2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC)

Word Level Domain-Diversity Attention Based LSTM Model for Sentiment Classification

Haoliang Zhang[1,2], Hongbo Xu[1], Jinqiao Shi[3], Tingwen Liu[1,2], Chun Liao[1,2]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]School of Computer Science, Beijing University of Posts and Telecommunications
{zhanghaoliang, hbxu, liutingwen, liaochun}@iie.ac.cn, shijinqiao@bupt.edu.cn

Abstract—Sentiment classification is an important task in Natural Language Processing research and it has considerable application significance. The complexity of human sentimental opinion implies that the hidden information such as application scenes or domains that behind the text may play an important role in the prediction of sentiment polarity. This paper presents a novel model for Sentiment Classification, Domain-Diversity Attention Mechanism based LSTM Model (DDAM-LSTM), integrating word level domain relevant features into an input-side attention mechanism of LSTM model. Firstly, we propose a representing and calculating method of domain relevant features for each word according to its context. Then we find that the common words and certain domain-specific words show obvious different distribution states as for domain tendency. On this basis, an attention mechanism is designed to assign scale weights to the words at the input side of LSTM network according to their diversity of domain tendency. By combining this unique attention mechanism with the LSTM model, we achieve the goal of fusing the implied domain knowledge with the Neural Network. Experimental results on three public benchmark datasets show that our proposed model yields obvious performance improvement.

Index Terms—deep learning, attention mechanism, sentiment analysis, natural language processing.

I. Introduction

Sentiment analysis, also known as opinion mining, is a sub task of text classification in Natural Language Processing (NLP). The research goal of this technology is to obtain the views, emotions and other tendentious opinion expressed in the text, which has important social application significance and value. With the continuous development of NLP technology, the research of sentiment analysis technology has made great progress. By now, the research of NLP technology can be divided into three stages. The core research methods of each stage are rule-based method [1] [2], statistical method [3] [4] [5] [6] and Neural Network method [7] [8] [9]. If we discuss this development process from the perspective of thinking hierarchy, we can find that the thinking level of Machine is constantly rising. In order to illustrate this process, we define the thinking levels as Knowledge Level, Feature Level, Rule Level and Calculation Level according to the thinking objectives and tasks of each level, as shown in Fig. 1.

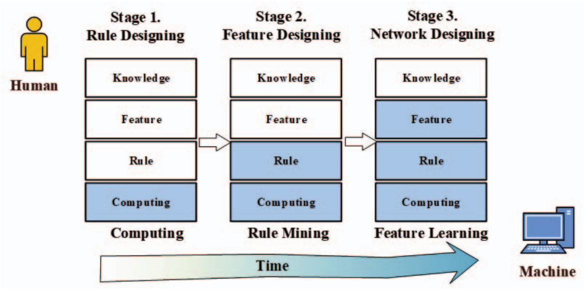


Fig. 1. The main evolution stages of Human-Machine Cooperation in Natural Language Processing.

In the first developing stage of NLP technology, thinking is mainly undertaken by Human, such as transforming their expert knowledge into useful features and logical rules which can conduct the computing process of Machine directly. In the second stage, Human doesn't need to edit the concrete rules, while on the other side Machine acquires the ability of generating appropriate rules by fitting statistical models from the training data that provided by Human. In the third stage, Human only need to raise questions and design appropriate Neural Network structures at the Knowledge Level, and then let Machine capture the features hidden in the data, generate rules and calculate the results. According to this general trend, if we want to further develop the learning ability of Machine, we should not be content to limit the Machine's thinking mode to the Feature Level, but try to make Machine further extend its vision by introducing external knowledge and even creating its knowledge through observation and comparison.

As we know, sentimental polarity is a kind of Human subjective opinion expressed by language, which means that even subtle changes in grammar or language style have a close impact on the expression of sentimental information. For example, when discussing hot issues in social media such as Twitter, texts with formal and complete grammatical structures and texts with casual and simple language styles reveal different semantic back-

This work was supported by The National Key Research and Development Program of China 2016YFB0801003.

