



Research on the Design of News Text Classification Algorithm Based on Bidirectional GRU Neural Network

Xiaonan Gu

School of Computer Science and Technology, Zhejiang Normal University, Jinhua, China
Email: 632848541@qq.com

How to cite this paper: Gu, X.N. (2025) Research on the Design of News Text Classification Algorithm Based on Bidirectional GRU Neural Network. *Open Access Library Journal*, 12: e13226.
<https://doi.org/10.4236/oalib.1113226>

Received: March 7, 2025

Accepted: April 14, 2025

Published: April 17, 2025

Copyright © 2025 by author(s) and Open Access Library Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This study addresses the growing demand for news text classification driven by the rapid expansion of internet information by proposing a classification algorithm based on a Bidirectional Gated Recurrent Unit (BiGRU) neural network to enhance classification accuracy and efficiency. Traditional text classification methods often suffer from inefficiency and insufficient accuracy when handling large-scale news data, whereas the application of deep learning techniques provides a novel approach to improving text classification. The proposed algorithm first preprocesses news texts through tokenization, stop-word removal, and low-frequency word filtering to optimize text representation. Subsequently, the BiGRU model is employed for feature extraction and classification. Experimental results demonstrate that the model achieves an accuracy of over 90% across 11 news categories, with an average classification time per news item of less than five seconds, indicating strong classification performance and computational efficiency. This study offers an effective solution for automated news classification on news platforms and holds significant potential for broader applications.

Subject Areas

Complex Network Models

Keywords

Neural Network, News Text Classification, Algorithm Design

1. Introduction

Text classification is an important task in the field of natural language processing (NLP), which aims to classify a given text into predefined categories [1]. With the rapid development of the Internet, news, social media, e-commerce and other

platforms generate a large amount of text data every day. How to efficiently and accurately classify these texts has become an urgent problem to be solved. Text classification has a wide range of application scenarios, such as spam filtering, sentiment analysis, news classification, topic detection, etc. [2]. Through automated text classification technology, users can quickly obtain the required information and improve the efficiency of information retrieval. At the same time, it also provides enterprises with tools for data analysis and decision support. In recent years, with the rapid development of information technology, the news industry has also ushered in unprecedented changes. The Internet has become the main channel for people to obtain news information, and major news portals publish a large amount of news content every day. However, with the continuous growth of news data, how to effectively classify and manage this massive data has become a huge challenge. Traditional text classification methods mainly rely on artificial rules or simple machine learning algorithms. These methods often show problems such as low efficiency and low accuracy when processing large-scale and diverse news data. Especially when faced with complex language structures, polysemous words, contextual dependencies and other problems, the limitations of traditional methods are more obvious. Therefore, how to use advanced natural language processing technology and deep learning models [3] to improve the accuracy and efficiency of news text classification has become a hot topic in current research. In recent years, deep learning technology has made significant progress in the field of text classification, especially models such as recurrent neural networks (RNN) [4], long short-term memory networks (LSTM) [5], and gated recurrent units (GRU) [6], which can effectively capture the temporal information and contextual dependencies in text, significantly improving the performance of text classification.

2. Research Objectives

This study aims to design and implement an efficient news text classification system based on deep learning to enhance classification accuracy and computational efficiency. The system processes news headlines and content as input and automatically categorizes them into predefined categories, including finance, real estate, education, technology, military, automobiles, sports, gaming, and entertainment. By developing this system, news platforms can significantly improve their ability to manage large-scale news data, optimize classification performance, reduce manual intervention, and lower operational costs. Moving forward, continuous optimization of classification algorithms will further enhance classification precision and execution efficiency, promoting the broader application of text classification technology in real-world scenarios.

3. Research Content

3.1. Requirements Analysis

The requirement analysis phase is crucial in the development of a news text clas-

sification system, as it determines the system's functionality and performance criteria, directly influencing subsequent design and implementation. This phase involves an in-depth understanding of the target users' needs and expectations while defining the specific application scenarios and technical requirements.

First, this system is primarily designed for news platforms, aiming to enhance the automation of news text classification, thereby optimizing news content management and retrieval efficiency. The system can be widely applied in scenarios such as news recommendation, topic clustering, and sentiment analysis, leveraging intelligent classification techniques to reduce manual intervention and improve data processing efficiency.

Second, the system's functional requirements must be well-defined. The news text classification system should encompass end-to-end automation capabilities, including text preprocessing, feature extraction, deep learning model training, and classification prediction. Additionally, a user-friendly interface and interactive visualization tools should be provided to facilitate news classification and result interpretation.

