Automatic Essay Scoring Model Based on Two-Layer Bi-directional Long-Short Term Memory Network

Linzhong Xia
Shenzhen Institute of Information Technology
No.2188, Longxiang Road, Longgang District, Shenzhen, China
0755-89226577, 86
43966506@qq.com

Jun Liu
Shenzhen Institute of Information Technology
No.2188, Longxiang Road, Longgang District, Shenzhen, China
0755-89226244, 86
liuj@sziit.edu.cn

Zhenjiu Zhang
Shenzhen Institute of Information Technology
No.2188, Longxiang Road, Longgang District, Shenzhen, China
0755-89226591, 86
Zhangzhenjiu@sziit.edu.cn

ABSTRACT

Automatic essay scoring provides a cost-effective and consistent alternative to human correction. However, in order to obtain good performance, human experts are needed to extract features of text manually in traditional ways. We propose a two-layer bi-directional long-short term memory model that a fully automatic essay scoring model. The agreement of essay scores which marked by our model and human raters respectively has achieved to 0.870 based on metrics of Cohen's κ. This model can achieve excellent results like human beings' professional raters.

CCS Concepts

•Applied computing→Document management and text processing→Document capture→Document analysis.

Keywords

Automatic essay scoring; long-short term memory; word embeddings.

1. INTRODUCTION

The development of Chinese society has been deeply integrated into the trend of world development, and its internationalization is becoming more and more high. Therefore, the popularity of English learning in China has become more and more widespread. And English courses have been offered in primary school, middle school and university in China. Now, the test of students' English level has become an important thing. There are four main indexes which are the ability of listening, speaking, reading and writing to reflect the students' English level. Based on the statistical data, it is found that students' writing scores are generally low. There are two reasons for this result. Firstly, a lot of English writing practice is a necessary condition to improve English level. With this, teacher's correction will be huge. Therefore, it is easily leading to teachers unable to give timely feedback. Secondly, if students can't get quick feedback, their motivation to continue to practice writing will be insufficient. Therefore, it is particularly important to develop a kind of automatic essay scoring system to release teachers from the enormous composition correction.

2. RELATED WORK

With the rapid development of the computer science and natural language processing (NLP), automatic essay scoring (AES) by computer becoming reality. Therefore, a series of AES systems have been developed, such as Project Essay Grader (PEG) [1], Intelligent Essay Assessor (IEA) [2], E-rater [3], IntelliMetric [4], Bayesian Essay Test Scoring System (BETSY) [5], and so on. Those AES systems mainly make use of the features of lexical, syntax, readability, topic relevance, discourse coherence and cohesion to predict score for essays. For example, PEG applies the shallow linguistic features of the texts to analyze and score the essays. IEA compares content similarity between a student's essay and other essays at the same topic marked by human beings' raters to predict how closely they match. E-rater uses statistical natural language processing techniques to extract linguistic features and then score the essays. BETSY is a very mature English composition scoring system. BETSY not only integrates the advantages of PEG, E-rater and IEA and other systems, but also has its own unique characteristics. In addition, it is worth mentioned that BETSY is the only automatic essay scoring system that can be downloaded for free at present.

Recently, with the rapid advances in deep learning and computing power, AES based on neural network becoming more and more powerful. Recurrent neural network has been shown to perform very well on the task of language modeling [6,7,8]. Zhou et al. proposed a neural network automatic English composition scoring algorithm which can well perform the complex relationship between features and essay score [9]. Dimitrios Alikaniotis et al. proposed a score-specific word embeddings (SSWEs) bi-directional long-short term memory network (BLSTM) model whose ability of automatic essay scoring had surpassed similar state-of-the-art systems [10]. Li et al. proposed a cognitive-based automated essay scoring model which is composed of sensory acquisition, score analyzer and background knowledge constructor [11]. In a word, the AES systems based on neural network have two important advantages. Firstly, it can learn the required feature representations for each dataset automatically. Secondly, it is unnecessary for manual tuning. Therefore, we propose a google word2vec two-layer BLSTM model to predict score of essays which produced by middle-school English-speaking students.