Corresponding author

978-1-7281-9558-2/20/$31.00 ©2020 IEEE
DOI 10.1109/DSC50466.2020.00032

grounds and implied features. Therefore, perceiving the changes of language style has unique value for sentiment analysis research. If Machine derives methods of perceiving the language style differences of text, it will be able to represent the hidden knowledge of language style at the Feature Level and integrate it into the learning process of Neural Network. This kind of method for representing external knowledge may grow into a bridge crossing the Feature Layer and the Knowledge Layer. In this paper, we propose a Domain Diversity Evaluation method to represent the relative differences of word level language style based on Domain Correlation Calculation method. We take a word as the center word, and call the word and its context as Local Text for the word. The Local text with high domain diversity is considered to have more significant domain tendency, while the local text with low domain diversity shows low domain tendency.

On the basis of observing the obvious relationship between domain tendency feature and text semantics, we propose an Attention Mechanism based on domain diversity of word level text, and combine it with Bi-LSTM network by scaling word vectors at input side, and finally we propose a new model Domain-Diversity Attention Mechanism based LSTM(DDAM-LSTM) for Sentiment Classification.

The main contributions of this paper are as follows:

- We study from the perspective of the domain-related grammatical features and propose a method to represent and quantify the features of text's language style.
- We integrate the grammatical style tendency features with Neural Network model in the form of attention mechanism, and propose an sentiment analysis model DDAM-LSTM.
- By comparing the experimental results of Bi-LSTM and DDAM-LSTM, we demonstrate that the Domain-Diversity Attention Mechanism (DDAM) can improve the classification accuracy of LSTM-based sentiment analysis model.
- We further optimize the DDAM method and then analyze and discuss its optimizing effect.

II. Related Work

A. The Value of Word Vector

Compared with traditional machine learning, deep learning methods can automatically learn some effective classification features from the data. Deep learning methods initially achieved great success in image and voice processing, while it was not applied in text classification straightly. This is mainly because the problem of text's continuous representation has not been solved. As we know, image and voice are sensory signals from the real world, which naturally have the characteristics of continuity and local correlation, while words are artificial abstract signals. Although words in a sentence have continuity in the abstract semantic space, there is no directly available correlation between words from text perspective.

Hinton [10] proposed distributed representation ideology, which represented word with a dense continuous vector trained through unsupervised approaches. This concept was first introduced into Neural Network language models by Bengio [11], and when Mikolov released word2vec [12], a tool kit for training word vectors, the semantic representation ability of word vectors was verified. Then many word embedding approaches [13] [14] were developed to demonstrate the powerful capabilities of distributed representation of words. And since then Neural Network and deep learning technology have been introduced into NLP field [7] [15] [8].

Word vector models not only solve the problem of continuous representation of words, but also provide rich semantic information for the downstream NLP tasks as usually they are generated in the language model training tasks. During this process, the grammatical features concealed in the training corpus are gradually encoded in word vectors. In fact, the word vectors have been used as carriers of grammatical and semantic information, just like memory. In addition, the unsupervised training of word vector models also facilitates the use of large-scale corpora and feature information expansion, which make up for the lack of labeled data in downstream tasks [16]. For example, in recent years, a series of outstanding models based on BERT have emerged [9] [17] [18], which are usually trained by hundreds of gigabytes data.

However, there is still great potential for research work on releasing the value of word vectors.

- First, word vector is only used as a container of text semantics, the extraction and utilization of the semantic information are completely dependent on the performance of Neural Network model in the downstream tasks.
- Second, although larger models and more parameters can continuously improve the application effect, they also bring huge calculation costs and limited application areas.
- Last, as the memory of text features, how to release the semantic information encoded in word vector actively deserves more research work [19].

B. Neural Network and Deep Learning

By now, Neural Network models in NLP tasks can be divided into three categories, CNN-based [7] [15], RNN-based [8] [20] and Transformer-based [21] [9] [17] [18]. CNN and RNN based models are good at capturing local features. For capturing long-range text features, attention mechanism is employed to work with Neural Networks together. While Transformer is a unique Neural Network model which is completely based on self attention mechanism. It has achieved excellent application results in most NLP tasks with the help of its powerful feature-capturing ability and its large-scale network parameters.

However, as the initial motivation of the research on these Neural Network models is to learn the internal characteristics from the training data, less attention is paid to integrating external knowledge.

