Automatic Essay Scoring with Recurrent Neural Network

Changzhi Cai
School of Telecommunication Engineering
Xidian University
Xi'an, Shaanxi Province, China
Tel: +86 13298367860
caichangzhi97@gmail.com

ABSTRACT
As deep learning has developed rapidly in recent years, the automatic essay scoring system, based on deep learning models, has become more reliable than previous feature-based systems. Recent researchers have developed an approach based on recurrent neural networks to learn the relationship between an essay and its assigned score, without any feature engineering. In this paper, we use an ASAP essay dataset, combining feature scoring and a recurrent neural network. The results show that we can compare the result of quadratic weighted Kappa of each experience to get the best model. GloVe significantly improves the results, and feature extraction can affect the result slightly. In future work, we will apply transfer learning, one-shot learning, and adversarial inputs in our model to get better performance.

CCS Concepts
• Computing methodologies → Information extraction

Keywords
Deep learning; neural network; feature extraction; automatic essay scoring; data processing; word embedding

1. INTRODUCTION
Manually scoring English papers written by students is a daunting task for evaluators. In large exams, such as the TOEFL and GRE, manual scoring requires a lot of time and effort. Consequently, Automated Essay Scoring (AES) [1] became established, supporting a large-scale use of computers to give scores and to automatically evaluate students' articles, thereby reducing the pressure on scorers. With the rapid development of artificial intelligence and deep learning, AES has also made great progress. More and more researchers in natural language processing have invested in this research, greatly increasing AES's accuracy and reliability. Simultaneously, in many English classes in middle schools and universities, teachers will use AES to correct English writing assignments in order to obtain more accurate scores.

However, AES requires significantly different scoring information for different scoring tasks. If the system gives the article an overall score, it needs to consider all the information in the article; if the system only judges one aspect of the article (e.g., no syntax error), then only this information should be evaluated. A few years ago, the Automated Student Assessment Award (ASAP) Competition, sponsored by the Hewlett Foundation in 2012, reinvigorated interest in the subject, and many subsequent studies in AES were based on this. In this project, we have also used ASAP as our dataset. The correlation between the scores assigned by the most advanced AES system and the scores assigned by human evaluators has proven to be relatively high.

In this paper, we have optimized the existing AES system based on recurrent neural network training, so that the automatic scoring results are closer to the level of English writers, thereby resulting in more accurate results. We have used ASAP as our training set and test set, extracting relevant features, performing analyses, and using mathematical models such as Quadratic Weighted Kappa (QWK). Although the method of extracting relevant features for scoring by Taghipour and Ng [2] has already been used, the improvements we have made here largely avoids the occurrence of low-probability errors, such as high-quality essays with high scores.

HP3C '19, March 8–10, 2019, Xi'an, China

ACM ISBN 978-1-4503-6638-0/19/03…$15.00
https://doi.org/10.1145/3318265.3318296

2. RELATED WORK

2.1 Particular Dimension
For a long time, users have been increasing the feature dimension of AES in order to improve its accuracy. In earlier work, Page [3] developed an AES tool called Project Essay Grade (PEG) using only language-surface features. The most recent AES system is E-Rater [4], which uses more natural language processing (NLP) techniques. Later, Attali and Burstein [5] released E-Rater V2, where they created a new set of features to represent language features related to organization and development, lexical complexity, and the use of specific vocabulary. Similar to Page [3], this system uses regression equations to evaluate student articles.

2.2 Feature Extraction
Feature extraction refers to the process of converting raw data that cannot be recognized by machine-learning algorithms into features that the algorithm can recognize. For example, in natural language processing, text is composed of a series of words that form a collection of words after segmentation.

In our project, we have extracted the characteristics of corpus articles, including article length, grammatical errors, spelling errors, and word complexity, as an important basis for judging students' English writing skills.

Previous researchers have used many feature extraction methods. For instance, EASE uses NLTK [6] for POS tagging and stemming, Aspell for spellchecking, and WordNet [7] to find synonyms. Correct POS tags are generated using a grammatically correct text (provided by EASE). The POS tag sequences not included in the correct POS tags are considered bad POS. EASE

also uses scikit-learn [8] for extracting unigram and bigram features.