Furthermore, performance requirements must be carefully considered. Given the large-scale and ever-growing volume of news data, the system should be capable of efficient data processing while maintaining high classification accuracy. Moreover, the system should demonstrate strong stability and scalability to accommodate future increases in data volume and evolving classification demands.

3.2. Model Establishment

This study presents a detailed implementation plan for the news text classification system. In the text preprocessing phase, standardized and structured data preparation is crucial for improving the model's training quality and classification accuracy. This stage includes tokenization, stop-word removal, low-frequency word filtering, text encoding, and sentence length normalization, ensuring the extraction of high-quality text features for classification.

In the model development phase, a Bidirectional Gated Recurrent Unit neural network is utilized for news text classification. The architecture consists of an input layer, embedding layer, BiGRU layer, dropout layer, and output layer. The functions of each layer and their interconnections will be elaborated. Proper hyperparameter selection ensures that the BiGRU model effectively captures the features of news texts, improving classification accuracy and generalization capability.

During model training, a structured training pipeline is designed, incorporating suitable loss functions and optimization algorithms. The model's performance is thoroughly analyzed and evaluated through iterative optimization to ensure robust classification performance and computational efficiency in real-world applications.

3.3. Key Technologies

This study integrates various technologies to support the development and imple-

mentation of the news text classification system. The key technologies utilized are as follows:

PyQt5: PyQt5 is a Python graphical user interface (GUI) building library based on the Qt framework. The library provides a rich set of UI components, allowing developers to efficiently build desktop applications with cross-platform features. In this research project, PyQt5 was used to build the user interface of the news text classification system to realize the functions of news text input and classification result display.

Flask framework: Flask is a lightweight Python web framework that is particularly suitable for developing small web applications. Its design is based on Werkzeug and Jinja2 and is known for its simplicity and flexibility [7]. In this research project, Flask is used to build a Web service to support users to access the system through a browser and perform news text classification tasks.

Multi-class classification and Softmax function: News text classification belongs to the category of multi-class classification problems, whose goal is to assign text data to multiple predetermined categories. The Softmax function is a commonly used multi-class classification activation function [8] that can convert the model output into a probability distribution of each category, ensuring that the sum of the probabilities of all categories is 1.

Word Embedding: Word embedding technology involves converting words in a text into low-dimensional vectors to capture the semantic associations between words. Common word embedding models include Word2Vec and GloVe. In this research project, word embedding technology is used to convert words in news text into vector form, which facilitates subsequent model training and classification work.

Pickle persistence: Pickle is a module in Python for object serialization and deserialization. It can store complex data structures (such as models, dictionaries, etc.) persistently to files and reload them when needed. In this research project, Pickle is used to save trained models and vocabularies to achieve fast loading and application during system runtime.

Keras: Keras is an advanced deep learning framework that supports backends such as TensorFlow, Theano, or CNTK. Keras provides a concise API that facilitates the rapid construction and training of deep learning models [9]. In this research project, Keras is used to build and train a bidirectional GRU neural network model.

3.4. Bidirectional GRU Neural Network Model

Recurrent Neural Network (RNN) is a fundamental class of deep learning models widely applied in sequential data processing tasks, such as machine translation and speech recognition. RNNs leverage recursive connections to process current input information along with previously computed hidden states, enabling the modeling of temporal dependencies in sequential data.

However, traditional RNNs suffer from the long-term dependency problem,

where they struggle to capture distant dependencies in long input sequences. This issue arises due to vanishing and exploding gradients, which hinder the model's ability to retain information over extended sequences. To address this limitation, researchers have introduced GRU and Long Short-Term Memory (LSTM) architectures, incorporating gating mechanisms to regulate information flow, thereby enhancing the model's ability to learn long-range dependencies.

BiGRU is an advanced variant that integrates bidirectional RNN and GRU structures to further improve sequence modeling [10]. By propagating information in both forward and backward directions, BiGRU captures richer contextual dependencies compared to standard unidirectional models. This bidirectional structure is particularly effective in natural language processing (NLP) tasks, as it allows the model to utilize both past and future information when making predictions. **Figure 1** shows a BiGRU neural network structure unfolded along the time axis.

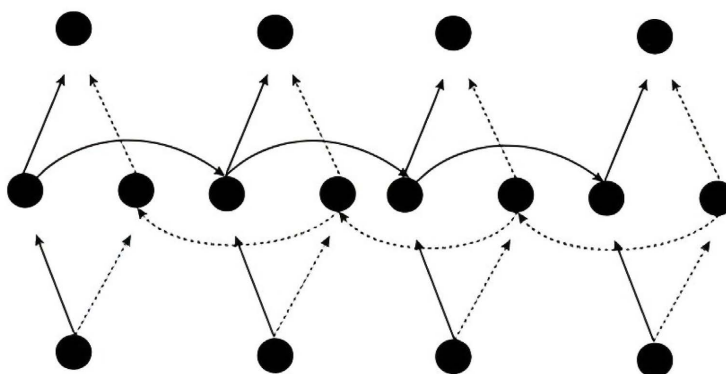


Figure 1. Bidirectional GRU neural network structure diagram.