CSAI2019, December 6–8, 2019, Beijing, China

ACM ISBN 978-1-4503-7627-3/19/12…$15.00
DOI: https://doi.org/10.1145/3374587.3374596

3. DATA

The data which is used in this paper was made publicly available to users of Kaggle, sponsored by the Hewlett Foundation[1]. The used dataset contains 1785 essays and the average length of essays is 350 words. The type of essay is about Persuasive/ Narrative/ Expository responses. Those essays were written by students from Grade 8. The scores of the essays come from two raters. In this paper, we have used the resolved score and the score range from 2 to 12. The data has been divided into three parts: training set (70%), validation set (12%), and test set (18%). The training set has been used for actual training and the validation set has been used for validation. The test set has been used for testing the performance of our model.

4. MODEL

4.1 Word Embeddings

In natural language processing tasks, how to represent text is very important. Usually, there are two ways to represent text: one-hot representation and distribution representation.

Traditional rule-based or statistical-based natural semantics approach treats words as atomic symbols. Therefore, each word has been represented by a long vector. The dimension of the vector is the size of the vocabulary. Only one dimension in the vector has a value of 1, and this dimension represents the current word. For example, we can use the way of one-hot representation represent words ‘Good’, ‘OK’ and ‘Cat’ as Figure 1 shown.

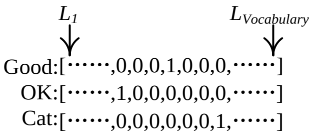

Figure 1. One-hot representation.

In Figure 1, the length of the vector ($L_{vocabulary}$) equal to the length of the vocabulary. This will cause very large feature space and generate dimension disaster. In addition, the one-hot representation can’t show the relationship between words.

Word embeddings refers to the transformation of words into a distributed representation, also known as word vector. Distributed representation represents a word as a continuous dense vector of fixed length. In applications, the value of fixed length is generally between 50 to 500. Distributed representation can reflect the relationship between words and overcome the problem of dimension disaster. For example, we can use the way of distributed representation represent words ‘Good’, ‘OK’ and ‘Cat’ as Figure 2 shown.

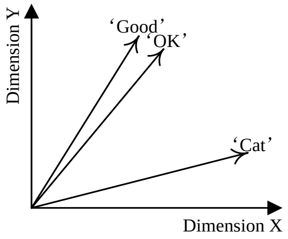

Figure 2. Distributed representation.

In Figure 2, the more similar of the meanings of different words, the smaller of the angle between them in vector space. In this paper, we have used the word embeddings which trained on the Kaggle dataset and the pre-trained word embeddings which trained on the Google News Corpus respectively.

4.2 Long-Short Term Memory Network

Language is a kind of sequential information. Recurrent Neural Network (RNN) is a best choice to process sequential information. However, with the increase of the length of the language sequence, the using effects of RNN will gradually deteriorate. In order to overcome the shortboard that RNN can’t process long-sequence text, a special kind of RNN has been invented that is called long-short term memory network (LSTM) [12,13,14]. The architecture of a LSTM cell as Figure 3 shown.

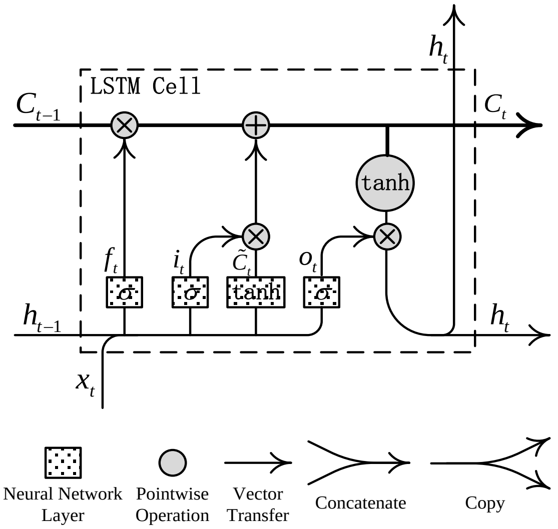

Figure 3. LSTM Cell.

