Automatic Essay Scoring System

Yidi Zhang(yz3464) **Zhaopeng Liu(zl1732)** **Binqian Zeng(bz866)**
New York University Center for Data Science

Abstract

Automated essay scoring (AES) system tremendously improves efficiency and guarantee fairness. Neural network becomes popular these days for its ability to recognize complex pattern, semantic and syntactic information from context. This project explains how different neural network structures and mechanisms help on the AES task by comparing several published popular models.

1 Introduction

Essay is considered to play an essential role in education. One of important problems in essay grading is subjectivity (Valenti et al., 2003) that is hard to be perfectly avoid. subjectivity can be reduced by taking multiple graders' opinions into consideration but it is highly cost and inefficient. An automatic scoring(AES) system would be fair and consistent in grading efficiently.

There are mainly two types of AES system: feature selection based machine learning approaches and neural network approaches which capture complex patterns from context. Neural network performs better than traditional machine learning on tasks like pattern recognition, natural language understanding so that it becomes more and more popular in designing artificial intelligent systems.

This project is aim to implement several published models on Kaggle 'Automated Student Assessment Prize'(ASAP) dataset and compare performances to get better understanding about how different structures and mechanisms help on natural language understanding. Details of code and best hyper-parameters settings are in our GitHub[1].

2 Related Work

2.1 Machine Learning Approach

Landauer, Holtz, and Laham (Landauer et al., 1998), and Burstein[(Burstein et al., 1998b) (Burstein et al., 1998a) (Burstein et al., 2001)] reported high correlations between human grading and computer generated scores.

S.Yenaeng, S.Saelee, and W.Samai (Yenaeng et al., 2014) use SVM with Genetic Algorithms (GA-SVM) to assess the quality medical case study essays written by medical students. Phandi (Phandi et al., 2015) and Attali (Attali and Burstein, 2004) proposed a domain adaptation AES technique and the E-Rater scoring system that both based on Bayesian linear ridge regression. Loukina, Zechner, Chen, Heilman (Loukina et al., 2015) compare several methods of feature selection for AES problem for rapidly constructing models that achieve good performance with satisfied validity and interpretability. Cummins (Cummins et al., 2016) published the constrained multi-task learning approach to get accurate prediction with unsubstantial data.

2.2 Neural Network Approach

With the advent of distributed and GPU systems, neural network based approaches have become popular for AES system. Neural network based approaches offer an ability to capture complex patterns, semantic and syntactic features without manually feature engineering. Taghipour and

[1] https://github.com/bz866/Automated-Scoring-System

Ng (Taghipour and Ng, 2016) compare performance of several neural network models and their combinations of mechanism show Long Short-Term Memory network outperform in AES tasks. Dong and Zhang (Dong and Zhang, 2016) employ convolutional neural network for the effect of automatically learning features especially improvement upon discrete features. Dong, Zhang and Yang (Dong et al., 2017) proposed an attention-based recurrent convolutional neural network for AES task, which outperforms the previous state-of-the-art methods on 'Automated Student Assessment Prize'(ASAP) dataset[2].

3 Task & Data Description

3.1 Task

The task of automated essay scoring(AES) is usually treated as a supervised learning problem. The score of essay can be treated as continuous values, especially for a large range of possible scores. AES models are trained to minimize the difference between generated scores and reference scores graded by human. Mathematically can be expressed as:

$$argmin_{\delta}\Sigma_{i=1}^{N}L(y_i^*, \delta(X_i))$$

where X_i is the representation of essay and y_i^* is the corresponding reference score assigned by human, δ is the mapping function from X_i to predicted score $\hat{y_i}$, L is the training loss function.

3.2 Data Description

Our dataset contains 21453 essays from Automated Student Assessment Prize(ASAP) dataset[3], which is divided as follows: 70% training set, 15% validation set, and 15% test set. The essays are written by students ranging from grade 7 to grade 10 with average length of 150 to 550, each with distinct marking criteria and score range.

4 Models

We implement several model architectures for comparison. In this section, we only introduce four kinds of representative models. The main idea of the rest will be mentioned in Experiment.

We use pre-trained GloVe[4] embedding matrix to represent words as vectors by $z_i = E_w w_i$ in all of our models.

