博 士 學 位 論 文

Deep Learning for Automated Essay Scoring

群山大學校 大學院

國際創業學科 國際創業學 專攻

梁 國 喜

指導教授 溫 炳 原

2020年8月

Deep Learning for Automated Essay Scoring

指導敎授　溫 炳 原

이 論文을 創業學 博士學位

請求論文으로 提出함

2020年4月

群山大學校 大學院

國際創業學科 國際創業學 專攻

梁 國 喜

梁國喜의 創業學 博士學位
請求論文을 認准함

2020年6月

學位論文審査委員會

審査委員長 정 동 원 印

審査委員 김 현 철 印

審査委員 온 병 원 印

審査委員 최 규 상 印

審査委員 시 용 찬 印

群山大學校 大學院

TABLE OF CONTENTS

Chapter 1 Introduction ····· **1**
1.1 Background and motivation ····· 1
1.2 Related work ····· 4
1.2.1 Feature-based approaches ····· 4
1.2.2 Deep learning approaches ····· 6
1.3 Thesis outline ····· 9
1.4 Contributions ····· 11
1.4.1 Self-learning representation mechanisms ····· 11
1.4.2 A novel neural network architecture for AES ····· 13
1.4.3 Creativity essay mining exploration ····· 13
1.5 Summary ····· 15
Chapter 2 Theoretical basics of deep learning ····· **16**
2.1 Introduction ····· 16
2.2 Fully connected network ····· 17
2.3 Autoencoders ····· 19
2.4 Convolutional neural network ····· 20
2.4.1 Convolutional layer ····· 22
2.4.2 Pooling layer ····· 22
2.4.3 Fully connected layer ····· 23
2.4.4 Receptive field ····· 23
2.4.5 Weights ····· 24

2.5 Recurrent neural network ····· 24
2.6 Generative adversarial network ····· 28
2.7 Attention mechanisms ····· 31
2.8 Backprogagation algorithm ····· 33
2.9 Summary ····· 36
Chapter 3 Self-learning representation mechanisms ····· 37
3.1 Introduction ····· 37
3.2 Self-learning features ····· 38
3.2.1 Syntactic and semantic features ····· 38
3.2.2 Consistency and coherence features ····· 40
3.2.3 Scoring related information ····· 45
3.2.4 Other features ····· 49
3.3 Mechanisms application ····· 50
3.4 Summary ····· 51
Chapter 4 A novel neural network architecture for AES ····· 52
4.1 Introduction ····· 52
4.2 Model architecture for AES ····· 57
4.2.1 Input definition ····· 57
4.2.2 Evaluation ····· 60
4.2.3 Model architecture ····· 61
4.3 Training ····· 66
4.4 Experiment ····· 67
4.4.1 Setup ····· 68
4.4.2 Baseline ····· 69
4.4.3 Result and discussion ····· 70
4.5 Summary ····· 78

Chapter 5 Exploration of creativity essay mining ····· **80**
5.1 Introduction ····· 80
5.2 Model architecture for creativity essay mining ····· 86
5.2.1 K-fold mask ····· 87
5.2.2 Evaluation ····· 88
5.2.3 Model architecture ····· 90
5.3 Training ····· 96
5.4 Experiment ····· 101
5.4.1 Setup ····· 102
5.4.2 Compared methods ····· 104
5.4.3 Result and discussion ····· 107
5.5 Summary ····· 136
Chapter 6 Conclusion and future work ····· **138**
6.1 Conclusion ····· 138
6.2 Future work ····· 140
References ····· **142**
ABSTRACT ····· **156**

