Journal of computer science and software applications
https://www.mfacademia.org/index.php/jcssa
ISSN:2377-0430
Vol. 4, No. 6, 2024

The Frontier Research on Text Classification Using Deep Learning

Michael Bennett
University of Delaware, Department of Computer and Information Sciences
Newark, USA
mbennett.b9767@udel.edu

Abstract:

Text classification is a core task in natural language processing, and significant advancements have been made through machine learning and deep learning research. This paper presents an overview of the current research landscape and examines the application of deep learning networks to text classification. It discusses the foundational concepts of text classification, feature extraction, text representation, and other related techniques and methods employed in text classification.

Keywords:

Neural Networks; Text Classification; Deep Learning.

1. Introduction

The widespread use of Internet technology has led to an exponential increase in the amount of online text data generated, and the extraction of valid information from this huge amount of text data has played an important role in the management and service of many industries, such as medical advice, hotel services, cinema screenings and business management, leading to the important issue of automatic text classification. It has become impossible to classify and organise text using manual methods [1]. Of all this information, text information is the easiest and most convenient way for people to use it, and it is also the most numerous. Users can express their opinions directly through words without further processing. However, due to the chaotic structure of textual information, it is difficult to organise and differentiate it by manual means. Therefore, how to extract the basic features in text information accurately and efficiently is a key issue in text information processing.

Text classification is the automatic classification of text sets by a computer according to a specific classification system or standard[2]. The purpose of text classification techniques is to express and distinguish text information quickly and efficiently, mainly to accomplish automatic classification of text, to deal with the problem of confusing text information, and to provide an efficient way to obtain and classify text information. Moreover, researchers can use text classification techniques to achieve data mining in natural language. There are two main types of traditional text classification methods, one is based on machine learning and the other is based on deep learning. In the traditional machine learning field, the main algorithms used for text classification are logistic regression, K-nearest neighbour, plain Bayes, decision trees, support vector machines and topic models. The basic idea is to use feature engineering to represent text, and then train the classifier to predict the class of text.

The advantage of deep learning over machine learning is that it uses its own network structure to automatically learn the feature representation of data without the need for manual feature engineering, and commonly used models include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and Graph Neural Networks (GCNs).

2. Deep Learning based Text Classification

2.1. Convolutional Neural Networks (CNN)

TextCNN, a text classification model for convolutional networks. Convolutional Neural Networks (CNNs) were previously mentioned in the field of machine vision until Kim proposed the TextCNN text classification model based on CNNs [3]. Compared with traditional image CNN networks, TextCNN maintains the original network structure and simplifies the convolutional layers. The simple structure of TextCNN network results in fewer parameters, less computation and faster training. The disadvantage is that the model is not very interpretable, and it is difficult to target specific features based on the training results when tuning the model, so it is difficult to evaluate the importance of each feature.

Convolution is the use of filters to learn or detect features in an image. The convolution equation is as follows, where S represents the result of the operation, I is the original image, K is the convolution kernel, m and n are the height and width of the convolution kernel, and the two values in parentheses represent the positions of the elements.

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n)$$

The TextCNN model[4]solves the task of sentence-level text classification by treating a sentence as a feature matrix composed of multiple word vectors and convolving it in one dimension using convolutional kernels of different sizes, and then extracting the most important features in each feature map through a maximum pooling layer. The structure of TextCNN is relatively simple, where the input data is first passed through an embedding layer to obtain an embedding representation of the input utterance[5], then through a convolution layer to extract the features of the utterance, and finally through a fully connected layer to obtain the final output. The structure of the whole model is shown in Figure 1.