In Figure 3, the output at time t is conditioned on the input both at time t and at time $t-1$. C_{t-1} is the long-term memory line at time $t-1$. C_t is the long-term memory line at time t. h_{t-1} is the short-term memory line at time $t-1$. h_t is the short-term memory line at time t. x_t is the t-th input word. f_t, i_t, o_t and $\widetilde{C_t}$ are the forget, input, output gates and the cell activation vectors respectively. The functions that are computed by the LSTM cell are given by (1) to (6):

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_f) \tag{1}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_i) \tag{2}$$

$$\widetilde{\boldsymbol{C}}_t = tanh(\boldsymbol{W}_C \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_C) \tag{3}$$

$$\boldsymbol{C}_t = \boldsymbol{f}_t * \boldsymbol{C}_{t-1} + \boldsymbol{i}_t * \widetilde{\boldsymbol{C}}_t \tag{4}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_o) \tag{5}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t * \tanh(\boldsymbol{C}_t) \tag{6}$$

where σ and $tanh$ are element-wise non-linear functions; $*$ is the Hadamard product; $\boldsymbol{W}$ and $\boldsymbol{b}$ are the learned weights and biases respectively.

4.3 Bi-directional LSTM

The essay is a kind of sequence information. Some front part of text information may be lost when training the LSTM in a uni-

[1] http://www.kaggle.com/c/asap-aes/

directional manner. To alleviate this issue, train the LSTM in a bi-directional manner has been proposed [15,16,17]. In bi-directional LSTM (BLSTM), feeding words to neural network from left to right and from right to left at the same time. The architecture of the BLSTM as Figure 4 shown. There are T words in input layer. In Embedding layer, the T words have been converted into word embeddings respectively. The BLSTM layer contains two sub-networks for the left and right sequence context, which are forward and backward pass respectively. The output of BLSTM layer as (7) shown:

$$\boldsymbol{h}_T = Concat(\overleftarrow{\boldsymbol{h}_T}, \overrightarrow{\boldsymbol{h}_T}) \tag{7}$$

where $\overleftarrow{\boldsymbol{h}_T}$ is the output state of backward hidden vectors at time T, $\overrightarrow{\boldsymbol{h}_T}$ is the output state of forward hidden vectors at time T, $\boldsymbol{h}_T$ is the concatenation of the forward and backward hidden vectors at time T.

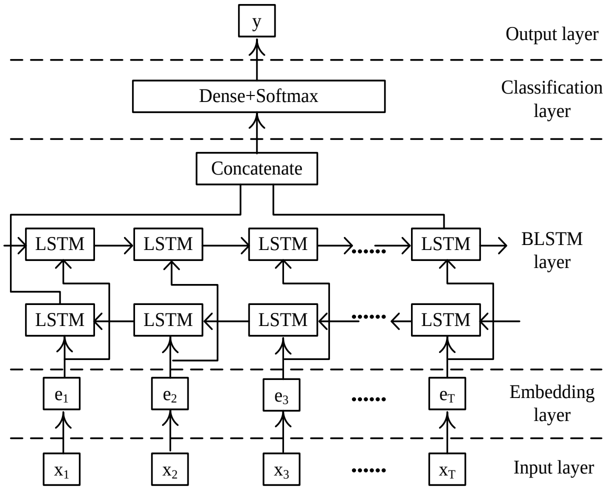

Figure 4. The architecture of the BLSTM.

5. EXPERIMENTS

5.1 Evaluation Metrics

The experiment results are evaluated upon mean absolute error (MAE), Pearson's correlation (γ), and quadratic weighted kappa (κ). Cohen et al. proposed using Kappa value as an index to evaluate the agreement of our scores and the human annotator's gold-standard. Practice has proved that it is an ideal index to describe the agreement of scores. 0 represents only random agreement between the raters and 1 is full agreement. The κ is calculated by (7):

$$\kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}} \tag{7}$$

where $w_{i,j}$ is calculated by (8):

$$w_{i,j} = \frac{(i-j)^2}{(N-1)^2} \tag{8}$$

In equation (7) and (8), N delegates possible essay ratings, an $N \times N$ matrix O is constructed where $O_{i,j}$ represents the number of essays receiving grade i from the first rater and j from the second rater. We can use the same way to construct matrix E, but assuming there is no correlation.

5.2 Results

The hyperparameters for our model are as follows: the learning rate η, the size of the LSTM layer L_{LSTM} and the dropout rate r. The word2vec (word embeddings) has been obtained by training on Kaggle dataset. The google word2vec (word embeddings) has been obtained by training on the Google News Corpus. We report the Pearson's product-moment correlation coefficient γ, and the MAE between the predicted scores and the gold-standard on our test set, which are considered more suitable metrics for evaluating essay scoring system. Meanwhile, we also report the Cohen's κ with quadratic weights, which is the evaluation metric used in the Kaggle competition. Performance of different models is shown in Table 1.

