Journal of Electronics and Information Science (2024)
Clausius Scientific Press, Canada

DOI: 10.23977/jeis.2024.090205
ISSN 2371-9524 Vol. 9 Num. 2

A Comprehensive Review of Text Classification Algorithms

Yachang Song[1], Xinyu Liu[1], Ze Zhou[1]

[1]*College of Electronic Information, Xijing University, Xian, China*

Keywords: Text Classification, Deep Learning, Pre-trained Models, Natural Language Processing, CNN, RNN, BERT

Abstract: This paper reviews the development of text classification algorithms, from rule-based and traditional machine learning methods to the evolution of deep learning and pre-trained models. As a crucial aspect of natural language processing, text classification is essential for various applications, such as information retrieval and sentiment analysis. The application of deep learning models like CNNs, RNNs, and pre-trained models such as BERT in text classification is highlighted, showcasing their advantages in processing large corpora. The challenges and future research directions in text classification are also discussed, offering guidance to researchers and practitioners in the field.

1. Introduction

Text classification[1], a fundamental task in natural language processing, categorizes texts into predefined categories. Initially dependent on manually crafted rules like keyword matching, the field has advanced with statistical learning methods such as Naive Bayes[2] and SVMs[3], enhancing automation and performance. The 21st century saw deep learning technologies like CNNs and RNNs capture deeper textual features. Recently, Transformer-based pre-trained models like BERT and GPT have emerged, setting new benchmarks by learning rich language representations. Despite significant progress, challenges remain, including low-resource language processing, enhancing model generalization, and reducing resource consumption.

2. Fundamental Knowledge

2.1. Text Classification Paradigms

Text classification has evolved from rule-based, machine learning, to deep learning approaches. Initially, rule-based methods classified text by defined rules, precise but costly and less adaptable. Machine learning methods, such as supervised learning algorithms like Naive Bayes, SVMs, and decision trees, as well as unsupervised learning techniques like cluster analysis and automated learning and adaptation, require extensive training data. Deep learning methods, especially CNNs, RNNs, and Transformer models, have excelled by learning complex features and capturing deep semantics, notably improving text classification performance in the pre-training and fine-tuning framework.

2.2. Fundamentals of Text Classification

Key to text classification is converting raw text into a model-processable format, including text cleaning, tokenization, stop word removal, and stemming. Feature extraction transforms preprocessed text into feature vectors, using models like Bag of Words (BoW), TF-IDF, and embeddings like Word2Vec. Model training and evaluation use labeled datasets, assessing performance through accuracy, recall, and F1 scores, further optimized by parameter tuning and cross-validation.

3. Text Classification Models

This section delves into key machine learning models, including Naive Bayes, SVMs, CNNs, RNNs, and Transformer models. Each model's development background, theoretical principles, main applications, and achievements are explored, demonstrating their effectiveness in solving text classification challenges.

3.1. Naive Bayes Classifier

In the field of natural language processing, especially in the task of text classification, the Naive Bayes classifier holds an indispensable position. This simple probabilistic classifier, based on Bayes' theorem, has become a powerful tool for addressing issues such as spam email detection and document categorization due to its straightforward mathematical principle and efficient implementation. Despite the foundational assumption of the Naive Bayes classifier — the independence between features — which often does not hold true in real applications, this model still demonstrates outstanding classification performance across various datasets.

The working principle of the Naive Bayes classifier is based on Bayes' theorem, a simple yet influential formula in probability theory. It is used to calculate the probability of a hypothesis given some evidence, as shown in the following equation:

$$P(Ck \mid x) = \frac{P(x|Ck)P(Ck)}{P(x)} \tag{1}$$

In this formula, $P(C_k \mid x)$ represents the posterior probability that the given text x belongs to category Ck. $P(x \mid C_k)$ indicates the likelihood of text x appearing, given that it belongs to category C_k.$P(C_k)$ is the prior probability of category C_k, that is, the probability of category C_k occurring before any text information is provided. Finally, $P(x)$ is the probability of the occurrence of text x. By calculating these probabilities, the Naive Bayes classifier can determine the most likely category to which the given text belongs.