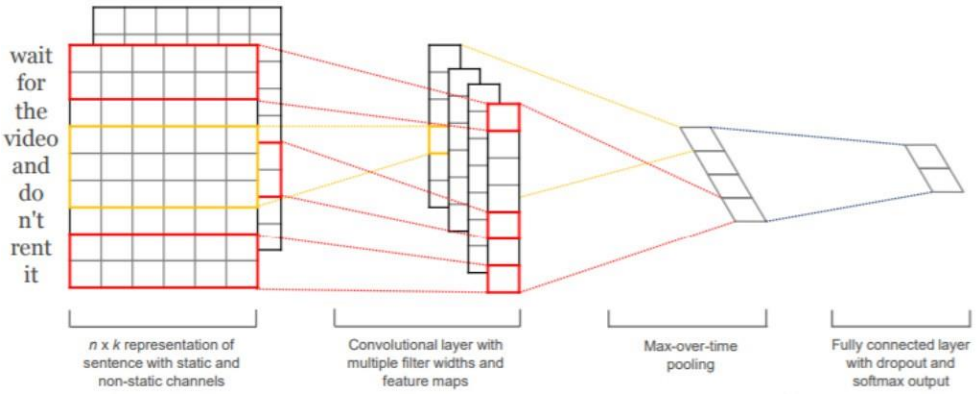

Figure 1. Structure of TextCNN network

Convolutional neural network (CNN) models were originally designed for computer vision and were later shown by Meek [6] to be useful for natural language processing with good results in semantic analysis. Since then, LeCun et al [7-8] proposed a character-level convolutional neural network model for different classification datasets for semantic analysis and topic classification tasks. However, the method is very slow to train and work for Chinese text classification because the term sets and word N-grams for Chinese text classification are much larger than those for English text classification. Moreover, the character-level feature processing gives up the semantic information that words have, and for Chinese, there are many overlapping semantics between words and characters, and this feature extraction approach is flawed.

2.2. Recurrent Neural Networks (RNN)

In some natural language processing tasks, when processing sequences, we generally use recurrent neural networks RNNs, and in particular some of their variants such as LSTM (more commonly used), GRU, and of course we can also apply RNNs to text classification tasks. Here the text can be a sentence, a document (short text, several sentences) or a chapter (long text), so that each segment of text is of different length. When classifying text, we generally specify a fixed input sequence/text length: this can be the length of the longest text(sequences), at which point all other text/sequences are padded to reach that length, or it can be the average of the lengths of all the text(sequences) in the training set, at which point the text(sequences) that are too long are truncated and the text that is too short is padded. In summary, to make all text(sequences) in the training set the same length, this length can be any other reasonable value in addition to the previously mentioned settings. The same needs to be done for the text(sequences) in the test set during testing.

RNN-based text classification models are very flexible and come in a variety of structures. In the following, this paper focuses on two typical structures.

2.2.1. First Class TextRNN

Then the mean value of the stitched hidden states over all time steps is taken and a SoftMax layer (with a SoftMax activation function in the output layer) is used to perform a multi- classification (sigmoid activation function for 2-classification). As shown in Figure 2.

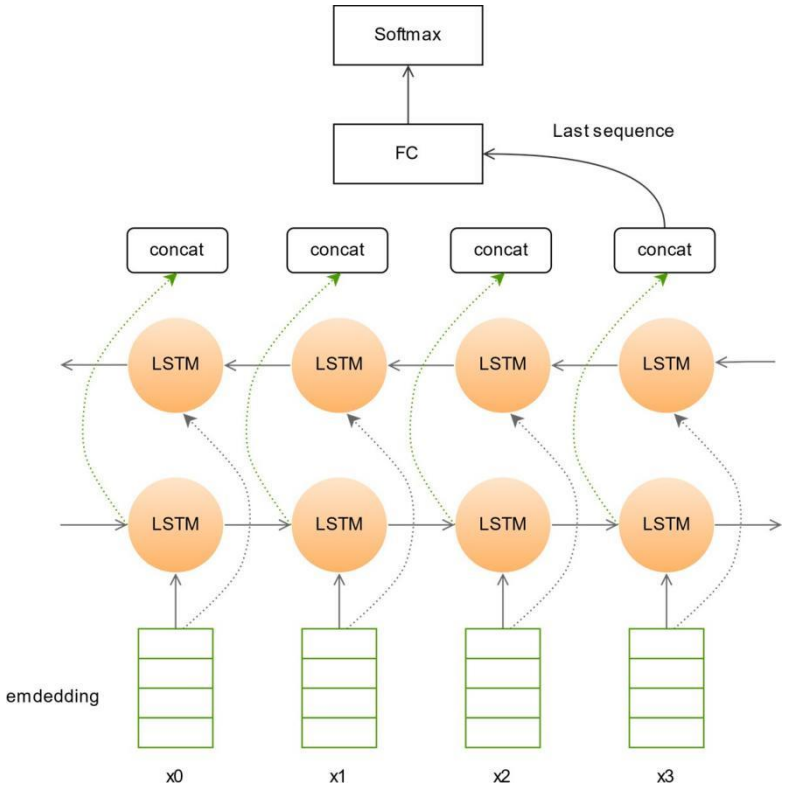

Figure 2. Flow chart of the first class of TextRNN classification

Dropout/L2 regularisation or BatchNormalization can also be added to the above structures to prevent overfitting and to speed up model training.

