment expressed in textual data. With the exponential growth of user-generated content on e-commerce platforms, movie review sites, and social media, there is a growing demand for automated sentiment analysis systems that can accurately classify opinions as positive, or negative [2].

Traditional sentiment analysis approaches relied on lexicon-based and classical machine learning methods such as Naïve Bayes, Support Vector Machines (SVM), and decision trees. While these techniques demonstrated reasonable performance, they often struggled with challenges like contextual understanding, sarcasm, and ambiguous language. Recent advancements in deep learning have significantly improved sentiment analysis by leveraging neural networks capable of learning complex patterns in text [3]. Models such as Recurrent Neural Networks (RNNs), Multi-Channel Convolutional Neural Network (CNN), and transformer-based architectures like Bidirectional Encoder Representations from Transformers have set new benchmarks in sentiment classification tasks [4]. Deep learning is a subfield of artificial intelligence (AI) that uses neural networks with multiple layers to learn patterns and



representations from data. In sentiment analysis, deep learning models process large volumes of textual data to classify sentiments as positive, or negative [5]. Unlike traditional machine learning models, which rely on handcrafted features, deep learning models automatically extract meaningful features from raw text, leading to improved accuracy and robustness. Sentiment analysis on product is an important application of deep learning. Companies use it to analyze customer feedback, improve products, and enhance user experience [6].

This paper focuses on applying deep learning and NLP techniques to analyze sentiment in product and movie reviews. The study aims to compare the effectiveness of different deep learning models, assess their performance using various evaluation metrics, and highlight the impact of advanced NLP techniques in sentiment classification. The datasets used in this study include Amazon product reviews, which provide diverse and real-world textual data for model training and evaluation. The overall framework of sentiment analysis is illustrated in Figure 1.

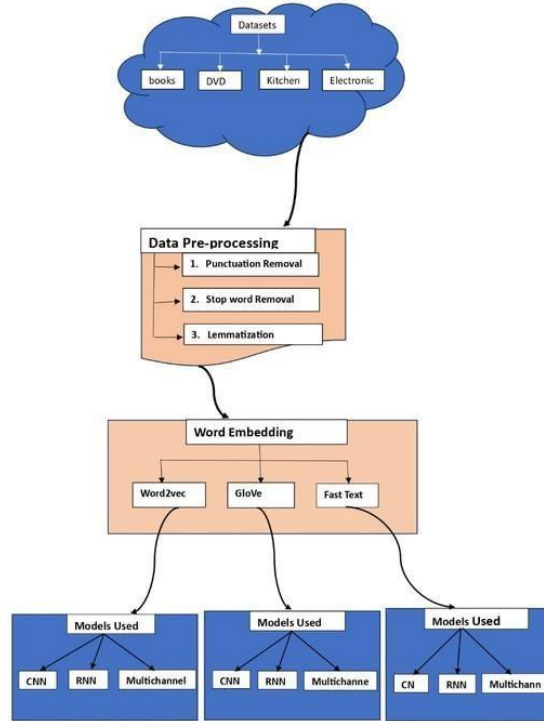


Figure 1. Framework of Sentiment Analysis.

By leveraging the power of deep learning and NLP, this research aims to contribute to the advancement of sentiment analysis methodologies, offering improved accuracy and adaptability to various real-world applications. The findings of this study can help enhance automated sentiment analysis systems, leading to more effective decision-making in businesses, social platforms, and other domains [7].

2. Related Works

Birjali et al. [8] used natural language processing (NLP), text analysis and computational techniques to automate the extraction or classification of sentiment from sentiment reviews. Analysis of these sentiments and opinions has spread across many fields such as consumer information, marketing, books, applications, websites, and social media. Sentiment analysis has become a hot area in decision making. With the accelerated evolution of the internet, websites, social networks, blogs, online portals, reviews, opinions, recommendations, ratings, and feedback are generated by writers. This writer-generated sentiment content can be about books, people, hotels, products, research, events, etc. [9].