Table 1. Results of different models

Word Embeddings	model	Pearson γ	MAE	Cohen's κ
word2vec	LSTM	0.837	0.65	0.808
word2vec	BLSTM	0.819	0.65	0.811
word2vec	Two-layer BLSTM	0.833	0.628	0.824
Google word2vec	LSTM	0.854	0.579	0.849
Google word2vec	BLSTM	0.861	0.557	0.856
Google word2vec	Two-layer BLSTM	0.872	0.546	0.870

In term of Pearson γ, MAE, and Cohen's κ, Two-layer BLSTM produce competitive results, outperforming LSTM and BLSTM. Meanwhile, the models which based on pre-trained google word2vec produce better performance than the models which based on word2vec. More specifically, we find google word2vec two-layer BLSTM is the best configuration, increasing Pearson γ to 0.872 and Cohen's κ to 0.871, reducing MAE to 0.546.

5.3 Discussion

The Cohen's κ is 0.721 between rater1 and rater2. Therefore, our google word2vec two-layer BLSTM can score the essays in a very human-like way. We have used Bayesian Optimization to find optimal hyperparameters configuration in fewer steps than in regular grid search. Using this approach, we found some parameters which are associate with better models. Finally, we find the hyperparameters (= 0.000009, L_{LSTM} = 2, r = 0.5) for the best scoring model. However, the optimal value of η and L_{LSTM} may be corpus dependent.

To improve the predictive performance of our model, there are three important key points have been considered. Firstly, due to the local dataset is small, the local word2vec which trained on local dataset is weaker than google word2vec which trained on Google News Corpus. Therefore, we use the google word2vec as the input of our model to enhance the represent ability of the text. Secondly, multi-layer neural networks are known for automatically learning useful features from data. In our model, we use the architecture of two-layer LSTM, with the first layer learning basic features and the second layer learning more high-level abstract features. Thirdly, BLSTM combines the forward hidden layer and the backward hidden layer, which can abstract both the preceding and succeeding contexts. The experiment results have proved the above considerations are effectively to improve the predictive performance of our model as Table 2 shown. The relative improvement ratio Δ based on the Cohen's κ are used as the evaluation metric. The relative improvement ratio Δ is calculated by (9):

$$\Delta = (\kappa_{GTBLSTM} - \kappa_{OM})/\kappa_{OM} \qquad (9)$$

where $\kappa_{GTBLSTM}$ is the value of the model of google word2vec two-layer BLSTM and κ_{OM} is the value of other models.

Table 2 Effect of each component on the performance of google word2vec two-layer BLSTM. % is omitted

Word Embeddings	model	Cohen's κ	Δ
word2vec	LSTM	0.808	7.6
word2vec	BLSTM	0.811	7.2
word2vec	Two-layer BLSTM	0.824	5.5
Google word2vec	LSTM	0.849	2.4
Google word2vec	BLSTM	0.856	1.6
Google word2vec	Two-layer BLSTM	0.870	-

In Table 2, the google word2vec has a powerful influence on the performance of the models. Among the three architectures mentioned above, Two-layer BLSTM obtains the best results. In a word, the importance of google word2vec is highest, next is the architectures of the model.

6. CONCLUSION

In this paper, we propose a novel deep neural network model, named google word2vec two-layer BLSTM, for scoring essays. The results have shown that this kind of architecture is able to surpass similar systems based on manual feature engineering. We also analyzed the performance of different word embeddings, and the results show that the pre-trained google word2vec is better than the word2vec which trained based on the Kaggle dataset. In a word, the model which introduced in this paper can replace human beings as a rater of essays.

7. ACKNOWLEDGMENTS

This work is supported by Engineering Applications of Artificial Intelligence Technology Laboratory of Shenzhen Institute of Information Technology (Number: PT201701), the Guangdong Province higher vocational colleges & schools Pearl River scholar funded scheme (2016), and Research Funds of Shenzhen Institute of Information Technology (No. ZY201708).