2.2.2. The Second Type of TextRNN

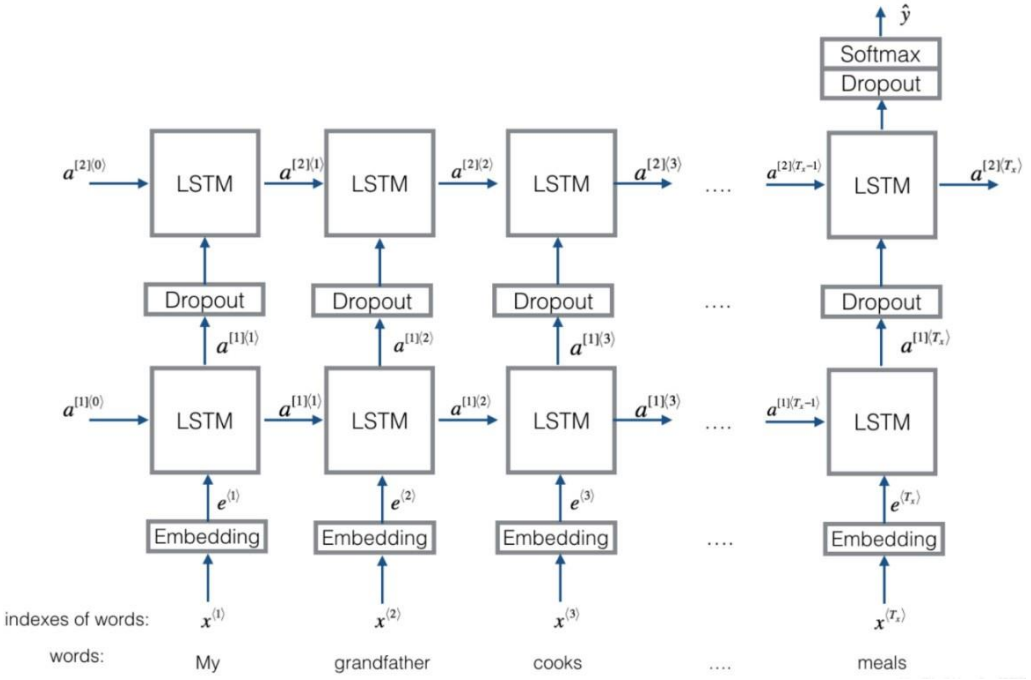

Figure 3. Flow chart of the second class TextRNN classification

Unlike the previous structure, a unidirectional LSTM is stacked on top of the bidirectional LSTM (the above figure is not quite accurate, the underlying layer should be a bidirectional LSTM). two hidden states of the bidirectional LSTM at each time step are stitched together and used as an input to the upper unidirectional LSTM at each time step, and finally the last time step of theupper unidirectional LSTM is taken as the last hidden state at the last time step of the upper one-way LSTM is then taken and passed through a softmax layer (softamx activation function for the output layer, or sigmoid for binary classification) to perform a multi-classification.

The structure of the TextRNN is very flexible and can be changed at will. TextRNN is very goodat text classification tasks and is comparable to TextCNN, but the training speed of RNN is relatively slow and two layers are usually enough.

2.3. Graphical Convolutional Neural Networks (GCNN)

Graph Convolutional Neural Networks, as the name suggests, are inspired by CNN, a convolutional neural network in deep learning, which has become a powerful tool in the field of image processing by using convolutional kernels with shared parameters to convolve the perceptual field for feature extraction, with two main features: local connectivity and weight sharing. Inspired by the application of convolutional networks in computer vision, researchers have started to investigate how to build convolutional operators on graphs for graph embedding.