C. External Knowledge Integration

In order to integrate the external knowledge into the training process, some researches try to involve the external knowledge such as lexicon features [22], emojis [23], knowledge graph [24] or domain information [25] [26] into the learning model, and others try to transfer the knowledge by training cross lingual [27] or cross tasks [28]. Although integrating external knowledge can improve the learning effect, it needs a lot of work to find a suitable way for injecting knowledge into Neural Networks. In addition, the cross task and cross lingual learning methods still depends on labeled data, so it is hard to collect enough labeled data.

D. Domain Relevance Evaluation Method

For the purpose of quantifying the relevance between the target text and the specific domain, our previous work [29] proposed a method called Domain Relevance Evaluation (DRE) for short.

The core idea is to learn the implied grammatical features of the specific domain text according to the domain word vector, and predict the probability of matching the local text context with the central word. The higher the predicted matching probability is, the more the text conforms to the specific domain, and the lower the predicted probability is, the less the target text conforms to the specific domain.

Through this way, we have developed a kind of Machine capability of evaluating the implied correlations with various domains for a text with the help of domain word vector models, which are used as external knowledge bases of specified domains. That is to say, the word vector model is no longer limited to provide Machine with a continuous representation of words, it can also tell Machine how much the target text style fitting with a domain, just like a sensor.

The key process of Domain Relevance Evaluation method is to train domain word vector models with Negative-Sampling based CBOW model, which will optimize the formula of (e.g., (1)).

$$g(w) = \sigma(x_w^T \theta^w) \prod_{u \in NEG(w)} \left[1 - \sigma(x_w^T \theta^u)\right] \qquad (1)$$

and maximize the value of $g(w)$ through training on domain corpora. The first half formula $\sigma(x_w^T \theta^w)$ indicates the probability of word w occurring as central word when given the context $Context(w)$, that can also be explained as the probability of meeting the grammatical habits or features of the pre-trained corpus. As formula (e.g., (2)).

$$p(w|Context(w), E_{domain}) = \sigma(x_w^T \theta^w) \qquad (2)$$

where x_w is the sum of word vectors of the words in Context(w) and $\theta^w \in R^m$ (m is the size of the word vector) is the auxiliary vector for word w. E_{domain} is a given word vector model trained on a specific domain corpus.

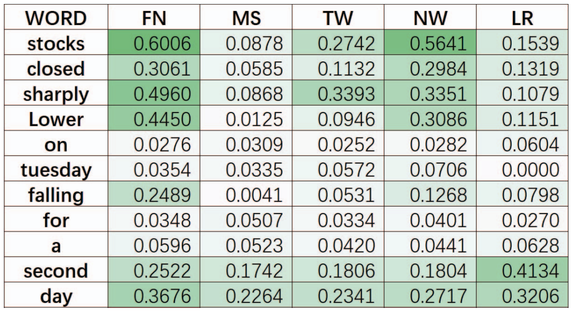
Being trained with Negative-Sampling based CBOW model on corpus of Specific Domain, the domain word vector has some new characteristics such as:

- Due to the large-scale domain text training, the statistical features encoded in the domain word vector model can not build a comprehensive feature set. In addition to the general grammatical features, the domain word vector model mainly contains domain-oriented text features.
- The domain word vector has the ability to quantitatively evaluate domain relevance for the given target text, according to e.g., (2), the parameters in the domain word vector model can be used to calculate the correlation degree of the target text to a specific domain.

III. Proposed Model

A. Domain-Diversity of Word

Applying Domain Relevance Evaluation (DRE) method to calculate the relevance value respectively with multiple domains for sample sentences, we find that the multi-domain relevance of some words are relatively close in value, while some are significantly different. Taking the sentence ”stocks closed sharp lower on Tuesday falling for a second day” as an example, we calculated the domain relevance values in five domains, including Financial News (FN), Movie Subtitles (MS), Twitter (TW), Comprehensive News (NW) and Literature Works (LR), as shown in Fig. 2.



