Overview of Deep Learning Methods for Sentiment Analysis

Yifei Zhao[1,*]

[1]College of Electronic and Information Engineering, Tongji University, Shanghai, China

*Corresponding author e-mail: zhaoyifei6543@gmail.com

Abstract. With the development and improvement of Web 2.0, user-driven Internet products are rapidly increasing, generating huge amounts of data, especially text data. Sentiment analysis can extract and analyse sentiment tendencies from text data, and is one of the most valuable research directions in natural language processing. In recent years, deep learning has been widely applied to sentiment analysis tasks, using advanced model architectures to achieve better results than ever before. In this paper, we review various deep learning methods and their extensions for sentiment analysis tasks in recent years synthetically, especially the Transformer models and Pre-trained models. In addition, the advantages and disadvantages of these models as well as the limitations of their use are illustrated. Finally, the paper summarized the main challenges of the current sentiment analysis tasks and possible future research directions.

Keywords: Emotional Analysis, Deep Learning, Web 2.0.

1. Introduction

Natural language processing (NLP) is an important research direction in the field of artificial intelligence and computer science. As Web 2.0 comes, a large number of users have generated large amounts of text information through various Internet products. Processing and analyzing text information have become an increasing demand, and sentiment analysis has received a lot of attention and research as a branch of NLP. The purpose of sentiment analysis is to automatically mine information in the text, such as emotional orientations and standpoints. Sentiment analysis has a wide application background. By analyzing the comments of the public under social software news topics, the government can better investigate public opinion. Sentiment analysis can also count the overall customer evaluation and promote merchants to make adjustments toward customer feedback, which has great business value.

Sentiment analysis consists of five main elements: entity, aspect, sentiment, holder and time, where entity and attribute are together called evaluation objects [4]. Usually emotions can be divided into different categories, like positive and negative emotions.

The tasks of sentiment analysis can be divided into three categories: document-level, sentence-level and aspect-level [4, 5]. Among them, aspect-based sentiment analysis (ABSA) is the focus and difficulty of the sentiment analysis task because it needs to analyze different sentiments among different attributes in a sentence. For example, in the sentence "the picture is good but the soundtrack is bad", the positive sentiment should be extracted from the "picture" aspect, but the negative sentiment should be extracted from the "soundtrack" aspect.

Early research on sentiment analysis used lexicon and corpus-based methods. Then machine learning approaches appeared to classify words into corresponding sentiment label. With the rise of deep learning research, deep learning methods are widely used in sentiment analysis. Early deep learning methods include convolutional neural networks (CNN) [14], recurrent neural networks (RNN) [18] and recursive neural networks (ReNN) [22]. Recent years have seen the emergence of methods based on pre-trained models and Transformer, such as BERT [1], XLNet [28] and GPT [32], etc.

The overall framework of the article is as follows: Part 2 collates traditional sentiment analysis methods based on lexicon and machine learning. Part 3 collates sentiment analysis methods based on deep learning. Part 4 discusses the challenges and possible development directions of sentiment analysis tasks.

2. Traditional Methods

2.1 Lexicon-based

The lexicon-based approach was the first traditional sentiment analysis method to be proposed and was first used in 1962 [6]. A pre-prepared lexicon contains the tendency weights of various sentiment words, and the analysis extracts the sentiment words from the text, calculates the sum of weights, and gets the overall sentiment score of the text. Later, different sentiment lexicons were developed, and the well-known ones include WordNet, SenticNet, MPQA [7], etc. Chinese sentiment analysis lexicons include NTUSD, HOWNET, etc. The lexicon-based methods are unsupervised and easy to understand, and the accuracy is high when the text has a high coverage of sentiment words. However, large numbers of special words on the Internet greatly increases the difficulty of lexicon construction. It excessively relies on the quality of lexicons, and perform poorly in identifying implicit sentiment and irony.

2.2 Machine Learning

The machine learning approach uses a supervised training model to classify text sentiment. In the training process, a labeled corpus is provided as training data, and then the classifier uses vectorized text words for classification. Common approaches include naive Bayesian model, maximum entropy model, and support vector machine (SVM) [5].