The graph convolutional nerve network (GCN) is a model evolved from the spectral convolutional nerve network (Spectral CNN) and the Chebyshev network (ChebNet). Spectral CNNs [9] were the first generation of convolutional neural networks to be applied to graph data. The model is based on spectral graph theory [10] and graph signal processing [11], and implements the convolution operation of node information and convolution kernel in the spectral domain to obtain node embeddings according to the convolution theorem [12]. Assuming that there are N nodes in the graph g, the graph convolution is defined as the product of the node features $x \in R$ Nand the convolution kernelg_θ (convolution kernel parameter $\theta \in R$ N) in the Fourier domain, as shown in equation .

$$g_\theta \star x = U g_\theta(\Lambda) U^T x$$

where $\star$ is the convolution operation, U is the eigenvector matrix of the normalised Laplacian matrixL, and fl is the diagonal matrix of its eigenvalues. The convolution kernelg_θ can be interpreted as a function of the matrix Λ, i.e., g_θ (fl). However, this model is computationally expensive and does not have the property of local connectivity. Therefore, Defferrard et al [13] proposed the Chebyshev network. The principle is based on the first generation, using the K- order truncated expansion of Chebyshev polynomials to fit the convolution kernelg_θ (fl)whose graph convolution is shown in equation :

$$g_{\theta'} \star x \approx \sum_{k=0}^{k} \theta'_k T_k(\tilde{L}) x$$

Among them, $\tilde{L} = \frac{2}{\lambda_{max}} L - I \in [-1,1]$, λ_{max} denotes the largest eigenvalue of L; $\theta' \in R^{k+1}$ is the vector of Chebyshev coefficients, and the Chebyshev polynomial is defined as $T_k(x) = 2x * T_{k-1}(x) - T_{k-2}(x)$, where $T_0(x) = 1$, $T_1(x) = x$ Compared to Spectral CNN, this model avoids the feature decomposition of the Laplacian graph operator L, thus reducing the computational cost and difficulty; also, since the operation is a K-order polynomial on the Laplacian matrix, the result of the graph convolution depends only on the distance to the target node in steps of no more than all nodes of order K, which is sufficient to show that ChebNet has good K-local connectivity.

In general, a graph consists of a number of nodes and edges, with the nodes representing real-world entities and the edges representing the relationships between entities. A graph convolutional neural network updates the feature representation of the current node by aggregating information from neighbouring nodes through the connection of edges between nodes. Although the network can extract information about nodes and graphs using convolutional operations, it still has some limitations. In order to further improve the generalisation capability of the network, many variants have emerged, such as the Graph Attention Network (GAT), Graph Gated Neural Network (GGNN), Graph Transformer Network (GTN), etc, GTN), etc.

Unlike images, text, etc., where non-gridded structured data often limits the expressive power of traditional neural networks, the emergence of GCN models and their variants has greatly advanced the field of natural language processing, especially for text classification tasks. This section focuses on the development and improvement of graph convolutional neural networks for text classification tasks in terms of different types of graph models. the Textgcn model [13] is the first application of graph convolutional networks to text classification tasks, where both words and text are considered as nodes and an undirected weighted heterogeneous graph is constructed for the whole corpus, and both word embeddings and text embeddings are learned.The relationship between word nodes in this model [14] depends on word co-occurrence information, and the edge weights are calculated by the Pointwise Mutual Information (PMI) algorithm. The edge weights between text and word nodes are defined as TF-IDF values, which

are used to evaluate the importance of words in the text. The adjacency matrix $\tilde{A}$ (with self- loop) and PMI of the model are calculated as follows .

$$\tilde{A}_{ji} = \tilde{A}_{ij} = \begin{cases} PMI(i,j) & i,j \text{ are words}, PMI(i,j) > 0 \\ TFIDF_{ij} & i \text{ is a word}, j \text{ is a doc} \\ 1 & j = i \\ 0 & otherwise \end{cases}$$

where $\tilde{A}$ is a symmetric matrix, P(i)denotes the probability of word i occurring in the corpus, and P(i, j) denotes the probability of word i and word j occurring in the corpus at the same time. As shown in Fig. 1, the input graph contains two types of nodes, word (Word) and text(Doc), and the Textgcn model acquires the final representation of the nodes, R(·), through the learning of a two-layer GCN model [15] to achieve the multi-classification task. Compared withtraditional approaches (e.g. TextRNN), this model can capture global semantic information. In addition to word co-occurrence relations, the TensorGCN model [16] also introduces Long Short Time Memory (LSTM) and syntactic dependencies between words for expressing semantic and syntactic relations between words. The model constructs three types of heterogeneous graphs based on these three relationships, and uses both intra- and inter-graph information propagation for information aggregation of nodes in a single graph and coordination of heterogeneous information between graphs, respectively. The SGC model [17], on the other hand, proposes to eliminate the activation operations between hidden layers and convert the intermediate processes into simple linear transformations to reduce the complexityand redundant computations of the model.