2.3 Neural Network System

In many fields, such as computer vision and natural language processing, neural networks gradually replace the feature-extracting methods, which can generate better results. Neural networks, in the field of machine learning and cognitive science, are mathematical models or computational models that mimic the structure and function of biological neural networks (the central nervous system of animals used to estimate or approximate functions, especially the brain). Neural networks consist of a large number of artificial neuron connection calculations. In most cases, artificial neural networks can change the internal structure based on external information, which is an adaptive system, i.e., in Layman's terms, has the ability to learn. The application of neural network model was carried out in the 20th century. In the late 1980s, the transformation from highly symbolic artificial intelligence (represented by expert systems that express knowledge using conditional rules) to low-symbolic machine learning (represented by the use of dynamic system parameters).

In recent years, with the deepening of machine learning and artificial intelligence research, neural network models have been introduced into the AES system. Alikaniotis et al [9] and Taghipour and Ng [10] proposed an AES model using long-term / short-term memory (LSTM) networks and gated recurrent unit (GRU) networks [11]. Long-term and short-term memory (LSTM) is a time recurrent neural network (RNN) originally proposed in 1997 [12]. Due to its unique design structure, LSTM is suitable for processing and predicting important events with long intervals and time series delays. These researchers have studied various recursive and convolutional structures on the same data set and found that the LSTM layer followed by the mean averaging operation was implemented. For GRU, it does not use output gate control for memory, but passes it directly to the next unit without any control. Unlike the aforementioned researchers, Dong and Zhang [13] used the Convolutional Neural Network (CNN) model [14] to score papers by applying two CNN layers at the word level and sentence level. The use of CNN and RNN, which will be introduced in the next two subsections, greatly increases the reliability of the scoring system by using a large number of databases as training sets, so that we can obtain more accurate scoring results.

2.4 Convolutional Neural Network

The convolutional neural network (CNN) is a class of multilayer neural networks specifically designed to process two-dimensional data. CNN is considered to be the first truly successful deep-learning method using a multi-layer hierarchical network [15]. It reduces the number of trainable parameters in the network by mining the spatial correlation in the data, and improves the efficiency of the backpropagation algorithm of the forward propagation network. Because CNN requires very little data preprocessing, researchers also treat it as a way of deep learning.

CNN has some advantages that traditional technologies do not: good fault tolerance, parallel processing capability, and self-learning ability [16]. It can handle complex environmental information, unclear background knowledge, and unclear inference rules, allowing samples to have large defects, distortion, fast running speed, good adaptive performance, and high resolution. It integrates the feature extraction function into the multi-layer perceptron by structural reorganization and reducing the weight, and omits the complicated image feature extraction process before recognition.

The generalization ability of CNN is significantly better than other methods. Convolutional neural networks have been applied to pattern classification, object detection, and object recognition. A convolutional neural network is used to establish a pattern classifier, and the convolutional neural network is used as a general pattern classifier for direct use in grayscale images. Although CNN plays a pivotal role in deep learning, in the field of natural language processing, RNN [17] is more popular than CNN.

2.5 Recurrent Neural Network

In many practical applications, data is interdependent. For example, when we understand the meaning of a sentence, it is not enough to understand each word in isolation; rather, we need to deal with the entire sequence of the words. When we think about a problem, we are basing this thought on past experience and knowledge, combined with the current actual situation in consideration. As with these sequential problems, if you use a feedforward neural network (such as a convolutional neural network), there are significant limitations. In order to solve such problems, a temporal neural network was designed. The recurrent neural network (RNN) is a very popular neural network model and has shown excellent results in many tasks of natural language processing.

Recurrent neural networks are sometimes used indiscriminately as two types of networks with similar structures, one of which is finite impulse and the other infinite impulse. Both types of networks exhibit temporal dynamic behavior [18]. The finite impulse recursive network is a directed acyclic graph that can be expanded and replaced by a strict feedforward neural network, which is a directed cyclic graph that cannot be expanded.

With the continuous improvement of computing power, the situation has changed a lot in recent years. With the emergence of some important architectures, especially the LSTM proposed in 1997, RNN has a very powerful application, which can successfully perform sequence tasks in many fields, such as speech recognition, robot translation, human-machine dialogue, speech synthesis, video processing, and other aspects.