NOTATION OF TABLES

σ	Activation function
θ^T	The transpose of a vector θ
$P_z(z)$	Distribution function of z
$E_{z\sim\ p_z(z)}$	The expectation of distribution $p_z(z)$
$\exp(x)$	Function e^x
$\log x$	The base-2 logarithm $\log 2^x$
$[a:b]$	Concatenation of column a and b
W^l	Weights matrix between layer $l-1$ and l
C	Cost function
$ReLU$	Rectified Linear Unit
δ^l	The gradient of the weighted input of layer l
$\partial C/\partial w$	The gradient of C on w
$\|X\|$	Norm of a vector X
$p(A)$	Probability of event A
$p(A\|B)$	Probability of event A, conditioned on event B
$\arg max(\theta)$	The max value of the vector θ
$tr(A)$	The trace of matrix A
∇	Vector differential operator

LIST OF TABLES

Table 4-1. Training hyper-parameters. ····· 67
Table 4-2. Statistics of ASAP dataset. ····· 69
Table 4-3. The Quadratic Weight Kappa (QWK) value compared with the baseline model. ····· 70
Table 4-4. The Kappa value under different module combinations. 73
Table 4-5. The sample set was used in the experiment. ····· 74
Table 4-6. The mean value and standard deviation of each prompt's Kappa value at the first 100 epochs under Ma + Mc, Mb + Mc, and Ma + Mb + Mc. ····· 75
Table 5-1. Confusion matrix. ····· 90
Table 5-2. Statistics of selected essays from ASAP dataset. ····· 103
Table 5-3. Training hyper-parameters. ····· 103
Table 5-4. Average F1 score under proposed method with different K value. ····· 109
Table 5-5. p-value under different combination of K values. ····· 110
Table 5-6. Average indicators of each prompt under Autoencoders. ····· 111
Table 5-7. Average indicators of each prompt under Attention. ··· 111
Table 5-8. Average indicators of each prompt under proposed method. ····· 112
Table 5-9. The ranked the prompts based on AVG. ····· 116
Table 5-10. F1 score under three methods with K=5. ····· 118
Table 5-11. Average indicators of prompt 5. ····· 119
Table 5-12. Most creative and most common essays ID in other prompts. ····· 125

LIST OF FIGURES

Figure 1-1. Self-learning representation mechanism. ····· 12
Figure 1-2. AES framework after adding self-learning mechanisms. ····· 12
Figure 1-3. AES framework after adding self-learning mechanisms and creativity essay mining. ····· 14
Figure 2-1. A fully connected layer. ····· 17
Figure 2-2. Multiple fully connected networks. ····· 18
Figure 2-3. An Autoencoder. ····· 20
Figure 2-4. A convolutional neural network. ····· 21
Figure 2-5. Rolled Recurrent neural network. ····· 25
Figure 2-6. Unrolled recurrent neural network. ····· 26
Figure 2-7. A single tanh layer. ····· 27
Figure 2-8. LSTM chain structure. ····· 27
Figure 2-9. The overviews of GANs. ····· 30
Figure 2-10. The attention mechanisms used in a seq2seq model. 32
Figure 2-11. Backpropagation method. ····· 35
Figure 3-1. Representation vector model. ····· 40
Figure 3-2. Similarity features concatenation. ····· 45
Figure 3-3. Scoring related information added deep learning model. ····· 46
Figure 3-4. Scoring related information concatenation. ····· 48

Figure 3-5. Different ways to use the self-learning mechanism. ····· 51
Figure 4-1. The overall framework of the approach. ····· 53
Figure 4-2. Siamese bidirectional long short-term memory architecture model architecture. ····· 54
Figure 4-3. Each prompt's Kappa value comparison under Ma + Mc and Mb + Mc at the first 100 epochs (prompt7 and prompt8 are 300epochs). ····· 76
Figure 4-4. The mean value of each prompt's Kappa value at the first 100 epochs under Ma + Mc, Mb + Mc, and Ma + Mb + Mc. ··· 77
Figure 4-5. The standard deviation of each prompt's Kappa value at the first 100 epochs under Ma + Mc, Mb + Mc, and Ma + Mb + Mc. ····· 77
Figure 4-6. Kappa value comparison under Ma + Mc, Mb + Mc and Ma + Mb + Mc (prompt2 and prompt3). ····· 78
Figure 5-1. Overall of creativity essay mining. ····· 85
Figure 5-2. A K-fold mask example of sequence with 10 positions, K=5. ····· 87
Figure 5-3. The framework of LSTM generator. ····· 92
Figure 5-4. The framework of CNN discriminator. ····· 94
Figure 5-5. The framework of essay vector representation. ····· 95
Figure 5-6. Transformer. ····· 105
Figure 5-7. Transformer-Encoder. ····· 106
Figure 5-8. Distance convergence of the 8 prompts under the three methods. ····· 113