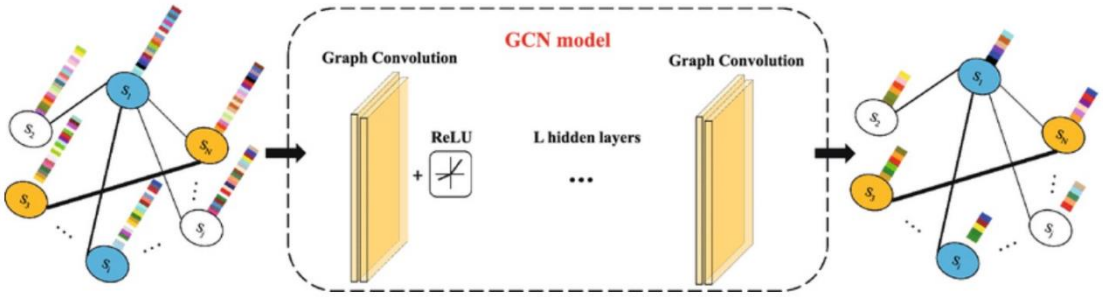

Figure 4. Flow diagram of the Text GCN model

However, existing models construct input graphs for the whole corpus, ignoring word interactions within the text. Therefore, Huang et al [18] proposed a Text-level GCN model, which constructs a directed graph for each text, and designs a globally shared node feature matrix and edge weight matrix. The Text-level GCN model can effectively solve the problem of high memory consumption and has good generalization ability to new samples.

2.4. Attentional Mechanisms

Attention mechanisms were first used in image processing and have since been introduced into natural language processing. When modelling question and answer pairs (AS) tasks, Attention mechanisms can be designed based on AS relationships. In contrast to text classification tasks, which are modelled as labels, attention mechanisms are introduced to extract significant words to represent sentences and to aggregate these representations of information words to form sentence vectors [19]. Similarly, to reward cued sentences for correctly classifying documents, the attention mechanism can be used again to measure the importance of sentences to obtain a document vector from which to classify documents.Although the self-attentive mechanism performs well in text classification tasks, however, due to the different training positions of the input text vectors during the training of the traditional self-attentive mechanism, the text vectors in the top positions suffer from a smaller initial observation window and lower information content during the training process.Therefore, there are limitations in directly sequentially computing the attention scores for the input text vectors. the Transformer structure proposed by Vaswani [20] et al.

in 2017, due to the fully attentional mechanism approach to feature extraction, the model proves to have a better ability to capture long-range dependencies compared to LSTM, and is able to more adequately model the information in sentences. The model can model the information in the sentence more adequately.

As an example, Yang et al [21] proposed a Hierarchical Attention Networks for Document Classification (HAN). HAN). The HAN model is based on the fact that the structure of a document is formed by words forming sentences and sentences forming documents, so the modeling is also divided into these two parts. Since the information content between words and sentences is different and cannot be unified, the Attention mechanism is introduced to improve the accuracy of the model. Practice has shown that the introduction of the Attention mechanism not only improves the accuracy of the model, but also enables the analysis and visualisation of the importance of words and sentences, which helps to deepen our understanding of the internal implementation process of text classification.

3. Text Representation

Computers can be more difficult to understand when dealing with textual data than with images. Humans are able to understand the underlying meaning and hidden emotional tendencies when reading text data because of their own reading comprehension abilities, but computers do not have the same ability as humans. Therefore, effective representation of text is a fundamental and most important part of natural language processing tasks, and the inclusion of as much semantic information as possible in the representation of text can significantly improve the ability of computers to process text data. The first task in text classification is to vectorise the text to obtain a vector expression that can be processed by the computer, on the basis of which an effective and accurate text classifier can be constructed to complete the text classification task.