3. METHOD

3.1 Dataset

To test our approach, we used Kaggle's (ASAP) dataset. The ASAP data set contains eight different types of article sets. Since the official test data for the ASAP competition was not released to the public, we and others before us (Phandi et al. [19]; Dong and Zhang [13]; Dong et al. [20]) only used the training data for experiments.

3.2 Data Preprocessing and Feature Extraction

In order to remove inappropriate data and avoid errors during runtime, we need data preprocessing. Many characteristics can reflect the level of students' English writing abilities, and we mainly choose spelling mistakes, article length, and vocabulary range as standard components for scoring.

Since there are eight different topics in the ASAP article database, the rating criteria for each topic is different, so we have normalized the ratings of all articles here, ranging from 0 to 1. When evaluating the QWK score, we transform the essay score from a 0 - 1 rating to the original rating scale.

First, the article in the ASAP database is checked for misspelling. Each essay is scanned and counts the spelling errors by using an 'autocorrect' package, and the number of spelling errors is stored as an extra feature. During the correction process, punctuation and unique nouns (e.g. @location) that appear in the article remain unchanged. Students with high and solid writing levels have less spelling errors than inferior students, and the amount of misspelling can be helpful in grading one's essay. Moreover, without this spelling-correction process, misspelled word will be represented by unknown tokens in the word embedding, which might impede the deep learning model from knowing the actual meaning of essays with lots of spelling errors.

After correcting the spelling, the Natural Language Toolkit (NLTK) [21] is used to count the amount of unique vocabulary, where each word is stemmed and different words with the same root are counted as one vocabulary type. For instance, 'computer', 'computation', and 'computational' all have the same stemming result, 'comput', and hence are counted as only one vocabulary. The number of unique vocabulary in each essay is also stored as an extra feature. This feature is important, because the size of vocabulary used in an essay can reflect the vocabulary richness and writing ability of the writer.

Typically, the longer the length of an essay is, the more content its writer can express and the higher his/her writing level is. Correlation between length of an essay and its score is high (0.74) in this dataset, and the effectiveness of these features will be analyzed in the next section.

After extracting the features, a regression algorithm is used to build a model based on the training data. The details of the features and the results of using support vector regression (SVR) and Bayesian linear ridge regression (BLRR) are reported in by Phandi et al.[2, 22].

3.3 Features and Their correlation

We will take a few extracted features as examples and analyze their previous correlations. When we use rater1_domain1 to represent the scores obtained by the article from essay set #1 and then visualize the data, we get the following figures:

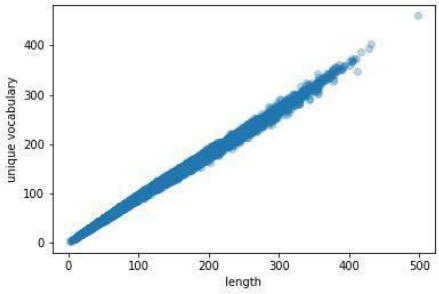

Fig 1.a. Relationship between essay length and unique vocabulary

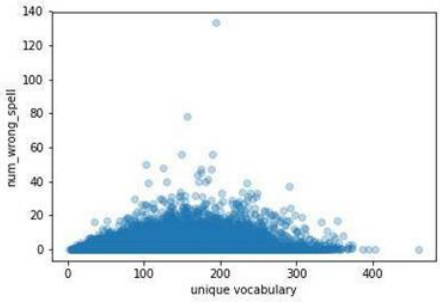

Fig 1.b. Relationship between unique vocabulary and num_wrong_spell

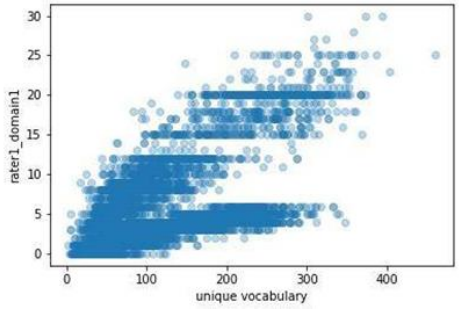

Fig 1.c. Relationship between unique vocabulary and rater1_domain1

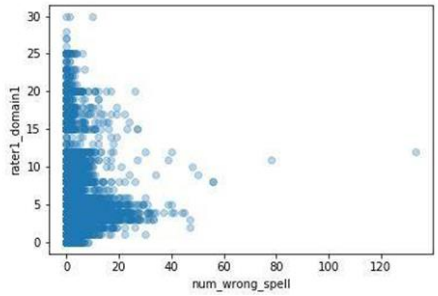

Fig 1.d. Relationship between rater1_domain1 and num_wrong_spell

Figure 1. Some features and their correlation

According to the data in the above figure, we can see that the length of the article is basically proportional to the complexity of the article. However, the number of misspellings is somewhat different from our estimate: although the number of spelling errors is positively correlated with the length of the article, the number of spelling errors decreases as the length increases, because there is a higher chance of making spelling mistakes in a longer essay. This shows that students with high writing levels can write longer articles while avoiding spelling mistakes.

In terms of scoring, although there are some special examples, basically the score of the article is positively correlated with the unique vocabulary of the article and negatively correlated with the number of misspellings. Because of these observations, combining these features with deep learning, for example, assigns very low

score to all short essays, regardless of the content might provide more indicative and robust results.

3.4 Model

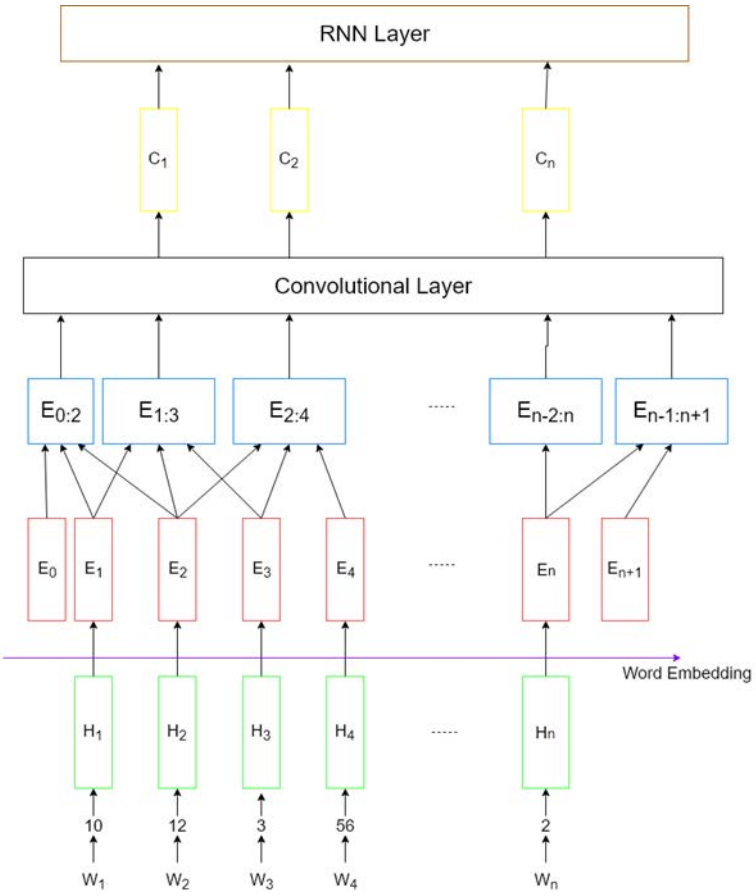

Fig 2. The model of word embedding and convolutional layer

In Figure 2, W1-Wn represents n words (including punctuation), and the number above each word represents the index of the word. First, we convert each word into one hot vector encoding, and use the letter H in the figure to indicate the relevant content. Next, we perform word embedding (purple line in Figure 2) on one hot vector (green block) to get the result after word embedding. Then, we combine the three consecutive word embedding vectors to form a matrix (blue block), which contains the line E0:2. We take the result of word embedding as data, put it into the convolutional layer (black block), which results in the form of the 1-by-d vector, where d is the hidden size of the recurrent neural network. We use the letter C (yellow block) to represent the vector from the convolutional layer. After getting the vectors from the convolutional layer, we use an RNN layer (brown block) for the final operation.