A linear layer with sigmoid function is set up as our decoder for all models that generate scores for each essay.

$$Score(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x} + b)$$

4.1 Bidirectional Long Short-term Memory Neural Network(Bi-LSTM)

Long short-term memory neural network is popular today for learning long-term dependencies across long time steps. In bidirectional LSTM (Schuster and Paliwal, 1997) neural network, future information is reachable from the current state, which is deem to be difficult to do for uni-directional LSTM. Word vectors are fed into Bi-LSTM to learn.

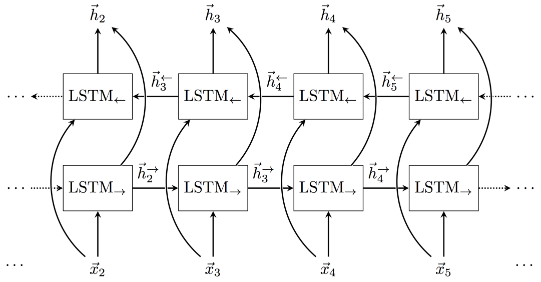

Figure 1: Bidirectional Long Short-term Memory Neural Network

4.2 Hierarchical Convolutional Neural Network with Mean-over-Time Pooling (CNN-CNN-MoT)

Hierarchical CNN model with MoT (Dong and Zhang, 2016) contains two convoluational layers and each layer is followed by a pooling layer. The first convolutional layer is employed on word level to learn sentences representations.

$$c_i = f(\mathbf{w} \cdot z_{i:i+h-1} + b)$$

where a filter $w \in R^{h*k}$ is applied to a window of h words to produce n-grams features and f is the non-linear activation function rectified linear unit (ReLU).

The convolutional layer is followed by an pooling layer that reduces the output dimensionality but

[2] https://www.kaggle.com/c/asap-aes

[3] https://www.kaggle.com/c/asap-aes

[4] https://nlp.stanford.edu/projects/glove/

keeps the most salient information to generate sentences representations.

$$s^j = max\{\mathbf{c}^j\} \bigoplus avg\{\mathbf{c}^j\}$$

The second convolutional layer is used to extract essay representation from sentences vectors. It's followed by an pooling layer(max-pooling and average-pooling) and a full-connected hidden layer. The hidden layer directly connects to the mean-over time output layer. The MoT function in output layer is defined as (Taghipour and Ng, 2016):

$$MoT(H) = \frac{1}{M}\Sigma_{t=1}^{M}\mathbf{h}_t$$

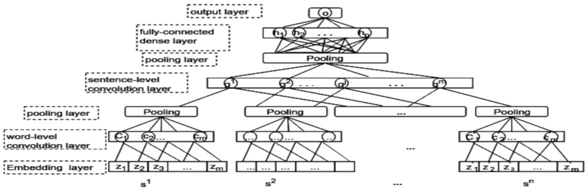

Figure 2: Hierarchical CNN structure

4.3 Attention-based Recurrent Convolutional Neural Network(LSTM-CNN-Att)

Sentence Representation

According to (Dong et al., 2017) , after word embedding, one convolutional layer is employed on each sentence:

$$\mathbf{z}_i = f(\mathbf{W}_z \cdot \left[\mathbf{x}_i^j : \mathbf{x}_i^{j+h_w-1}\right] + \mathbf{b}_z)$$

where W_z, b_z are weight matrix and bias vector, respectively, h_w is the window size in the convolutional layer and z_i is the result feature representation.

The attention pooling (Bahdanau et al., 2014) layer follows the convolutional layer to learn a sentence representation as:

$$\mathbf{m}_i = tanh(\mathbf{W}_m \cdot \mathbf{z}_i + \mathbf{b}_m)$$

$$u_i = \frac{e^{\mathbf{w}_u \cdot \mathbf{m}_i}}{\Sigma e^{\mathbf{w}_u \cdot \mathbf{m}_j}}$$

$$\mathbf{s} = \Sigma u_i \mathbf{z}_i$$

where $\mathbf{W}_m$, $\mathbf{w}_u$ are weight matrix and vector, respectively, $\mathbf{b}_m$ is the bias vector, $\mathbf{m}_i$ and u_i are attention vector and attention weight respectively for i-th word. s is the final sentence representation, which is the weighted sum of all the word vectors.