Text representation is the starting point for natural language processing tasks, in which words are processed through a series of operations to obtain a vector or matrix representation that can be processed by a computer. Depending on the processing method, this can be broken down into discrete and distributed representations. Bag-of-words models are a typical approach to discrete representations, and common bag-of-words models include one-hot coding (also called unique hot coding), TF-IDF, n-gram, etc. [22]. In the traditional use of machine learning methods for natural language processing tasks, one-hot encoding is often utilised as a text vectorization representation method, which first divides all text into words and then removes the recurring words to form a large word list. The first is data sparsity and dimensional catastrophe. When there are millions of words in a dictionary, it is obviously difficult for a computer to handle such a large dimension, and for shorter sentences, most of the elements in the vector are zero, so the data sparsity problem is also very obvious. Secondly, this encoding ignores the order of the words and assumes that the words are independent of each other, so it is likely that two sentences with completely different meanings will end up with the same vector after the one-hot encoding, and therefore will not meet the requirements at all. Finally, the encoding does not take into account the relative importance of the word, as it only deals with whether the word appears in the lexicon or not, and does not take into account the frequency of the word, which in general means that the more frequently a word appears in the text, the more important it should be (with the exception of stop words that have no real meaning).The TF-IDF method has been developed to better assess the importance of words in a document. The main principle of this method is to proportion the importance of a word to the number of times it appears in a document and inversely to the number of times it appears in all documents. Although this method is an improvement over one-hot coding, the overall problem of data sparsity should not be underestimated, and the information before and after words is not sufficiently considered [23]. In order to take into account the key factor of word order, the concept of n-gram was subsequently introduced, where n consecutive words are extracted from a sentence to obtain the information before and after the word.

With the n-gram representation, it is possible to obtain rich features and extract pre/post information. However, as n increases, the dimensionality of the word list becomes successively larger, and there are problems of data sparsity and high word list dimensionality, so it is not yet common to use this method in practice.

The use of deep learning techniques to obtain text representations has become an inevitable choice due to the continuous progress and development of deep learning. 1986 Hinton [24] et al. first used a distributed approach to text representation, proposing a method that uses deep neural networks to learn to obtain text vector representations. The main difference from one- hot is that text vectorisation represents words as a continuous, dense vector of real numbers, which has the advantage that the vector representation contains richer semantic features. In 2013, Mikolov [25] et al. of Google proposed an open source word vector tool, Word2Vec, based on their work on feed-forward neural network language models, which further optimised the network structure. In 2014, Pennington [26] et al. proposed a new word embedding method, GloVe, which is based on the statistical information of global lexical co-occurrences to learn word vectors, and which combines the advantages of statistical information and local context windows methods to In 2016 Facebook open-sourced a new text vector representation, FastText. While previous word vector models process words to obtain a separate vector for each word, which does not make good use of the internal structural features of words, FastText takes into account the morphological composition of words, by adding sub-word information so that some low frequency words can be represented as word vectors through sub-words, greatly enriching the semantic information. The BERT model proposed by Google team Devlin
[27] et al. in 2018, deeply increases the generalization capability of the word vector model, and also fully describes character-level, word-level, sentence-level and even inter-sentence relationship features. In 2019, Yang et al. proposed the XLNet [28] model, which uses a log- likelihood approach that maximises all possible factorisation orders to learn bidirectional contextual information, while also exploiting the features of autoregression itself to The model uses a log-likelihood approach that maximises the order of all possible factorisations for the purpose of learning bidirectional contextual information, while also exploiting the features of autoregression itself to further overcome the shortcomings of BERT and finally incorporating ideas from Transformer-XL. The model proved to be better at representing words and performed well in NLP downstream tasks.

4. Conclusion

Text classification has become a hot topic of research in the field of natural language processing in the face of the huge amount of textual information available on the Internet. Although machine learning and traditional neural networks have made great progress in this task, there are still drawbacks. In contrast to machine learning, which is computationally expensive to extract features, and traditional neural networks, which can only extract local word sequence information, text classification methods based on graph convolutional neural networks have certain advantages, as they can not only map complex semantic relationships in text, but also capture global graph information. In general, deep learning methods have received a lot of attention in the field of text classification. In this paper, we look at the future research directions from five aspects: improving existing deep learning models, fine-tuning language models, proposing migration learning, relying on the Transformer architecture and model interpretability issues, which are believed to be of great significance for the deeper application of the models.

References

[1] Lin Yiyang. The importance of natural language processing technology [J]. Digital Communication World, 2018, 158(02):186-187.

[2] Wang Kui, Liu Bosong. A Review of Text Classification Research [J]. Data Communication, 2019, 11(03): 37-47.