The mathematical formulation of BiGRU is as follows:

Forward GRU computation:

$$\vec{h}_t = GRU(x_t, \vec{h}_{t-1}) \quad (3.1)$$

Backward GRU computation:

$$\overleftarrow{h}_t = GRU(x_t, \overleftarrow{h}_{t+1}) \quad (3.2)$$

The calculation formula for the final output of the BiGRU neural network is:

$$h_t = [\vec{h}_t; \overleftarrow{h}_t] \quad (3.3)$$

where \vec{h}_t and \overleftarrow{h}_t represent the hidden states of the forward and backward GRUs.

3.5. Model Description

The BiGRU neural network model structure can be divided into the following five levels according to its function:

- 1) Input Layer

This layer inputs text data into the template. Under our model parameter setting, the length of the text is 200 as the input of the model.

2) Embedding Layer

The data first enters the embedding layer, which converts the original input data into a vector. Under our model parameter settings, each word is converted into a 100-dimensional vector with a total dimension of 200×100 .

3) BiGRU Layer

The bidirectional GRU layer includes the forward GRU layer and the backward GRU layer. In our model, this layer is a 1×28 -dimensional bidirectional GRU layer.

4) Dropout Layer

The Dropout layer prevents complex co-adjustment of the training data, *i.e.*, prevents overfitting, by making other hidden units unreliable.

5) Output layer (Dense Layer)

The dense output layer performs dimensionality conversion and converts the features of different spaces into a new feature space through the dense layer. In our model, this layer will output the probability that the original data set belongs to 11 text categories (See **Table 1**).

Table 1. Five-layer structure of the BiGRU neural network model.

Layer (type)	Output Shape	Param
input_2 (InputLayer)	(None, 200)	0
embedding_1 (Embedding)	(None, 200, 100)	1,000,100
bidirectional_1 (Bidirection)	(None, 128)	63,744
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 11)	1419

3.6. Model Training

Model training is the core link in the news text classification system, which directly determines the performance and effect of the classification model. In this project, we used a BiGRU neural network as the classification model and trained it with a large amount of news text data to improve the classification accuracy and generalization ability of the model. This section will introduce in detail the model training process, key parameter settings during training, loss function selection, and optimization strategies during training.

3.7. Training Process

The model training process mainly includes the following steps:

Data preparation: Before model training, the preprocessed text data needs to be divided into training set, validation set and test set. The training set is used for model training, the validation set is used to adjust the model's hyperparameters and prevent overfitting, and the test set is used to evaluate the model's performance.

Model initialization: Before training begins, the parameters of the bidirectional GRU model need to be initialized. Usually, the weight matrix and bias terms can be set by random initialization or pre-training. In this project, we used the random initialization method.

Forward propagation: In each round of training, the model will perform forward propagation based on the input news text data to calculate the predicted probability of each category. The forward propagation process includes the calculation of the input layer, embedding layer, bidirectional GRU layer, Dropout layer, and output layer.

Loss calculation: By comparing the model's prediction results with the true labels, the value of the loss function is calculated. The loss function is used to measure the prediction error of the model. Commonly used loss functions include cross-entropy loss [11] and mean squared error. In this project, we used the cross-entropy loss function because it is particularly suitable for classification tasks.

Back propagation and parameter update: After calculating the value of the loss function, the model calculates the gradient of each parameter through the back propagation algorithm and uses an optimization algorithm (such as stochastic gradient descent SGD [12], Adam [13], etc.) to update the model parameters. Through continuous iteration, the prediction error of the model gradually decreases and the classification performance gradually improves.

Validation and early stopping: After each round of training, the model will be validated on the validation set to calculate the loss and accuracy of the validation set. If the loss of the validation set no longer decreases or the accuracy no longer increases, the early stopping strategy can be used to terminate the training early to prevent the model from overfitting.

3.8. Loss Function and Optimizer

In this project, we use the cross-entropy loss function, which is particularly suitable for multi-classification tasks. The cross-entropy loss function calculates the model's prediction error by comparing the model's predicted probability distribution with the distribution of the true label. The formula of the cross-entropy loss function is as follows:

$$\text{Loss} = -\sum_{i=1}^n y_i \log(\hat{y}_i) \quad (3.4)$$

Among them, y_i is the true label and \hat{y}_i is the predicted probability of the model.