3. Proposed Work

In this study, we propose a deep learning-based sentiment analysis model to classify reviews from Amazon using Natural Language Processing (NLP) techniques. The proposed methodology consists of data preprocessing, feature extraction, model selection, and evaluation to enhance sentiment classification accuracy. Initially, raw text data undergoes preprocessing, including removal of special characters, stop-words, and punctuation, followed by tokenization and lemmatization to standardize text representation. Since reviews may contain noise and informal language, handling imbalanced datasets through oversampling or under sampling techniques is also considered. For feature extraction, we leverage word embeddings such as Word2Vec, GloVe, and FastText. The classification task is performed using deep learning architectures, including Convolutional Neural Networks (CNNs) for local feature extraction, Recurrent Neural Networks (RNNs), and Multi-channel CNN. By leveraging deep learning and

NLP techniques, our approach aims to enhance sentiment analysis on Amazon reviews, capturing complex linguistic patterns and improving sentiment classification accuracy.

3.1 Data Collection

The dataset consists of user reviews collected from one of the major platforms: Amazon (<https://www.kaggle.com/datasets/bittlingmayer/amazonreviews>). Amazon reviews encompass a wide range of product categories, including electronics, books, and household items [10]. These reviews provide a rich and diverse textual data set for sentiment analysis, containing both positive and negative sentiments expressed in natural language.

The data is sourced from publicly available datasets, such as the Amazon Product Reviews dataset, which is widely used for sentiment classification tasks. The Amazon dataset includes millions of product reviews with associated star ratings (1-5), where ratings of 1-2 are considered negative, 3-5 are positive equally distributed between positive and negative sentiments, making it ideal for binary classification tasks.

To ensure data quality, we perform preprocessing steps, including duplicate removal, filtering out extremely short or irrelevant reviews, and normalizing text formats. The dataset is split into training, validation, and test sets, typically in an 80-20 ratio, ensuring sufficient data for model training and evaluation. Additionally, we address potential class imbalances using techniques like Train-Test-Split.

By collecting and processing real-world reviews from Amazon, our study ensures a diverse dataset that captures varying writing styles, opinions, and linguistic patterns, ultimately improving sentiment analysis performance.

3.2 Data Pre-processing

Data preprocessing is a crucial step in sentiment analysis as it helps clean and standardize textual data for effective model training. The raw reviews often contain noise, irrelevant words, and inconsistencies that must be addressed before analysis [11]. One of the primary steps is stop-word removal, where commonly used words such as —the,| —is,| and —and| are eliminated, as they do not contribute significant meaning to sentiment classification. However, words that impact sentiment, such as —not|, are carefully retained. Next, tokenization is performed to break down sentences into individual words or sub-words, making them easier to process. For deep learning models, advanced tokenization techniques like sub-word tokenization (e.g., Byte Pair Encoding or WordPiece) are used to handle out-of-vocabulary words effectively. To further refine the text, stemming is applied, which reduces words to their base form by stripping suffixes; for example, —running| becomes —run| and —happiness| becomes

—happi.| However, stemming may produce non-standard words, so lemmatization is used as an alternative to map words to their dictionary root forms while preserving meaning, such as converting —better| to —good|. These preprocessing techniques ensure that the input data is clean, structured, and semantically meaningful, ultimately improving the performance of deep learning models in sentiment classification.

The basic pre-processing steps are:

1. Removal of stop words
2. Removal of punctuations

3. Removal of digits
4. Removal of spaces
5. Removal of expressions
6. Removal of URLs
7. Standardizing words
8. Stemming
9. Lemmatization

3.3 Word Embedding

Word embedding is a technique used in Natural Language Processing (NLP) to represent words in a continuous vector space, capturing semantic relationships and contextual similarities. Unlike traditional methods such as one-hot encoding and TF-IDF, word embeddings provide dense and meaningful representations of words, improving performance in deep learning models for sentiment analysis [12]. Among the most widely used word embedding techniques are Word2Vec, GloVe, and FastText.