2.2.1. Naive Bayesian Model.

Naive Bayesian model is a classification method based on Bayes' theorem and the assumption of conditional independence of features. It calculates the probability of classification by features and selects the cases with high probability, which is a machine learning method based on probability theory, and is widely used in the field of sentiment classification. Let the text vector be $w = (w_0, w_1, \dots, w_n)$ and the target sentiment be c_i, then the naive Bayes theorem is expressed as:

$$P(c_i|w) = \frac{P(w|c_i)P(c_i)}{P(w)} = \frac{P(w_0|c_i)P(w_1|c_i)\cdots P(w_n|c_i)P(c_i)}{P(w)} \quad (1)$$

Use the above formula to find out the probability that the text belongs to each emotion, and select the one with the highest probability as the final emotion classification result.

2.2.2. Maximum Entropy Model.

According to the maximum entropy principle, among all possible probabilistic models, the one with the highest entropy is the best probabilistic model. The process of solving the maximum entropy model can be transformed into a constrained optimization problem. Let the text vector be x and the set of sentiment labels be $l = (l_1, l_2, \dots, l_n)$, then the maximum entropy model is expressed as:

$$P(l|x) = \frac{1}{Z(x)} exp\left(\sum_{i=1}^{n} \lambda_i f_i(l, x)\right) \quad (2)$$

where $Z(x)$ is the normalization constant, the feature function f_i indicates whether the sentiment label l_i, and λ_i is the model parameter, which is solved by maximizing the likelihood function.

2.2.3. Support Vector Machine.

The essence of SVM is to find the best hyperplane in high dimension to minimize the classification error rate of different data categories. In order to maximize the distinction between different categories, the hyperplane should be selected at a position in the middle of the classification interval. Let the hyperplane equation be $w^T x + b = 0$, x_i be the text vector and y_i be the sentiment label, then the optimization objective is:

$$\min_{w,b} \frac{||w||^2}{2}$$
$$s.t.\ (w^T x_i + b) \geq 1 \quad (3)$$

The optimization objective can be solved using the Lagrange multiplier method.

Compared with lexicon-based methods, machine learning do not require manual construction of lexicons, thus avoiding subjectivity, and can handle different types of features with wider applicability. However, machine learning has limited ability to handle large scale data. With the rise of deep learning research, deep learning methods are gradually being applied to sentiment analysis tasks instead of traditional machine learning.

3. Deep Learning

This chapter is divided into two main parts to sort out the application of deep learning in sentiment analysis: traditional deep learning methods and pre-trained model methods.

3.1 Traditional Deep Learning Models

3.1.1. Word2Vec.

Deep learning models need to transform textual information into input features. This transformation process is called word embedding. Word embedding usually transforms a high-dimensional sparse word vector into a low-dimensional dense embedding vector, and the dimensions of the embedding vector represents the potential semantic features of a word. A widely used word embedding method is Word2Vec [8]. It consists of two models: continuous bag-of-words model (CBOW) and continuous Skip-gram model. CBOW uses the contextual words to predict the current word, and Skip-gram uses the current word to predict the contextual words. Word2Vec is a shallow neural network with a single hidden layer. Typically, a one-hot encoding of words is used as the input of the network, and the weight matrix between the input and hidden layers is the embedding vector to be learned, as Figure 1 shows.

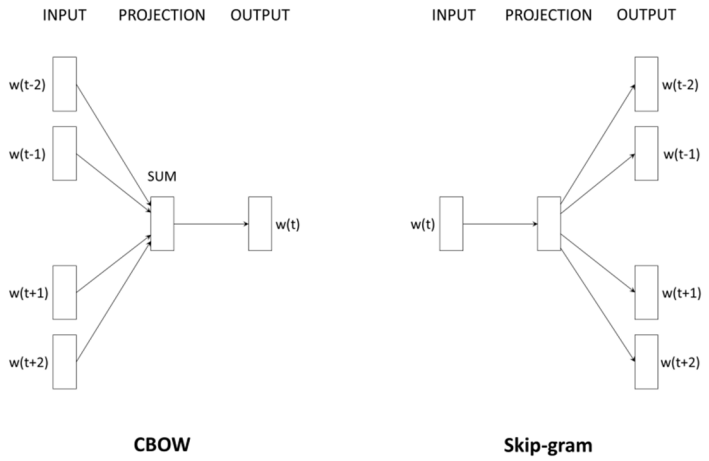

Figure 1. CBOW predicts the current word from the given contextual words, and Skip-gram predicts the contextual words from the given current word.

Other word embedding methods extend Word2Vec, such as GloVe [10] and FastText [11], etc. GloVe learns semantic relationships between derived words in a co-occurrence probabilities matrix. FastText does not consider each word as a whole, but as a N-gram made up of characters. In addition, RNN-based and Transformers-based word embedding methods can achieve contextual relevance, such as ELMo [12], CoVe [13], and BERT [1]. Among them, ELMo uses a character-based encoding layer and two BiLSTM layers to learn the contextual representation, while CoVe uses a deep LSTM encoder.

