Contextual Features and Their Application in Sentiment Classification Models (Postprint)

Authors: Liu Dong, Zhang Caihuan

Date: 2018-11-29T00:00:00+00:00

Abstract

To investigate the role of contextual features in text classification, this paper proposes a hierarchical bidirectional LSTM model for sentiment classification tasks. The model first tokenizes sentences and employs word vectors as input to the first-layer bidirectional LSTM; subsequently, it extracts dense, continuous vectors from the document as contextual features; then, the output vectors from the first-layer model and the contextual vectors are jointly fed into the second-layer bidirectional LSTM; finally, the output vectors of this hierarchical bidirectional LSTM model are classified using a sigmoid function. The contextual vectors act upon each sentence, enhancing consistent sentiments while weakening inconsistent ones, thereby improving classification accuracy. Experimental results on two public datasets demonstrate that the hierarchical bidirectional LSTM integrating contextual features achieves superior accuracy. Furthermore, evaluation on a public dataset comprising over 20,000 Chinese reviews indicates that the model' s test accuracy improves compared to both standard LSTM and bidirectional LSTM, suggesting that contextual features contribute significantly to enhancing sentiment classification.

Full Text

Context Features and Their Application in Sentiment Classification Models

Liu Donga, **Zhang Caihuan**b
aSchool of Information Technology, bSchool of Mathematical Sciences, Luoyang Normal University, Luoyang, Henan 471022, China

Abstract

To investigate the role of context features in text classification, this paper proposes a hierarchical bidirectional LSTM model for sentiment classification. The model first segments sentences into words and uses word embeddings as input to the first-layer bidirectional LSTM. It then extracts dense, continuous vectors from the entire document as context features. The output vectors from the first layer are concatenated with these context vectors and fed into a second-layer bidirectional LSTM. Finally, the output from this hierarchical bidirectional LSTM model is passed through a sigmoid function for classification. The context vector acts on each sentence, enhancing consistent sentiments while weakening inconsistent ones, thereby improving classification accuracy. Experiments on two public datasets demonstrate that the hierarchical bidirectional LSTM with integrated context features achieves superior accuracy. Additionally, testing on a public dataset containing over 20,000 Chinese reviews shows that the model' s test accuracy improves compared to both standard LSTM and bidirectional LSTM, indicating that context features significantly enhance sentiment classification performance.

Keywords: context features; sentiment classification; hierarchical Bi-LSTM

0 Introduction

Sentiment classification of text is an important research direction in text analysis. With the development and popularization of the Internet and mobile applications, massive amounts of text—particularly reviews expressing user opinions on products, services, movies, and current affairs—have emerged online. In-depth analysis and mining of these texts hold significant application value and can generate substantial economic benefits. For instance, analyzing product reviews can provide merchants with product feedback to improve service quality and optimize marketing strategies, while sentiment analysis of online reviews enables timely understanding of public opinion. Therefore, research on text sentiment classification is highly meaningful.

Sentiment classification can be categorized in several ways. In terms of classification schemes, most studies adopt binary polarity (positive and negative), though some involve ternary classification (positive, negative, neutral) and multi-class classification that quantifies user sentiment intensity into multiple levels. From the perspective of text granularity, it can be divided into document-level, sentence-level, and word-level classification. From a language perspective, it can be categorized as monolingual or cross-lingual sentiment classification. From a domain perspective, it can be divided into single-domain and cross-domain classification.

From a technical standpoint, early research primarily relied on sentiment lexicons, selecting sentiment words from text as features for classification. However, such methods did not achieve ideal accuracy. Subsequently, machine learning methods—including supervised, unsupervised, and semi-supervised approaches—were applied to text sentiment classification, yielding improved results. Following the tremendous success of deep learning in image and speech processing, these techniques were adapted for sentiment classification problems with promising outcomes. Severyn et al. used unsupervised language models to train word embeddings, which served as initialization parameters for a deep CNN to classify Twitter sentiment, achieving excellent results in two Semeval-2015 subtasks. Tang et al. integrated target information into LSTM to model the correlation between targets and their contexts, improving sentiment classification accuracy. Nguyen et al. proposed an RNN-based sentiment classification method that considers dependency and constituent parse trees. These results demonstrate the potential of deep learning for natural language understanding. However, these methods only consider phrase-level or sentence-level relationships and cannot model emotional continuation, intensification, or transition between sentences. While deep neural networks can obtain features at different abstraction levels, increasing network depth imposes new requirements on computational resources and dataset size. For natural language understanding, hierarchical models with repeated stacking are more appropriate, such as the hierarchical CNN used by Denil et al. to extract important sentences from reviews, the hierarchical LSTM autoencoder applied by Li et al. to paragraph generation and reconstruction, and the RNN and CNN-based short text classification method by Ji et al., though this was sentence-level and only considered a limited number of sentences before and after the target. Most similar to our approach is the hierarchical model by Ruder et al., who also used a hierarchical bidirectional LSTM for sentiment classification but incorporated product aspect information at the sentence level, whereas we add contextual information to each sentence.