The choice between Word2Vec, GloVe, and FastText depends on the application. Word2Vec is effective for capturing syntactic and semantic relationships in large datasets, GloVe is beneficial for applications that require a

global understanding of word relationships, and FastText is ideal for handling rare words and morphologically rich languages. For sentiment analysis, FastText often provides better performance due to its ability to handle unseen words, while Word2Vec and GloVe remain strong choices for capturing context-based sentiment patterns [13].

In summary, word embeddings play a crucial role in improving sentiment analysis models by providing meaningful word representations. The selection of an embedding technique should be based on data set characteristics and the specific requirements of the deep learning model.

3.4 Deep Learning Process

Deep learning is a powerful set of algorithms that can automatically learn to understand complex data. Think of it as a way for a computer to learn in layers, building on what it knows to grasp bigger, more complicated ideas [14].

This ability makes deep learning a perfect match for Big Data. The algorithms can handle huge amounts of raw, unorganized information and learn from it without needing a human to label everything. This is a big advantage for Big Data analytics, where there's often too much data to sort through manually [15].

When you're working with this data, you first need to clean it up. Then, the deep learning model goes to work, pulling out key features or —aspects. For instance, in a sentence, it might identify certain words or phrases that reveal an opinion. It can then use this information to figure out the sentiment, or whether the sentence has a positive or negative tone. To do this, it often uses different models like n-grams to find important words and phrases. It might even use a convolutional neural network (CNN) to create the different layers it needs to learn from the data.

3.4.1 CNN

Convolutional Neural Network (CNN) is a clever and efficient algorithm designed to recognize patterns, especially in images. It's becoming a go-to tool for everything from text analysis to image recognition because it's straightforward, easy to adapt, and doesn't require a huge amount of data to be trained on.

One of the coolest things about a CNN is its **“weight-sharing” structure**, which makes it a lot like how a biological brain works. This smart design dramatically simplifies the model and reduces the number of parameters it needs to learn [16].

A CNN generally has two main parts:

3.4.1.1 First, there's the **feature extraction layer**. This part acts like a detective, zeroing in on small, local areas of an image to pick out specific details. It identifies local features like edges or textures and figures out how they relate to each other.

3.4.1.2 The second part is where it uses these extracted features to classify the image.

What makes CNNs so powerful is their ability to recognize an object even if it's moved around, zoomed in, or slightly distorted. This is because the network **implicitly learns** these features on its own from the training data, so you don't have to manually tell it what to look for. The CNN architecture is shown in Figure 2.

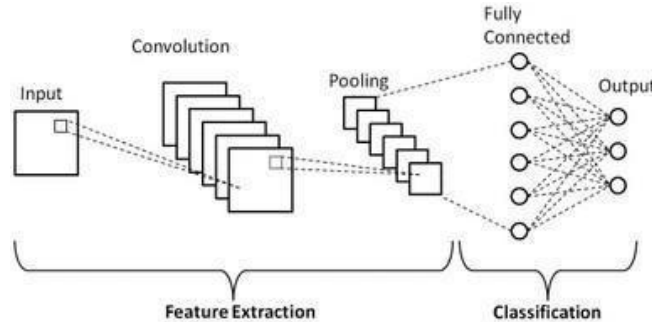


Figure 2. CNN Architecture.

3.4.2 RNN

The core idea of an RNN is its recurrent loop. Instead of a feed-forward structure where information only moves in one direction, an RNN feeds its own output from a previous time step back into itself as an input for the next time step. This loop is what gives the network its —memory. You give the RNN a piece of sequential data, like the first word of a sentence. The network processes this word and produces an output. In the —hidden state (memory), at the same time, it generates a hidden state, which is a vector that contains a summary of the

information it has processed so far. Think of this as the network's —memory or —context. In the recurrence state, when you give the network the next word in the sequence, it doesn't just use that new word as input. It also uses the hidden state (the —memory) it created from the previous step. This allows it to understand the new word in the context of the words that came before it. The iteration step process repeats for every word in the sequence. Each new hidden state is a culmination of all the information from the beginning of the sequence up to that point. The RNN architecture is shown in Figure 3.