3.2. Classification Method Based on Support Vector Machine Model

The Support Vector Machine (SVM), proposed by Cortes and others, is a powerful machine learning model widely used in text classification tasks. Its fundamental principle is to find the optimal decision boundary, namely the maximum-margin hyperplane, to distinguish between different categories of data points. Due to its excellent generalization ability, SVM performs outstandingly in many high-dimensional data classification tasks, including text classification.

When applied to text classification tasks, the core concept of SVM is to transform each text in the dataset into a feature vector, and then to construct a separating hyperplane to maximize the distance between support vectors, thereby achieving the purpose of classifying text data. Given a set of text data, each text is transformed into a high-dimensional feature vector $x_i \in R^p$, where each dimension represents a feature (such as the importance of a word or its frequency), and $y_i \in \{-1,1\}$ represents

the category label for each text. SVM seeks to find a hyperplane in this high-dimensional space:

$$w \cdot x + b = 0 \tag{2}$$

where w is the normal vector to the hyperplane, and b is the bias term. The choice of the hyperplane follows the principle of maximizing the margin, that is, minimizing $//\mathrm{w}//^2$ while ensuring that all data points satisfy:

$$y_i(w \cdot x_i + b_i) \geq 1, i = 1, \ldots \ldots l \tag{3}$$

Solving this minimization problem ensures that the distance from the hyperplane to the nearest data points (i.e., the support vectors) is maximized, thus enhancing the model's generalization ability.

However, in actual text classification tasks, it is often difficult to find a perfect classification hyperplane that completely separates all samples. This challenge can be addressed by introducing slack variables $\xi i \geq 0$ to meet the following conditions:

$$y_i\big((w \cdot x_i) + b\big) \geq 1 - \xi_i, i = 1, \ldots \ldots l \tag{4}$$

After introducing slack variables, the optimization goal of SVM simultaneously considers maximizing the margin and minimizing classification errors, thereby adapting to complex and noisy real text data while maintaining model generalization ability. This method provides a robust model framework for text classification, especially suitable for dealing with high-dimensional, sparse text datasets, making it an effective tool for solving text classification problems.

3.3. Classification Method Based on Decision Tree Model

The application of decision tree models in text classification aims to predict the category of texts by learning decision rules from text features. Initially, it is necessary to extract features from the text data, such as word frequency and key terms within documents. Subsequently, these features are used to construct a decision tree. The decision tree model employs a supervised learning approach, building a tree structure from the top down through recursive partitioning. Its core objective is to analyze the training dataset to learn a series of decision rules, thereby constructing a model capable of predicting the category of target variables. After the complete construction of the decision tree, pruning is often applied to the model to reduce the negative impacts of noise and overfitting phenomena on classification performance.

The decision tree model is a method that recursively builds a tree structure from the data in a supervised learning manner, aiming to predict the category of target variables by learning decision rules from the training data. This model is constructed in a top-down approach. Once the decision tree is fully built, pruning techniques can be utilized to reduce noise and overfitting, thereby enhancing the model's accuracy in classifying new data.

4. Deep Learning Models in Text Classification

Deep learning has revolutionized fields like image recognition and machine translation by learning complex feature relationships. This section discusses foundational network models like FNNs, RNNs, and CNNs in text classification, their unique advantages, and the further enhancement of model performance through text embedding techniques.

4.1. Text Classification Method Based on Basic Network Models

Deep learning technologies, by automatically discovering hidden feature patterns, have reduced the need for feature engineering, showcasing unique advantages and significant achievements in text

classification through feedforward neural networks, recurrent neural networks, and convolutional neural networks.

Feedforward Neural Networks (FNN)[4] serve as the basic framework for deep learning in text classification, consisting of input layers, embedding layers, multiple stacked neural network processing layers, and output layers. The FastText[5] model builds on this by incorporating character-level n-grams as features, enhancing the understanding and generalization of infrequent words, though it has limitations in handling long-distance sequence dependencies and complex semantics.

