



Text Sentiment Classification Based on Feature Fusion

Chen Zhang, Qingxu Li*, Xue Cheng

School of Cyber Security, Gansu University of Political Science and Law, Lanzhou 730070, China

Corresponding Author Email: zc6454@gsli.edu.cn

<https://doi.org/10.18280/ria.340418>

Received: 21 April 2020

Accepted: 1 July 2020

Keywords:

word vector, convolutional neural network (CNN), bidirectional long short-term memory (BiLSTM) network, CNN_BiLSTM parallel hybrid model

ABSTRACT

The convolutional neural network (CNN) and long short-term memory (LSTM) network are adept at extracting local and global features, respectively. Both can achieve excellent classification effects. However, the CNN performs poorly in extracting the global contextual information of the text, while LSTM often overlooks the features hidden between words. For text sentiment classification, this paper combines the CNN with bidirectional LSTM (BiLSTM) into a parallel hybrid model called CNN_BiLSTM. Firstly, the CNN was adopted to extract the local features of the text quickly. Next, the BiLSTM was employed to obtain the global text features containing contextual semantics. After that, the features extracted by the two neural networks (NNs) were fused, and processed by Softmax classifier for text sentiment classification. To verify its performance, the CNN_BiLSTM was compared with single NNs like CNN and LSTM, as well as other deep learning (DL) NNs through experiments. The experimental results show that the proposed parallel hybrid model outperformed the contrastive methods in F1-score and accuracy. Therefore, our model can solve text sentiment classification tasks effectively, and boast better practical value than other NNs.

1. INTRODUCTION

Information is being generated all the time in this era of technological development and innovation. With the advancement of science and technology, information is disseminated in various forms, namely, text, image, and audio. Among them, text is the primary carrier of information dissemination [1].

The boom of information technology has spawned many emerging Internet platforms, namely, Weibo, Baidu Tieba, and blogs, creating a huge amount of data. Many types of the data, including product/service reviews, hot news, and social network comments, belong to text sentiment classification.

Currently, sentiment classification manifests in complex and diverse forms, and brings many kinds of information, ranging from Weibo replies to news comments. These information often exist as short texts, which contain fewer words and involve irregular use of words than the traditional long texts.

To realize sentiment classification of massive data, many researchers have carried out text sentiment classification through statistical methods, traditional machine learning (ML), and deep learning (DL) [2]. The main bases of text sentiment classification are sentiment dictionary, traditional ML, and DL.

As its name suggest, the sentiment dictionary method requires a sentiment dictionary. However, few mature sentiment dictionaries are available, and building a sentiment dictionary is time-consuming and laborious. As a result, this method has rarely been adopted for sentiment classification.

The popular traditional ML methods include naive Bayes [3], decision tree (DT) [4], k-nearest neighbors (k-NN) [5], support vector machine (SVM) [6], etc. These methods usually represent the text as high-dimensional sparse vectors, which

requires manual annotation to construct features [7].

The DL constructs a neural network (NN) model for text data analysis. In recent years, this approach has been gradually introduced to sentiment classification. Through distributed representation [8], the text data are trained into low-dimensional dense vectors, and the features are automatically extracted via the NN, eliminating the need for manual annotation in traditional ML.

At present, convolutional neural network (CNN) [9] and long short-term memory (LSTM) are two main NNs for sentiment classification. The application of CNN to sentiment classification is a common practice. However, the CNN only extracts the local features of text data, failing to consider the semantic association between contexts. With sequential inputs, the LSTM can infer the semantics between words in the context. Nonetheless, the LSTM, owing to its complex structure, has difficulty in learning the association between words in high-dimensional text. The ensuing problem of long-term dependence will bring about exploding and vanishing gradients, two common problems of the CNN [10].

To sum up, sentiment dictionary methods can improve the accuracy of sentiment classification. The accuracy depends on the maturity of sentiment dictionary. But there are few mature sentiment dictionaries. The traditional ML methods rely on manual annotation to acquire the features of text data. Although it helps to make accurate sentiment analysis, the manual annotation is time-consuming and labor-intensive, reducing the applicability of traditional ML. The DL methods can automatically extract text features, without needing manual annotation. Nevertheless, a single NN (e.g. CNN and LSTM) cannot extract the global and local features from the text data at the same time.

