Postprint of Text Sentiment Analysis Based on Recurrent Neural Network and Attention Model

Authors: Ronglei Hu, Rui Lu, Qi Xiao, Zhang Xinran

Date: 2018-08-13T00:00:00+00:00

Abstract

Text sentiment analysis is a major branch in the field of natural language processing. Its development not only has a significant impact on the natural language processing domain, but also has profound influence in fields such as politics, economics, and social sciences that are greatly affected by people' s subjective viewpoints. The role of deep learning in text sentiment analysis is becoming increasingly important. A text sentiment analysis scheme is proposed by combining the Long Short-Term Memory network from recurrent neural networks with a feed-forward attention model. A feed-forward attention model is incorporated into the basic Long Short-Term Memory network, and the scheme is implemented under the TensorFlow deep learning framework. According to evaluation metrics such as accuracy, recall, and F1 measure, comparison with existing schemes demonstrates that the proposed scheme has significant advantages over traditional machine learning methods and pure Long Short-Term Memory network approaches.

Full Text

Preamble

Text Sentiment Analysis Based on Recurrent Neural Networks and Attention Model

Hu Ronglei, Rui Lu, Qi Xiao, Zhang Xinran
(Department of Electronic & Communication Engineering, Beijing Electronic Science and Technology Institute, Beijing 100070, China)

Abstract: Text sentiment analysis represents a major branch in natural language processing, with developments that exert profound influence not only within the NLP field itself but also in domains heavily affected by subjective viewpoints, such as politics, economics, and social sciences. Deep learning plays an increasingly important role in text sentiment analysis. This paper proposes a text sentiment analysis scheme that combines Long Short-Term Memory (LSTM) networks from the recurrent neural network family with a feedforward attention model. The feedforward attention model is integrated into the basic LSTM network, and the proposed scheme is implemented under the TensorFlow deep learning framework. Evaluation metrics including precision, recall, and F1-measure demonstrate that our proposed scheme offers significant advantages over both traditional machine learning methods and standalone LSTM networks.

Keywords: sentiment analysis; deep learning; long short-term memory; attention model

0 Introduction

With the explosive growth of internet information in recent years, natural language processing has assumed a position of critical importance. Both academia and industry have shown considerable interest in processing massive volumes of textual data. As a component of natural language processing, text sentiment analysis exerts profound influence not only within the NLP field but also in domains such as politics, economics, and social sciences that are heavily impacted by subjective human viewpoints. Consequently, it has become a hot research topic among scholars both domestically and internationally in recent years.

As internet development continues to expand the scale of textual data, sentiment analysis using deep learning has gradually matured. Inspired by the successful application of deep learning across various domains and its growing adoption in text sentiment analysis, this paper proposes a structure based on Long Short-Term Memory networks and attention models for text sentiment analysis. Through experimentation and validation, we demonstrate the effectiveness of our proposed method.

1 Related Work

Text sentiment analysis refers to the detection, analysis, and mining of subjective texts containing user-expressed opinions, preferences, and emotions. With the development of the internet, people increasingly express their subjective views and opinions online regarding products, news topics, and current events. Analyzing the sentiment within these viewpoints has become essential.

Common approaches to text sentiment analysis primarily include dictionary-based methods and traditional machine learning algorithms. Dictionary-based approaches mainly operate by matching sentiment words from emotional lexicons and applying certain rules to score texts, ultimately determining sentiment polarity. Kim and Hovy proposed a dictionary-based algorithm that constructs new sentiment dictionaries using bootstrapping strategies based on synonym and antonym relationships in WordNet, along with some given positive and negative sentiment words. Sentiment orientation is determined by multiplying the scores of emotional words in sentences. Dictionary-based methods dominated the field for a period and produced various lexicons (see reference [2]), but they require extensive manual annotation, making them time-consuming and labor-intensive. Consequently, such methods are now less commonly used.

Conventional machine learning approaches primarily treat text sentiment analysis as a text classification problem. Since sentiment classification divides texts into positive and negative categories, sentiment analysis places greater emphasis on studying words and sentences that express emotions. Pang et al. were the first to publish a paper proposing the use of supervised learning algorithms for sentiment classification, applying naive Bayes, maximum entropy, and support vector machine (SVM) methods to movie review sentiment classification. Numerous studies have shown that SVM algorithms achieve excellent results in text sentiment analysis. However, these methods suffer from poor generalization capabilities—algorithms that perform well in one domain may perform poorly when applied to another.

