Proceedings of CONF-APMM 2025 Symposium: Controlling Robotic Manipulator Using PWM Signals with Microcontrollers
DOI: 10.54254/2753-8818/2025.AD26239

Sentiment Analysis of Twitter Comments Based on Deep Learning

Wanxin Ma

Division of Intelligence and Computing, Tianjin University, Tianjin, China
mwxmwx152@gmail.com

Abstract. As the web continuously evolves, the information quantities online grow progressively. Therefore, the automatic processing of massive online data has become a research hotspot. Text classification serves as an effective method for analyzing unstructured web data. This study focuses on employing deep learning techniques for sentiment classification of textual information in social networks, aiming to develop an automated sentiment analysis tool. The primary objective is to analyze and process massive volumes of unstructured online data while overcoming the limitations of traditional text analysis methods, such as statistical or machine learning methods which heavily depend on manually engineered features. This paper constructs a sentiment classification model based on a Bidirectional Long Short-Term Memory network (Bi-LSTM) integrated with an attention mechanism, which classifies user comments into positive or negative categories. Experimental results demonstrate that, compared with single-layer LSTM, Bi-LSTM, and two-layer Bi-LSTM models, the proposed model achieves more accurate text classification performance.

Keywords: Deep-learning, Bidirectional Long Short-Term Memory, Text classification

1. Introduction

In recent years, with the rapid advancement of internet technologies, the amount of data on the internet has increased significantly. By 2024, the global scale of internet usage has reached the level of billions of terabytes, containing vast amounts of unstructured data, such as videos, audio, images, and text. The analysis and application of online data have also become prominent areas of research. Confronted with massive volumes of unstructured and disordered data, traditional analytical methods suffer from significant challenges, including high complexity, extended processing time, and substantial labor costs. Consequently, the development of automated processing and detection tools has become an inevitable trend. Among such methods, text classification provides an effective approach for organizing and analyzing unstructured textual corpora. By assigning appropriate labels to disordered data, text classification facilitates easier use and development of downstream applications.

In the field of Natural Language Processing (NLP), text classification is a fundamental task, with broad applications across domains such as medicine, government policy implementation, news recommendation, and spam filtering. Text classification refers to the process of assigning texts to

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2753-8818/2025.AD26239

predefined categories or labels. Common tasks include sentiment analysis, news categorization, and topic modeling. In recent years, tasks such as question answering and other natural language understanding tasks have also been regarded as forms of text classification [1]. Traditionally, text classification relied heavily on statistical and machine learning methods, which in turn depended on manually crafted features. However, deep learning's advancement has prompted the development and broad adoption of multiple deep learning models in text classification. Deep neural networks, the foundation of deep learning, allow machines to automatically learn sample features. Unlike traditional methods, for example, machine learning, which heavily rely on user-provided domain knowledge, deep learning primarily depends on labeled datasets. Convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep belief networks (DBNs) are a few examples of common deep neural network types [2].

Within text classification, sentiment analysis stands as a pioneering and ubiquitous application. This task typically involves several key stages: data preprocessing, feature extraction, and classification [3]. From an analytical perspective, sentiment analysis can be roughly categorized into analysis on phrase-level, analysis on sentence-level, analysis on document-level, and analysis on aspect-level [4]. At present, several effective classification algorithms exist, such as Naive Bayes and Support Vector Machines (SVM) [5]. However, due to the massive volume of online textual data —often varying in length, complex in syntactic structure, and containing a high proportion of irrelevant information—traditional methods typically struggle to extract sufficient semantic content for accurate sentiment detection in the domain of online public opinion analysis [6]. As a result, the construction of deep learning-based automatic sentiment analysis tools has become a research hotspot both domestically and internationally. Conneau et al. proposed a model based on VDCNN using small convolutions and pooling operations [7]. Kowsari et al. employed a hierarchical deep learning model to classify large-scale textual data [8]. Yao et al. constructed a text classification model based on fastText [9], while Sun et al. optimized text classification tasks using large language models [10].

In this research, a sentiment analysis tool based on large-scale deep learning models is designed to automatically assign sentiment labels—either positive or negative—to short-form textual comments from social media. The experimental dataset is sourced from the social platform Twitter, comprising 41,158 entries. The experimental procedure is as follows: first, the heterogeneous and disordered corpus is cleaned through data preprocessing. Then, a Word2Vec model is employed to learn features from the cleaned text, generating word vectors. Subsequently, a Bidirectional Long Short-Term Memory (Bi-LSTM) network is used to train the sentiment analysis model.