WORD	FN	MS	TW	NW	LR
stocks	0.6006	0.0878	0.2742	0.5641	0.1539
closed	0.3061	0.0585	0.1132	0.2984	0.1319
sharply	0.4960	0.0868	0.3393	0.3351	0.1079
Lower	0.4450	0.0125	0.0946	0.3086	0.1151
on	0.0276	0.0309	0.0252	0.0282	0.0604
tuesday	0.0354	0.0335	0.0572	0.0706	0.0000
falling	0.2489	0.0041	0.0531	0.1268	0.0798
for	0.0348	0.0507	0.0334	0.0401	0.0270
a	0.0596	0.0523	0.0420	0.0441	0.0628
second	0.2522	0.1742	0.1806	0.1804	0.4134
day	0.3676	0.2264	0.2341	0.2717	0.3206

Fig. 2. Domain relevance of a sample sentence on 5 domains.

Cells with high relevance value are marked with darker color, while cells with low relevance value are marked with lighter color. For example, the word ”stocks” has the highest relevance in the domains of Financial News (FN) and Comprehensive News (NW), while it has very low relevance in the domain of Movie Subtitles (MS). And another words like ”on”, ”Tuesday”, ”for” and ”a” do not show significant correlation with all domains.

As we know that ”stocks closed sharp lower” in this example is a typical domain specific term for stock analysis, which obviously contribute more semantic information, and these words also show great diversity in correlation with multiple domains. Therefore, we propose the hypothesis that there is a close correlation between the domain diversity and semantic contribution for the words in a sentence. In order to verify this hypothesis, we define the conception of Domain Diversity of words which can be calculated by (e.g., (3)).

$$d = \sqrt{\frac{1}{n-1}\sum_{1}^{n}(x_i - \overline{x})} \tag{3}$$

where n is the number of domains involved in the calculation, and x_i is the correlation value between the word and the specific domain.

B. Domain-Diversity Attention

Regarding word vector as a signal to represent word meaning, a reasonable way to achieve enhancement effect is enlarging signal amplitude for important signals and reducing it for unimportant signals.

According to our hypothesis, larger weights should be assigned to the words with high Domain Diversity. So we set the weight assigned to each word as scaling weight. For important words, weight values that more than 1 should be assigned, and for unimportant words, values less than 1 should be assigned. For the mean weight to be 1, the sum of all weights should be equal to the sentence length, and the weight calculation formula is as (e.g., (4)).

$$\alpha_i = \frac{e^{d_i}}{\sum_{k=1}^{n} e^{d_k}} n \tag{4}$$

where d_i is the domain diversity corresponding to the target word, n is the length of the sentence, that is, the number of words. Under normal conditions, the attention mechanism is used to sum the weights of the hidden layer to generate the sentence representation vector, so the sum of the attention weights is required to be 1. While, this paper proposes a slightly different way to allocate attention, because the purpose of application here is to scale the signal rather than weighted sum. The complete calculation flow of attention weight is illustrated in Fig. 3. We use multi-domain word vectors to evaluate the word level domain relevance of the text. After multi-domain dispersion calculation and scaling range calculation, we get the word level scaling weight.

C. DDAM-Optimized Algorithm

In the case of words with similar Domain Diversity value and different mean values, it is necessary to make some slight adjustments. Since the value of domain relevance represents the probability of a word appearing in the current context, the greater the average value of multiple domain relevance, the more likely the word is to be a common or idiomatic word, and its contribution to the semantic information should be smaller. Based on the above assumptions, we use the mean value of domain-relevance as an adjustment parameter to further optimize the DDAM algorithm. The formula is as (e.g., (5)).

$$d_{opt} = \frac{1}{\overline{x}}\sqrt{\frac{1}{n-1}\sum_{1}^{n}(x_i - \overline{x})} \tag{5}$$

where $\overline{x}$ is the mean value of domain relevance.

D. Network Architecture

We combine Domain-Diversity Attention with Bi-LSTM network to construct a sentiment classification model, DDAM-LSTM model. The domain diversity weight is multiplied by the word vector for the input word at the input end, as if the signal of a word is scaled. Then the scaled word vector S_i is input into the Neural Network for training, as shown in Fig. 4. where w_i is the domain discrete attention weight calculated from the input text.