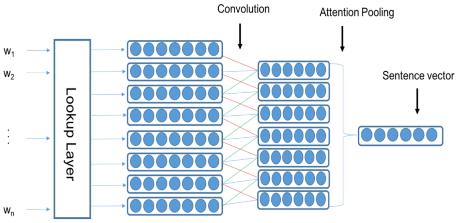

Figure 3: Sentence representation using ConvNet and attention pooling

Text Representation

Sentence representations $s_1, s_2, s_3, \ldots, s_t$ are fed into LSTM that is same as traditional LSTM (Hochreiter and Schmidhuber, 1997). Hidden states $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \ldots, \mathbf{h}_T$ that obtained from LSTM are fed as input into another attention pooling layer to extract essay representations. The attention pooling on sentence level is defined as:

$$\mathbf{a}_i = tanh(\mathbf{W}_a \cdot \mathbf{h}_i + \mathbf{b}_a)$$

$$\alpha_i = \frac{e^{\mathbf{w}_\alpha \cdot \mathbf{a}_i}}{\Sigma e^{\mathbf{w}_\alpha \cdot \mathbf{a}_j}}$$

$$\mathbf{o} = \Sigma \alpha_i \mathbf{h}_i$$

where $\mathbf{W}_a$, $\mathbf{w}_\alpha$ are weight matrix and vector respectively, $\mathbf{b}_a$ is the bias vector, $\mathbf{a}_i$ is attention vector for i-th sentence, and α_i is the attention weight of i-th sentence. $\mathbf{o}$ is the final text representation, which is the weighted sum of all the sentence vectors. Text representations are fed into decoder to generate scores.

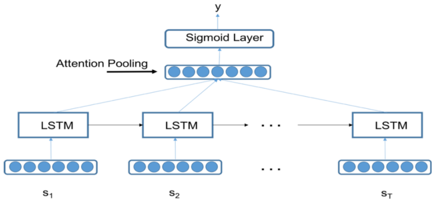

Figure 4: Document (Text) representation using LSTM and attention pooling

4.4 Recurrent Convolutional Neural Network with Mean-over-Time Pooling (LSTM-CNN-MoT)

The recurrent convolutional neural network is quite similar with LSTM-CNN-Att in 4.3. Differences appear at pooling layer and output layer. LSTM-CNN-MoT employs a convolutional layer after word embedding same as LSTM-CNN-Att does.

$$\mathbf{z}_i = f(\mathbf{W}_z \cdot \left[\mathbf{x}_i^j : \mathbf{x}_i^{j+h_w-1}\right] + \mathbf{b}_z)$$

The attention pooling layer is replaced by average pooling layer to extract sentence representation as:

$$s^j = avg\{\mathbf{c}^j\}$$

Sentence representations $s_1, s_2, s_3, \ldots, s_t$ are fed into LSTM that is same as in LSTM-CNN-Att. Hidden states $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \ldots, \mathbf{h}_T$ are input for the mean-over time output layer. The function in output layer is defined same as that in 4.2:

$$MoT(H) = \frac{1}{M}\Sigma_{t=1}^{M}\mathbf{h}_t$$

Above MoT layer, decoder generates scores.

5 Training & Evaluation

5.1 Training

RMSProp optimization algorithm (Dauphin et al., 2015) and Adam optimization algorithm (Kingma and Ba, 2014) are introduced in our models to minimize the mean squared error (MSE) loss function over the training data. The loss function is defined as :

$$MSE\left(\hat{s}_i, s_i^*\right) = \frac{1}{N}\Sigma_{i=1}^{N}\left(\hat{s}_i - s_i^*\right)^2$$

RMSProp optimization algorithm applies learning rates that are based on the average of recent magnitudes of the gradients for the weight, which performs good for data with noise (Jas,), while Adam achieve good results fast.

Additionally, we leverage dropout regularization to avoid over-fitting and *clip-grad* in *Pytorch* to improve the performance of neural network. In the Attention-based Recurrent Convolutional Neural Network, we also use early stopping to avoid over-fitting.

We train all models with a fixed number of epochs and use the validation set to choose the best set of hyper-parameters. The final result is evaluate on the test set. Essays are tokenized with NLTK[5] tokenizer, lowercase the text. During training, both reference scores and generated scores are normalized into [0,1].

5.2 Evaluation Metric

The most popular way (Williamson et al., 2012) to evaluate automated scoring is Quadratic Weighted Kappa(QWK), which is adopted as evaluation metric in (Dong and Zhang, 2016) (Taghipour and Ng, 2016) (Vaughn and Justice, 2015). QWK, modified from weighted kappa, quantify disagreements. The quadratic weight matrix in QWK is defined as (Cohen, 1968):

$$W_{i,j} = 1 - \frac{(i-j)^2}{(R-1)^2}$$

where *i* and *j* are the reference rating (assigned by a human rater) and the system rating (assigned by an AES system), respectively, and R is the number of possible ratings.