3.1.2. Convolutional Neural Network.

CNN is a special feedforward neural network using convolutional operations. Its design is inspired by animal vision mechanisms: Visual cells have their own perceptual fields, which have similar effects to filters. CNN has input layer, feature processing layer and output layer, where the feature processing layer is divided into a convolutional layer and a pooling layer. The convolution layer uses a filter matrix to slide-scan the input of the layer, and the numbers of the filter matrix are used as weights to multiply with the input values at the corresponding positions to extract local features. The pooling layer samples the convolution results and reduces the feature dimension.

CNNs were initially applied in the field of computer vision (CV). In 2008, CNNs were used on NLP tasks [17]. In 2014, a CNN-based approach was first used to solve the problem of sentiment analysis [2]. The input is a three-dimensional matrix in which different embedding results constitute multiple channels. The filter uses one-dimensional convolution to extract features for two to three words at a time. After convolution, the most significant features are selected as features of the text using max-1-pooling. Finally, a fully connected layer and a softmax layer are passed to obtain the classification result output, as Figure 2 shows.

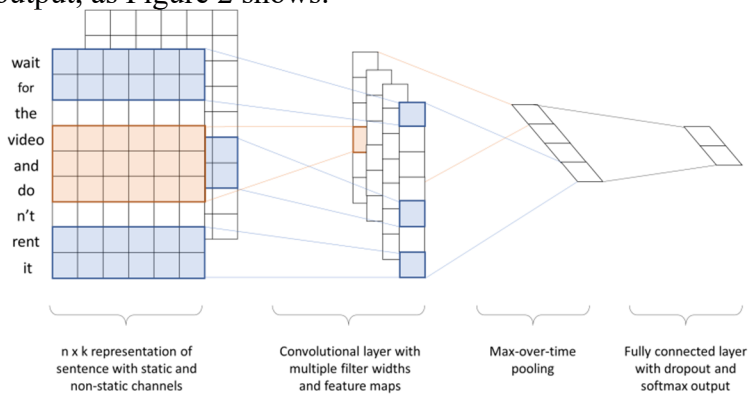

Figure 2. CNN in sentiment analysis.

3.1.3. Recurrent Neural Network.

RNN is a feedforward neural network with directed recurrence which has the ability to model sequences. Corresponding to a time series, the state value of the hidden layer at the tth moment is expressed as:

$$h_t = \tanh(Wx_t + Uh_{t-1} + b) \quad (4)$$

where x_t is the tth input at the tth moment, and h_{t-1} is the state value of the hidden layer at the previous t-1th moment.

Since texts are sequences and contain contextual associations between sequences, RNNs have natural applicability to NLP problems. When dealing with NLP problems, each word vector is used as an input at each moment.

However, when the sequence length is too long, RNNs can generate gradient disappearance and gradient explosion problems [20]. To solve the gradient problem, the long short-term memory network(LSTM) was proposed [19]. Compared with the simple RNN, LSTM adds memory unit c, input gate i, forget gate f, and output gate o, as Figure 3 shows.

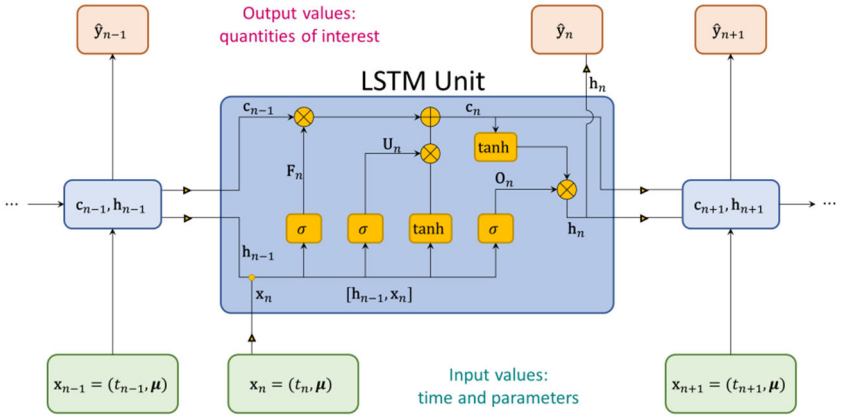

Figure 3. LSTM model architecture

The gates and the memory units of LSTM greatly enhance RNN's ability to process long sequence data. Gated Recurrent Unit (GRU) [21] is a simplified version of the LSTM that combines input and forgetting gates into update gates.