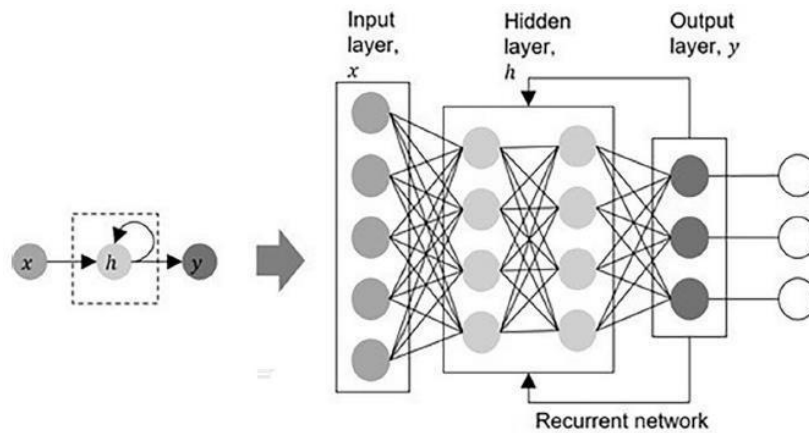


Figure 3. RNN Architecture.

3.4.3 Multi-Channel CNN

Multichannel models in deep learning are advanced architectures designed to process multiple representations of input data simultaneously. In the context of sentiment analysis, a multi-channel model utilizes multiple word embedding techniques (such as Word2Vec, GloVe, and FastText) to enhance the model's understanding of linguistic patterns. Unlike single-channel models, which rely on a single embedding for word representations, multichannel models incorporate multiple embeddings, allowing the network to learn richer semantic and syntactic features.

The architecture of a multichannel model typically consists of parallel convolutional or recurrent neural network layers, each processing a different embedding of the input text. These channels operate independently, extracting unique features from each representation before merging them in a later stage. This approach enables the model to capture diverse linguistic characteristics, improving the robustness and accuracy of sentiment classification. The final feature representations from all channels are combined, usually through concatenation or attention mechanisms, and passed through fully connected layers to generate sentiment predictions, as shown in Figure 4.

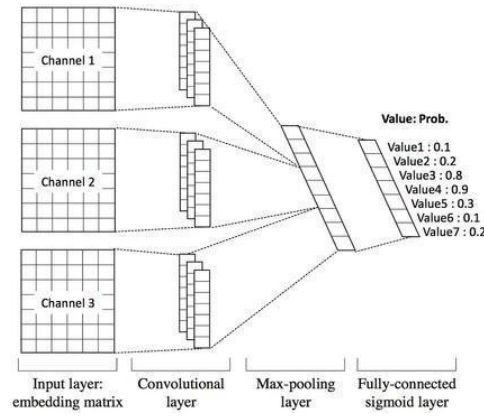


Figure 4. Multichannel CNN Architecture.

One of the key advantages of multichannel models is their ability to mitigate the limitations of individual embedding techniques. For example, Word2Vec captures contextual similarity, GloVe leverages global word co-occurrence statistics, and FastText considers sub-word information. By integrating these representations,

multichannel models enhance generalization, leading to improved performance, as evidenced in our experimental results, where the Multichannel model with FastText embeddings achieved 93.3% accuracy, outperforming single-channel CNN and RNN models.

3.5 Sentiment Analysis

Sentiment analysis is the process of figuring out the emotional tone of a piece of writing—whether it's positive, negative, or neutral. It's often called opinion mining because its main goal is to extract people's opinions or attitudes from text. For example, a company might use sentiment analysis to understand how customers feel about a new product by analyzing social media comments or reviews. To do this, the system processes the text and converts it into a feature vector, which is a way of representing the words as numbers. This allows a computer to classify the review as positive or negative, helping the company get a quick sense of public opinion.

4. Experimental Results

4.1 Dataset Description

For evaluating the performance of deep learning models on sentiment analysis, we used a widely recognized dataset:

- **Amazon Reviews Dataset:** This dataset contains customer reviews from various product categories, labeled as positive or negative based on user sentiment.

The datasets were preprocessed using techniques such as stop-words removal, tokenization, stemming, and lemmatization to ensure optimal text representation.