Moreover, we use the Adam optimizer to update the model parameters. Adam is an adaptive learning rate optimization algorithm that combines the advantages of momentum method and RMSProp [14]. It can automatically adjust the learning rate during training and improve the convergence speed and stability of the model.

3.9. Training Results

During the training process, we recorded the loss function values and classifica-

tion accuracy of the training set and validation set of each epoch. By observing the iterative curve of the loss function (as shown in **Figure 2**), we can judge the training effect of the model. When the loss of the validation set no longer decreases, it means that the model has converged and training can be stopped. Through the above training process and optimization strategy, we have successfully trained an efficient and accurate bidirectional GRU neural network model that can accurately classify news texts. In the subsequent model evaluation, we will further verify the performance of the model.

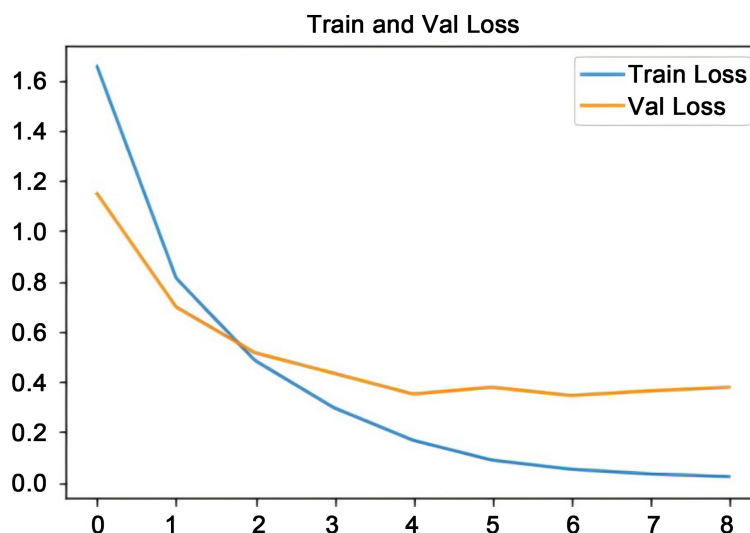


Figure 2. Loss function iteration graph.

4. Model Evaluation

4.1. Experimental Evaluation Indicators

In order to comprehensively evaluate the classification performance of the model, we use the following four commonly used evaluation indicators: Accuracy, Precision, Recall and F1-Score. These indicators can reflect the classification effect of the model from different angles. TP, TN, FP, and FN represent True Positive, True Negative, False Positive and False Negative, respectively.

Accuracy indicates the percentage of correct predictions in the total samples. Although accuracy can determine the overall accuracy, it cannot be used as a good indicator to measure results when the samples are unbalanced. The formula is as follows:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

Precision P (Precision) is a measure of the precision of a category, the probability of actually being a positive sample among all samples predicted to be positive. Precision represents the accuracy of the prediction of the positive sample results, while accuracy represents the overall prediction accuracy, including both positive and negative samples. The formula is as follows:

$$P = \frac{TP}{TP + FP} \quad (4.2)$$

Recall rate R (Recall) is a measure of the recall rate of a category, which is the probability of actually positive samples being predicted as positive samples. The higher the recall rate, the higher the probability that the actual bad user is predicted. Its meaning is similar to: it is better to kill a thousand by mistake than to let one go. The formula is as follows:

$$R = \frac{TP}{TP + FN} \quad (4.3)$$

The F1 value is a comprehensive measure of recall and precision, and is an indicator used in statistics to measure the accuracy of a binary classification model. The F1 score can be seen as a weighted average of the model's accuracy and recall. Its maximum value is 1 and its minimum value is 0. The larger the value, the better the model. The formula is as follows:

$$F_1 = 2 \times \frac{R \times P}{R + P} \quad (4.4)$$

4.2. Experimental Results and Analysis

Through experiments, we obtained the classification performance indicators of the model on the training set and test set, and analyzed the results in detail.

1) Dataset processing

In the data preprocessing stage, we divided the 14,632 news data into training sets and test sets, and ensured the correspondence between category labels and text content. The specific data distribution is as follows:

Number of training set samples: 13,168.

Number of test set samples: 1464.

2) Preprocessing of category labels

We assigned each category a unique numeric code as follows:

#label: 11 {0: "Sports", 1: "Sports Focus", 2: "Military", 3: "Entertainment", 4: "Real Estate", 5: "Education", 6: "Car", 7: "Game", 8: "Technology", 9: "Comprehensive Sports Latest", 10: "Finance"}.

