Development of Text Classification Methods Based on Deep Learning

Xiaoxi Jiang
School of Computer Science, Central South University, Changsha, China

Keywords: Text Classification, Deep Learning, Text Processing, NLP.

Abstract: As the Internet continues to grow, more and more text data are being produced online. Proper classification can be beneficial for mining the valuable information contained in text, so management and classification on text data is quite crucial. Text categorization, or the process of adding labels or tags to text units, is one of the classic issues associated with Natural Language Processing (NLP). In the early stage of the research, some researchers put forward the methods of keyword matching and expert rules for text classification, but the effect is poor due to the limitation of matching rules. Fortunately, deep learning as an emerging technology has attracted a lot of attention from researchers. Deep learning-based text classification has shown better categorization performance in the processing of text data. This paper explores the information of various deep learning models in design and application in detail and introduces the relevant methods to improve the efficiency and accuracy of text classification, and finally summarizes the future research directions of deep learning algorithms in the field of text classification.

1 INTRODUCTION

Text processing has seen tremendous advances in deep learning in recent years. Algorithms and computing power improving, breaking improvement has been made in processing and understanding the complexity of human language. Text units, such as phrases, queries, paragraphs, and documents, should be given tags or markings as part of text classification, sometimes referred to as text categorization. Tickets, user comments, emails, chat logs, insurance claims, web data, social media and customer service department queries and answers are just a few of the many sources from which text data is gathered. Text is a very comprehensive source of knowledge. However, because text is unstructured, it may be difficult and time-consuming to extract insights from it. Challenges for processing text include designing models capable of understanding the deeper meanings of texts, handling texts in different languages and dialects, and overcoming issues with noisy data and biases, and so on. These challenges are at the forefront of current research.

2 DEEP LEARNING MODULES FOR TEXT CLASSIFICATION

2.1 RNN

Recurrent neural networks, or RNNs for short, were developed by Jordan et al. using the Hopfield network in conjunction with the notion of distributed parallel processing and storage (Jian & Sun 2021). One fundamental type of recurrent neural networks (RNNs) is the Simple Recurrent Neural Networks (SRNs). Their primary characteristic is their recurrent connection, which enables the network to handle sequential input while preserving a given amount of memory. SRNs usually have input layers, hidden layers, and output layers. For the hidden layers, each neuron receives not only information from input layers but hidden states from the previous time step. This structure allows the network to process current input considering previous information. So that SRNs can capture temporal dependencies in the given sequence. In the SRNs, the hidden state is update by:

$$h_t = f(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \quad (1)$$

$$y_t = f\left(W_{hy}h_t + W_{xy}x_t + b_y\right) \quad (2)$$

Jiang, X.
Development of Text Classification Methods Based on Deep Learning.
DOI: 10.5220/0012829500004547

In *Proceedings of the 1st International Conference on Data Science and Engineering (ICDSE 2024)*, pages 518-523
ISBN: 978-989-758-690-3

where h_t is the hidden state at time step t, x_t is the input, y_t is the output. The activation function is denoted by f, the bias term by b, and the weight matrix by W. f is usually a nonlinear function such as tanh or ReLU.

In text classification tasks, RNNs, especially SRNs, are used to process and understand text sequences. These networks are able to learn the temporal relationships between words, which is essential for understanding sentence structure and semantics. Text classification using RNN models is commonplace in applications including spam detection, subject categorization, and sentiment analysis. For example, in topic classification tasks, RNNs can help to recognize text topics. The processing procedure for these tasks usually have following steps: First the text sequence is encoded (i.e., text vectorization). Then the encoded text sequence is fed to the model. Ultimately, the decision on classification is reached via one or more completely linked layers.

Although SRNs are theoretically capable of handling long-distance temporal dependencies, in practice they often experience gradient vanishing or gradient explosion during training. This makes it difficult for the network to learn long-term dependencies. Also, RNN is a biased model. This means the later words have a greater impact on the results than the earlier words. As a result, it may be less effective when used to capture the semantics of long texts, such as entire documents. This is because the key constituents may appear anywhere in the document, not simply at the end.

2.2 LSTM

Among the several RNNs variations, it is Long Short-Term Memory (LSTM) that is the most popular architecture being used widely. LSTM can solve the problem of gradient vanishing or gradient explosion suffered by vallina RNNs when dealing with long-term dependencies by introducing three gates (input gate, output gate, forget gate) and cell state. The cell state is the key to realizing long-term memory in LSTM. Only a small amount of linear manipulation is done during the transmission of cell states, allowing information to be easily passed on without change. This enables long term memory retention. The addition of information to the cell state is controlled by the input gate. It controls the inflow of information through a sigmoid activation function. The forget gate determines which portion of the data in the cell state should be forgotten. The forget gate regulates the information outflow using a sigmoid function, same as the input gate. Using two activation functions, the output gate determines which portion of the cell state should be transferred to the following time step. The sigmoid function controls the passing of information, and the tanh function scales the information. With these three gate units, the cell state is updated at each time step and passed between time steps (Staudemever & Morris 2019).