[3] Kim Y. Convolutional neural networks for sentence classification[C]//Proceedings of the 2014 Conference on Empir.

[4] Ding Zeyuan. Chinese Biomedical Text Information Extraction Based on Deep Learning [D]. Dalian: Dalian University of Technology, 2020.

[5] KIM Y. Convolutional neural networks for sentence classification[C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. 2014: 1746-1751.

[6] ZHANG X, ZHAO J, LECUN Y. Character-level convolutional networks for text classification[C]// Advances in Neural Information Processing Systems. 2015: 649-657.

[7] LIU P, QIU X, HUANG X. Recurrent Neural Network for Text Classification with Multi-Task Learning[C]// Proceedings of the 25th International Joint Conference on Artificial Intelligence. 2016: 2873– 2879.

[8] YANG Z, YANG D, DYER C, et al. Hierarchical Attention Networks for Document Classification[C]// Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016: 1480–1489.

[9] BRUNA J, ZAREMBA W, SZLAM A, et al. Spectral networks and locally connected networks on graphs [J]. Proceedings of the International Conference on Learning Representations. arXiv: 1312.6203, 2013.

[10] CHUNG FRK. Spectral Graph Theory[EB/OL]. http://www.ams.org/books/cbms/092/.

[11] SANDRYHAILA A, MOURA J.M.F. Discrete Signal Processing on Graphs[J]. IEEE Trans. On Signal Processing, 2014, 62(12): 3042-3054.

[12] SHUMAN D I, NARANG S K, FROSSARD P, et al. The EmergingField of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains[J]. IEEE Signal Processing Magazine, 2013, 30(3):83-98.

[13] DEFFERRARD M, BRESSON X, VANDERGHEYNST P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering[C]// Advances in Neural Information Processing Systems. 2016: 3844-3852.

[14] YAO L, MAO C, LUO Y. Graph Convolutional Networks for Text Classification[C]// Proceedings of the 33rd AAAI Conference on Artificial Intelligence. 2019: 7370–7377.

[15] KIPF T N, WELLING M. Semi-Supervised Classification with Graph Convolutional Networks[C]// Proceedings of the 5th International Conference on Learning Representations. 2017: 1-13.

[16] LIU X, YOU X, ZHANG X, et al. Tensor graph convolutional networks for text classification[C]// Proceedings of the 34th AAAI Conference on Artificial Intelligence. 2020: 8409-8416

[17] Ma, Y., Sun, D., Gao, E., Sang, N., Li, I., & Huang, G. (2024). Enhancing Deep Learning with Optimized Gradient Descent: Bridging Numerical Methods and Neural Network Training. arXiv preprint arXiv:2409.04707.

[18] Choi K, Fazekas G, Sandler M, et al. Convolutional recurrent neural networks for music classification[C]. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017: 2392-2396.

[19] Yang Z,Yang D,Dyer C,et al. Hierarchical attention net-works for document classification [C]// Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. San Diego,USA: NAACL,2016: 1480-1489

[20] Agnihotri D, Verma K, Tripathi P. An automatic classification of text documents based oncorrelative association of words [J]. Journal of Intelligent Information Systems, 2018, 50(3): 549-572.

[21] Feng G, Li S, Sun T. A probabilistic model derived term weighting scheme for text classification [J]. Pattern Recognition Letters, 2018, 110(15): 23-29.

[22] Mohasseb A, Bader-El-Den M, Cocea M. Question categorization and classification using grammar based approach [J]. Information Processing and Management,2018,54(6):1228-1243.

[23] Yang Z,Yang D,Dyer C,et al. Hierarchical attention networks for document classification [C]// Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. San Diego,USA: NAACL,2016: 1480-1489.

[24] Xu K, Ba J, Kiros R, et al. Show, attend and tell: Neural image caption generation with visual attention[C]. International conference on machine learning. 2015:2048-2057.

[25] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]. Advances in neural information processing systems. 2017: 5998-6008.

[26] Hou Hanqing, Huang Gang. Computer and Document Classification[J]. Modern Library and Information Technology, 1982,2(1):5-14.

[27] Choi K, Fazekas G, Sandler M, et al. Convolutional recurrent neural networks for music classification[C]. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017: 2392-2396.

[28] Mnih V, Heess N, Graves A. Recurrent models of visual attention[C]. Advances in neural information processing systems. 2014: 2204-2212.