Bi-LSTM model is an improved network model based on RNN. It is good at learning sequential features, but it is not suitable for learning long distance features. While combining with DDAM, the Bi-LSTM model integrates the external knowledge of contribution weights at the word level to learn the hidden features in training data, thereby helping the model gain a global perspective.

IV. Experiments

In this paper, the performance of the model is compared and analyzed according to the classification accuracy. The main experimental objectives are as follows:

1) By comparing DDAM-LSTM model with Bi-LSTM model, the application effect of Domain-Diversity Attention Mechanism is verified.
2) Through the comparison of the classification experiment results, it is verified whether the application of DDAM optimized algorithm can bring improvement.

A. Data Sets

We tested and compared the results on three open sentiment classification datasets.

- MR [30]: Sentence polarity dataset which we used for sentiment classifier model training.
- Sentiment140 [31]: A popular dataset for sentiment analysis on tweets, which has three types of polarity label, negative, neutral and positive. For validating our model, we only use the negative and positive sentences in our experiments.
- SST-2 [32]: Stanford Sentiment Treebank data, which is published by Stanford University and commonly used for sentiment classification. It consists of sentences that come from people's review texts of movies. There are only two categories of label, positive and negative in this dataset.

B. Experimental Setup

We use the Bi-LSTM model as the basic Neural Network structure, apply the DDAM and DDAM-Optimized methods to the input side of the network, insert the Scaled Embedding Layer, and propose the DDAM-LSTM model and Optimized DDAM-LSTM model for Sentiment classification.

In this experiment, Bi-LSTM, DDAM-LSTM and the Optimized DDAM-LSTM model are trained with the same Neural Network parameter settings as follows:

TABLE I
Experimental Settings for LSTM based Networks

Parameters	Settings
Word Embedding for input	Google word2vec
Dimension of word vector	300
Activation function	ReLU
Batch size	64
Gradient Learning method	Adam
Dropout rate	0.2

V. Results and Analysis

Since the purpose of this experiment is limited to verifying the enhancement effect of the DDAM method on the Bi-LSTM model under the same experimental parameter settings, we have not performed complex model parameter optimization and training techniques, and the experimental results cannot be compared with the most advanced BERT based models.

A. Numerical Comparative Analysis

We compare the performance of the three methods as the results in Table II. Through the comparison of classification accuracy results, we can see that our model gets better results than Bi-LSTM model in all the comparisons on the three data sets.

TABLE II
Table of our Experimental Results

Data Set	Accuracy of Models		
	Bi-LSTM	DDAM-LSTM	Opt-DDAM-LSTM
MR	79.15	79.72	80.27
SST-2	83.49	84.29	84.04
Sent140	78.85	79.35	79.36

Among them, DDAM-LSTM on SST-2 data set has the most obvious improvement effect. The experimental results show that the Domain-Diversity Attention Mechanism at the input can improve the performance of the classification model, which mainly comes from the expansion and application of domain related knowledge. In addition, it also proves that our hypothesis that the higher the Domain-Diversity degree is, the more important the word is, is valid.

In addition, we compare the results of DDAM-LSTM model and Optimized DDAM-LSTM model, and find that DDAM-Optimized method has obvious improvement effect on MR data set, and has no obvious improvement effect on Sentiment140 data set, while as to SST-2 data set its performance is not so good as that of DDAM method.

Through this comparison, although we can not firmly prove that DDAM-Optimized method performs superior to DDAM, it also can be considered as a supplement to DDAM method.

B. Comparative Analysis of Training Curve

We can acquire a deeper understanding of the performance of the models by comparing the accuracy curve in the training process upon the target models.