In recent years, the application of deep learning to natural language processing has significantly advanced the field, with text sentiment analysis increasingly employing deep learning algorithms. Bengio pioneered the use of neural networks for language modeling, though this model had many parameters and high training costs. Mikolov et al. from Google simplified this model and proposed word2vec, implementing CBOW and Skip-gram frameworks that have been applied across multiple domains. Kim proposed using Convolutional Neural Networks (CNN) for sentence modeling to solve sentiment classification tasks, achieving excellent results across multiple datasets. Santos et al. utilized deep CNNs for sentiment analysis of short texts, while Irsoy et al. employed Recurrent Neural Networks (RNN) for sentence modeling. Long Short-Term Memory (LSTM), a special RNN model, was also validated by Tai et al. for solving sentiment analysis problems. LSTM is a special type of recurrent neural network that outperforms standard RNNs in many tasks.

The attention model, proposed by Bahdanau et al., was first applied to machine translation and achieved promising results before being incorporated into Google' s neural machine translation system. As deep learning frameworks have become increasingly applied to sentiment analysis, deep learning approaches have gained prominence in this domain. Detailed discussions of deep learning-based text sentiment analysis can be found in reference [11].

With the development of natural language processing for Chinese, traditional algorithms based on sentiment dictionaries and machine learning have also advanced considerably for Chinese text sentiment analysis. Chinese sentiment analysis typically adapts foreign methods while incorporating characteristics of the Chinese language, with most supervised learning-based research achieving good results. Deep learning methods also play an important role in Chinese sentiment analysis, as surveyed in reference [12].

2 Proposed Scheme Based on LSTM and Attention Model

To achieve the goal of text sentiment analysis, this paper proposes a network structure based on Long Short-Term Memory networks and attention models. The model structure is shown in [Figure 1: see original paper]. The model consists of two main components: a Long Short-Term Memory neural network structure and an attention model structure. The overall workflow of our proposed model is as follows: First, the input text sentences are encoded using a word vector model to transform the text into word vector representations. These representations then pass through the LSTM and attention model structure, followed by a fully connected layer, and finally a classifier completes the text sentiment analysis task.

2.1 Model Representation

Given a text sentence dataset D containing texts $X\{x_1, x_2, ..., x_m\}$ and corresponding sentiment labels $Y\{y_1, y_2, ..., y_m\}$, where each text sentence X_i consists of n words represented as $\{x_{i1}, x_{i2}, ..., x_{in}\}$, the final objective function is expressed as:

$$\mathcal{L} = \sum_{i=1}^{m} \log p(y_i | x_i; \theta)$$

where θ represents all parameters involved in the model, and $f(\cdot)$ represents the formal expression of the model.

2.2 Long Short-Term Memory Network (LSTM)

A core concept of recurrent neural networks is connecting previous information to the current task, effectively functioning as a memory mechanism. Due to the special contextual nature of text, employing recurrent neural networks for textual tasks yields better results.

LSTM is a special type of recurrent neural network proposed by Hochreiter and Schmidhuber in 1997. In many tasks, LSTM solves the gradient vanishing and gradient explosion problems of standard RNNs, thus outperforming standard recurrent neural networks. Because of LSTM' s excellent performance, we adopt LSTM for our sentiment analysis task. The ingenuity of LSTM lies in its addition of input gates, forget gates, and output gates, which make the self-looping weights variable. This allows the integration scale to change dynamically across different time steps while model parameters remain fixed, thereby avoiding gradient vanishing or explosion.

[Figure 2: see original paper] shows the schematic diagram of the Long Short-Term Memory network, where x represents input data, h denotes LSTM unit output, and C indicates memory cell values. In the LSTM dynamic gate structure, the forget gate determines what information to discard. After reading

h_{t-1} and x_t, this gate outputs a value between 0 and 1, where f_t represents the percentage of information to be discarded (0 means complete discard, 1 means complete retention). The calculation formula for f_t is:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

where σ represents the sigmoid function, W_f denotes forget gate weights, and b_f represents forget gate bias.

In [Figure 2: see original paper], i_t refers to the update value that controls the influence of current input on the memory cell state. After passing through a tanh layer, a new candidate value vector is created and added to the state. The two formulas for this process are:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C_t} = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

where W_i denotes update gate weights, b_i represents update gate bias, tanh is the hyperbolic tangent function, W_c represents update candidate values, b_c denotes update candidate bias, and $\tilde{C_t}$ refers to the candidate value.

Next, the cell state is updated by multiplying the old state with f_t to discard already determined information, with changes made according to the degree of state updating. The state update formula is:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C_t}$$

where C_t represents the new state value.

Finally, the output value is determined. The output gate value o_t controls the output of the memory cell state value. The output information process formulas are:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t \cdot \tanh(C_t)$$

where W_o denotes output value weights, b_o represents output value bias, and h_t indicates the final determined output.