When discussing the sentiment expressed in a document, one cannot ignore its context—the situation in which sentences reside. Therefore, incorporating document context into sentiment classification models is undoubtedly beneficial. This paper discusses the distributed representation of document context, a method previously used by Choi et al. in neural machine translation models with good effect. Ghosh et al. integrated context features (topics) into their model, evaluating the CLSTM model on three NLP tasks: next word prediction, next sentence prediction, and sentence topic prediction, achieving a 20% improvement. The contributions and innovations of this paper are: (a) we represent contextual information as a continuous dense vector applied to sentence sentiment expression, where sentiments consistent with the global context are enhanced while inconsistent ones are weakened, resulting in better classification performance; (b) experiments demonstrate that sentiment classification models with contextual information not only outperform standard bidirectional LSTM models but also achieve considerable improvements over attention-based models when hyperparameters are properly tuned.

1 Methodology

1.1 Distributed Representation of Context

Each review is treated as a document. Documents are truncated to contain sentences (10 based on the corpus), with sentences containing only empty strings added if insufficient. Each sentence is segmented and truncated to contain m words (5), with empty strings added if insufficient. Thus, the i-th sentence can be represented as a concatenation of word vectors:

$$\mathbf{s}_i = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m)$$

where $\mathbf{x}_i \in \mathbb{R}^k$ is the k-dimensional word vector for the i-th word.

Similarly, the i-th document can be represented as a concatenation of sentences:

$$\mathbf{d}_i = (\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_\ell)$$

When analyzing the sentiment of a review, the entire document constructs a context. Therefore, context can be represented as an unordered collection of all words in the document. Following the approach of Choi et al., we define the context $\mathbf{c}$ as:

$$\mathbf{c} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_t$$

where T is the number of words in the document.

1.2 Hierarchical Bidirectional LSTM Model

LSTM is a special type of RNN that can learn long-term dependencies. By adding input, output, and forget gates to RNN units, LSTM models long-term dependencies and has achieved great success in many problems. In sentiment analysis of reviews, it is recognized that long-term dependency is also important. Not only can preceding sentences in a review reflect the sentiment the customer wants to express, but subsequent sentences often either intensify or reverse the sentiment. Therefore, we use a hierarchical bidirectional LSTM, divided into sentence-level and document-level.

(a) Sentence-level bidirectional LSTM: For word w_t in a sentence, its word embedding $\mathbf{x}_t$, the previous unit' s output $\mathbf{h}_{t-1}$, and cell state $\mathbf{C}_{t-1}$ are input to the forward LSTM to compute the next output $\mathbf{h}_t$ and cell state $\mathbf{C}_t$. First, we compute the input gate i_t, candidate cell state values, and forget gate f_t:

$$\begin{aligned} i_t &= \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i) \\ \tilde{\mathbf{C}}_t &= \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c) \\ f_t &= \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \end{aligned}$$

From these, we compute the new cell state value:

$$\mathbf{C}_t = f_t * \mathbf{C}_{t-1} + i_t * \tilde{\mathbf{C}}_t$$

Finally, we compute the output gate value and the LSTM unit' s output:

$$o_t = \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o)$$
$$\mathbf{h}_t = o_t * \tanh(\mathbf{C}_t)$$

where $\mathbf{W}_{i,c,f,o}$ and $\mathbf{U}_{i,c,f,o}$ are weight matrices, and $\mathbf{b}_{i,c,f,o}$ are bias terms.

This forward LSTM processes words in their original order, thus fully utilizing preceding information. Similarly, the backward LSTM processes in reverse order, fully utilizing subsequent information. The output at a given time step is formed by concatenating the corresponding states from both LSTMs.