To delve deeper into the rich semantic information contained in texts and effectively address the context dependency issues in longer texts, researchers have turned to Recurrent Neural Networks (RNN)[6] to model the sequential nature of text. Text is understood as a series of words arranged in temporal order, such as text $S = \{w_1, w_2, ..., w_n\}$, where each word w_i represents its state at a specific time t_i. RNNs are widely regarded as effective tools for handling such sequential data. However, standard RNNs often encounter vanishing gradient problems when processing long sequences, making it difficult for the model to retain information about long-distance dependencies. To address this challenge, variants of RNNs, such as Gated Recurrent Neural Networks and Long Short-Term Memory networks (LSTM)[7], are commonly used in text classification research. LSTMs manage the flow of information effectively through memory cells and three special gates (input gate, forget gate, output gate), thereby preserving long-term dependency information.

Recurrent Neural Networks (RNN) excel in processing the temporal characteristics of text, while Convolutional Neural Networks (CNN) demonstrate advantages in phrase or keyword identification through the TextCNN model, using multi-size filters to capture local features of text. TextCNN integrates key information through max pooling to form advanced representations, effectively revealing local semantics but is limited in handling long-distance dependencies.

Furthermore, the development of text embedding technologies has further enhanced model performance, including character-level, word-level, and sentence-level embeddings, each addressing the issues of out-of-vocabulary words, learning grammatical and semantic information, and capturing the potential connections between sentences.

Overall, compared to traditional machine learning methods, deep learning's basic network models show significant advantages in processing word order information and deep semantic analysis. However, these foundational models also have their own limitations, such as the simplicity of the model structure and the capacity to model specific types of information.

4.2. Classification Method Based on Graph Neural Network Models

Graph Neural Networks (GNNs)[8] enhance the performance of classification tasks by mining the internal graph structure information of texts, such as syntactic and semantic dependency trees. GNNs utilize an information propagation mechanism, iteratively improving node representation by aggregating information from neighboring nodes and combining it with the node's own representation. Various aggregation and combination functions have led to a diversity of GNN architectures, adaptable to different application scenarios.

GNNs demonstrate flexibility and effectiveness in text classification tasks across different scales, from words to documents. TextRank, a pioneer in applying graph structures to text classification, constructs graph models with nodes and edges representing text units and their relationships, catering to a wide range of text processing needs from sentiment analysis to topic classification.

On a more macroscopic, document-level, Peng and others introduced a graph-based CNN model in 2018. This model represents texts as word graphs and uses graph convolutions to feature these word graphs for document classification. This method effectively captures discontinuous and long-distance dependencies within texts, leveraging the powerful semantic learning capabilities of CNNs.

In 2019, Yao and his team adopted another approach by constructing a heterogeneous graph containing both word and document nodes to classify text data, a novel innovation at the document granularity. In this graph, the connections between words and documents are determined by word frequency, while connections among words are based on their semantic associations. Using Graph Convolutional Networks (GCNs) to process this structure and facilitating information transfer between documents to learn embeddings for unlabeled words and documents, this strategy improved performance by an average of 1.2% across multiple standard datasets, achieving an average accuracy of 84.5%.

In sentence-level classification tasks, in 2019, Zhang and his team used the Dependency Tree (DT) structure within sentences to construct text graphs, subsequently applying GNNs for sentence classification. This method significantly enhanced the accuracy of the classification model by effectively utilizing long-distance dependencies between words. Continuing in this research direction, in 2020, Zhang and others constructed independent graph representations for each document and used GNNs to learn representations of words within documents. This approach, emphasizing the capability to generate embeddings for unknown words in documents, exhibited exceptional performance across multiple standard datasets.