2. Methodology

2.1. Data source and description

The dataset used in this research were collected from the social media platform Twitter, focusing on user comments posted in March 2020. A total of 33,444 comments were obtained from the public data platform Kaggle. The dataset centers on discussions related to COVID-19, with all texts written in English. Sentiment labels are categorized into two classes: positive and negative, with 18,046 entries labeled as positive and 15,398 entries labeled as negative. This study divides the dataset into training and testing subsets using an 8:2 split.

DOI: 10.54254/2753-8818/2025.AD26239

2.2. Data preprocessing

As the data derive from a social network, the textual corpus contains substantial noisiness and irregularities. To address this, this study undertook a series of data cleaning and tokenization procedures.

The cleaning process involved: Removing stop words while retaining negation terms; Eliminating punctuation marks, except for exclamation points and question marks; Stripping HTML tags and web URLs. Additionally, because social media comments frequently include usernames and account handles, words that appeared only once were treated as low-frequency terms and were removed from the corpus.

The distributions of word frequency before and after cleaning are shown in the tables below. Table 1 and Table 2 present the top five most frequent words before and after the cleaning process, respectively:

Table 1. Top 5 most frequent words before cleaning

Word	the	to	t	and	co
Count	34811	32904	23185	20460	18491

Table 2. Top 5 most frequent words after cleaning

Word	covid	coron-avirus	prices	food	super-market
Count	17602	14205	6676	6374	5661

2.3. Method overview

2.3.1. Word2vec

The Word2Vec model was proposed by Google in 2013 and is an unsupervised model designed to learn semantic representations from large-scale text corpora. Addressing the high-dimensional sparsity and lack of semantic information inherent in one-hot encoding, Word2Vec maps words into dense, low-dimensional vectors.

Word2Vec, as a classic word vector generation framework, includes two core architectures: the continuous bag-of-words model (CBOW) and Skip-Gram. The former predicts the target word depended on the surrounding words, and the latter infers the context distribution based on the given target word. They form a complementary semantic representation learning paradigm together. The resulting word embeddings are byproducts of the training process and serve as semantically meaningful vector representations.

2.3.2. Bi-LSTM model with integrated attention mechanism

The Long Short-Term Memory (LSTM) is an improved architecture of RNN that effectively alleviates the gradient vanishing and gradient explosion problems in sequential text processing through a unique gating mechanism. Compared with traditional RNN, this structure significantly enhances the ability to model long-distance semantic dependencies, enabling it to more accurately capture deep contextual associations in text sequences.

The structure is illustrated in Figure 1:

DOI: 10.54254/2753-8818/2025.AD26239

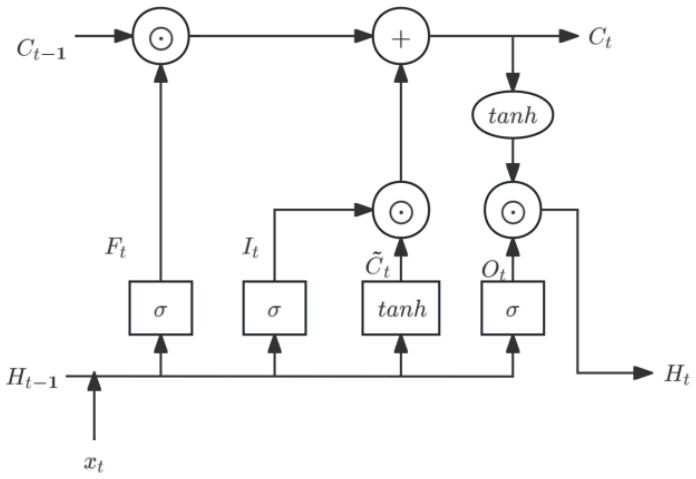


Figure 1. Structure of the LSTM neural network (picture credit: original)

In this structure, h_t represents the hidden state at time step t, while C_t represents the memory cell state at the corresponding time. Employing dynamic filtration of past memory data, the forget gate F_t selects what to eliminate from the prior cell state, enabling the system to vary its memory retention tactics using real-time input. The integration of new information is jointly controlled by the input gate I_t and the candidate memory $\widetilde{C}_t$, which jointly regulate the writing process of current features to the cell state. Finally, the output gate O_t determines the specific output value of the hidden state ht through a gated filtering mechanism, thereby completing the information transmission at the current time step.