For the one-dimensional convolutional layers, the kernel size is 5, and padding is 2; thus, the input dimension and output dimension of the convolutional layers stay the same.

We use the box with the letter E to represent word embedding for a word (red block), with the number next to E indicating the order, and the punctuation is also counted. The number above the word indicates the index of the word. We combine three consecutive word embeddings to form a matrix.

Since E1 and En cannot be in the middle of the matrix, we place E0 in front of E1 and En+1 in the back of En. E0 and En+1 are not contents of the article. The padding E_0 and E_n+1 are vectors filled with zeros, which have no meanings in this context.

3.5 Training Details

The learning rate of deep learning models determines the speed of weight update. Setting it too large will make the result exceed the optimal value, while setting it too small will make the falling speed too slow. In the training phase, we initialize the learning rate as 0.001, and the learning rate is decreased by 10% in each epoch. A total of 50 epochs is used.

The batch size is the number of training examples in one forward/backward pass. The higher the batch size, the more memory space it will need; the lower the batch size, the longer it will need. Depending on our GPU memory limit, we set the batch size to 32.

GloVe is a widely used word embedding model [23]. To utilize it, we download the trained word embedding model from Wikipedia 2014 + Gigaword 5 (https://nlp.stanford.edu/projects/GloVe/) [23] , which includes six billion tokens and 400,000 vocabularies. It is divided into 50, 100, 200, or 300 dimensions. We attempted all dimensions of word embedding and found that the result of 50 and 300 dimensions are similar; to save more time, the embedding size we set here is 50.

The depth of the neural network is usually determined by the number of layers. The more layers there are, the deeper the neural network is and the better the effect of deep learning. However, due to the limitations of running computer equipment, we have used a layer of neural network to conduct experiments here. Simultaneously, with one-way experiments, the bidirectional option is set to False. We have tried different hidden sizes of 50 and 200, and the results are provided in the Results section.

Dropout is a regularization technique that reduces overfitting in neural networks by preventing the complex adaptation of training data. This is a very effective method of model averaging using neural networks. Depending on the depth and size of the RNN models, the dropout ratio should be different. We used a validation set to determine a reasonable dropout ratio before cross validation, where the same dropout ratio is adapted for the same RNN model. For instance, when there are over three layers of LSTM and the hidden size is greater than 200, a dropout ratio is best set as 0.9; when there is only one layer of GRU cells and the hidden size is smaller than 100, the dropout ratio should be below 0.3.

The next table shows the average results from five runs of the program in our experiment.

4. RESULTS

Table 1. The result of all models(Compared with Ng.et al.)

Model	Essay1	Essay2	Essay3	Essay4	Essay5	Essay6	Essay7	Essay8
LSTM-200-300-2+feature	**0.75**	**0.60**	**0.53**	**0.76**	**0.74**	**0.76**	**0.75**	**0.63**
LSTM-50-50-1+feature	**0.73**	**0.60**	**0.58**	**0.76**	**0.78**	**0.80**	**0.75**	**0.61**
LSTM-50-50-1	**0.75**	**0.60**	**0.58**	**0.76**	**0.78**	**0.80**	**0.74**	**0.60**
GRU-50-50-1+feature	**0.71**	**0.62**	**0.58**	**0.77**	**0.78**	**0.79**	**0.76**	**0.62**
GRU-50-50-1	**0.72**	**0.60**	**0.58**	**0.76**	**0.78**	**0.79**	**0.75**	**0.62**
GRU-50-300 (Ng)	**0.62**	**0.59**	**0.67**	**0.79**	**0.80**	**0.80**	**0.75**	**0.57**
LSTM-50-300 (Ng)	**0.76**	**0.69**	**0.68**	**0.80**	**0.82**	**0.81**	**0.80**	**0.59**

In the model names, the first number represents the GloVe embedding size, the second number represents the number of hidden neurons, and the last number represents number of layers in the recurrent neural network.