Through the above analysis, this paper combines CNN and

bidirectional LSTM (BiLSTM) into a parallel hybrid model called CNN_BiLSTM, in which the traditional LSTM is replaced with BiLSTM. The proposed model can effectively extract and fuse both local and global features, making text sentiment classification much more accurate. To demonstrate its advantages, CNN_BiLSTM was compared with single NNs like CNN and LSTM, as well as other DL NNs through experiments. The comparison shows the excellence of CNN_BiLSTM in solving text sentiment classification tasks, and confirms the effectiveness of this parallel hybrid model.

The main contributions of this paper are as follows:

(1) The traditional LSTM was replaced with BiLSTM. The semantics of a word are not only related to the information before it, but also correlated with the information after it. However, the traditional unidirectional LSTM only considers the information before the current word, ignoring that after the word. The BiLSTM can overcome this limitation, and obtain all contextual information from both directions.

(2) The CNN_BiLSTM is a text sentiment classification method that couples CNN with BiLSTM. This parallel hybrid model combines the merits of both NNs in feature extraction, and makes effective extraction of both local and global features, and fuses the features obtained by both NNs, thereby improving the accuracy of text sentiment classification.

2. LITERATURE REVIEW

With the proliferation of DL, many scholars have begun to explore deep into deep NNs in the field of natural language processing (NLP), and discovered that DL is superior than traditional ML in many tasks. The application of deep NNs greatly promotes the development of DL.

Many researchers at home and abroad are striving to improve the accuracy of text classification. Kim [11] presented a multi-channel CNN for sentiment classification tasks, and experimentally proved that the design of multiple convolutional layers can extract features more effectively. Taking character sequence as the basic unit, Santos and Gatti [12] obtained the relevant features of the data through training, and thus improved the accuracy of text sentiment classification.

To solve the vanishing gradient problem of LSTM, Zhang et al. [13] added gated units to adjacent layers, such that the information can propagate between layers. To obtain deeper semantic features, Liang et al. [14] proposed a sentiment classification model based on polarity transfer and LSTM: firstly, the LSTM was extended to a tree-shaped recurrent NN (RNN); then, a polarity transfer model was introduced through the association between each word and its previous and subsequent information. Considering the excellent effect of LSTM in sentiment classification, Lu et al. [15] designed the P-LSTM model, which vectorizes each phrase and takes the vector as the input; Moreover, the phrase factor mechanism was integrated to fuse the features of the embedding layer and the hidden layer of the LSTM, further improving the accuracy of text classification.

All the above studies solve classification tasks with CNN or LSTM alone. Despite the good classification effect, the single NN only obtains the local features of the text. With the wide application of hybrid NNs in many fields, CNN is being combined with other NNs and implemented in the field of NLP. For instance, Zhang et al. [16] merged CNN and LSTM into the CNN-LSTM sentiment classification model, and achieved good results in text sentiment classification tasks. Li et al. [17]

proposed a hybrid model combining BiLSTM and CNN: the information before and after each word in the text are obtained by BiLSTM, and the CNN feature extraction is adopted for classification. This NN fusion approach gives inspiration to the design of our model.

The in-depth research on NNs reveals some problems: the training speed of the model attenuates, with the growing difficulty in training. This suppresses the accuracy of sentiment classification. To solve the problem, this paper builds a parallel hybrid model called CNN_BiLSTM, which combines CNN with BiLSTM, a substitute of the traditional LSTM.

3. MODEL CONSTRUCTION

3.1 Overview of text classification

Text classification consists of the following steps: data preprocessing, feature extraction, vectorization, construction of classification model, and model evaluation. The original texts, which are comments crawled from Weibo, were split into a training set and a test set. The two datasets were preprocessed, and then subject to feature extraction. The extracted features were imported to a classifier as matrix vectors. The classifier was trained by the training set, and verified by the test set, producing the final evaluation results of the model.

(1) Data preprocessing: The crawled text data are free in form, without any fixed grammar or pattern. Therefore, the text data were preprocessed through cleaning, word segmentation, and removal of stop words.

(2) Vectorization: The text data were expressed as digital vectors that can be recognized and processed by the computer.

(3) Feature extraction: The feature vector obtained by the convolution is processed by the nonlinear activation function of rectified linear unit (ReLU) to generate the features to be output by the text.