Bi-LSTM model extends the standard LSTM by incorporating a reverse-direction neural network, allowing it to capture backward dependencies in the data. Model based on this bidirectional architecture understand contextual information in both forward and backward directions more comprehensively.

The attention mechanism helps the model focus on the key information within a sequence while ignoring less relevant elements, thus mitigating the limitations of hidden layer capacity when dealing with long sequences. It assigns a weight to each time step, and uses this weighted information to compute the output at the next time step.

2.3.3. Model architecture

In this study, word2vec is employed to convert unstructured textual data into word embeddings, generating an embedding matrix. The Bi-LSTM's capability in sequential modeling is then utilized to capture contextual semantics. Figure 2 depicts the complete architecture of the sentiment analysis model:

DOI: 10.54254/2753-8818/2025.AD26239

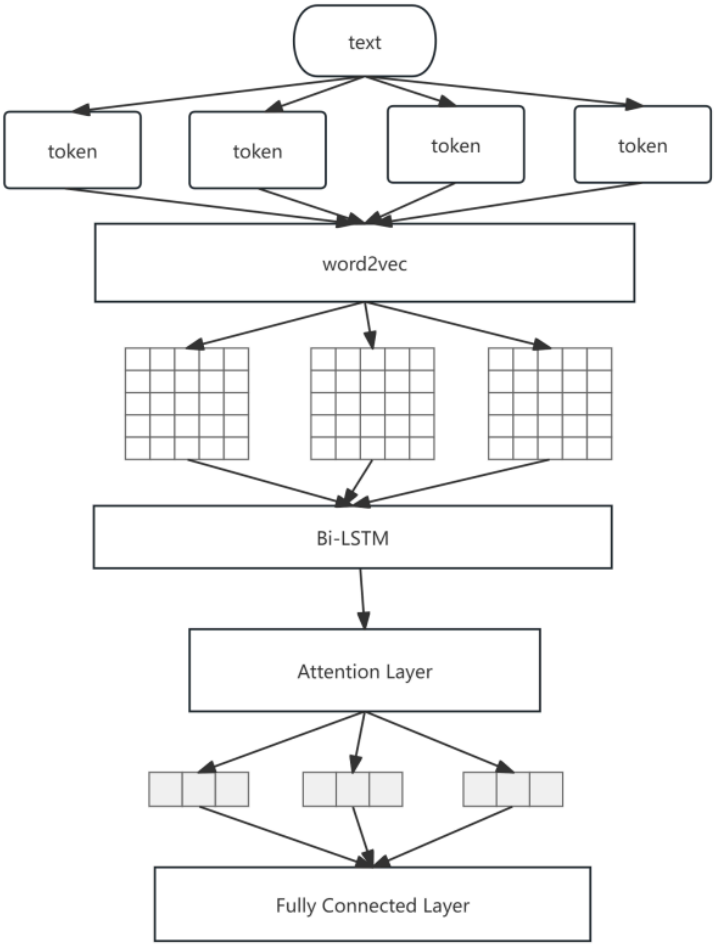


Figure 2. Architecture of the Bi-LSTM Model with Attention Mechanism (picture credit: original)

3. Results and discussion

3.1. Model parameter validation

This study constructed a four-dimensional evaluation system to comprehensively verify the model performance: accuracy was used to measure the overall classification accuracy, precision served as the metric for assessing positive class prediction dependability, recall was used to reflect the ability to identify positive class samples, and F1-score was used to comprehensively examine the balanced performance of precision and recall. This study verifies three critical model parameters: the hidden layer dimensionality of the Bi-LSTM model, learning rate, and batch size, using the f1 score as the evaluation metric.

In the Bi-LSTM model, selecting too few neurons may lead to underfitting, whereas an excessive number may cause overfitting. Additionally, training time for the model lengthens with increasing neurons. Figure 3 illustrates how the model's performance stays largely constant as the Bi-LSTM model's dimensionality increases. However, starting from 128 dimensions, performance begins to decline. Further increasing the hidden layer dimensionality not only reduces training efficiency but also results in a rapid performance drop. Therefore, this study selects 32 dimensions as the optimal size for the hidden layer.