2.3 Feedforward Attention Model

In text analysis tasks, attention models represent the relevance between words in a text sentence and the output result. This model was first applied to machine translation tasks. The feedforward attention model employed in this paper represents a direct simplification of conventional attention models, constructing a single vector c from the entire sequence as follows:

$$c = \sum_{t=1}^{T} \alpha_t h_t$$

$$\alpha_t = \frac{\exp(a(h_t))}{\sum_{k=1}^{T} \exp(a(h_k))}$$

where a is a learning function that currently depends only on h_t. In the above formulas, the attention mechanism can be considered as constructing a fixed-length embedding layer c for the input sequence by computing an adaptive weighted average of the state sequence h.

[Figure 3: see original paper] shows a schematic diagram of the feedforward attention model. In this diagram, vectors from the hidden state sequence h_t are fed into the learning function $a(h_t)$, producing a probability vector α. Vector c is the weighted average of h_t with weights α.

The advantage of the attention mechanism model lies in its ability to integrate information over time. Therefore, by using this simplified attention mechanism, the model can handle variable-length sequences even when h_t is computed in a feedforward manner. Since the computation can be fully parallelized, using feedforward attention also leads to substantial efficiency gains.

The model proposed in this paper connects the output h of the Long Short-Term Memory model with the attention text c as input to the fully connected layer, which finally outputs the result after passing through the fully connected layer.

3 Experiments

3.1 Experimental Data Preparation

To validate the effectiveness of our proposed model, we selected the Chinese sentiment mining hotel review corpus (ChnSentiCorp) as our test set. This corpus was collected and organized by Dr. Tan Songbo from the Chinese Academy of Sciences, comprising 10,000 reviews automatically collected from Ctrip.com and subsequently processed. The corpus is organized into four subsets. Due to the specific nature of text sentiment analysis, this paper employs the ChnSentiCorp-Htl-ba-6000 dataset for experiments. This is a balanced corpus containing 3,000 positive and 3,000 negative samples. shows sample data from the dataset.

3.2 Text Preprocessing

Due to the special characteristics of Chinese, individual characters mostly cannot independently express meaning, requiring Chinese text to be segmented. This paper employs the Jieba segmentation system for text analysis and processing. Text preprocessing also includes removing stop words—words that carry no substantial meaning.

3.3 Modeling Process

The modeling process for our proposed scheme primarily utilizes the TensorFlow deep learning framework. [Figure 4: see original paper] illustrates the model framework construction process. The implementation scheme for sentiment analysis based on LSTM and attention models has been introduced previously. The model construction employs TensorFlow' s sequential model framework. First, an Embedding layer is added as model input, followed by an LSTM model. A feedforward attention mechanism layer is added after the LSTM model to assign different weights to different words in review texts when obtaining sentence vector representations. These differently weighted word vectors are then combined through weighted summation to obtain the sentence vector representation. Subsequently, a Dense layer is added, and the Activation layer uses a sigmoid function for text classification. Additionally, dropout is employed during model training to prevent overfitting. Finally, the compilation process uses gradient descent algorithms for weight update iterations.

The feedforward attention mechanism layer is constructed according to specified formats. [Figure 5: see original paper] shows partial code for constructing the feedforward attention mechanism, which strictly follows the steps introduced earlier and adheres to TensorFlow framework requirements.

After modeling completion, the prepared text test data is used to apply the model to text sentiment analysis and evaluate it using assessment metrics.

3.4 Experimental Evaluation Metrics

Precision and recall are two metrics used to evaluate classification results, defined as:

$$\text{Precision} = \frac{\text{Correct}}{\text{Output}}$$

$$\text{Recall} = \frac{\text{Correct}}{\text{Labeled}}$$

where Correct represents the number of correctly returned results, Output represents the total number of returned results, and Labeled represents the total number of samples in that class in the test set. Precision measures classifier accuracy, while recall measures whether the classifier can find all samples of a class. These two metrics should be balanced, which is achieved using the F1-measure (the weighted harmonic mean of precision and recall), defined as:

$$\text{F1-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

3.5 Experimental Results and Analysis

This paper conducted two sets of experiments on the dataset. The first set classified texts using different text representation methods to determine the optimal word vector training scheme. The second set classified texts using different classification methods to validate the effectiveness of our proposed sentiment analysis model.

In the first set of experiments, three approaches were compared: - **BOW-LSTM**: Words are converted to vectors using the bag-of-words model, then trained with LSTM - **if-idf-LSTM**: Words are converted to vectors using the if-idf model, then trained with LSTM - **W2V-LSTM**: Words are converted to vectors using word2vec training, then trained with LSTM

shows the classification results based on different text representation methods. Analysis reveals that the if-idf model weights words in the bag-of-words representation. However, both methods ignore semantic relationships between words in the text. Consequently, W2V-LSTM demonstrates the best performance among the three. Since word2vec yields optimal results, our scheme adopts word2vec for word vector training.