4.2 Model Training

We implemented three deep learning architectures: CNN, RNN, and a Multichannel model. The training parameters of these models are given in Table 1, and their performance was evaluated using standard classification metrics.

Table 1. Model training parameters.

Parameter	CNN	RNN	Multi-channel CNN
Batch_size	128	128	128
Embedding_dim	100	100	300
Filters	128	128	128
Kernel_size	3	3	5, 6, 7
Epoch	5	5	5
Dropout	0.5	0.5	0.5
Activation Function	Sigmoid	Sigmoid	Sigmoid

4.3 Performance Comparison

The models were trained using an 80-20 train-test split, with hyper-parameters such as learning rate, batch size, and number of epochs fine-tuned for optimal performance, as listed in Table 1. Table 2 presents the comparative results of the models on the dataset. The bold values indicate the best performance for each measure.

Table 2. Comparative performance analysis.

Model	Embedding	Accuracy	Precision	Recall	F1-Score
CNN	Word2Vec	88.5	87.9	87.2	87.5
	GloVe	89.2	88.6	87.9	88.2
	FastText	90.1	89.5	89.0	89.2
RNN	Word2Vec	86.3	85.7	85.0	85.3
	GloVe	87.8	87.1	86.5	86.8
	FastText	88.6	88.0	87.4	87.7
Multi-channel CNN	Word2Vec	91.5	91.0	90.3	90.6
	GloVe	92.1	91.7	91.0	91.3
	FastText	93.3	92.8	92.2	92.5

The confusion matrix and accuracy of the models are shown in Figures 5 and 6.

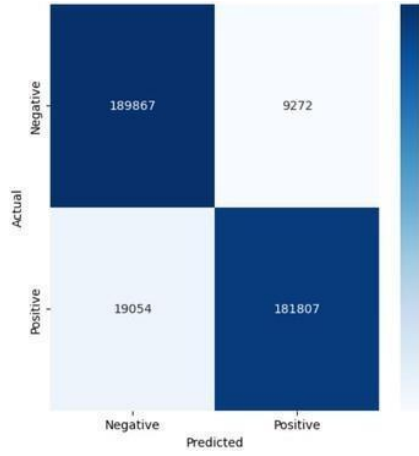


Figure 5. Confusion Matrix.



Figure 6. Accuracy.

4.4 Results and Discussion

The label distributions of the training and validation sets are shown in Figures 7 and 8. The model accuracy and loss curves of the Multi-channel CNN model are given in Figure 9, and the ROC curve for the Amazon reviews dataset is given in Figure 10.

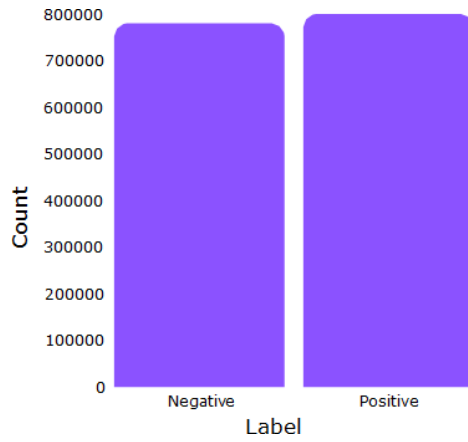


Figure 7. Training set label distribution.

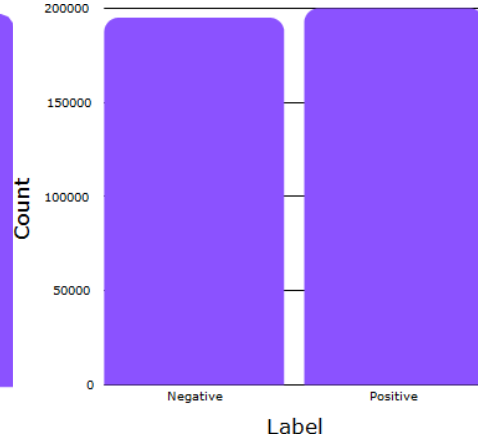


Figure 8. Validation set label distribution.