Chapter 1 Introduction

1.1 Background and motivation

In recent years, deep learning has made significant progress in the field of artificial intelligence. The accuracy of recognition in areas such as speech recognition, image recognition, and video capture has dramatically improved. Artificial intelligence has made substantial applications in medical, financial, and autonomous driving fields, etc. These applications also have much promoted the theoretical research of deep learning. The achievements of deep learning are increasing all the time, from scientific research to industrial applications, the application of deep learning is becoming more and more widespread.

Natural language processing (NLP) is quite a challenging technology, which is considered as the jewel in the crown of artificial intelligence. There has been no major breakthrough in decades in the research area of NLP until the introduction of deep learning in recent years. The research on natural language processing has been made new progress on the semantic level. Breakthroughs have been made in the text representation, machine translation, intelligent answering, and content recommendation. Automated essay scoring (AES), as an important part of natural language processing, has also made remarkable progress in recent years, and a series of achievements have emerged. Nowadays, it is the best historical period to study

automated essay scoring.

A successful automated essay scoring method applied to practice can not only reduce a lot of workloads but also accelerate the learning process and improve the learning effect. For example, using AES to online learning or online test, such as MOOC, APP, IELTS, and TOEFL, etc., can speed up the process of evaluation and interaction. Meanwhile, AES can enhance the learning effect and raise the interest of the study. On the other hand, if a machine could recommend some creativity essays in AES, which will make online learning more intelligent. From thousands of essays, the machine recommends several excellent texts to share with students for learning, which is a crucial thing for education.

However, there are so many fruitful achievements and promising prospects; in the long run, natural language processing is still in the initial stage of development. More advanced and intelligent automated scoring applications still need to be studied. In general, the current shortcomings of automated essay scoring methods are mainly in the following three aspects.

(1) The integration of existing achievements in natural language processing for automated essay scoring is not enough. Automated essay scoring is quite a comprehensive work, which needs to consider all kinds of aspects comprehensively. At present, semantic text analysis, sentence classification, sentence generation, machine translation, new language models, etc. have made progress. There are pieces of evidence (Dong et al., 2017; Tay et al., 2018; Ke and Ng, 2019) that the comprehensive application of these results may promote

the automated essay scoring. However, the existing research on the integration of these results is not timely. There is something new to explore.

(2) The existing automated essay scoring methods still need to be improved. The space to enhance the average accuracy is still available. After the text data are mapped into a vector representation, the cost of computing is high, the training method and parameter optimization are complicated. Therefore, developing a new AES approach with higher average accuracy, lower learning complexity, better generalization performance, and even can be implemented in semi-supervised or unsupervised learning is still one of the most expected problems to be solved.

(3) As an innovative work in automated essay scoring area, creativity essay mining is also one of the most challenging tasks. At present, there are very few related research literature (Darwish et al., 2020). Through creativity essay mining, automated essay scoring can become more "intelligent." It is a gratifying work to find out creativity essays. Therefore, there is still a lot of work to be expected in creativity essay mining.

Based on the above three issues, this thesis intends to study in the following three aspects.

(1) Because of the insufficient integration and possible application of the achievements in natural language processing for automated essay scoring, this thesis studies how to integrate the research results in the text. These include similarity analysis, sentiment analysis, semantic analysis, etc., to apply to the automated essay scoring method.