Comparing the training curves of DDAM-LSTM model and Bi-LSTM model on MR data set, as shown in Fig. 5 , we can see that DDAM-LSTM model learns faster

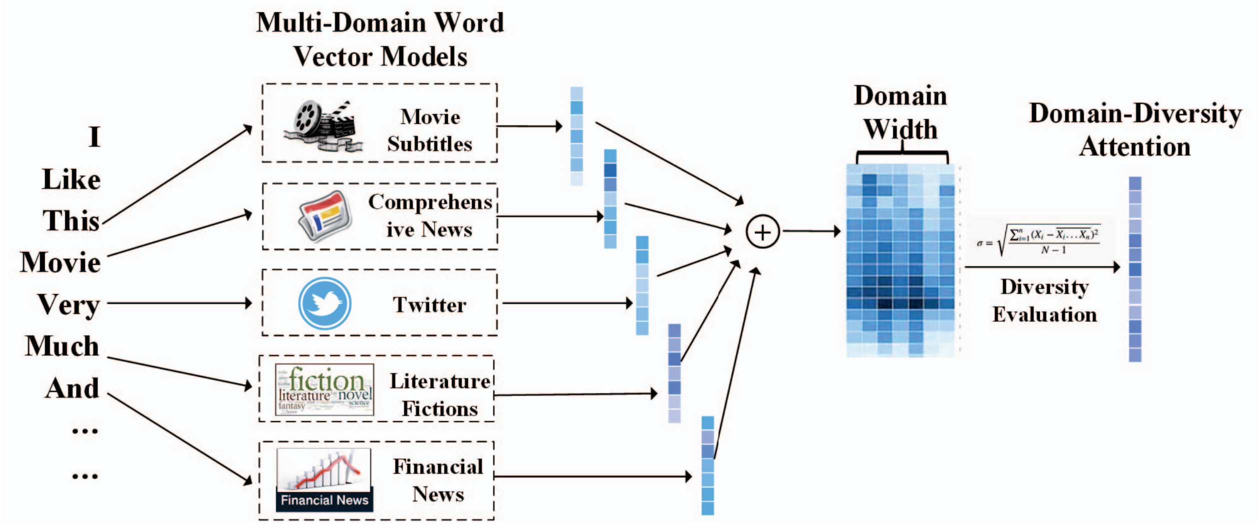

Multi-Domain Word Vector Models
I
Like
This
Movie
Very
Much
And
...
...
Movie Subtitles
Comprehensive News
Twitter
fiction literature novel fantasy
Literature Fictions
Financial News
Financial News
+
Domain Width
$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \overline{X_i \ldots X_n})^2}{N-1}}$
Diversity Evaluation
Domain-Diversity Attention

Fig. 3. The generation method of DDAM-Attention weights.

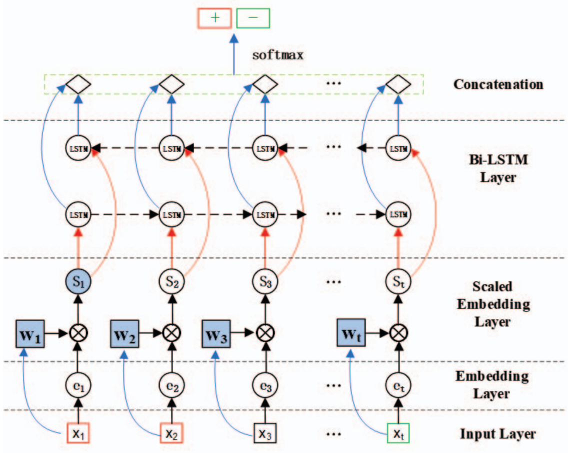


Fig. 4. The structure of DDAM-LSTM Neural Network.

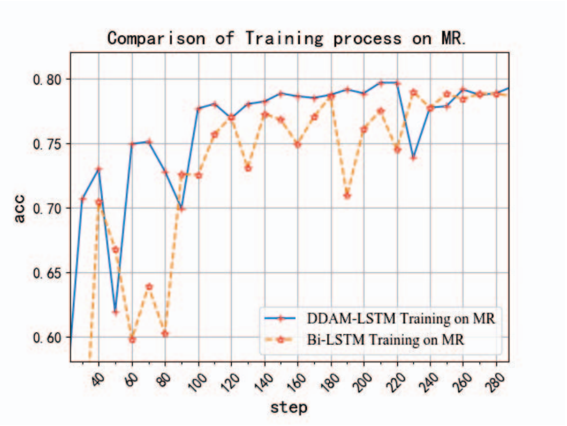


Fig. 5. The training processes of DDAM-LSTM compared with Bi-LSTM on MR.

than Bi-LSTM model, and the performance fluctuation of DDAM-LSTM is significantly smaller than that of Bi-LSTM model.