(b) A bidirectional LSTM stacked at the sentence level: The difference is that this bidirectional LSTM takes as input the concatenation of the sentence-level LSTM' s output with the context vector $\mathbf{E}$, forming the second-layer bidirectional LSTM. After adding the context vector, the input gate i_t, candidate cell state values, and forget gate f_t are modified as:

$$i_t = \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{W}_e\mathbf{E} + \mathbf{b}_i)$$
$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1} + \mathbf{W}_e\mathbf{E} + \mathbf{b}_c)$$
$$f_t = \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{W}_e\mathbf{E} + \mathbf{b}_f)$$

The new cell state value is computed as:

$$\mathbf{C}_t = f_t * \mathbf{C}_{t-1} + i_t * \tilde{\mathbf{C}}_t$$

Finally, the output gate value and unit output are:

$$o_t = \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{W}_e\mathbf{E} + \mathbf{b}_o)$$
$$\mathbf{h}_t = o_t * \tanh(\mathbf{C}_t)$$

(c) The vector formed by concatenating the second-layer LSTM' s outputs is passed to a sigmoid function, producing the probability distribution for document sentiment classification. The model structure is shown in Figure 1 [Figure 1: see original paper].

2 Experiments

2.1 Datasets and Training

We evaluated our model on two datasets: IMDB and Yelp 2013. These datasets were constructed and preprocessed by Tang et al., with a training/development/test split ratio of 8:1:1. Following Chen et al., we used the same datasets, though Chen et al. used 200-dimensional pretrained word embeddings, whereas we added an embedding layer to generate 256-dimensional word vectors. Basic statistics are shown in Table 1 .

Table 1 Statistics of datasets

Dataset	Classes	#Documents	Sentences/Doc
IMDB	Binary		
Yelp2013	Binary		

The baseline model used in experiments is Bi-LSTM. The evaluation metric is test accuracy, and the training objective is to minimize the binary cross-entropy function. We used the Adam optimizer with mini-batches of size 32. To facilitate experimental analysis, we also applied the model to another Chinese dataset containing over 20,000 reviews across 6 domains, primarily product and hotel service reviews. The shortest reviews contain only one character (e.g., "good" , "like"), while the longest review contains 2,968 characters about travel and hotel service complaints. All reviews are labeled as positive or negative. After shuffling, the first 15,000 were used as the training set and the remainder as the test set. In the model' s input layer, we used 256-dimensional word vectors and 256-dimensional context vectors; the LSTM hidden layer had 128 units, and dropout was set to 0.5. All texts were segmented using jieba before training.

2.2 Results and Comparative Analysis

Previous experiments on the IMDB and Yelp 2013 datasets compared our model with single-layer LSTM and Bi-LSTM, showing accuracy improvements through the context vector' s effect. This experiment primarily uses Bi-LSTM as the baseline to validate the context vector' s effect, comparing it with Bi-LSTM with context and HB-LSTM with context. Results are shown in Table 2 . As before, we employed early stopping with a patience of 2 epochs.

Table 2 Results on IMDB and Yelp2013

Model	Yelp 2013 Train Acc.	Yelp 2013 Test Acc.	IMDB Train Acc.	IMDB Test Acc.
Bi-LSTM				

Model	Yelp 2013 Train Acc.	Yelp 2013 Test Acc.	IMDB Train Acc.	IMDB Test Acc.
Bi-LSTM with Context				
HB-LSTM with Context				

Table 2 shows that compared to Bi-LSTM, HB-LSTM achieves varying degrees of improvement in test accuracy on both datasets.

Since we used the same datasets as Chen et al. for sentiment classification, we can compare results. Chen et al. used RMSE as the loss function, while we use cross-entropy. Table 3 shows the comparison between HB-LSTM and the attention-based NSC model (NSC+UPA stands for Neural Sentiment Classification with User and Product Attention) under both loss functions.

Table 3 Effect of HB-LSTM and NSC+UPA with RMSE or binary_{crossentropy} as loss function

Model	Yelp 2013 RMSE	Yelp 2013 Binary Crossentropy	IMDB RMSE	IMDB Binary Crossentropy
NSC+UPA				
HB-LSTM				

Table 3 reveals substantial differences. Using RMSE as the loss function, HB-LSTM' s accuracy is 3-7 percentage points worse than NSC+UPA. However, when using binary_{crossentropy}, accuracy nearly doubles. This discrepancy may be due to different hyperparameter settings, which warrants further investigation.

The Chinese dataset concerns sentiment classification of product and service reviews, a binary classification problem where LSTM models have already achieved good results. To verify whether context vectors can further improve classification, we selected single-layer LSTM as the baseline and compared it with bidirectional LSTM (Bi-LSTM) and hierarchical bidirectional LSTM (HB-LSTM). Results are shown in Table 4 .