4.3. Classification Method Based on Transfer Learning Models

Transfer learning technology[9], by leveraging relevant knowledge, enhances the model's performance on new tasks and reduces the dependency on labeled data. Especially in cross-domain or multi-level text classification tasks, such as fake news detection and topic identification, transfer learning has shown significant advantages, overcoming the limitations of traditional deep learning models and has proven effective in various fields like sentiment analysis and computer vision.

In 2018, Howard and others proposed ULMFiT[10], an innovative fine-tuning method for universal language models, which implemented transfer learning in natural language processing (NLP) tasks. It is applicable to a variety of text classification tasks without the need for additional data support and significantly reduced error rates in small sample data environments. In 2019, Banerjee and his team introduced HTrans, a hierarchical transfer learning strategy for multi-label text classification problems. By fine-tuning classifiers from top to bottom within a hierarchical structure, they significantly improved classification performance. In the same year, Houlsby and others addressed the issue of parameter efficiency when fine-tuning pre-trained models, introducing an efficient adapter module strategy that requires tuning only a small number of parameters to achieve good performance. In 2020, Raffel and others proposed a framework that unifies all text-based tasks into a text-to-text format, further expanding the application of NLP transfer learning. In 2021, Cao and others proposed a new fine-grained cross-domain sentiment classification deep transfer learning mechanism to better utilize unlabeled data in the target domain. This approach uses a domain adaptation model to minimize the feature distribution differences between the source and target domains.

Transfer learning reduces the reliance on data and annotations in text classification, facing challenges such as the complexity of cross-domain knowledge transfer, the assessment of cross-domain transfer potential, and handling content privacy. Specifically, the context dependency of word meanings may lead to negative transfer from a weakly related source domain to the target domain, necessitating the development of new strategies to avoid such impacts.

4.4. Classification Method Based on Pre-trained Models

Pre-trained models[11], by learning general data characteristics on large datasets, provide a strong starting point for specific natural language processing tasks. This approach allows downstream tasks

to fine-tune based on foundational knowledge, achieving high performance and reducing the need for training data. The effectiveness of pre-trained models stems from their rich inherent general information, leading to rapid convergence, low data dependency, outstanding performance, and reduced risk of overfitting.

Pre-trained models learn general characteristics on large datasets, offering an optimized starting point for specific tasks such as text classification. This strategy reduces the need for training data, speeds up model convergence, improves performance, and lowers the risk of overfitting due to the extensive information contained within.

Unsupervised learning techniques are widely applied in pre-trained models because they can be trained on a large amount of unlabeled data. Self-supervised learning, a combination of unsupervised and supervised methods, generates training signals from unlabeled data to facilitate learning. This enables models to learn feature extraction without manual annotation, significantly reducing data preparation costs. Self-supervised learning has proven effective in large-scale language models like BERT[12] and GPT, advancing natural language processing technology.

Word2Vec[13], an early pre-trained language model, trained through unsupervised skip-gram and continuous bag-of-words strategies, provides features for text classification. However, it produces fixed word vectors, ignoring the polysemy of words, such as different meanings of "apple" in various contexts, which may limit model performance. Therefore, Word2Vec generates a static word embedding for "apple," which might restrict the model's capability in tasks like text classification.

To address the issue of capturing polysemy in models like Word2Vec, researchers introduced the Masked Language Model (MLM), as used in BERT. MLM randomly masks words in the input, training the model to predict these words based on context, allowing the model to learn context-related word embeddings without manual annotation. This requires the model to utilize bidirectional context, enhancing sensitivity to semantic changes and generalization capability.

When using BERT for text classification tasks, a special token [CLS] is added at the front of the input sequence. This token has a specific meaning within BERT's architecture, used to generate an embedding vector representing the entire input sequence for classification tasks. In the process of text classification, analyzing the final embedding vector corresponding to the [CLS] token, and passing it to a simple classifier (such as a softmax function), allows for the prediction of the entire text sequence's category label.

$$p(y_i|h) = softmax(Wh_{[CLS]}) \tag{5}$$

The MLM (Masked Language Model) training strategy has endowed BERT with powerful language understanding capabilities. After completing pre-training, BERT can be fine-tuned to adapt to a wide range of NLP tasks, including text classification, named entity recognition, and question-answering systems. In these tasks, BERT's bidirectional context comprehension significantly enhances model performance.