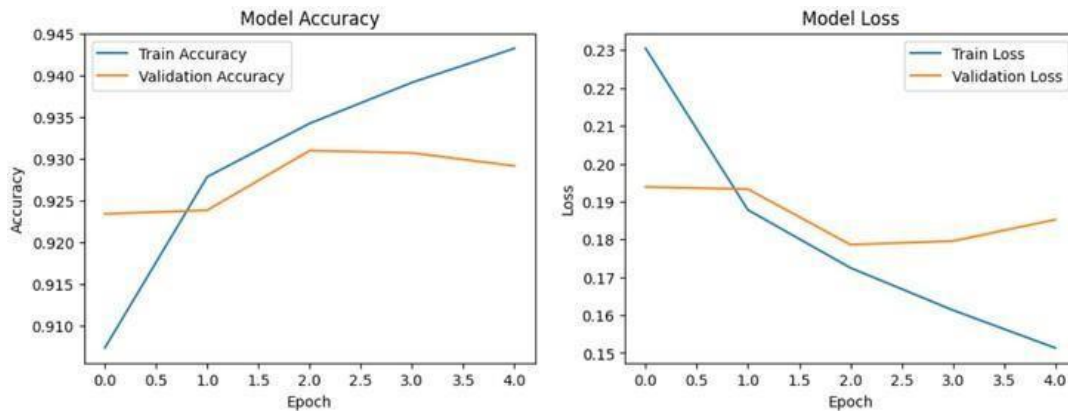


Figure 9. Model accuracy and model loss of Multi-channel CNN model for the dataset.

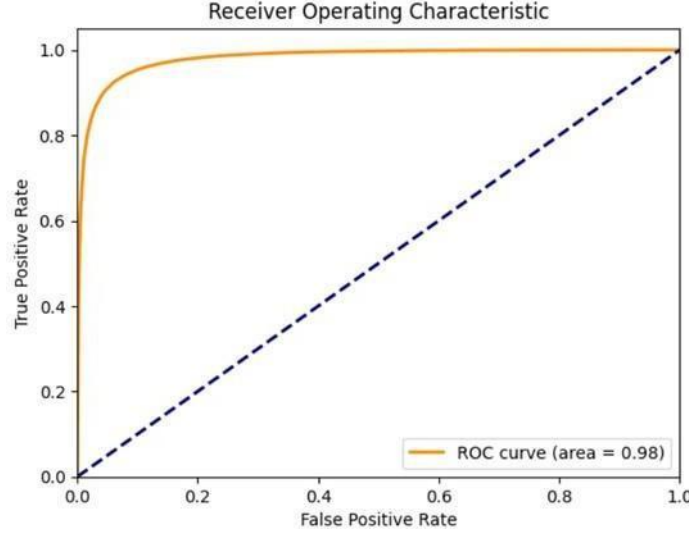


Figure 10. ROC for Amazon reviews.

The results indicate that the Multi-channel CNN model with FastText embedding outperformed all other configurations, achieving an accuracy of 93.3%. The CNN models also performed well, especially with FastText embeddings, suggesting that convolutional architectures effectively capture local dependencies in text data. RNN models performed slightly lower than CNN and Multichannel models, likely due to their sequential nature, which may not efficiently capture long-range dependencies in text. Additionally, FastText embeddings consistently outperformed Word2Vec and GloVe, highlighting their effectiveness in handling sub-word information, which is particularly useful for noisy or informal text data.

5. Conclusion

In this study, we explored sentiment analysis on reviews using deep learning and NLP techniques. Our approach involved data preprocessing techniques such as stop-words removal, tokenization, stemming, and lemmatization, followed by word embedding using Word2Vec, GloVe, and FastText. We employed three deep learning models—CNN, RNN, and a Multichannel model—to classify sentiments in reviews from Amazon datasets.

The experimental results demonstrated that deep learning models effectively capture sentiment in text, with the Multichannel model using FastText embedding achieving the highest accuracy of 93.3%. The findings highlight the importance of word embedding choices, as FastText consistently outperformed Word2Vec and GloVe, particularly in handling out-of-vocabulary words and sub-word information. Additionally, CNN-based models performed well due to their ability to capture spatial relationships in text, while RNN models showed slightly lower performance due to their sequential nature.