We used Jieba word segmentation tool to segment the news text and counted the length of the text. The results showed that the longest text length was 2010 words, the shortest was 1 word, and the average length was 238 words.

3) Build a dictionary and delete low-frequency words

During the text preprocessing phase, a comprehensive vocabulary was constructed, containing 81,718 unique words. The frequency distribution of these words is illustrated in **Figure 3**.

In practical applications, textual data often include a substantial number of low-frequency or noisy words. These words appear infrequently in the dataset and contribute minimally to feature representation. Moreover, they may introduce unnecessary noise, potentially reducing the stability and effectiveness of the classification model. To address this issue, a frequency-based filtering strategy was

implemented, where words with low occurrence frequencies were removed, and only the most informative words were retained. Specifically, we preserved the top 10,000 most frequently occurring words, ensuring that the retained vocabulary captures the most salient linguistic features of the dataset.

This process effectively reduces redundant information, enhances computational efficiency, and optimizes the feature space for the classification model. Additionally, by eliminating low-frequency words, we mitigate data sparsity issues, improving the model's ability to distinguish between different text categories. Consequently, this refinement contributes to better classification accuracy and enhances the model's generalization capability in real-world applications.

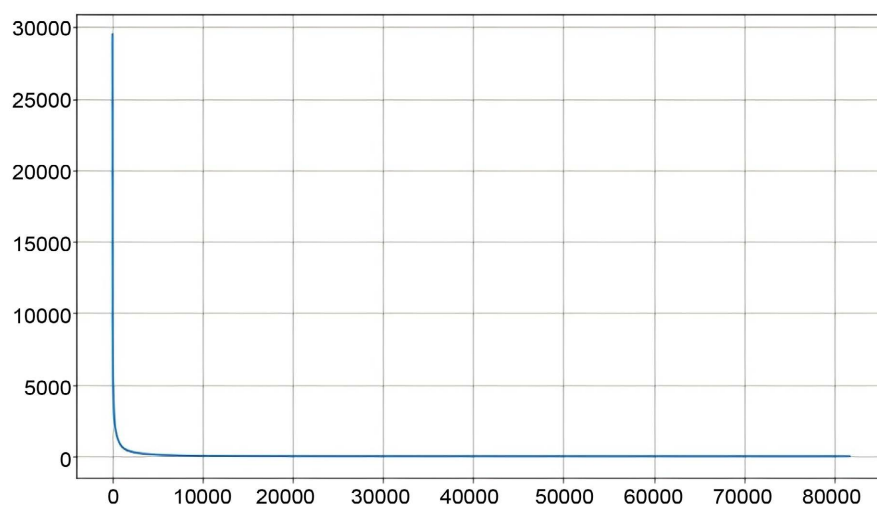


Figure 3. Word frequency distribution.

4) Training of Bidirectional GRU Model

At the end of each epoch, the accuracy of the validation set is calculated. When the accuracy no longer improves, the training is stopped. The general practice is to record the best validation set accuracy so far during the training process. When the best accuracy is not reached for n consecutive epochs, it can be considered that the accuracy is no longer improving, and the iteration can be stopped at this time.

During the training process, we recorded the loss function values and classification accuracy of the training set and validation set for each epoch. By observing the iterative curve of the loss function, we found that the loss of the model on the training set gradually decreased, while the loss on the validation set tended to stabilize after reaching a certain value, indicating that the model has converged.

5) Model training effect

Performance on the training set and test set:

The classification accuracy of the model on the training set reached 99%, and the precision, recall, and F1 value were all close to 1, indicating that the model had a very good fitting effect on the training set. The classification accuracy of the model on the test set is 91%, and the precision, recall and F1 value are 0.90, 0.91 and 0.90 respectively, indicating that the model also has good generalization abil-

ity on the test set.

To evaluate the model's classification performance across different categories, we analyze the provided confusion matrices and prediction results for both the training and test sets.

The confusion matrix for the training set (Figure 4) reveals that the model performs well overall, with high values along the diagonal, indicating accurate classifications for most categories. However, some off-diagonal values suggest that the model faces challenges in distinguishing certain categories. This observation is further supported by the prediction results for the training set (Figure 5), where the predicted values align closely with the true values, but minor fluctuations are still present. These discrepancies indicate that while the model has learned the primary patterns, further adjustments are needed to reduce errors and improve consistency in classifying specific categories.

$$\begin{bmatrix} 165 & 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 357 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 139 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 690 & 0 & 0 & 0 & 1 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 176 & 0 & 0 & 0 & 1 & 0 & 5 \\ 0 & 0 & 0 & 3 & 0 & 447 & 0 & 0 & 1 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 1 & 570 & 0 & 1 & 0 & 13 \\ 0 & 0 & 0 & 2 & 0 & 0 & 1 & 1165 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 700 & 0 & 35 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 898 & 0 \\ 0 & 0 & 0 & 2 & 2 & 2 & 15 & 0 & 31 & 0 & 7713 \end{bmatrix}$$

Figure 4. Confusion matrix of the training set.