The second set of experiments compares our proposed system with: - **W2V-SVM**: Traditional machine learning using word2vec-trained vectors with SVM classification - **W2V-Att-CNN**: A structure combining CNN and attention models, referencing the sentiment analysis model proposed by Feng Xingjie et al. [14] - **W2V-LSTM**: Using word2vec-trained vectors with LSTM - **W2V-Att-LSTM**: Our proposed structure using word2vec-trained vectors with LSTM and attention model

Using 10-fold cross-validation on the ChnSentiCorp-Htl-ba-6000 dataset, the results are shown in .

presents four typical sentiment-bearing examples to illustrate our model' s workflow. After text preprocessing with Jieba segmentation, word2vec training converts words to vectors that serve as model input. Passing through our LSTM and feedforward attention model, the classification layer finally categorizes text data as positive or negative. Comparison with manually labeled data validates our model' s effectiveness, with sample analysis clearly demonstrating accuracy guarantees.

Examining the results in , W2V-SVM outperforms CBOW-SVM because word2vec-trained vectors incorporate contextual semantic information, yielding better sentiment analysis results. W2V-LSTM surpasses W2V-SVM because

LSTM, as a deep training model, extracts more meaningful relevant features while considering contextual relationships. W2V-Att-LSTM outperforms W2V-LSTM because the introduced feedforward attention mechanism considers relevance between text sentences and results, achieving better performance when combined with LSTM. Furthermore, comparing W2V-Att-LSTM with W2V-Att-CNN reveals that our LSTM-with-attention model performs better than the CNN-with-attention model, proving more suitable for text sentiment analysis.

4 Conclusion

This paper proposes a text sentiment analysis method based on Long Short-Term Memory networks and attention models. Experimental results demonstrate the effectiveness and feasibility of our approach, enabling better research on text orientation and contributing to deep learning-based text sentiment analysis.

Deep learning models for Chinese sentiment analysis remain in early development stages. The complexity of Chinese grammatical structures and semantic diversity of words pose challenges for text sentiment analysis. Future work will further investigate deep learning models for Chinese sentiment analysis.

References

[1] Kim S M, Hovy E. Determining the sentiment of opinions [C]// Proc of the 20th International Conference on Computational Linguistics. Stroudsburg, PA: Association for Computational Linguistics, 2004: 1367-1374.

[2] Ahmad M, Aftab S, Muhammad S S, et al. Tools and techniques for lexicon driven sentiment analysis: a review [J]. International Journal of Multidisciplinary Sciences And Engineering, 2017, 8 (1): 17-23.

[3] Pang Bo, Lee L, Vaithyanathan S. Thumbs up?: sentiment classification using machine learning techniques [C]// Proc of Conference on Empirical methods in Natural Language Processing. Stroudsburg, PA: Association for Computational Linguistics, 2002: 79-86.

[4] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model [J]. Journal of Machine Learning Research, 2003, 3: 1137-1155.

[5] Mikolov T, Sutskever I, Chen Kai, et al. Distributed representations of words and phrases and their compositionality [C]// Proc of the 26th International Conference on Neural Information Processing Systems. USA: Curran Associates Inc, 2013: 3111-3119.

[6] Kim Y. Convolutional Neural Networks for Sentence Classification [J]. ArXiv preprint arXiv, 2014: 1408.5882.

[7] Dos Santos C N, De Bayser M G. Deep convolutional neural networks for

sentiment analysis of short texts [C]// Proc of the 25th International Conference on Computational Linguistics. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, 2014: 69-78.

[8] Irsoy O, Cardie C. Opinion mining with deep recurrent neural networks [C]// Proc of Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA: Association for Computational Linguistics, 2014: 720-728.

[9] Tai Kaisheng, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks [J]. Computer Science, 2015, 5 (1): 36.

[10] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate [J]. ArXiv preprint arXiv, 2014: 1409.0473.

[11] Zhang Lei, Wang Shuai, Liu Bing. Deep learning for sentiment analysis: A survey [J]. Wiley Interdisciplinary Reviews Data Mining & Knowledge Discovery, 2018, 8 (7): 1-34.

[12] Peng Haiyun, Cambria E, Hussain A. A review of sentiment analysis research in Chinese language [J]. Cognitive Computation, 2017, 9 (4): 423-435.

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

[14] 冯兴杰, 张志伟, 史金钏. 基于卷积神经网络和注意力模型的文本情感分析 [J]. 计算机应用研究, 2018, 35 (5): 1434-1436. (Feng Xingjie, Zhang Zhiwei, Shi Jinchuan. Text sentiment analysis based on convolutional neural networks and attention model [J]. Application Research of Computers, 2018, 35 (05): 1434-1436.)

Note: Figure translations are in progress. See original paper for figures.