In practical applications, the length of text data to be learned is usually not too short. Therefore, achieving long-term memory of textual information is particularly important for classification tasks. LSTM performs well in text classification tasks because it has a good understanding of the long-term contextual information in the document. To be specific, LSTM can learn the deeper meaning of text, such as grammar and syntax. It can also learn the complex relationships between words and how their order in a sentence affects the semantics. As one of the variants of Simple RNNs, LSTMs are also capable of handling variable-length input sequences.

As the most common and fundamental models using in text classification tasks, LSTM has many variations to boost its ability to learn long distance dependencies. Sentence-State LSTM improves its parallel processing ability by updating all the states of words in each loop instead of processing one by one. Also, the model introduces a global sentence-level state to represent the sentences for exchanging information between words. These makes S-LSTM can learn the utter sentence features faster and maintaining high efficiency in processing long sentences at the same time (Zhang et al. 2018). Tree-LSTM is a variant that extends the LSTM network to tree topologies. This architecture can process the information providing by the child node on each node at the same time. Tree-LSTM is particularly good at dealing with the data with hierarchical structure (for example syntactic trees for sentences) because it can the semantic and syntactic of sentences better than vallina LSTM. Instead of unidirectional LSTM, there are Bidirectional LSTM. The Bidirectional LSTM can benefit more from the contextual information. So it gets higher marks than unidirectional LSTM on the classification accuracy. In addition, LSTMs can further enhance their performance through Attention Mechanism.

2.3 GRU

Gated Recurrent Unit (GRU) is a very effective variation of LSTM network. Comparing to the LSTM, it only has two gate units: update gate and reset gate. The update gate determines how the input

sequences and the former hidden state affect the current hidden state. The reset gate determines how much information of the former hidden state can be forgotten, which means the model has the ability to ignore some unimportant former messages. To summarize, the structure of the GRU is simpler than that of LSTM and training GRU usually costs less time. So the GRU is also a very popular model for the text classification tasks.

Like LSTM, GRU has many variants. In 2015 researchers constructed a GRU neural network model based on a stack structure, using a tree structure to capture long-term dependent information. In 2018 a Multi-sentiment-resource Enhanced Attention Network (MEAN) was introduced. MEAN incorporates emotion lexicon, negation words, and intensity words—three different categories of sentiment linguistic knowledge—into the deep neural network through attention processes. The experimental results on Movie Review (MR) and Stanford Sentiment Treebank (SST) demonstrate that MEAN has better appearance than competitors and achieves best performance on both datasets (Lei et al. 2018). A completely attention-based bidirectional CRU neural network (Bi-GRU) was suggested in 2019 (FABG). The model learns the text's semantic information using Bi-GRU, and at each step it learns the weights of the Bi-GRU's previous and current outputs using the complete attention method. such that the model performs better on text classification problems by enabling the representation to pick up crucial information at each stage and disregard irrelevant information (Tang et al. 2019)

2.4 Transformer

Self-attention, also called intra-attention, is a mechanism used in models to learn dependencies between different positions within a sequence. With this mechanism, the model is able to not only focus on current position but also considering other places of the sequence while processing text sequences. For self-attention can simulate complex relationships of words in sentences, it becomes crucial in NLP tasks. It is the parallel processing capability of this mechanism that makes the model more efficient when processing long sequences.

Based on the self-attention mechanism, the transformer model was proposed in 2017. The Transformer model utilizes mere attention method for recognizing global relationships between input and output, eschewing repetition. For long-distance dependencies in texts, the self-attention mechanism allows the model pay attention to all other positions in the input sequence, which means that each position of the input sequence is able to obtain information directly from the whole sequence. As for the traditional recurrent neural network, information can only be passed between time steps in chronological order. Moreover, RNN experiences gradient expansion and gradient vanishing while working with lengthy sequences. As for the convolutional neural network, multiple layers or large convolutional kernels are required to capture long-distance dependencies. Undoubtably, it will increase the model’s complexity and computation cost. Considering the parallelization, parallel computation is hard for the recurrent neural network because dealing current information needs the result of the preceding time step. The convolutional neural network usually uses local receptive field, which means the convolutional kernels need to be moved in certain order while computing. So the CNN can hardly realize parallelization either. The Transformer model computes attention weights in parallel for each position's relationship with all other positions. Additionally, the model utilizes positional encoding to represent each word in the text sequence. These positional encodings can be computed and stored once for the entire text sequence input to the model. Subsequently, during computations, the model can directly employ these positional encodings. This design also facilitates parallel computation. In terms of its representational capacity, the Transformer model employs a multi-head attention mechanism, where each attention head possesses its own weight matrix. This setup enables the model to concurrently learn different attention patterns across multiple heads. The attention output formed by the weighted combination of all attention distributions often contains richer information, enhancing the model's representational capacity and generalization ability. In summary, the Transformer model has already become the mainstream model for NLP tasks today for its prominent advantages in handling long text sequences and its excellent performance in training efficiency.