(4) Construction of classification model: The vectorized text was expressed as a matrix, and imported with text tags into the classification model for continuous training of parameters.

(5) Model evaluation: The model was evaluated by the accuracy and F1-score on test set.

3.2 Data preprocessing

In Chinese, there is no clear boundary between words. Hence, Chinese texts must be preprocessed through data cleaning, word segmentation, and removal of stop words. By contrast, English texts do not need to go through word segmentation, for English words are already separated by spaces.

The research data, as comments crawled from Weibo, are free in form, without any fixed grammar or pattern. Before application, the original data must receive data cleaning, word segmentation, and removal of stop words. Here, the cleaned data are segmented by Jieba, a famous word segmentation tool for Chinese texts.

The segmentation results contained some meaningless words and special characters, such as the auxiliary words “le” and “de”. These words and characters are collectively referred to as stop words, which need to be removed. Thus, the stop words were removed from the Weibo comments after word segmentation, keeping only meaningful words.

3.3 Feature extraction

The CNN is a common extraction tool of text features. It usually contains the following layers: input layer, convolutional layer, pooling layer, and fully-connected layer. Specifically, the input layer converts the word vectors in the text into a matrix. The convolutional layer, as the core of the CNN, extracts advanced features from the preprocessed text through convolution, i.e., processes the convoluted eigenvectors with the ReLU activation function, and generates the output features. The pooling layer reduces data operations through down-sampling; the most popular pooling strategy is max pooling. The fully connected layer stitches the max pooled features to avoid the loss of some features. The output layer exports the evaluated emotional tendency.

As a variant of RNN, the LSTM improves the efficiency of the original network by adding a memory neuron [18]. As shown in Figure 1, a typical LSTM consists of multiple memory neurons, where x_{t-1} is the input of the current neuron, h_{t-1} is the output of the previous neuron, and δ is the activation function. These modules work together to decide whether to retain the current time information. If the current

information is retained, the importance of the current input will be measured by the tanh function and other parameters. In the LSTM, information can be memorized or forgotten selectively.

Despite its ability to avoid vanishing gradient [19] and long-term dependence, the traditional unidirectional LSTM could not effectively solve the problem of exploding gradient, and only considers the information before the current word, while ignoring the information after the word. In reality, the semantics of words are not only related to the information before them, but also related to the information after them. Thus, this paper proposes to replace LSTM with BiLSTM [20]. As shown in Figure 2, the BiLSTM is a superposition of two LSTMs. In the BiLSTM, the context information can be obtained from two directions.

In this paper, CNN is combined with BiLSTM into a parallel hybrid model called CNN_BiLSTM, which has better extraction effect than CNN or BiLSTM alone. The addition of BiLSTM aims to make up for the insufficiency of CNN in feature extraction, and obtain global text features, which further improve the accuracy of text sentiment classification.