DOI: 10.54254/2753-8818/2025.AD26239

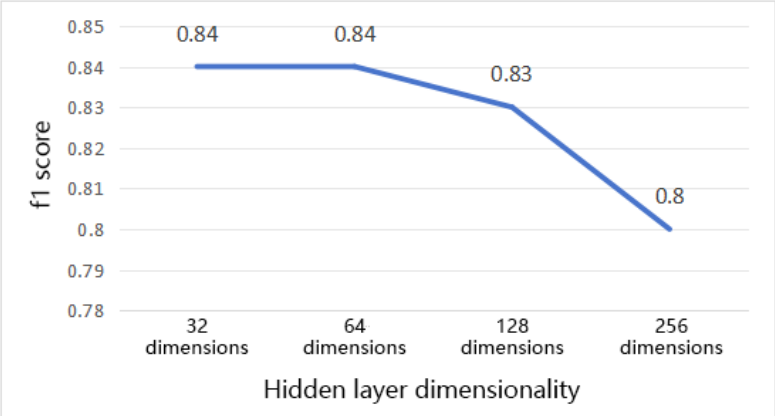


Figure 3. Model performance with different hidden layer dimensionalities (picture credit: original)

The learning rate is a crucial hyperparameter in deep learning. As the learning rate decreases, the model's adjustment rate slows down, which helps prevent oscillation or divergence of the loss function. The learning rates tested are 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001, 0.00005, and 0.00001. Figure 4 demonstrates enhanced model performance corresponding to lower learning rates. The optimal performance is appeared when the model's learning rate is set to 0.001, at which point the f1 score reaches 0.84. Beyond this point, model performance remains essentially unchanged. Thus, a learning rate of 0.001 is adopted in this model.

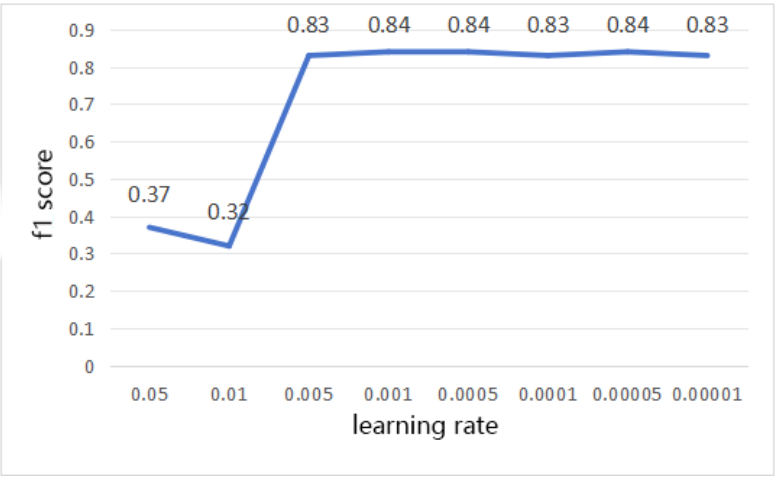


Figure 4. Model performance under different learning rates (picture credit: original)

The batch size means the amount of data samples processed in each training iteration. A larger batch size generally leads to faster training, while a smaller batch size tends to enhance the generalization ability of deep learning model. This study evaluates model performance under batch sizes of 16, 32, 64, 128, and 256. Illustrated in Figure 5, the x-axis denotes batch size, while the y-axis signifies the model's F1 metric. The experimental results show that when the batch size is 16, the model achieves peak performance. As the batch size continues to increase, the model's generalization ability weakens significantly, and the F1 score shows a clear downward trend, dropping to 0.80 when the size increases to 256. When the batch size reaches 256, the F1 score drops to 0.80. Therefore, this study adopts a batch size of 16.

DOI: 10.54254/2753-8818/2025.AD26239

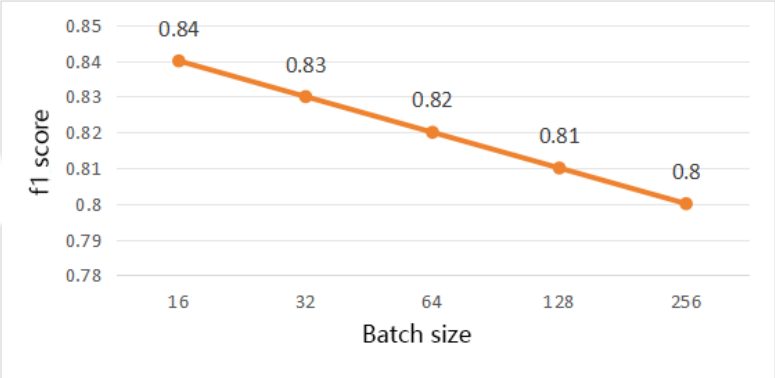


Figure 5. Model performance with different batch sizes (picture credit: original)

3.2. Comparative validation of LSTM models

The classification performance of the following four models is compared in this study:

Unidirectional LSTM model, Bi-LSTM model, Bi-LSTM model with an attention mechanism, Two-layer Bi-LSTM model with an attention mechanism.