An observed score matrix O is calculated such that $O_{i,j}$ refers to the number of essays that receive a rating i by the human rater and a rating j by the AES system. Finally, given the three matrices W , O and E, the QWK value is calculated as:

$$K = 1 - \frac{\Sigma W_{i,j}O_{i,j}}{\Sigma W_{i,j}E_{i,j}}$$

The final performance is evaluated by QWK using scores that rescaled to original score range.

6 Experiments

6.1 Setup

Architectural Choices

We have implemented several experiments to identify how different architectures and mechanisms help us on our task. We also implement one machine learning model that leverage features extracted by Forward Feature Selection as comparison. These architectural choices are summarized below:

- Regression model with feature engineering vs.

[5] http://www.nltk.org/

Neural Network without feature engineering
• Convolutional vs. recurrent neural network
• Using mean-over-time vs. an attention mechanism
• Using a recurrent layer vs. a convolutional recurrent layer

Hyper-parameters

Best sets of parameters for each models are shown in our GitHub[6].

Hyper-parameters Names	Value
word embedding dim	**50**
window size	3, **5**
number of filters	**100**
hidden units	50, **100**
dropout rate	0.3, **0.5**
epochs	12, **50, 100**
batch size	**10, 32**
learning rate	0.001, **0.0001**
optimizer	Adam, **RMSProp**
lstm units	50, **100**

Table 1: Hyper-parameters

6.2 Quantified Results

Model	Avg QWK
Linear Regression	0.72
LSTM	0.751
Bi-LSTM	0.756
LSTM-MoT	0.823
CNN-CNN-MoT	0.792
LSTM-CNN-MoT	0.821
LSTM-LSTM-MoT	0.823
LSTM-CNN-Att	**0.833**

Table 2: Comparison of Average QWK on Different Models

6.3 Analysis

Neural Network vs. Feature Engineering

Results with neural network prove that they capture more complex patterns and establish better semantic and syntactic understanding.

Global and Local

LSTM-CNN-MoT outperforms CNN-CNN-MoT. AES is a task that requires coherent understanding. LSTM can learn more global context information than CNN in the stage of generating text representations. From the structure of Hierarchical CNN, we can see that only information that in window size is reachable for hidden states in CNN. On the other hand, previous information are reachable for hidden states in LSTM. The final state is the representation that contains all context information. LSTM-LSTM-Mot acquires a sightly higher result than LSTM-CNN-MoT does. This also demonstrate the importance of the ability to ensemble global information in document representation. LSTM is more appropriate than CNN in this for its structure.

Mean-over-Time(MoT) vs. Attention pooling(Att)

LSTM-CNN-Att outperforms LSTM-CNN-MoT. Attention pooling evaluate key words and key sentences with higher weights. This is consistent with human graders' accessing process. However, mean-over-time pooling layer tread words and sentences equally, which doesn't conform the general grading logic. Hence, attention layer serves on AES task better. We extract representative examples in Table 3 and Table 4 (word level and sentence level) to show how attention mechanism helps in understanding.

Examples
well computers are not just a pass of time it also makes your life easier
computers are essential to our culture
with modern technology, the world is better connected and united.

Table 3: Example of attention pooling over topic 1 (word level)

Sentences	Attention Weights
dear UNK UNK, computers have advanced our world so far into the future that practically everything is just a click away.	0.14942
in addition to students, doctors also greatly benefit from computers.	0.33125
technology has re-shaped the medical world dramatically with x-rays, UNKś, surgeries and medicines.	0.20231
UNK, the head surgean at UNK says, ””’UNK technology, our UNK would be useless. we wouldń be able to do anything.	0.04973

Table 4: Example of attention pooling over topic 1 (sentence level)

[6] https://github.com/bz866/Automated-Scoring-System

Visualization Samples

Sample 1; RS[7]: 0.0; PS[8]: 0.3

”Dear readers, I think that its good and bad to use the computer to much”

Neural network performs not good on extremely short documents. This essay only has one sentence.