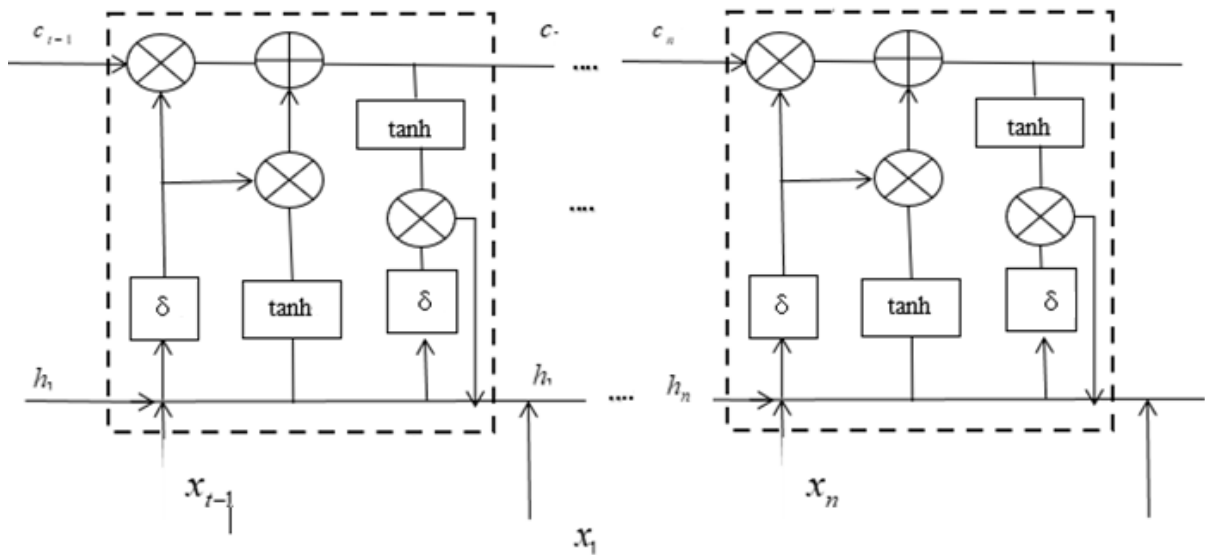


Figure 1. The structure of the LSTM

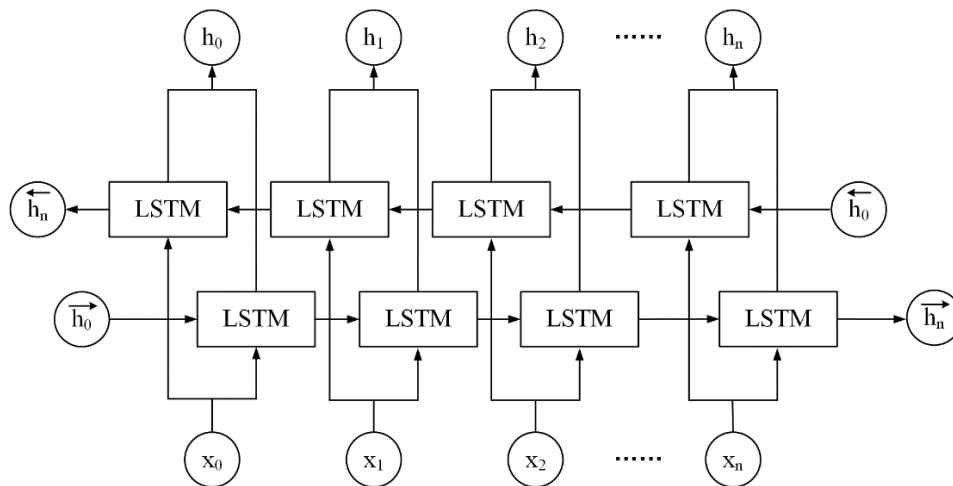


Figure 2. The BiLSTM model

3.4 Vectorization

In NLP, it is impossible for computers to directly recognize text information. Therefore, the first step of text sentiment classification is to convert the text data into numerical vectors that can be recognized and processed by the computer.

In China, the earliest text representation method is distributed representation. The most typical strategy is one-hot coding. As an effective encoding method, one-hot coding vectorizes the text by expressing every time a word appears in a sentence as 1, and the rest as 0. Despite its simplicity, this representation method might induce problems like the curse of dimensionality, because it produces high-dimensional sparse vectors, ignoring the semantic information between words.

Later, the word vector method appeared thanks to scientific development. This technique represents high-dimensional sparse vectors into low-dimensional dense vectors, providing an effective solution to the curse of dimensionality. In addition, the word vector method can clarify the ambiguous words, and overcome matrix sparseness in eigenvector extraction.

Weibo comments, written by different users, are free in form, with no fixed grammar or pattern. Thus, the texts of these comments must be transformed into vectors or a matrix. In this paper, Word2vec [21] is selected to train the representation relationship between words, turning text words into vectors with semantic value.