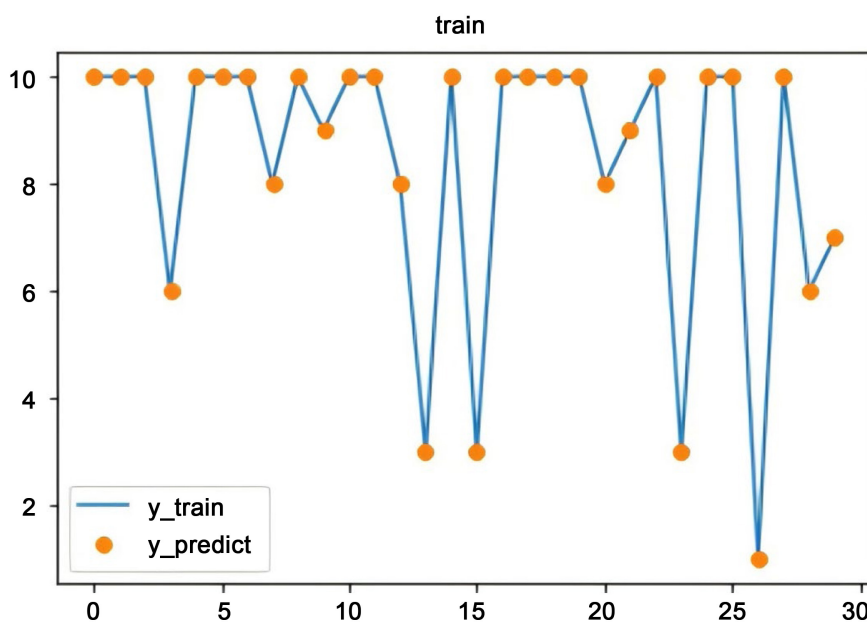


Figure 5. Prediction results of the training set.

The confusion matrix for the test set (Figure 6) displays similar trends to that of the training set, but with more prominent off-diagonal values, highlighting the model's difficulty in classifying certain categories in unseen data. This is further reflected in the prediction results for the test set (Figure 7), where there is a greater variation between the predicted and actual values compared to the training set. These discrepancies suggest that the model may struggle to generalize to new data, potentially due to overfitting on the training set or challenges in handling the nuances of the test data. To improve the model's performance, further optimization and generalization efforts are necessary to enhance its robustness for real-world applications.

$$\begin{bmatrix} 21 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 39 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 13 & 2 & 0 & 1 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 90 & 0 & 1 & 0 & 0 & 2 & 0 & 9 \\ 0 & 0 & 0 & 0 & 14 & 0 & 0 & 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 1 & 0 & 34 & 1 & 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 0 & 0 & 0 & 40 & 0 & 0 & 0 & 21 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 128 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 55 & 0 & 37 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 102 & 0 \\ 0 & 0 & 0 & 4 & 2 & 1 & 6 & 0 & 29 & 0 & 790 \end{bmatrix}$$

Figure 6. Confusion matrix of the test set.