3 TEXT CLASSIFICATION MODELS

3.1 Pre-trained Models

Pre-trained models have evolved from the earliest Word2vec and GloVe embeddings to more recent universal language models such as ULMFiT and ELMo for text classification. Pre-training often consists of two steps: first, train the model on a bigger dataset to get it to a satisfactory state; second, modify the pre-training model to fit different tasks and refine it with the task-specific dataset. Models such as BERT, XLNet, and RoBERTa are all adaptations of the Transformer.

BERT is a transformer-based bidirectional deep language model that captures bidirectional contextual semantics proposed by Devlin et al. in 2018 (Devlin et al. 2018). The design ideas of BERT include a transformer-based architecture, bidirectional pre-training and two pre-training tasks. Since the advantages of the transformer model have been described in detail above, the paper will not repeat them here. One of the main breakthroughs of the BERT model is the bidirectional pre-training mechanism, which allows the model to take into account the input text's left and right contexts during the pre-training stage. The model will be able to understand the text's context more fully by using this procedure. It is Masked Language Model (MLM) and Next Sentence Prediction (NSP) that are the two pre-training tasks of BERT. The model must learn the bidirectional representation of every word in the context by predicting words that are randomly blocked in the input text for the MLM problem. The NSP task assists the model understand the link between sentences through requesting it to predict whether two sentences are consecutive or not. After finishing NSP tasks, the model may well comprehend the logic of the sentences, even the text structure.

Built based on Transformer-XL, XLNet is a generalized autoregressive language model (Yang et al. 2019). Transformer-XL is an improvement of Transformer, solving the problem that the maximum dependency distance between characters of Transformer is limited by the input length and the decrease of training efficiency. Text categorization, sentiment analysis, and natural language reasoning reach state-of-the-art levels in XLNet, which performs better in tasks involving lengthy text sequences due to Transformer-XL's speed and ability to record larger context durations. In 2019, Liu et al. proposed RoBERTa model by optimizing on the basis of BERT (Liu et al. 2019). RoBERTa model has longer pre-training and uses larger batch size compared to BERT. The model also removes the NSP task to improve the performance of downstream tasks and uses dynamic masking to improve model generalization. Ultimately, the best performance at the time was achieved on natural language processing tasks such as GLUE, SQuAD, and RACE, demonstrating that the BERT pretraining target is still competitive with the right design choices (Liu et al. 2019).

3.2 Model Light-Weighting

Through the process of training a compact model (the student) to mimic the behavior of a bigger model (the teacher), knowledge distillation is a compression approach (Sanh et al. 2019). In supervised learning, the probability distribution of the output of the teacher model is used to minimize the loss of cross-entropy, which trains the student model. Using this approach, there will be a student model with fewer parameters, lower computational requirements, but maintaining high performance. Applications for knowledge distillation are numerous and include speech recognition, computer vision, NLP tasks, and even more. Especially when deploying deep learning models on resource-constrained devices, knowledge distillation can effectively reduce model size and the time of inference.

Network quantization is the process that replaces full-precision network parameters with lower precision, e.g., replacing float numbers with 2-bit or 8-bit integers. However, the accuracy of network will usually decrease slightly after the compression of quantization. Except for quantization, another popular compression technique is pruning. The technique of pruning involves eliminating "unimportant" connections from a network and then adjusting the sparse network to bring it back to accuracy. Reducing the pruning-induced performance error is the goal of fine-tuning (Huang et al. 2022).

Various of approaches have been used in order to improve the applicability of models in resource-constrained environments. Besides knowledge distillation, model pruning and quantization, methods include the use of lightweight network architectures, optimization of the training process of the models, and the development of specific hardware accumulators. These approaches aim to make models more efficient, enabling them to work on low-power and low computational devices without sacrificing too much performance. For example, DistilBERT, a general-purpose pre-trained variant of BERT that is 40%

smaller and 60% quicker than BERT, was introduced by Sanh et al (Sanh et al. 2019). This shown that DistilBERT is a strong choice for edge applications by proving that a general-purpose language model can be effectively trained using distillation and that the individual components can be examined with an ablation research (Sanh et al. 2019).