3.1.4. Recursive Neural Network.

ReNN is a generalized form of recurrent neural network that allows a bottom-up representation of phrases in the form of trees or graphs as model inputs, and are compatible with the recursive structure of sentences. The recursive neural tensor network (RNTN) [23] is a model designed for sentiment analysis based on ReNN. It allows any length of phrase as input, represents phrases by word vectors and parse trees, and then classifies sentiment by analyzing the order of word combinations using a tensor-based combination function, as Figure 4 shows.

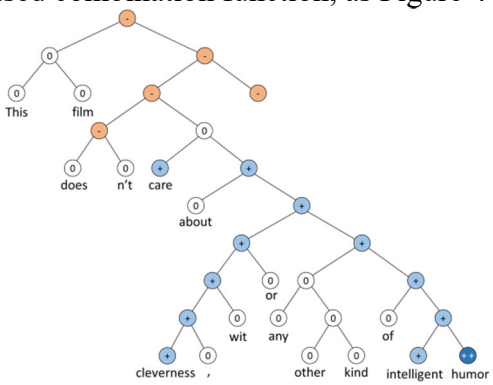

Figure 4. Recursive tree structure generated by RNTN. Blue colors indicates positive emotions, while red colors indicates negative emotions.

3.1.5. Transformer.

Transformer is a deep learning model which was initially used for machine translation sequence-to-sequence Seq2Seq tasks [24]. Due to its excellent performance, it is now widely used in various fields such as NLP and CV. Studies have shown that Transformer-based pre-trained models have the best performance on various tasks [25].Transformer consists of an encoder and a decoder, which are composed of multiple encoding and decoding layers, where a single encoding layer consists of a self-attention module and a feedforward neural network (FFN) , as Figure 5 shows.

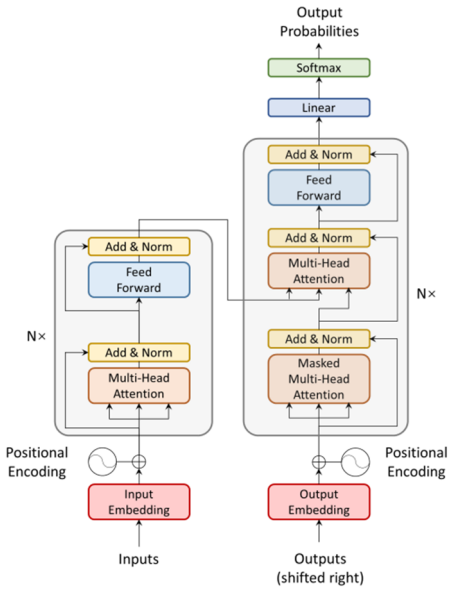

Figure 5. Transformer model structure. The left is the encoder which contains N encoding layers, and the right is the decoder which contains N decoding layers.

The self-attention module introduces contextual information to each input and has learnable weights. For each input, a query vector Q, a key vector K and a value vector V are generated by multiplying the word embedding vector with three weight matrices W^Q, W^K and W^V respectively. The dot product of the query vector and the key vector are dot multiplied to calculate the vector similarity, which indicates the relevance of the context to the location of that input. After the self-attention module, the word vector is represented as:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

where dividing by $\sqrt{d_k}$ is to prevent the gradient vanishing problem of the softmax function and to speed up the convergence. Residual connection is used after the attention module, and layer normalization is performed before FFN.

In Transformer, self-attention has been expanded to multi-head attention. It has the ability to focus on different contextual locations, while reducing the influence of the input vectors themselves. Multi-head attention has multiple sets of weight matrices. The input vectors are fed into multiple self-attentive layers, and the outputs are concatenated:

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\ldots,\text{head}_h)W^O$$
$$where \quad \text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{6}$$

Since Transformer does not have a circular structure like RNN, it does not have the natural ability to capture sequential sequences. Therefore, positional encoding needs to be introduced to represent the distance between two words when encoding word vectors. Main ways include sine-cosine absolute positional encoding [24] and learnable relative positional encoding [25].

3.2 Pre-trained Models

Model pre-training is a transfer learning method that uses a pre-trained model on a specific downstream target task. For sentiment analysis, the initial pre-trained model learns the meaning of the text by a self-supervised approach, and then uses the model on a sentiment analysis task by fine-tuning or prompting. Natural text languages have natural annotation features that enable context-based prediction, and thus can be self-supervised pre-trained, which achieved very good results on sentiment analysis tasks.