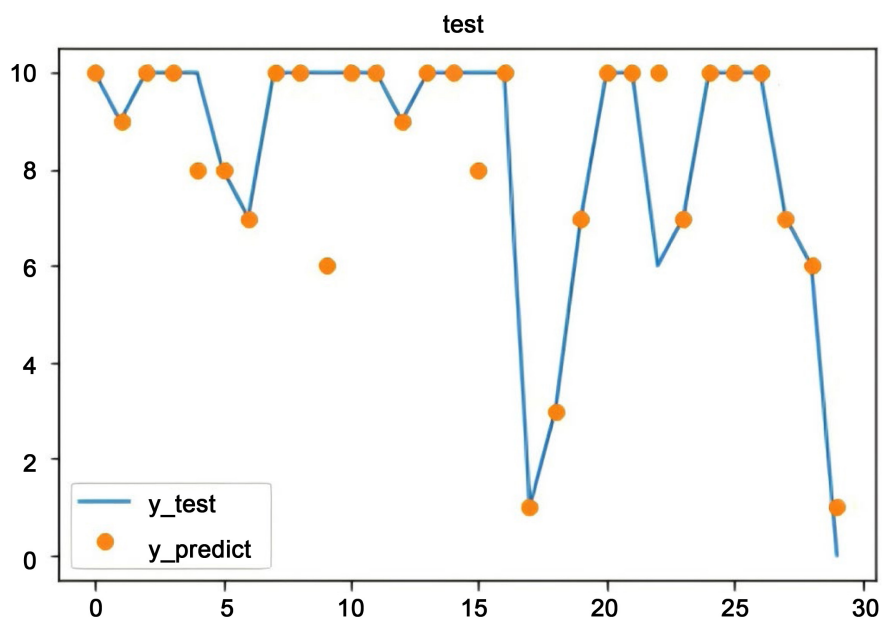


Figure 7. Prediction results of the test set.

Overall, the results demonstrate that the model performs well across most cat-

egories, achieving high classification accuracy and maintaining a low misclassification rate. However, for certain categories, such as “Technology,” the model’s performance is weaker, exhibiting a higher misclassification rate. This may be due to two main factors: 1) Overlap in text context between the “Technology” and “Finance” categories, leading to feature confusion; and 2) A limited amount of training data for the “Technology” category, resulting in the model’s weak generalization ability. To address these issues, techniques such as data augmentation can be applied to increase training samples for underrepresented categories, while incorporating attention mechanisms can improve the model’s ability to capture key textual features, thereby enhancing classification performance.

When evaluating the model, in addition to testing on the original dataset (14,632 news data), we also introduced an external public dataset, the THUCNews Chinese news classification dataset, for cross-validation. The experimental results show that the model’s test set accuracy on the original dataset is 91%, while the accuracy on the external dataset is 88%. Despite certain performance differences, the model still shows strong generalization ability. This indicates that the current dataset may have a certain degree of homogeneity, and further verification on more diversified datasets (such as multi-language, multi-field news) is needed in the future to fully evaluate the robustness of the model.

6) Model Convergence Analysis

As illustrated in **Figure 8**, during training, the loss function of the test set gradually decreases, while the classification accuracy steadily improves and stabilizes in the later stages. This trend indicates that the model has achieved convergence, effectively minimizing overfitting and maintaining a high level of generalization. These findings further validate the effectiveness of the BiGRU model in news text classification tasks.

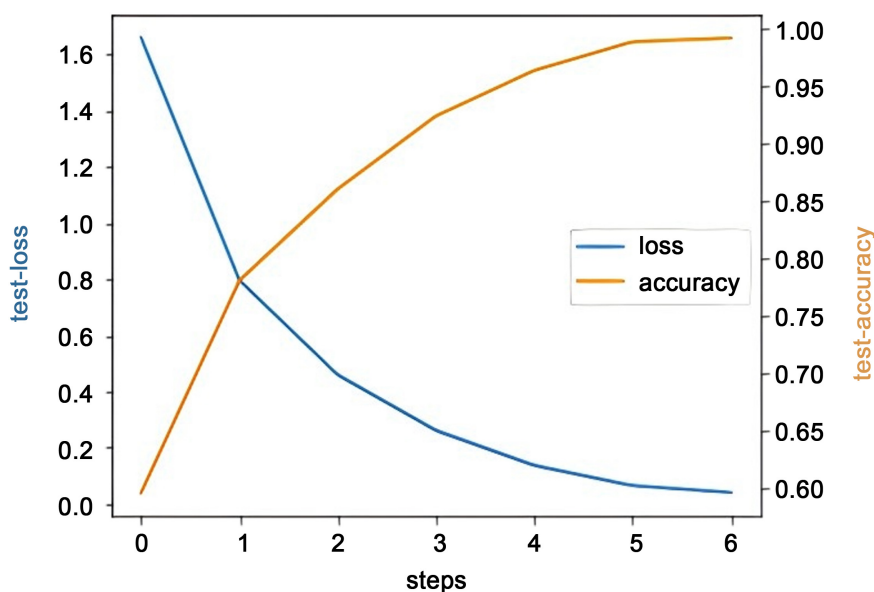


Figure 8. Relationship between loss function and accuracy of test set.

5. Conclusions

This study focuses on the design and implementation of a news text classification system, systematically exploring the complete process from requirements analysis, text preprocessing, model construction, model training, to performance evaluation. By developing a classification model based on the BiGRU neural network and integrating deep learning techniques, an efficient automatic classification method for news text is proposed. The experimental results demonstrate that the model achieves a classification accuracy of 91% on the test set, with an F1-score close to 0.90, indicating strong classification performance.