Despite the high performance of deep learning models, challenges such as sarcasm detection, negation handling, and domain-specific language interpretation remain open areas for improvement. Future work can explore hybrid models combining CNN and RNN architectures or transformer-based approaches such as BERT and GPT to enhance sentiment classification accuracy further. Additionally, incorporating attention mechanisms and adversarial training may help improve model robustness and generalization.

Overall, our study confirms that deep learning techniques, when combined with effective NLP preprocessing and embedding strategies, provide a powerful approach to sentiment analysis. These findings contribute to the advancement of automated sentiment classification and have practical applications in fields such as e-commerce, customer feedback analysis, and social media monitoring.

REFERENCES

1. O. Faruqe, M. Jahan, M. Faisal, M. S. Islam and R. Khan, —Bangla Hate Speech Detection System Using Transformer Based NLP and Deep Learning Techniques, *Asian Conference on Innovation in Technology*, 2023.
2. Liu, Y., & Wu, A. POI recommendation method using deep learning in location-based social networks. *Wireless Communications and Mobile Computing*, 2021, 9120864. <https://doi.org/10.1155/2021/9120864>
3. Jindal K, Aron R. A systematic study of sentiment analysis for social media data. *Mater Today: Proc*, 2021.
4. Kundu R, Singh S et al. Automatic image captioning using different variants of the long short-term memory (LSTM) deep learning model. In *Deep learning research applications for natural language processing*, IGI Global, 2023, pp. 1–24.
5. Joshi, Rohit, and Raj Kumar Tekchandani. —Comparative analysis of Twitter data using supervised classifiers. *International Conference on Inventive Computation Technologies (ICICT)*, Vol. 3, IEEE, 2016.
6. Dang NC, Moreno-Garcia MN, Prieta FDL. Sentiment analysis based on deep learning: a comparative study. *Electronics*, 9(3), 2020.
7. Bai Q, Dan Q, Mu Z, Yang M. A systematic review of emoji: current research and future perspectives. *Frontiers in Psychology*, 10, p. 2221, 2019.
8. Birjali M, Kasri M, Beni-Hssane A. A comprehensive survey on sentiment analysis: approaches, challenges and trends. *Knowledge-Based Systems*, 226, p. 107134, 2021.
9. S. Zahoor and R. Rohilla, —Twitter Sentiment Analysis Using Machine Learning Algorithms: A Case Study, *International Conference on Advances in Computing, Communication & Materials*, pp. 194-199, 2020.
10. Meng, J., Long, Y., Yu, Y., Zhao, D., & Liu, S. Cross-domain text sentiment analysis based on CNN_FT method. *Information*, 10(5), p. 162, 2019.
11. O. Sharif, M. M. Hoque, and E. Hossain, —Sentiment Analysis of Bangla Texts on Online Restaurant Reviews Using Multinomial Naïve Bayes, *International Conference on Advances in Science, Engineering and Robotics Technology*, pp. 1-6, 2019.
12. N. N. Prachi, M. Habibullah, M. E. H. Rafi, E. Alam, and R. Khan, —Detection of Fake News Using Machine Learning and Natural Language Processing Algorithms, *Journal of Advances in Information Technology*, 13, pp. 652-661, 2022.
13. L. Mandloi and R. Patel, —Twitter Sentiments Analysis Using Machine Learning Methods, *International Conference for Emerging Technology*, pp. 1-5, 2020.
14. B. Seong and K. Song, —Sentiment analysis of online responses in the performing arts with large language models, *Heliyon*, 18, 2023.
15. Dang N C, Moreno-García M N, De la Prieta F. Sentiment analysis based on deep learning: A comparative study. *Electronics*, 9(3), p. 483, 2020.
16. Rani S, Kumar P. Deep learning based sentiment analysis using convolution neural network. *Arabian Journal for Science and Engineering*, 44(4), pp. 3305–3314, 2019.