8. REFERENCES

[1] Wresch, W. 1993. The imminence of grading essays by computer--25 years later. *Computers and Composition.* 10, 2 (1993), 45-48.

[2] Thomas, K. L., Darrell, L., and Peter, F. 2003. Automatic Essay Assessment. *Assessment in Education*. 10, 3 (Nov. 2003), 295-308. DOI=https://www.tandfonline.com/doi/abs/10.1080/0969594032000148154.

[3] Attali, Y., and Burstein, J. 2005. *Automated essay scoring with e-rater@v.2.0.* Research Report. ETS, Princeton, NJ.

[4] Rudner, L. M., Garcia, V., and Welch, C. 2006. An evaluation of the IntelliMetric essay scoring system. *Journal of Technology, Learning, and Assessment.* 4, 4 (Mar. 2006), 1-22.

[5] Rudner, L. M., Liang, T. 2002. Automated essay scoring using Bayes' theorem. *Journal of Technology, Learning, and Assessment.* 1, 2 (Jun. 2002), 1-21.

[6] Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., and Robinson, T. 2013. One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling. *ArXiv, Google.*

[7] Cao, W., Song, A., and Furuzuki, T. 2017. Stacked residual recurrent neural network with word weight for text classification. *IAENG International Journal of Computer Science*. 44, 3 (Aug. 2017), 277-284.

[8] Zhang, Y., Er, M. J., Venkatesan, R., Wang, N., and Pratama, M. Sentiment classification using Comprehensive Attention Recurrent models. In *Proceedings of the 2016 International Joint Conference on Neural Networks* (Vancouver, BC, Canada, July 24-29, 2016). IEEE, DOI=10.1109/IJCNN.2016.7727384.

[9] Zhou, Y., and Huang, G. 2014. An Automatic English Composition Scoring Model Based on Neural Network Algorithm. In *Proceedings of the 13th International Conference on Computer and Information Science* (Taiyuan, China, June 4-6, 2014). IEEE, DOI=10.1109/ICIS.2014.6912123.

[10] Alikaniotis, D., Yannakoudakis, H., and Rei, M. 2016. Automatic Text Scoring Using Neural Networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (Berlin, Germany, August 7-12, 2016). Association for Computational Linguistics, 715-725. DOI=10.18653/v1/P16-1068.

[11] Li, L., and Sugumaran, V. 2019. A cognitive-based AES model towards learning written English. *Journal of Ambient Intelligence and Humanized Computing.* 10, 5 (May, 2019), 1811-1820. DOI=https://doi.org/10.1007/s12652-018-0743-1.

[12] Hochreiter, S., and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation.* 9, 8 (Dec. 1997), 1735-1780. DOI= 10.1162/neco.1997.9.8.1735.

[13] Nowak, J., Taspinar, A., and Scherer, R. LSTM Recurrent Neural Networks for Short Text and Sentiment Classification. In *Proceedings of the 16th International Conference on Artificial Intelligence and Soft Computing* (Zakopane, Poland, June 11-15, 2017). Springer Verlag, 553-562. DOI=10.1007/978-3-319-59060-8_50.

[14] Alduayj, S. S., and Smith, P. Sentiment Classification and Prediction of Job Interview Performance. In *Proceedings of the 2019 2nd International Conference on Computer Applications & Information Security* (Riyadh, Saudi Arabia, May 1-3, 2019). IEEE, DOI=10.1109/CAIS.2019.8769559.

[15] Mohammed, N. A. A., Tan, G., and Hussain, A. 2018. Bidirectional Recurrent Neural Network Approach for Arabic Named Entity Recognition. *Future Internet.* 10 (Dec. 2018), 1-12. DOI=https://doi.org/10.3390/fi10120123.

[16] Fu, L., Yin, Z., Wang, X., and Liu, Y. A Hybrid Algorithm for Text Classification Based on CNN-BLSTM with Attention. In *Proceedings of the 2018 International Conference on Asian Language Processing* (Bandung,

Indonesia, November 15-18, 2018). IEEE, DOI=10.1109/IALP.2018.8629219.

[17] Zhou, P., Shi, W., Tian, J., Qi, B., Li, B., Hao, H., and Xu, B. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational* Linguistics (Berlin, Germany, August 7-12, 2016). Association for Computational Linguistics, DOI=10.18653/v1/P16-2034.