In terms of system design, a stable and efficient news classification framework was established through text preprocessing, feature extraction, and deep learning-based classification. During model optimization, hyperparameter tuning, a cross-entropy loss function, and the Adam optimizer were employed, along with an early stopping strategy to enhance the convergence speed and generalization ability of the model. Furthermore, experimental evaluations highlight the superior performance of the BiGRU model in news text classification tasks, particularly in short-text classification. However, certain misclassification issues remain for specific categories, suggesting that improvements in feature extraction and category differentiation are still necessary. Overall, this study effectively enhances the accuracy and efficiency of news text classification, providing strong support for automated text classification technology and laying a solid foundation for further optimization and expansion.

While the proposed BiGRU model shows promising performance in news text classification, there is potential for further optimization. Future research could explore several avenues: 1) Integrating advanced pre-trained language models like BERT or Transformer to improve semantic understanding; 2) Incorporating attention mechanisms for refined feature extraction and enhanced classification accuracy; 3) Using data augmentation techniques, such as synonym replacement and back-translation, alongside larger-scale datasets, to increase data diversity and improve generalization; 4) Optimizing computational efficiency to improve real-time processing and scalability for large-scale news data applications. Overall, future work will aim to enhance the model's intelligence, adaptability, and cross-domain applicability, advancing automated text classification technology for practical use.

Conflicts of Interest

The author declares no conflicts of interest.

References

- [1] Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. and Brown, D. (2019) Text Classification Algorithms: A Survey. *Information*, **10**, Article 150. <https://doi.org/10.3390/info10040150>
- [2] Aggarwal, C.C. and Zhai, C. (2012) A Survey of Text Classification Algorithms. In:

- Mining Text Data*, Springer, 163-222. https://doi.org/10.1007/978-1-4614-3223-4_6
- [3] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M. and Gao, J. (2021) Deep Learning-Based Text Classification. *ACM Computing Surveys*, **54**, 1-40. <https://doi.org/10.1145/3439726>
- [4] Liu, P.F., Qiu, X.P. and Huang, X.J. (2016) Recurrent Neural Network for Text Classification with Multi-Task Learning.
- [5] Bai, X. (2018) Text Classification Based on LSTM and Attention. 2018 *Thirteenth International Conference on Digital Information Management (ICDIM)*, Berlin, 24-26 September 2018, 29-32. <https://doi.org/10.1109/icdim.2018.8847061>
- [6] Yu, S., Liu, D., Zhu, W., Zhang, Y. and Zhao, S. (2020) Attention-Based LSTM, GRU and CNN for Short Text Classification. *Journal of Intelligent & Fuzzy Systems*, **39**, 333-340. <https://doi.org/10.3233/jifs-191171>
- [7] Lakshmanarao, A., Babu, M.R. and Bala Krishna, M.M. (2021) Malicious URL Detection Using NLP, Machine Learning and Flask. 2021 *International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, Chennai, 24-25 September 2021, 1-4. <https://doi.org/10.1109/icses52305.2021.9633889>
- [8] Martins, A. and Astudillo, R. (2016) From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification. *International Conference on Machine Learning*, New York, 19-24 June 2016, 1614-1623.
- [9] Gulli, A. and Pal, S. (2017) Deep learning with Keras. Packt Publishing Ltd.
- [10] Duan, J., Zhao, H., Qin, W., Qiu, M. and Liu, M. (2020) News Text Classification Based on MLCNN and Bigru Hybrid Neural Network. 2020 *3rd International Conference on Smart Block Chain (Smart Block)*, Zhengzhou, 23-25 October 2020, 1-6. <https://doi.org/10.1109/smartblock52591.2020.00032>
- [11] Hui, L. and Belkin, M. (2020) Evaluation of Neural Architectures Trained with Square Loss vs Cross-Entropy in Classification Tasks.
- [12] Prasetijo, A.B., Isnanto, R.R., Eridani, D., Soetrisno, Y.A.A., Arfan, M. and Sofwan, A. (2017) Hoax Detection System on Indonesian News Sites Based on Text Classification Using SVM and SGD. 2017 *4th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, Semarang, 18-19 October 2017, 45-49. <https://doi.org/10.1109/icitacee.2017.8257673>
- [13] Takeru, M., Dai, A.M. and Goodfellow, I. (2016) Adversarial Training Methods for Semi-Supervised Text Classification.
- [14] Sari, W.K., Rini, D.P. and Malik, R.F. (2020) Text Classification Using Long Short-Term Memory with Glove Features. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, **5**, 85-100. <https://doi.org/10.26555/jiteki.v5i2.15021>