Comparing the training curves of optimized DDAM-LSTM model to that of DDAM-LSTM model on MR data set, as shown in Fig. 6, we find that while the accuracy rate is slightly improved, the fluctuation range of the curve is obviously more stable.

Through the comparisons of the training curves, the practical value of DDAM-LSTM model and Optimized DDAM-LSTM model is demonstrated.

VI. Conclusions And Future work

In this paper, we introduce a novel input-side Attention Method (Domain-Diversity Attention) and propose DDAM-LSTM model and Optimized DDAM-LSTM model. The Domain-Diversity Attention method is designed to assign weights for words according to their semantic contributions which can be reflected by domain dispersion features. The LSTM model is good at capturing text sequence features. If the word vectors are scaled according to the semantic contribution at the input side, it can help LSTM model to learn the correlation features between important words more effectively, thereby improving its sentiment classification effect. Based on this principle, we applied Domain-Diversity Attention to LSTM and develop DDAM-LSTM and Optimized DDAM-LSTM. Our experiments on three datasets for sentiment analysis demonstrate the effectiveness of our models, meanwhile it also prove that Domain-Diversity Attention can indicate the semantic contribution of words at the word level.

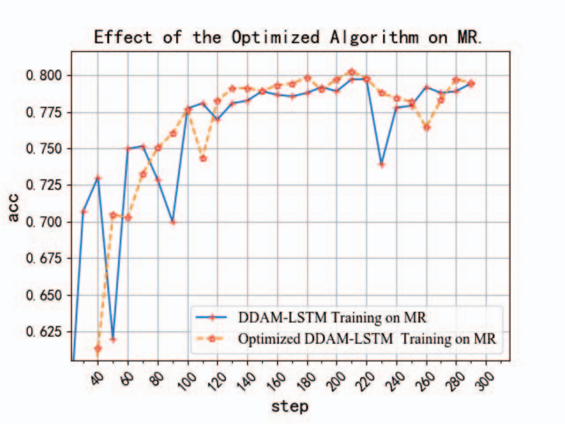


Fig. 6. The training processes of DDAM-LSTM compared with Optimized DDAM-LSTM on MR.

Acknowledgment

We would like to thank Xiaofei Zhou and Qiannan Zhu for their helpful suggestions and comments.

References

[1] L. Barque and F.-R. Chaumartin, “Regular polysemy in wordnet.” J. Lang. Technol. Comput. Linguistics, vol. 24, no. 2, pp. 5–18, 2009. [Online]. Available: http://dblp.uni-trier.de/db/journals/ldvf/ldvf24.html#BarqueC09

[2] C. Beierle, U. Hedtstück, U. Pletat, P. H. Schmitt, and J. H. Siekmann, “An order-sorted logic for knowledge representation systems,” IWBS Report, vol. 113, 1990.

[3] A. McCallum and K. Nigam, “A comparison of event models for naive bayes text classification,” 1998. [Online]. Available: citeseer.ist.psu.edu/article/mccallum98comparison.html

[4] S. F. Chen and R. Rosenfeld, “A survey of smoothing techniques for me models.” IEEE Trans. Speech and Audio Processing, vol. 8, no. 1, pp. 37–50, 2000. [Online]. Available: http://dblp.uni-trier.de/db/journals/taslp/taslp8.html#ChenR00

[5] A. L. BERGER, “A maximum entropy approach to natural language processing,” Computational Linguistics, vol. 22, no. 1, pp. 39–71, 1996.

[6] H. He, Z. Li, C. Yao, and W. Zhang, “Sentiment classification technology based on markov logic networks,” The New Review of Hypermedia and Multimedia, vol. 22, no. 3, pp. 243–256, 2016.

[7] Y. Kim, “Convolutional neural networks for sentence classification,” Eprint Arxiv, 2014.

[8] Z. C. Lipton, J. Berkowitz, and C. Elkan, “A critical review of recurrent neural networks for sequence learning,” arXiv preprint arXiv:1506.00019, 2015.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018, cite arxiv:1810.04805Comment: 13 pages. [Online]. Available: http://arxiv.org/abs/1810.04805

[10] G. E. Hinton, "Learning distributed representations of concepts," in Parallel Distributed Processing: Implications for Psychology and Neurobiology, R. G. M. Morris, Ed. Oxford, England: Clarendon Press, 1989, pp. 46–61.