Table 4 Experimental result and comparison on Chinese review dataset

Model	Accuracy
LSTM	
Bi-LSTM	
HBLSTM	

As shown in Table 4, models using LSTM perform well overall. LSTM integrates neighboring information and adequately considers dependency relationships within a certain range before and after each sentence, leaving limited room for improvement with bidirectional LSTM. However, the hierarchical bidirectional LSTM with context features (HB-LSTM) shows significant accuracy improvement, demonstrating that after adding context vectors, global contextual information plays a role in features, yielding better performance than the baseline.

Table 5 lists several examples misclassified by LSTM for comparison between LSTM and HB-LSTM.

Table 5 Some wrong classification examples by LSTM

Text	LSTM	HBLSTM
Very good, fast delivery. Called customer service in the morning, installer came in the afternoon. Speedy! But there was no water at the time, didn’ t test it, should be fine. If there are problems, can still contact manufacturer maintenance personnel.	negative	positive

Text	LSTM	HBLSTM
Original gas water heater broke, replaced it, much better now, no more waiting over 40 seconds for hot water with unstable temperature. Now with Midea electric water heater, hot water comes out after 5 seconds with stable constant temperature. Good. Highly recommend not buying gas ones!! Midea electric water heater is economical and very practical.	negative	positive

Text	LSTM	HBLSTM
Fast internet, convenient VPN connection to company intranet. IT manager set up the network well, praise. But front desk attitude doesn’ t meet satisfactory standards. Hotel has no iron, needed to meet client, had to ask them to iron a shirt, quoted 5 yuan, but charged 8 yuan at checkout, saying service and rush fees are 3 yuan. Opening the glass shower door, realized you need at least 2-meter-long right arm to reach the towel, had to come out dripping wet. Thought about it, can’ t bring towel into small shower stall either, don’ t know who designed this. Didn’ t find more advantages than other hotels, 560 yuan somewhat wasted, won’ t come again.	positive	negative

The first example expresses clear positive sentiment. LSTM misclassified it due to excessive negative words like “no water,” “no problem,” “problem,” etc. The second review compares before and after product usage and gives advice “definitely don’ t buy gas ones,” which may have misled LSTM, but HB-LSTM better grasped global information. The third review shows a turn in sentiment—praising the hotel’s network initially but expressing dissatisfaction with facilities and services later, stating “won’t come again.”Overall, negative sentiment should dominate, which HB-LSTM correctly identified by better considering the overall context' s weight in features.

3 Conclusion

A common approach in current sentiment analysis technology involves segmenting documents, removing stop words, representing words as dense continuous vectors through a feedforward neural network, and then performing feature extraction and sentiment classification via deep learning models like LSTM or CNN. This method has achieved certain results. This paper analyzes problems in this general approach, such as long-term dependency of sentiment in documents, and proposes the hypothesis of using document-level contextual information to enhance deep learning models' feature extraction capabilities for improved sentiment classification. Experimental results demonstrate that this context-feature-integrated model shows noticeable effects on sentiment classification compared to baseline models.

Future research will explore applying context features to broader classification problems, especially social network text analysis, which will have greater value in community detection, public opinion monitoring, and analysis. Additionally, analogous to character-level deep learning models in English and pixel-based models in image processing, we believe Chinese characters are the basic unit of Chinese, and deep learning models for Chinese character-based sentiment analysis deserve in-depth investigation.

References

[1] Tsytsarau M, Palpanas T. Survey on mining subjective data on the Web [J]. Data Mining and Knowledge Discovery, 2012, 24 (3): 478-514.

[2] 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. 2002: 79-86.

[3] Xia Rui, Xu Feng, Zong Chengqing, et al. Dual Sentiment Analysis: Considering Two Sides of One Review [J]. IEEE Trans on Knowledge and Data Engineering, 2015, 27: 2120-2133.

[4] Pang Bo, Lee L. Seeing stars: exploring class relationships for sentiment categorization with respect to rating scales [C]// Proc of the 43rd Annual Meeting on Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2005, 115-124.

[5] Zhao Yanyan, Qin Bing, Liu Ting. Sentiment analysis [J]. Journal of Software, 2010, 21 (8): 1834-1848.

[6] Li Shoushan, Xia Rui, Zong Chengqing, et al. A framework of feature selection methods for text categorization [C]// Proc of the 47th Annual Meeting of ACL. Stroudsburg: Association for Computational Linguistics, 2009, 692-700.

[7] Wan Xiaojun. Using bilingual knowledge and ensemble techniques for unsupervised Chinese sentiment analysis [C]// Proc of EMNLP. Stroudsburg: Association for Computational Linguistics, 2008, 553-561.

[8] Lin Chenghua, He Yulan, Everson R, et al. Weakly supervised joint sentiment-topic detection from text [J]. IEEE Trans on Knowledge and Data Engineering, 2012, 24 (6): 1134-1145.