3.3 Multitask Learning

Multitask Learning (MTL) is an approach from machine learning, which allows models to train multiple related tasks simultaneously to improve learning efficiency and performance by sharing information. The core of this method is that sharing some same presentation of features is possible when the training tasks have similarities in the distribution of data. By this means, performances on different tasks can be improved at the same time by sharing information. Connections between tasks are established through simultaneous training and optimizing in MTL (Zhang et al. 2020). This training mode fully facilitates the information exchange between tasks, so that each task can obtain inspiration from other tasks, and indirectly utilize the data of other tasks with the help of information migration in the learning process, thus alleviating the dependence on a large amount of labeled data and achieving the purpose of improving the learning performance of the respective tasks.

Multitask learning has a promising application in NLP tasks such as text classification. With multitask learning, the knowledge learned from one task can be utilized to improve other tasks (Zhang et al. 2020). For example, combining a tokenization task with a named entity recognition task can improve the efficiency of recognizing unfamiliar characters. In text classification tasks, multi-task learning can improve performance by sharing word embedding layers so that different tasks can share the ground-level semantic representation while maintaining specific parameters for the task at the top level. The technique can address issues with small sample sizes, lessen the requirement for a significant quantity of labeled data, and enhance the model's capacity for generalization.

4 CONCLUSION

In the area of deep learning-based text categorization, this study provides a more thorough analysis of the sample models, outlining the traits, concepts, and key technical aspects of each model. Deep learning-based text classification proposes a rich theory related to text classification and provides new technology and new ideas for research in other related fields, with a broad research prospect. Even though a lot of researchers have been working in this field recently and have produced a lot of study findings, there are still some issues that need to be resolved in the course of further research projects.

- Propose new models and methods. It will take new deep learning models to increase categorization accuracy.
- Interpretability of deep learning models. Deep learning models achieve remarkable results in text classification problems, thanks to their unique capabilities in semantic mining and feature extraction. However, these models are difficult to reproduce and offer poor interpretability during the training process.
- Cross-language or multi-language text classification. Cross-language text categorization is being used by businesses and multinational organizations more and more. It is challenging to apply classification models trained in the source language to the target language classification issue because between the data in the source and target languages, there is no feature space overlap.
- The sheer number of web pages that really need to be searched is compared to the quantity of data used for text categorization; techniques that work well on smaller data sets might not work as well on larger ones. This disconnect between text classification research and its practical applications is evident in the case of information retrieval. It is therefore worthwhile to research and develop ways to guarantee that the algorithm maintains a superior classification impact on huge data sets.

REFERENCES

P. Jia, and W. Sun, A Survey of Text Classification Based on Deep Learning, Computer and Modernization, 7, 29–37 (2021).

R.C. Staudemeyer, and E.R. Morris, Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks, ArXiv:1909.09586, (2019).

Y. Zhang, Q. Liu, and L. Song, Sentence-State LSTM for Text Representation, ArXiv.org, (2018).

Z. Lei, Y. Yang, M. Yang, and Y. Liu, A Multi-sentiment-resource Enhanced Attention Network for Sentiment Classification, ArXiv.org, (2018).

Q. Tang, J. li, J. Chen, H. Lu, Y. Du, and K. Yang, " Classifying news texts using a full attention-based bi-GRU neural network," in Proceedings of *2019 IEEE 5th International Conference on Computer and Communication*s (2019), pp 1970-1974.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, ArXiv.org, (2018).

Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R.R. Salakhutdinov, and Q.V. Le, XLNet: Expanded Autoregressive Pretraining for Linguistic Interpretation, Advances in Neural Information Processing Systems 32, 1-11 (2019).

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, RoBERTa: A Sturdily Enhanced BERT Pretraining Method, ArXiv.org, (2019).

V. Sanh, L. Debut, J. Chaumond, and T. Wolf, DistilBERT, a more condensed, quicker, less expensive, and lighter variant of BERT, ArXiv.org, (2019).

Z. Huang, S. Yang, W. Lin, J. Ni, S. Sun, Y. Chen, and Y. Tang, Knowledge Distillation: A Survey, Chinese Journal of Computer 45(3), 1789-1819 (2022).

Y. Zhang, J. Liu, and X. Zuo, Survey of Multi-Task Learning, Chinese Journal of Computer 43(7), 5586-5609 (2020).