According to the methods section, four experiments were conducted in the project, and the results are shown in Table 1, where each row represents the results of the same model for eight different corpus, with each column representing the result of using different models for the same corpus. The larger the number, the more accurate the results of the model run will be.

5. DISCUSSION

5.1 Effect of Structure

In this project, for the first six corpus, LSTM performance was not weaker than or even stronger than GRU; however, in the last two topics, LSTM performed slightly weaker than GRU. However, the overall difference between LSTM and GRU is small, and the impact is very slight.

Surprisingly, the size of deep learning models didn't affect the results too significantly. Deeper recurrent neural networks might overfit the small training data (i.e., only about 1600 datapoints for each essay topic) and yield more inferior results than shallower networks. Recurrent neural network with 50 hidden neurons might be powerful enough to learn latent information and features for this ASAP dataset, which is obtained from primary schools. For real-world scenarios where people write on diverse and complex topics, a deeper structure should perform better than shallow results.

The main focus of this study is to compare deep learning models with and without manual features. We acknowledge there might be better deep learning models, because hyper parameter tuning (e.g., number of layers, number of neurons, convolutional size, learning rate, etc.) can improve the testing of QWK scores. However, finding the best hyperparameters is not the ultimate goal of this study; we have used the same set of hyperparameters and learning schema for all of our models.

5.2 Effect of Feature Selection

To our surprise, when using LSTM the effect of feature extraction on results was not significant, with only the 7th and 8th corpus being slightly better than the results without the feature extraction. Moreover, for the first corpus, the result of adding feature extraction is slightly lower than not adding it. However, whether the original result is enhanced or weakened, the impact of feature extraction is not very obvious. As with LSTM, feature extraction is not obvious for the use of GRU model enhancements; there is only a slight improvement in the 4th and 7th corpus, and slightly weakened in the first corpus. For better results, we should also use a dataset called GloVe.

5.3 Effect of GloVe

According to the comparison, we can see that the use of GloVe in word embedding improves results. Word embedding is the collective name for a set of language modeling and feature learning techniques in natural language processing (NLP) where words or phrases from the vocabulary are mapped to the vectors of real numbers.

GloVe is trained from a large dataset, including Wikipedia, and is an unsupervised learning algorithm, with the initialization of word embedding from its results being beneficial for our supervised learning (i.e., essay scoring) training [24]. GloVe is also widely used in other NLP applications, such as question

answering [25] and machine reading [26]. There are other factors that should be considered in this project which are not mentioned here, such as consistency and adversarial input.

5.4 Future Work

Other factors also determine the quality of the article, which have been applied by some researchers of essay scoring; these include consistency and adversarial input, which we will include in our models in the future.

Some existing methods to measure document consistency which might be helpful in scoring essays include adjacent sentence representation [27], hierarchical recursive neural network [18], solid mesh representation [28], and argument-oriented techniques [29]. Adversarial inputs [30, 31] and jointly-training local consistency models [32] can eliminate confrontational interference in grammatical but incoherent sentences. Essay consistency can reflect the quality of the article to a certain extent and are one of the reference standards for essay scoring. We should consider and include these factors in future research. However, consistency is not the focus of this work.

6. CONCLUSION

In summary, deep learning model on automatic essay scoring outperforms traditional feature selection methods. However, when we tried to combine deep learning with feature selection, where we included essay length, vocabulary complexity, and typo counts as features, our results suggested that including even a small amount of feature selection into deep neural networks might benefit the automatic scoring system, because the recurrent neural network can barely learn some important features (e.g., typo count), given the size and complexity of the data.

Besides, compared with existing methods of essay scoring, after word embedding, we use both convolutional neural network and recurrent neural network in our model. This improvements we have made here largely avoids the occurrence of low-probability errors, such as high-quality essays with high scores.

We also discovered that deeper recurrent neural networks do not have significant advantages over shallower recurrent neural networks, which might be caused by the lack of data. The entire dataset only contains less than 2000 essays for each topic, which is far less than the 14 million images in the ImageNet dataset used for image classification tasks. The deeper and more complex models might be benefited significantly if more diverse data is collected for automatic essay scoring purpose. A pretrained GloVe model and other unsupervised learning techniques can improve the scoring system, and similar transfer learning / one-shot learning approaches can help scientists to develop better deep learning models without expensive data acquisition.