For instance, in sentiment analysis tasks, the rich contextual information learned through MLM allows BERT to understand the sentiment in texts more accurately. In question-answering systems, BERT's ability to understand bidirectional context enables it to more precisely locate the position of answers.

The introduction of the MLM training method for the BERT model has not only opened a new direction for the development of pre-trained language models but has also greatly advanced the field of NLP. Its ability to capture complex bidirectional contextual information is key to its outstanding performance across many tasks. However, the BERT model also faces some challenges, including the significant demand for computational resources and limitations in processing long texts. Despite these challenges, the successful application of BERT and its variants in the NLP field undoubtedly demonstrates the immense potential and value of deep learning-based pre-trained models.

5. Comparison and Experimental Analysis of Text Classification Algorithms

5.1. Evaluation Methods

Evaluating the strengths and weaknesses of classification algorithms often involves multiple aspects, including accuracy, generalization ability, computational efficiency, and model interpretability. To comprehensively assess classification algorithms, researchers use various evaluation metrics and methods. Here are some of the most commonly used evaluation methods:

Accuracy is one of the most intuitive evaluation metrics, defined as the proportion of correctly classified samples to the total number of samples. For binary classification problems, accuracy can be represented as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{6}$$

where TP (True Positives) and TN (True Negatives) represent the number of correctly classified positive and negative classes, respectively, while FP (False Positives) and FN (False Negatives) represent the number of samples incorrectly classified as positive and negative classes, respectively.

Precision focuses on the proportion of samples predicted as positive by the model that are actually positive; recall, on the other hand, focuses on the proportion of all actual positive samples that are correctly predicted as positive by the model. These two metrics are particularly useful in cases of data imbalance, with formulas as follows:

$$Precision = \frac{TP}{TP+FP} \tag{7}$$

$$Recall = \frac{TP}{TP+FN} \tag{8}$$

The F1 score is the harmonic mean of precision and recall, attempting to strike a balance between these two metrics. The formula for the F1 score is:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{9}$$

The F1 score is particularly applicable in scenarios where precision and recall are equally important.

5.2. Performance Comparison

Across several commonly used datasets:

IMDb Movie Review Dataset: Used for sentiment analysis, it contains a large number of movie reviews with their sentiment labels (positive or negative).

20 Newsgroups: A news group message dataset containing documents from 20 different topics, suitable for topic classification tasks.

Reuters-21578: A news wire dataset frequently used for news article topic classification.

AG's News Corpus: A news article dataset for news classification, containing multiple categories.

Different text classification models have shown their respective strengths and limitations:

FNN: While performing well on some simple text classification tasks, FNNs often fail to capture long-distance dependencies in text, limiting their application in complex tasks.

LSTM and RNN: Due to their recurrent structure, these two models can handle sequential data well, especially in capturing long-term dependencies, superior to FNN. However, LSTMs and RNNs may encounter gradient vanishing or exploding problems when processing very long sequences.

CNN: By leveraging local relevance, CNNs excel at extracting key local features in text, particularly suitable for tasks like sentiment analysis that require capturing local text patterns.

word2vec: As a pre-trained word embedding method, word2vec can effectively capture semantic relationships between words, but its performance largely depends on the capability of the downstream model.

BERT: As a Transformer-based pre-trained model, BERT captures rich linguistic features through pre-training and achieves state-of-the-art (SOTA) performance in multiple text classification tasks by fine-tuning. Its success is largely due to its deep bidirectional structure, enabling a comprehensive understanding of language context.

Experimental results show that models based on deep learning (such as CNN, LSTM, BERT) typically outperform traditional machine learning models (such as FNN based on word2vec) in text classification tasks. Among them, BERT, due to its powerful language understanding ability, performs exceptionally well on datasets with complex semantic relationships. However, the performance of deep learning models often relies on a significant amount of training data and computational resources.