3.2.1. Unsupervised Pre-trained Networks.

UPN use unsupervised algorithms to pre-train the data, and then fine-tune them using supervised algorithms. A commonly used UPN in sentiment analysis is the autoencoder (AE) [14].

Autoencoders were proposed in 1987 and have better ability in downscaling and feature representation than PCA and LSA which can only represent linearly. AE is a three-layer neural network with the same target and input. It's aim is to find a mapping as the result of self-coding. Common variants of AE include the denoising autoencoder (DAE) [15] and the stacked denoising autoencoder (SDA) [31]. DAE can reconstruct the input in the presence of noise by not allowing the encoder to simply learn identity transformations. In sentiment analysis, the DAE is able to change the semantics of the document while removing or adding some text. SDA stacks multiple self-encoding layers together, increasing the depth of the network.

3.2.2. BERT.

Bidirectional Encoder Representations from Transformers (BERT) is a Transformer-based pre-training model. BERT uses the encoder of Transformer, and pre-train on two unsupervised tasks. Masked language model (MLM) and Next Sentence Prediction (NSP).

The goal of MLM is to understand the relationships between words by predicting the target words. It allows to train a deep bidirectional model that can focus on both the left and right side of the context. During pre-training, 15% of the words in a sentence are randomly selected to predict. 80% of them are replaced with masks, 10% are replaced with random words, and 10% remain unchanged. Unlike DAE, MLM does not reconstruct the entire input. the goal of NSP is to understand the relationship between sentences by predicting the next sentence. In the pre-trained dataset sentence pairs, 50% of the later sentences are the correct following sentence of the previous ones, and 50% of the later sentences are random.

Recently, many improved models based on BERT have emerged and reached better results, such as RoBERTa [3], ALBERT [27] and T5 [29].

RoBERTa uses a larger number of model parameters and more training data based on BERT, and uses Byte-Pair Encoding (BPE) to mix character-level and word-level representations. In addition, RoBERT does not use NSP, but takes continuous sentences as input. Dynamic masks are used for the MLM task, allowing the model to adapt to different masking strategies. ALBERT introduces a factorization of word embeddings, and uses sharing cross-layer parameter to reduce the number of model parameters. The sentence order prediction (SOP) is also proposed to replace the random replacement in NSP with exchanging the order of sentence pairs.

The T5 model introduced by Google uses the BERT-style small segment replacement method for pre-training, selecting 15% for replacement and 3 for small segment length. T5 converts all NLP tasks into Text-to-Text tasks uniformly, and the input and output are always strings. For example, an output in sentiment analysis may be either a "positive" or "negative" string, rather than a numerical value. This framework allows any NLP task to use the same model, loss function, and hyperparameters.

3.2.3. XLNet.

XLNet is another model that perform well on sentiment analysis tasks. It predicts the current word from left to right based on the above word, while a portion of the following words are placed in the position of the above words by permutation, avoiding the problem of not seeing the following words. XLNet proposes a two-stream self-attention mechanism that separates the position information of the word from the content information. The content representation is represented by h_{z_t}, which contains both the content information of the context and the target; the query representation is represented by g_{z_t}, which contains the content information of the context and the location information of the target, but without the content information of the target:

$$g_{z_t}^{(m)} = \text{Attention}\left(Q = g_{z_t}^{(m-1)}, KV = h_{z<t}^{(m-1)}; \theta\right),$$
$$h_{z_t}^{(m)} = \text{Attention}\left(Q = h_{z_t}^{(m-1)}, KV = h_{z\leq t}^{(m-1)}; \theta\right) \quad (7)$$

In addition, XLNet also combines the relative positional encodings and segment-level recurrence of Transformer-XL [30]. It solves the dependency problem of ultra-long sequences, and has an obvious performance improvement on reading comprehension tasks of long texts.

3.2.4. GPT.

GPT uses Transformer's Decoder structure and removes the multi-head attention, keeping only masked multi-head attention, making GPT a one-way language model, i.e., using the above to predict the current word. For positional encoding, GPT uses learnable positional encoding.

Based on GPT, GPT2 [33] moves the layer normalization of each decoder layer to the input side, and adds an additional layer normalization after the final self-attention. It initializes the residual layer parameters according to the depth of the network, and expands the model parameters and data set. GPT3 [34] further expands the model, discards fine-tuning, and proposes task-agnostic meta learning to enhance model generalization.