3.5 CNN_BiLSTM model

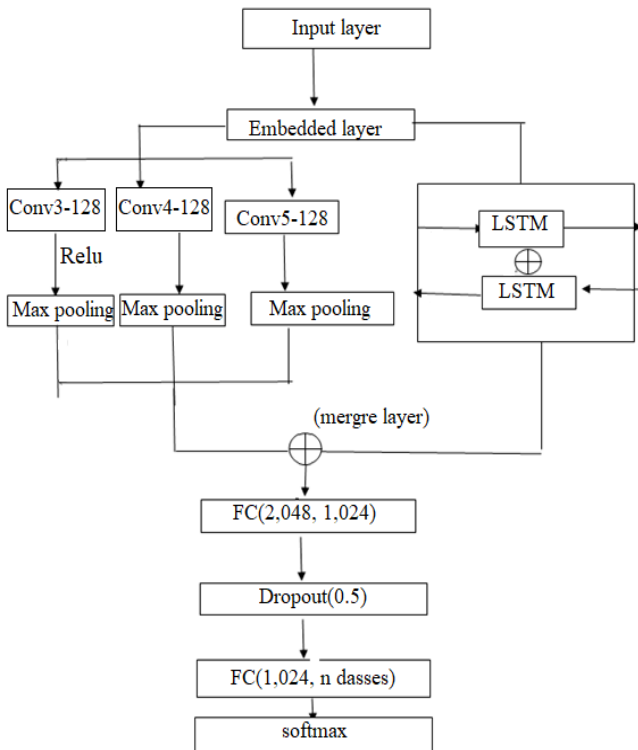


Figure 3. The structure of CNN_BiLSTM model

The CNN is only able to extract local features of the text, failing to acquire the global features. To make up for the defect, this paper integrates CNN with BiLSTM into a parallel hybrid model called CNN_BiLSTM. As shown in Figure 3, the proposed model consists of two parts: the CNN on the left, and the BiSLTM on the right. The left part extracts local features from the text, while the right part extracts the global features

of the text. In this way, our model could obtain the contextual semantics from the text, while mining the local text features. The features obtained by CNN are fused with those obtained by BiLSTM, further enhancing the accuracy of text semantic classification.

4. EXPERIMENTS AND RESULT ANALYSIS

4.1 Experimental environment

The experimental program was compiled in Windows 10, using Python3.7. The simplifyweibo_4_moods dataset was taken as the corpus, and segmented by Jieba. The dataset contains Weibo comments with semantic tags. The DL framework is PyTorch. The experimental environment for semantic classification is illustrated in Table 1 below.

Table 1. The experimental environment

System configuration	Name
Operating system	Windows 10
Word segmentation tool	Jieba
DL framework	PyTorch
Corpus	simplifyweibo_4_moods dataset
Programming language	Python3.7

4.2 Experimental data

To verify its effectiveness, the proposed parallel hybrid model CNN_BiLSTM was applied to simplifyweibo_4_moods dataset (Table 2). which provides over 360,000 Chinese texts of Weibo comments. The comments have four kinds of tags: pleased (~200,000), angry (~50,000), disgusted (~50,000), and depressed (~50,000). The dataset was divided into a training set (80%) and a test set (20%). The F1-score of repeated experiments was taken as the final evaluation result.

Table 2. The simplify weibo_4_moods dataset

Sentiment class	Sentiment	Number of comments
PA	Pleased	199,496
NA	Angry	51,714
	Disgusted	55,267
	Depressed	55,267

4.3 Hyperparameter settings

Jieba was utilized to segment the texts of Weibo comments into words, and assign part of speech (POS) tags to them. Then, Word2vec was adopted to train word vectors. Before model training, many model parameters need to be configured, namely, dropout rate, regularization method, kernel size, and number of kernels. Because model has different sensitivities to different parameters, the selection of parameter values greatly affects the accuracy and F1-score of text sentiment classification.

For our experiments, the word vector dimension was set as 128, the size of the hidden LSTM layer as 256, the activation function as ReLU, and the minimum batch size of training as 16. A total of 100 kernels were selected from the value range of $\{(2, 3, 4), (3, 4, 5), (4, 5, 6)\}$. Adam was chosen as the optimization function, the learning rate was initialized as 0.001, and the cross entropy was defined as the loss function.

To prevent overfitting, a dropout layer (dropout rate=0.5) was added between convolutional layer and fully-connected layer.

Considering the features of Chinese language processing, multiple model parameters and potential data factors were selected as the factors to be tested. By fixing all the other parameters, one factor was adjusted at a time to reflect its impact on text classification accuracy. Through thorough consideration of how parameters affect the accuracy of text classification, some parameters of the CNN_BiLSTM model were selected as follows (Table 3).

Table 3. The parameter selection

Parameter	Attribute
Word vector dimension	128
Dropout rate	0.5
Hidden layer size	256
Loss function	Cross entropy
Optimization function	Adam
Learning rate	0.001
Minimum batch size	16

4.4 Data preprocessing

The Weibo comments were crawled online. Due to the diversity of writers, these comments are free in form, with no fixed grammar or pattern. Besides, the text data of these comments are complicated, involving text information of semantic values, and numerous redundant data. To prevent the redundant data from affecting text semantic classification, the texts of Weibo comments were preprocessed in the following steps:

Step 1. Remove the punctuation marks and other special symbols from the text data, and only retain the key information with semantic values.