Sample 2; RS: 0.6 ; PS: 0.8

”. . . not only can you get information on certain people and places on these machines, you can also learn whatś the newest fads, whatś carrently happening in @LOCATION2, and the latest @CAPS6 gossip. . . . ”

This essay is full of sentence with semicolon and comma. Although comma and semicolon are generally symbol for advanced sentence structure, the author doesn’t use them properly. The attention mechanism could possibly give higher weights on comma and semicolon so that the model generates a high score.

Sample 3; RS: 1; PS:0.7

”. . . Daer @CAPS1 . . . peapre, . . . valueable . . . overwelm . . . fastect . . . remenber . . .

This essay is actually logical and languages are beautiful but it has so many misspells, which is regretful. Though any grader will praise it with a high score, according to the official grading rules, an essay with many misspells can’t achieve a full score. Here we see the subjectivity of human grader. The grader ignore misspells and give it full mark. The neural network also gives it a not bad score but maybe too much penalty on misspells.

The neural network still can not fully understand documents and sentences hierarchically. It follows some strict rules that learned from normal circumstances. One way to improve the robust and the flexibility is to modifying bad generated results by human and retrain our models by those specific samples.

7 Conclusion

According to comparison between several structures and mechanisms, the most proper way to represent documents is to treat it hierarchically. Attention mechanism performs well on weighting key words and key sentences. Larger datasets with possible special cases are required to train a reliable AES system.

Contribution*

Yidi Zhang: LSTM-MoT; CNN-CNN-MoT; LSTM-CNN-MoT; LSTM-LSTM-MoT; LSTM-CNN-Att

Zhaopeng Liu: LSTM; Bi-LSTM; LSTM-LSTM-MoT;

Binqian Zeng: CNN-CNN-MoT; LSTM-CNN-MoT; LSTM-CNN-Att

All three of us contribute in error analysis and report writing.

References

Yigal Attali and Jill Burstein. 2004. Automated essay scoring with e-rater® v. 2.0. *ETS Research Report Series*, 2004(2).

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Jill Burstein, Lisa Braden-Harder, Martin Chodorow, Shuyi Hua, Bruce Kaplan, Karen Kukich, Chi Lu, James Nolan, Don Rock, and Susanne Wolff. 1998a. Computer analysis of essay content for automated score prediction: A prototype automated scoring system for gmat analytical writing assessment essays. *ETS Research Report Series*, 1998(1).

Jill Burstein, Karen Kukich, Susanne Wolff, Chi Lu, Martin Chodorow, Lisa Braden-Harder, and Mary Dee Harris. 1998b. Automated scoring using a hybrid feature identification technique. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 206–210. Association for Computational Linguistics.

Jill Burstein, Karen Kukich, Susanne Wolff, Chi Lu, and Martin Chodorow. 2001. Enriching automated essay scoring using discourse marking.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Ronan Cummins, Meng Zhang, and Ted Briscoe. 2016. Constrained multi-task learning for automated essay scoring. Association for Computational Linguistics.

Yann Dauphin, Harm de Vries, and Yoshua Bengio. 2015. Equilibrated adaptive learning rates for nonconvex optimization. In *Advances in neural information processing systems*, pages 1504–1512.

Fei Dong and Yue Zhang. 2016. Automatic features for essay scoring-an empirical study. In *EMNLP*, pages 1072–1077.

Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

[7] RS refers to reference score.

[8] PS refers to predicted score.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Thomas K Landauer, Peter W Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Anastassia Loukina, Klaus Zechner, Lei Chen, and Michael Heilman. 2015. Feature selection for automated speech scoring. In *BEA@ NAACL-HLT*, pages 12–19.

Peter Phandi, Kian Ming A Chai, and Hwee Tou Ng. 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*, pages 431–439.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *EMNLP*, pages 1882–1891.

Salvatore Valenti, Francesca Neri, and Alessandro Cucchiarelli. 2003. An overview of current research on automated essay grading. *Journal of Information Technology Education: Research*, 2(1):319–330.

David Vaughn and Derek Justice. 2015. On the direct maximization of quadratic weighted kappa. *arXiv preprint arXiv:1509.07107*.

David M Williamson, Xiaoming Xi, and F Jay Breyer. 2012. A framework for evaluation and use of automated scoring. *Educational measurement: issues and practice*, 31(1):2–13.

S Yenaeng, S Saelee, and W Samai. 2014. Automatic medical case study essay scoring by support vector machine and genetic algorithms. *International Journal of Information and Education Technology*, 4(2):132.