[11] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model." J. Mach. Learn. Res., vol. 3, pp. 1137–1155, 2003. [Online]. Available: http://dblp.uni-trier.de/db/journals/jmlr/jmlr3.html#BengioDVJ03

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in Advances in Neural Information Processing Systems 26, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119. [Online]. Available: http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality

[13] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in EMNLP, vol. 14, 2014, pp. 1532–1543.

[14] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018, cite arxiv:1802.05365Comment: NAACL 2018. Originally posted to openreview 27 Oct 2017. v2 updated for NAACL camera ready. [Online]. Available: http://arxiv.org/abs/1802.05365

[15] M. J. Er, Y. Zhang, N. Wang, and M. Pratama, "Attention pooling-based convolutional neural network for sentence modelling." Inf. Sci., vol. 373, pp. 388–403, 2016. [Online]. Available: http://dblp.uni-trier.de/db/journals/isci/isci373.html#ErZWP16

[16] B. Li, A. Drozd, Y. Guo, T. Liu, S. Matsuoka, and X. Du, "Scaling word2vec on big corpus." Data Science and Engineering, vol. 4, no. 2, pp. 157–175, 2019. [Online]. Available: http://dblp.uni-trier.de/db/journals/dase/dase4.html#LiDGLMD19

[17] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," Bioinformatics, vol. 36, no. 4, pp. 1234–1240, 09 2019. [Online]. Available: https://doi.org/10.1093/bioinformatics/btz682

[18] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: Improving pre-training by representing and predicting spans," 2019, cite arxiv:1907.10529Comment: Accepted at TACL. [Online]. Available: http://arxiv.org/abs/1907.10529

[19] H. Zhang, H. Xu, and J. Shi, "Sentence-sfv model: Make full use of word embedding actively," in 2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS), 2019, pp. 179–183.

[20] X. She and D. Zhang, "Text classification based on hybrid cnn-lstm hybrid model," in 2018 11th International Symposium on Computational Intelligence and Design (ISCID), 2018.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, p. 5998–6008. [Online]. Available: https://papers.nips.cc/paper/7181-attention-is-all-you-need

[22] K. Margatina, C. Baziotis, and A. Potamianos, "Attention-based conditioning methods for external knowledge integration," 2019.

[23] Z. Chen, S. Shen, Z. Hu, X. Lu, Q. Mei, and X. Liu, "Emoji-powered representation learning for cross-lingual sentiment classification," in The World Wide Web Conference, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 251–262. [Online]. Available: https://doi.org/10.1145/3308558.3313600

[24] A. Kumar, D. Kawahara, and S. Kurohashi, "Knowledge-enriched two-layered attention network for sentiment analysis," 2018.

[25] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 513–520.

[26] N. X. Bach, V. T. Hai, and T. M. Phuong, "Cross-domain sentiment classification with word embeddings and canonical correlation analysis," in Proceedings of the Seventh Symposium on Information and Communication Technology, ser. SoICT '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 159–166. [Online]. Available: https://doi.org/10.1145/3011077.3011104

[27] G. Lample and A. Conneau, "Cross-lingual language model pretraining," arXiv preprint arXiv:1901.07291, 2019.

[28] G. Lee, E. Yang, and S. Hwang, "Asymmetric multi-task learning based on task relatedness and loss," in International Conference on Machine Learning, 2016, pp. 230–238.

[29] H. Zhang, H. Xu, J. Shi, T. Liu, and J. Ya, "Sfv-cnn: Deep text sentiment classification with scenario feature representation," in International Conference on Mathematical Aspects of Computer and Information Sciences. Springer, 2019, pp. 382–394.

[30] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in Proceedings of the 43rd annual meeting on association for computational linguistics. Association for Computational Linguistics, 2005, pp. 115–124.

[31] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," CS224N project report, Stanford, vol. 1, no. 12, p. 2009, 2009.

[32] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in Proceedings of the 2013 conference on empirical methods in natural language processing, 2013, pp. 1631–1642.