Ever since the automatic essay scoring problem was proposed 20 years ago, state-of-the-art models have yielded comparable results to human graders. However, there exists many areas for future improvement for automatic models.

7. REFERENCES

[1] Attali, Y., & Burstein, J. (2006). Automated essay scoring with e-rater® V. 2. The Journal of Technology, Learning and Assessment, 4(3).

[2] Taghipour, K., & Ng, H. T. (2015). Semi-supervised word sense disambiguation using word embeddings in general and specific domains. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies(pp. 314-323).

[3] Page, E. B. (1968). The use of the computer in analyzing student essays. International review of education, 14(2), 210-225.

[4] Burstein, J., Kukich, K., Wolff, S., Lu, C., & Chodorow, M. (1998, April). Computer analysis of essays. In NCME Symposium on Automated Scoring.

[5] Attali, Y., & Burstein, J. (2004). Automated essay scoring with e-rater® v. 2.0. ETS Research Report Series, 2004(2), i-21.

[6] Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.".

[7] Fellbaum, C. (1998). A semantic network of english: the mother of all WordNets. In EuroWordNet: A multilingual database with lexical semantic networks (pp. 137-148). Springer, Dordrecht.

[8] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.

[9] Alikaniotis, D., Yannakoudakis, H., & Rei, M. (2016). Automatic text scoring using neural networks. arXiv preprint arXiv:1606.04289.

[10] Taghipour, K., & Ng, H. T. (2016). A neural approach to automated essay scoring. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (pp. 1882-1891).

[11] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

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

[13] Dong, F., & Zhang, Y. (2016). Automatic features for essay scoring–an empirical study. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (pp. 1072-1077).

[14] Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188.

[15] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436.

[16] Qiu, X., & Huang, X. (2015, July). Convolutional Neural Tensor Network Architecture for Community-Based Question Answering. In IJCAI (pp. 1305-1311).

[17] Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In Eleventh Annual Conference of the International Speech Communication Association.

[18] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.

[19] Phandi, P., Chai, K. M. A., & Ng, H. T. (2015). Flexible domain adaptation for automated essay scoring using correlated linear regression. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 431-439).

[20] Dong, F., Zhang, Y., & Yang, J. (2017). Attention-based Recurrent Convolutional Neural Network for Automatic Essay Scoring. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017) (pp. 153-162).

[21] Bird, S., & Loper, E. (2004, July). NLTK: the natural language toolkit. In Proceedings of the ACL 2004 on Interactive poster and demonstration sessions (p. 31). Association for Computational Linguistics.

[22] Phandi, P., Chai, K. M. A., & Ng, H. T. (2015). Flexible domain adaptation for automated essay scoring using correlated linear regression. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 431-439).

[23] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation.

[24] Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[25] Xiong, C., Zhong, V., & Socher, R. (2016). Dynamic coattention networks for question answering. arXiv preprint arXiv:1611.01604.

[26] Cheng, J., Dong, L., & Lapata, M. (2016). Long short-term memory-networks for machine reading. arXiv preprint arXiv:1601.06733.

[27] Li, J., & Hovy, E. (2014). A model of coherence based on distributed sentence representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 2039-2048).

[28] Nguyen, D. T., & Joty, S. (2017). A neural local coherence model. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 1320-1330).

[29] Zhang, H., & Litman, D. (2018). Co-Attention Based Neural Network for Source-Dependent Essay Scoring. In Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications (pp. 399-409).

[30] Yannakoudakis, H., Briscoe, T., & Medlock, B. (2011, June). A new dataset and method for automatically grading ESOL texts. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 (pp. 180-189). Association for Computational Linguistics.

[31] Yannakoudakis, H., & Briscoe, T. (2012, June). Modeling coherence in ESOL learner texts. In Proceedings of the Seventh Workshop on Building Educational Applications Using NLP (pp. 33-43). Association for Computational Linguistics.

[32] Farag, Y., Yannakoudakis, H., & Briscoe, T. (2018). Neural Automated Essay Scoring and Coherence Modeling for Adversarially Crafted Input. arXiv preprint arXiv:1804.06898.