The performance differences between algorithms are primarily determined by their architectural features: for example, CNNs excel at capturing local features, LSTMs and RNNs are better at capturing long-term dependencies, and BERT's bidirectional Transformer structure enables it to thoroughly understand text context. Choosing which classification model to use depends on the specific circumstances and needs.

Overall, the main advancements in the text classification field currently focus on the integrated use of deep learning technology and pre-trained language models. By fine-tuning pre-trained models or designing optimized input prompts for specific tasks, researchers can significantly improve model performance across various application scenarios. With the continued progress of these technologies, the text classification field is expected to experience rapid development. This article provides valuable insights and guidance for the research community and practitioners by analyzing the performance of various text classification models on different datasets. These research findings have practical value in selecting, designing, and optimizing text classification models, contributing to the further development and innovation of text classification technology.

6. Future Directions and Challenges

The text classification field faces challenges with technological advancements and data growth, such as the complexity of processing unstructured text, understanding the implied meanings and semantics, and long-distance dependencies. Future directions include multimodal classification integrating information beyond text, cross-lingual classification breaking language barriers, and enhancing performance with deep learning technologies like pre-trained models BERT and GPT. Challenges involve processing unstructured text, understanding the complexity of natural language, adapting models to new vocabulary, and dealing with multimodal and cross-lingual text processing. Deep learning offers new avenues for complex text classification, indicating that through technological innovation and optimization strategies, text classification will achieve more precise and efficient processing capabilities.

7. Conclusion

This article reviews the evolution of text classification algorithms, from rule-based and statistical methods to deep learning models such as CNN, RNN, LSTM, BERT, and GPT, highlighting their contributions to improving classification accuracy and efficiency. Text classification technology has played a key role in various domains, including sentiment analysis and spam detection, aiding businesses in extracting valuable information. Despite facing challenges such as processing unstructured text and understanding complex semantics, future research will focus on innovations in model architecture, interpretability, cross-lingual capabilities, and multimodal classification methods.

The aim is to achieve more accurate, transparent, and widely applicable text classification solutions, continuing to advance the field of natural language processing.

References

[1] Sebastiani F. Machine learning in automated text categorization. ACM Computing Surveys (CSUR), 2002, 34(1):1-47.

[2] Maron M E. Automatic indexing: an experimental inquiry. Journal of the ACM (JACM), 1961, 8(3):404-417.

[3] Joachims T. Text categorization with support vector machines: Learning with many relevant features//Lecture Notes in Computer Science: volume 1398 Machine Learning: ECML-98, 10th European Conference on Machine Learning, Chemnitz, Germany, 1998: 137- 142.

[4] Unanue I J, Haffari G, Piccardi M. T3l: Translate-and-test transfer learning for cross-lingual text classification. arXiv preprint arXiv:2306.04996, 2023.

[5] Joulin A, Grave E, Bojanowski P, et al. Bag of tricks for efficient text classification//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, Volume 2: Short Papers, 2017: 427- 431.

[6] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model//INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Japan, 2010: 1045-1048.

[7] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8):1735-1780.

[8] SCARSELLI F, GORI M, TSOI A C, et al. The graph neural network model. IEEE Transactions on Neural Networks, 2008, 20(1):61-80.

[9] Zhuang F, Qi Z, Duan K, et al. A comprehensive survey on transfer learning. Proceedings of the IEEE, 2020, 109(1):43-76.

[10] Howard J, Ruder S. Universal language model fine-tuning for text classification//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, Volume 1: Long Papers, 2018: 328-339.

[11] Wu T, Caccia M, Li Z, et al. Pretrained language model in continual learning: A comparative study//The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, 2022.

[12] Devlin J, Chang M, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, USA, Volume 1 (Long and Short Papers), 2019: 4171-4186.

[13] Ikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, 2013, 26.