Step 2. Segment the text data into words with Jieba.

Step 3. Remove the stop words that simultaneously appear in the stop word lists released by Harbin Institute of Technology, Baidu, and Machine Intelligence Laboratory of Sichuan University [22], eliminating the redundant data.

Step 4. Digitize the tags of text data: assign 0 to pleased, 1 to anger, and 2 to disgusted and depressed.

4.5 Evaluation metrics

In ML, the common evaluation metrics include the area under the curve (AUC), precision, recall, F1-score, accuracy, etc. The performance of classification model is usually evaluated by the confusion matrix, a 2×2 table of true positive (TP), false positive (FP), true negative (TN), and false negative (FN). The precision, recall, F1-score, and accuracy can be respectively calculated by:

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$$F1 = \frac{2PR}{P + R}$$

$$\text{accuracy} = \frac{\text{correct}}{\text{total}}$$

where, $\text{Correct} = \text{True_Positive} + \text{True_Negative}$;
 $\text{Total} = \text{Positive} + \text{Negative}$.

To evaluate the performance of our model, accuracy and F1-score were selected as the metrics. The accuracy is the number of correctly classified text data as a proportion of the total number of text data.

4.6 Results analysis

The experiments aim to verify whether CNN_BiLSTM performs well on simplifyweibo_4_moods dataset, and whether this parallel hybrid model improves the accuracy of sentiment classification. For this purpose, CNN, GRU+CNN, LSTM, TextCNN, and TextCNNBN were compared with our model under the same experimental environment (GRU: gated recurrent unit; BN: Bayesian network). The cross entropy was selected as the loss function, owing to its relatively high accuracy.

As shown in Table 4, our model achieved an accuracy 2.794% higher than that of CNN, 2.285% higher than that of GRU+CNN, 3.497% higher than that of LSTM, 4.283% higher than that of TextCNN, and 3.877% higher than that of TextCNNBN. The results intuitively reflect the good performance of our model in text sentiment classification, and superiority in terms of F1-score.

Table 4. The experimental results

Model	Accuracy	F1-score
CNN	0.74645	0.8135714
LSTM	0.73942	0.8361344
TextCNN	0.73156	0.8125001
TextCNNBN	0.73562	0.82635789
CNN+GRU	0.75154	0.7973484
CNN_BiLSTM	0.77439	0.8720238

5. CONCLUSIONS

Despite its advantage in local feature extraction, the CNN alone tends to overlook the contextual semantics between words in sentiment classification. To solve the problem, this paper proposed a parallel hybrid model called CNN_BiLSTM. In the model, global features with contextual semantics are extracted by BiLSTM, and fused with the local features extracted by CNN. The effectiveness of our model was verified on simplifyweibo_4_moods dataset. The experimental results show that, under the same conditions, our model outperformed the other methods (e.g. TextCNN and LSTM) in the accuracy of text sentiment classification and F1-score. The future research will further optimize CNN_BiLSTM. To improve the accuracy of text sentiment classification, the attention mechanism could be introduced to apply the local features extracted by CNN to the sentiment representation features of BiLSTM, enhancing the ability of BiLSTM to analyze the semantic information between words. In addition, the generative adversarial network (GAN) could be added to CNN_BiLSTM, in an attempt to improve the accuracy of text sentiment classification.

ACKNOWLEDGEMENTS

This work was supported by National Social Science Foundation (Grant No.: 16XXW006) and Higher Education