[9] Sun Yan, Zhou Xueguang, Fu Wei. Unsupervised Topic and sentiment unification model for sentiment analysis [J]. Acta Scientiarum Naturalium Universitatis Pekinensis, 2013, 49 (1): 102-108.

[10] Li Suke, Jiang Yanbing. Semi-supervised sentiment classification based on sentiment feature clustering [J]. Journal of Computer Research and Development, 2013, 50 (12): 2570-2577.

[11] Zhou Zhihua. Disagreement-based Semi-supervised Learning [J]. Acta Automatica Sinica, 2013, 39 (11): 1871-1878.

[12] Zhang Xiaomei, Li Ru, Wang Bin, et al. Subjective and objective classification of micro-blog based on feature fusion [J]. Journal of Chinese Information Processing, 2014, 28 (4): 50-57.

[13] Zhao Yanyan, Qin Bing, Che Wanxiang, et al. Appraisal expression recognition with syntactic path for sentence sentiment classification [J]. International Journal of Computer Processing of Oriental Languages, 2011, 23 (1): 21-37.

[14] Song Yanxue, Zhang Shaowu, Lin Hongfei. Sentence sentiment analysis based on ambiguous words [J]. Journal of Chinese Information Processing, 2012, 26 (3): 38-44.

[15] Socher R, Perelygin A, Wu J Y, et al. Recursive deep models for semantic compositionality over a sentiment treebank [C]// Proc of Conference on Empirical Methods in Natural Language Processing, 2013, 1631-1642.

[16] Lu Bin, Wan Xiaojun, Yang Jianwu, et al. Using tongyici cilin to compute word semantic polarity [C]// Proc of the 7th International Conference on Chinese Computing, 2007: 17-23.

[17] Zhang Lei, Li Mengshi, Chen Li, et al. Features and opinions classification of Chinese product reviews based on two level HHMMs [J]. Journal of Sichuan University Engineering: Science Edition, 2013, 45 (2): 94-102.

[18] Severyn A, Moschitti A. UNITN: training deep convolutional neural network for Twitter sentiment classification [C]// Proc of the 9th International Workshop on Semantic Evaluation, 2015: 464-469.

[19] Vo D, Zhang Yue. Target-dependent Twitter sentiment classification with rich automatic features [C]// Proc of International Joint Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2015: 1347-1353.

[20] Nguyen T H, Shirai K. PhraseRNN: phrase recursive neural network for aspect-based sentiment analysis [C]// Proc of Conference on Empirical Methods in Natural Language Processing, 2015: 2509-2514.

[21] Denil M, Demiraj A, Freitas N D. Extraction of salient sentences from labelled documents [EB/OL]. [2018-04-25] https://arxiv.org/pdf/1412.6815.

[22] Li Jiwei, Luong M T, Jurafsky D. A hierarchical neural autoencoder for paragraphs and documents [C]// Proc of the 53rd Annual Meeting of the Association for Computational Linguistics, 2015: 1106-1115.

[23] Lee J Y, Dernoncourt F. Sequential short-text classification with recurrent and convolutional neural networks [C]// Proc of Conference of the North American Chapter of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2016, 515-520.

[24] Ruder S, Ghaffari P, Breslin J G. A hierarchical model of reviews for aspect-based sentiment analysis [C]// Proc of Conference on Empirical Methods in Natural Language Processing, 2016: 999-1005.

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

[26] Dataset source [EB/OL] (2016-06-29), [2018-04-25]. http://spaces.ac.cn/archives/3863/

[27] Choi H, Cho K, Bengio Y. Context-dependent word representation for neural machine translation [J]. Computer Speech & Language, 2016.

[28] Xiang Zhang, Yann LeCun. Text Understanding from Scratch [EB/OL], (2015) [2018-04-25] https://arxiv.org/abs/1502.01710

[29] Chen Huimin, Sun Maosong, Tu Cunchao, et al. Neural sentiment classification with user and product attention [C]// Proc of Conference on Empirical Methods in Natural Language Processing. Stroudsburg: Association for Computational Linguistics, 2016: 1650-1659.

[30] Tang Duyu, Qin Bing, Liu Ting. Learning semantic representations of users and products for document level sentiment classification [C]// Proc of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015: 1014-1023.

[31] Ghosh S, Vinyals O, Strope B, et al. Contextual LSTM (CLSTM) models for Large scale NLP tasks [EB/OL]. (2016) [2018-04-25]. http://arxiv.org/abs/1602.06291.

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