Table 3 shows the performance of each model. As indicated by the results, the unidirectional LSTM model considers only historical information while ignoring future context, resulting in a lower F1 score and inferior overall performance. Therefore, the Bi-LSTM model is employed in this study, which yields a slight improvement in performance by incorporating both past and future information. The F1 score increases from the original 0.80 to 0.81. The results of experiment indicate that integrating the attention mechanism into the Bi-LSTM model improves the model to focus more effectively on highly relevant parts of the input, leading to a substantial improvement in performance, with the F1 score increasing by 0.3. In contrast, replacing the Bi-LSTM model with a two-layer Bi-LSTM does not yield significant performance gains; instead, it results in a slight decline. This may be attributed to the increased complexity of the neural network, which can cause overfitting when dealing with small sample sizes. Therefore, this study adopts the Bi-LSTM model with an integrated attention mechanism for the short-text classification task.

Table 3. Comparison of Bi-LSTM models

Model	accuracy	precision	recall	F1-score
LSTM	0.8035	0.8025	0.8021	0.8012
Bi-LSTM	0.8124	0.8130	0.8124	0.8116
Two-layer Bi-LSTM	0.8370	0.8396	0.8370	0.8373
Bi-LSTM-attention	0.8451	0.8467	0.8451	0.8453

4. Conclusions

To address the growing demand for sentiment analysis on massive social media data, this study constructs a sentiment analysis model employed LSTM networks. By integrating an attention mechanism, the model enhances the machine's capability to understand sentiment in text. Textual data are converted into dense word vectors using the word2vec algorithm, and contextual dependencies are captured via a Bi-LSTM model, ultimately enabling effective sentiment classification of short-text online comments. Through a comparative evaluation of a unidirectional

DOI: 10.54254/2753-8818/2025.AD26239

LSTM model, a Bi-LSTM model, a Bi-LSTM model with an attention mechanism, and a two-layer Bi-LSTM model, the proposed model demonstrates a superior balance of efficiency and performance, significantly outperforming the alternatives. Considering the limitations of this study —namely, a relatively small dataset and suboptimal word vector representations—Subsequent research might explore employing expanded datasets for overfitting mitigation, and incorporate Transformer architectures in conjunction with Bi-LSTM to further improve model performance.

References

[1] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning--based text classification: a comprehensive review. ACM computing surveys (CSUR), 54(3), 1-40.

[2] Zulqarnain, M., Ghazali, R., Hassim, Y. M. M., & Rehan, M. (2020). A comparative review on deep learning models for text classification. Indones. J. Electr. Eng. Comput. Sci, 19(1), 325-335.

[3] Tan, K. L., Lee, C. P., & Lim, K. M. (2023). A survey of sentiment analysis: Approaches, datasets, and future research. Applied Sciences, 13(7), 4550.

[4] 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.

[5] Lin, Y. (2022). Support vector machines and the Bayes rule in classification. Data Mining and Knowledge Discovery, 6(3), 259-275.

[6] Sun, X. (2024).Research on Classification Method of Online Sentiment Text Based on Deep Learning (Master's Thesis, Xi'an University of Technology).https: //link.cnki.net/doi/10.27398/d.cnki.gxalu.2024.001115doi: 10.27398/d.cnki.gxalu.2024.001115.

[7] Conneau, A., Schwenk, H., Barrault, L., & Lecun, Y. (2016). Very deep convolutional networks for text classification. arXiv preprint arXiv: 1606.01781.

[8] Kowsari, K., Brown, D. E., Heidarysafa, M., Meimandi, K. J., Gerber, M. S., & Barnes, L. E. (2017, December). Hdltex: Hierarchical deep learning for text classification. In 2017 16th IEEE international conference on machine learning and applications (ICMLA) (pp. 364-371). IEEE.

[9] Yao, T., Zhai, Z., & Gao, B. (2020, March). Text classification model based on fasttext. In 2020 IEEE International conference on artificial intelligence and information systems (ICAIIS) (pp. 154-157). IEEE.

[10] Sun, X., Li, X., Li, J., Wu, F., Guo, S., Zhang, T., & Wang, G. (2023). Text classification via large language models. arXiv preprint arXiv: 2305.08377.