4. Discussions

More and more research has made progress on sentiment analysis tasks. However, for deep learning models, significant training costs are still required to achieve high accuracy rates. Moreover, dealing with affective phenomena in text, such as subjectivity, aspects, attitudes and feelings, has proven to be a complex interdisciplinary problem. Identifying irony, ambiguous expressions and implied emotions are challenging tasks for NLP. Currently, aspect-based sentiment analysis is still the focus and difficulty of research, and there is an increasing demand for multimodal and real-time sentiment analysis. Viewpoint mining is one of the most challenging development directions of sentiment analysis techniques, which raises higher requirements on discovering and extracting specific views of text.

References

[1] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[2] Chen Y. Convolutional neural network for sentence classification[D]. University of Waterloo, 2015.

[3] Liu Y, Ott M, Goyal N, et al. Roberta: A robustly optimized bert pretraining approach[J]. arXiv preprint arXiv:1907.11692, 2019.

[4] Liu B. Sentiment analysis: Mining opinions, sentiments, and emotions[M]. Cambridge university press, 2020.

[5] Pang B, Lee L, Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques[J]. arXiv preprint cs/0205070, 2002.

[6] Stone P J, Bales R F, Namenwirth J Z, et al. The general inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information[J]. Behavioral Science, 1962, 7(4): 484.

[7] Taboada M, Brooke J, Tofiloski M, et al. Lexicon-based methods for sentiment analysis[J]. Computational linguistics, 2011, 37(2): 267-307.

[8] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[9] Rong X. word2vec parameter learning explained[J]. arXiv preprint arXiv:1411.2738, 2014.

[10] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation[C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.

[11] Joulin A, Grave E, Bojanowski P, et al. Bag of tricks for efficient text classification[J]. arXiv preprint arXiv:1607.01759, 2016.

[12] Ilić S, Marrese-Taylor E, Balazs J A, et al. Deep contextualized word representations for detecting sarcasm and irony[J]. arXiv preprint arXiv:1809.09795, 2018.

[13] McCann B, Bradbury J, Xiong C, et al. Learned in translation: Contextualized word vectors[J]. Advances in neural information processing systems, 2017, 30.

[14] Ballard D H. Modular learning in neural networks[C]//Aaai. 1987, 647: 279-284.

[15] Vincent P, Larochelle H, Bengio Y, et al. Extracting and composing robust features with denoising autoencoders[C]//Proceedings of the 25th international conference on Machine learning. 2008: 1096-1103.

[16] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[17] Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning[C]//Proceedings of the 25th international conference on Machine learning. 2008: 160-167.

[18] Elman J L. Finding structure in time[J]. Cognitive science, 1990, 14(2): 179-211.

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

[20] Hochreiter S. The vanishing gradient problem during learning recurrent neural nets and problem solutions[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 1998, 6(02): 107-116.

[21] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 2014.

[22] Goller C, Kuchler A. Learning task-dependent distributed representations by backpropagation through structure[C]//Proceedings of International Conference on Neural Networks (ICNN'96). IEEE, 1996, 1: 347-352.

[23] Socher R, Perelygin A, Wu J, et al. Recursive deep models for semantic compositionality over a sentiment treebank[C]//Proceedings of the 2013 conference on empirical methods in natural language processing. 2013: 1631-1642.

[24] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[25] Qiu X, Sun T, Xu Y, et al. Pre-trained models for natural language processing: A survey[J]. Science China Technological Sciences, 2020, 63(10): 1872-1897.

[26] Shaw P, Uszkoreit J, Vaswani A. Self-attention with relative position representations[J]. arXiv preprint arXiv:1803.02155, 2018.

[27] Lan Z, Chen M, Goodman S, et al. Albert: A lite bert for self-supervised learning of language representations[J]. arXiv preprint arXiv:1909.11942, 2019.

[28] Yang Z, Dai Z, Yang Y, et al. Xlnet: Generalized autoregressive pretraining for language understanding[J]. Advances in neural information processing systems, 2019, 32.

[29] Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer[J]. The Journal of Machine Learning Research, 2020, 21(1): 5485-5551.

[30] Dai Z, Yang Z, Yang Y, et al. Transformer-xl: Attentive language models beyond a fixed-length context[J]. arXiv preprint arXiv:1901.02860, 2019.

[31] Vincent P, Larochelle H, Bengio Y, et al. Extracting and composing robust features with denoising autoencoders[C]//Proceedings of the 25th international conference on Machine learning. 2008: 1096-1103.

[32] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

[33] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

[34] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.