(2) For the existing automated essay scoring methods, the average accuracy still has room to improve, the training complexity is large, and the training optimization method is quite tricky. This thesis studied to find a scoring model that is more effective and has better performance to improve the average accuracy of AES.

(3) Finding a creative essay in the processing of automated essay scoring. This thesis also uses the text GANs network to study creativity essay mining.

The above three parts also correspond to the three chapters of this thesis, which are the core content of this thesis.

1.2 Related work

1.2.1 Feature-based approaches

Research on AES began decades ago. In the field of application, the first AES system named Project Essay Grade (PEG) (Ellis et al., 1966) for automating the educational assessment was seen in 1967. Intelligent Essay Assessor (IEA) (Foltz et al., 1999) adopts a Latent Semantic Analysis (LSA) (Landauer et al., 1998) algorithm to produce semantic vectors for essays and computes the semantic similarity between the vectors. The E-rater system (Attali et al., 2004), which can extract various grammatical structure features of the essay, now plays a facilitating role in the Graduate Record Examination and Test of English as a Foreign Language. The early research of AES was regarded as a semi-automated machine learning approach based on various feature extractions. Larkey (1998) and Rudner and Lawrence

(2002) treated AES as a kind of classification using bag-of-words features. Attali and Burstein (2004) and Foltz et al. (1999) used regression approaches to achieve AES. Yannakoudakis et al. (2011) took automated essay scoring as a ranking problem by ranking the order of pair essays based on their quality. Features such as words, Part-of-Speech (POS) tagging, n-grams features, sophisticated grammatical features are extracted. Tandalla (2012) used traditional machine learning approaches to extract multi-features to achieve AES, including regular expression from the text and trained on ensemble learning approaches like RF. Mehmood et al. (2017) also proposed a model performing AES using multi-text features and ensemble machine learning. Chen and He (2013) described AES as a ranking problem that took the order relation among the whole essays into account. The features contain syntactical features, grammar, and fluency features as well as content and prompt specific features. Shristi et al. (2017) proposed a regression-based approach for automatically scoring essays that are written in English; they use standard Natural Language Processing (NLP) techniques for extracting the features from the essays. Phandi et al. (2015) used a correlated Bayesian Linear Ridge Regression approach to tackle domain-adaptation tasks. Fauzi et al. (2017) evaluated the use of a hierarchical classification approach to the essays' automated assessment. This research computes the essay scores by using a hierarchical approach, analogous to an incremental algorithm for hierarchical classification. Fauzi et al. used an automatic essay scoring system based on n-gram and cosine similarity to extract features and considered the word order. Based on the existing

automated essay evaluation systems, Zupanc et al. (2017) proposed an approach that incorporates additional semantic coherence and consistency attributes. They extracted the coherence attributes by transforming sequential parts of an essay into the semantic space and calculating the changes between them to estimate the essay's coherence. All of these methods mentioned above are all kinds of machine learning that need handcrafted feature extraction. The application fields that have certain limits and the average accuracy is not always good.

1.2.2 Deep learning approaches

Since deep learning was introduced into natural language processing, more and more researchers have carried out related research. Santos and Gatti (2014) proposed a deep convolutional neural network that focuses on different levels of analysis from character-level to sentence-level information to perform sentiment analysis of short essays. Yin et al. (2016) investigated machine comprehension on a question answering (QA) benchmark called MCTest. They proposed a neural network framework, termed hierarchical attention-based convolutional neural network (HABCNN), to address this task without any handcrafted features. HABCNN employs an attention mechanism to weigh the key phrases, key sentences, and key snippets that are relevant to answering the question. Zhang et al. (2015) gave a sensitivity analysis of one-layer CNN to explore the effect of architecture components on model performance to distinguish between important and comparatively