REFERENCES

- [1] Wang, H., Zhou, C.D., Li, L.X. (2019). Design and application of a text clustering algorithm based on parallelized K-means clustering. *Revue d'Intelligence Artificielle*, 33(6): 453-460. <https://doi.org/10.18280/ria.330608>
- [2] Bansal, N., Sharma, A., Singh, R.K. (2019). An evolving hybrid deep learning framework for legal document classification. *Ingénierie des Systèmes d'Information*, 24(4): 425-431. <https://doi.org/10.18280/isi.240410>
- [3] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *European Conference on Machine Learning*. Berlin: Springer, pp. 137-142. <https://doi.org/10.1007/BFb0026683>
- [4] Sikder, S., Metya, S.K., Goswami, R.S. (2019). Exception-tolerant decision tree/rule based classifiers. *Ingénierie des Systèmes d'Information*, 24(5): 553-558. <https://doi.org/10.18280/isi.240514>
- [5] de Vries, A.D., Mamoulis, N., Nes, N., Kersten, M. (2002). Efficient KNN Search on vertically decomposed data. *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*. Madiso: ACM Press, pp. 322-333. <https://doi.org/10.1145/564691.564729>
- [6] Reddy, C.V.R., Reddy, U.S., Kishore, K.V.K. (2019). Facial emotion recognition using NLP and SVM. *Traitement du Signal*, 36(1): 13-22. <https://doi.org/10.18280/ts.360102>
- [7] Shang, C., Li, M., Feng, S.Z., Jiang, Q.S., Fan, J.P. (2013). Feature selection via maximizing global information gain for text classification. *Knowledge-Based Systems*, 54: 298-309. <https://doi.org/10.1016/j.knosys.2013.09.019>
- [8] Zhang, D.W., Xu, H., Su, Z.C., Xu, Y.F. (2015). Chinese comments sentiment classification based on word2vec and SVM perf. *Expert Systems with Applications*, 42(4): 1857-1863. <https://doi.org/10.1016/j.eswa.2014.09.011>
- [9] Chirra, V.R.R., Uyyala, S.R., Kolli, V.K.K. (2019). Deep CNN: A machine learning approach for driver drowsiness detection based on eye state. *Revue d'Intelligence Artificielle*, 33(6): 461-466. <https://doi.org/10.18280/ria.330609>
- [10] Neelapu, R., Devi, G.L., Rao, K.S. (2018). Deep learning based conventional neural network architecture for medical image classification. *Traitement du Signal*, 35(2): 169-182. <https://doi.org/10.3166/TS.35.169-182>
- [11] Kim, Y. (2014). Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Doha, Qatar, pp. 1746-1751.
- [12] Santos, C.N., Gatti, M. (2014). Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. *Proc of International Conference on Computational Linguistics*, pp. 69-78.
- [13] Zhang, Y., Chen, G.G., Yu, D., Yao, K.S., Khudanpur, S., Glass, J. (2016). Highway long short term memory RNNs for distant speech recognition. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5755-5759. <https://doi.org/10.1109/ICASSP.2016.7472780>
- [14] Liang, J., Chai, Y.M., Yuan, H.B., Gao, M.L., Zan, H.Y. (2015). Emotional analysis based on polarity transfer and LSTM recursive network. *Journal of Chinese Information Processing*, 29(5): 152-159. <https://doi.org/10.3969/j.issn.1003-0077.2015.05.020>
- [15] Lu, C., Huang, H., Jian, P., Wang, D., Guo, Y.D. (2017). A P-LSTM neural network for sentiment classification. *Computer Science*, 10234: 524-533. https://doi.org/10.1007/978-3-319-57454-7_41
- [16] Zhang, Y., Yuan, H., Wang, J., Zhang, X.J. (2017). Using a CNN-LSTM model for sentiment intensity prediction. *Proceedings of the 8th Work-Shop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 200-204.
- [17] Li, Y., Wang, X.T., Xu, P.J. (2018). Chinese text classification model based on deep learning. *Future Internet*, 10(11): 113. <https://doi.org/10.3390/fi10110113>
- [18] Bodapati, J.D., Veeranjanyulu, N., Shaik, S. (2019). Sentiment analysis from movie reviews using LSTMs. *Ingenierie des Systemes d'Information*, 24(1): 125-129. <https://doi.org/10.18280/isi.240119>
- [19] Greff, K., Srivastava, R.K., Koutnik, J., Steunebrink, B.R., Schmidhuber, J. (2016). LSTM: a search space odyssey. *IEEE Transactions on Neural Net-work and Learning Systems*, 28(10): 2222-2232. <https://doi.org/10.1109/TNNLS.2016.2582924>
- [20] Schuster, M., Paliwal, K.K. (2002). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11): 2673-2681. <https://doi.org/10.1109/78.650093>
- [21] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 3111-3119.
- [22] Data Hall. Stop Words Set [EB /OL]. <http://www.datatang.com/data/19300/>, accessed on Jul. 5, 2016.