inconsequential design decisions for sentence classification. Yang et al. (2016) proposed a hierarchical attention network for document classification. The model has a hierarchical structure that mirrors the hierarchical structure of documents, and it also has two levels of attention mechanisms that applied at the word and sentence level, enabling it to attend differentially to more and less important content when constructing the document representation. Dong and Zhang (Dong and Zhang, 2016) employed a convolutional neural network (CNN) for the effect of automatically learning features. Kumar et al. (2017) introduced a novel architecture for AES grading by combining three neural building modules: Siamese bidirectional LSTMs applied to a model and a student answer, a new pooling layer based on earth-mover distance across all hidden states from both LSTMs and a flexible final regression layer to output scores.

Especially in 2012, Kaggle launched a competition on AES called 'Automated Student Assessment Prize' (ASAP, https://www.kaggle.com/c/asap-aes/data) sponsored by the Hewlett Foundation. Hewlett hopes data scientists and machine learning specialists help solve fast, effective and affordable solutions for automated grading of student-written essays. At that time, the competitors mostly use machine learning algorithms that need handcrafted feature extraction. Recently, many researchers have conducted a series of neural network-based AES studies using ASAP data sets. Alikaniotis et al. (2016) employed a neural model to learn features for essay scoring automatically, which leverages a score-specific word embedding (SSWE) for word representations.

Alikaniotis's experiment shows that SSWE is better for word embedding compared with other pre-trained word embeddings like word2vec, and LSTM (Hochreiter and Schmidhuber., 1997) structure can capture the semantic information of the essay better than support vector machine (SVM). Taghipour et al. (2016) developed an approach based on recurrent neural networks to learn the relation between an essay and its assigned score, without any feature engineering. They combined convolutional neural networks and recurrent neural networks for AES and demonstrated that LSTM and CNN are capable of outperforming systems that extensively require handcrafted features. In this paper, CNN was taken as an optional layer before inputting into LSTM, especially for those essays with a long length. Dong et al. (2017) thought that, when using RNN and CNN to model input essays, the relative advantages of RNN and CNN cannot be compared based on the single vector representations of the essays. In addition, different parts of the essay can give a different contribution to the score. Therefore, they introduced the attention mechanisms on the basis of CNN and RNN and found that the attention mechanisms help to find the keywords and sentences that contribute to judging the quality of essays. By building a hierarchical sentence-document model to represent essays, the model uses the attention mechanisms to decide the relative weights of words and sentences automatically. The model can learn text representation with LSTMs, which could model the coherence and coherence among a sequence of sentences. Furthermore, attention pooling is used to capture more relevant words and sentences that contribute to the final quality of essays. Borrowing the idea from

Dong, we also use an attention mechanism at the LSTM layer. Tay et al. (2018) described a new neural architecture that enhances vanilla neural network models with auxiliary neural coherence features and proposed a new SKIPFLOW mechanism. The SKIPFLOW model alleviates two problems: one is to alleviate the inability of current neural network architectures to model flow, coherence, and semantic relatedness over time; the other one is to ease the burden of the recurrent model. To do so, the SKIPFLOW models the relationships between multiple snapshots of the LSTM's hidden state over time. As the model reads the essay, it models the semantic relationships between two points of an essay using a neural tensor layer. Eventually, multiple features of semantic relatedness are aggregated across the essay and used as auxiliary features for prediction. Then, they use the semantic relationships between multiple snapshots as auxiliary features for prediction. The SKIPFLOW mechanism based on LSTM architecture, which incorporates neural coherence features, implements an end-to-end AES approach.

Inspired by SKIPFLOW, furthermore, we put forward a self-information mechanism that is an extension from the essay to the essay and sample (rating criteria). Ref. (Dong et al., 2017; Tay et al., 2018) was also taken as a baseline of the experiment in chapter 4.

1.3 Thesis outline

The rest of the thesis is organized in the following way:

Chapter 2 states the mainstream neural network structures in deep learning and discusses their possible applications in AES. These models

and methods are also the theoretical basis of the full thesis.

In chapter 3, we propose a self-learning mechanism. In this thesis, we consider helping the neural networks to learn some specific information and representing some rating information beyond the essay to improve the accuracy of AES. We divide rating criteria into sample essays or abstract, keywords which are provided by domain experts, and propose three specific self-learning representation mechanisms: 1) syntactic and semantic representation mechanism, 2) consistency and coherence representation mechanism, 3) scoring related information representation mechanism, and some other preprocess technologies.

Chapter 4 proposes a relatively interpretable novel neural network AES approach called Siamese Bidirectional Long Short-Term Memory Architecture (SBLSTMA) that can accept the self-learning mechanism. Thus, the SBLSTMA model can capture not only the essay's semantic features but also some specific information and the rating criteria information behind the essays. We use the SBLSTMA model for the task of AES and take the ASAP dataset as evaluation. Experimental results show that our approach is better than the previous neural network methods.

In chapter 5, based on AES's results, we select those essays with high scores and employ the text GANs network for creativity essay mining. Based on the assumption that common or non-creativity essay is relatively easy to predict, the creative essay should be challenging to predict. We use GANs to train the essays that parts of it are masked and then use GANs to generate the masked part to judge the essay's creativity.

In the end, in chapter 6, a conclusion and further work are discussed.

1.4 Contributions

This thesis proposes a new self-learning representation mechanism and incorporates it into a newly proposed neural network architecture, which improves the average accuracy of AES. Besides, this thesis also explores the study of creativity essay mining.

1.4.1 Self-learning representation mechanisms

At present, in supervised training, almost all the neural network models, only focus on the learning of the training object and do not learn the possible information behind the object. While in AES, not only essay can be learned, information such as scoring criteria is also useful. The literature (Dong et al., 2017; Tay et al., 2018) has empirically demonstrated that prior to obtaining a certain scoring feature can greatly improve the accuracy of automated scoring. Inspired by this, this thesis investigates the characteristics of text semantic analysis, syntax analysis, the consistency and coherence of the essay, emotion analysis, etc. These are characterized by semantic characteristics, syntax characteristics, and the content of the essay. We represent these kinds of information and integrate them into the neural network so that the neural network training can learn more information and is more purposeful. This kind of self-learning mechanism is manually designed, but the relevant knowledge is learned by the neural network itself. In this way, we can "help" the

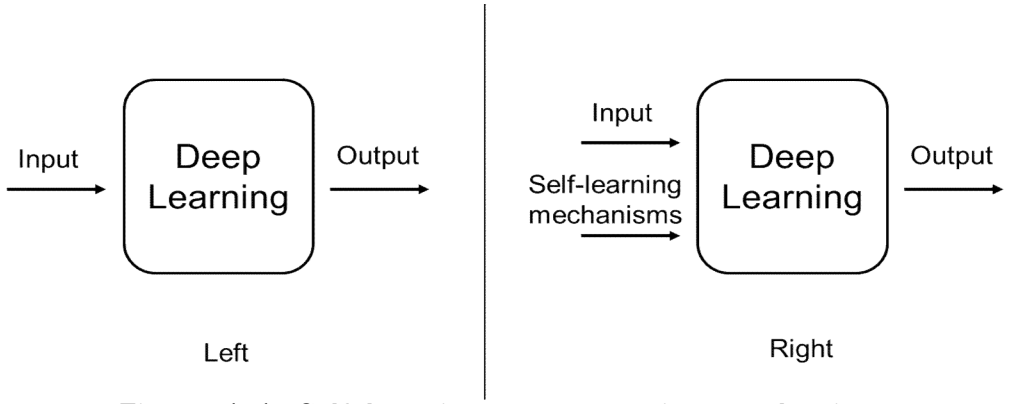

Figure 1-1. Self-learning representation mechanism.

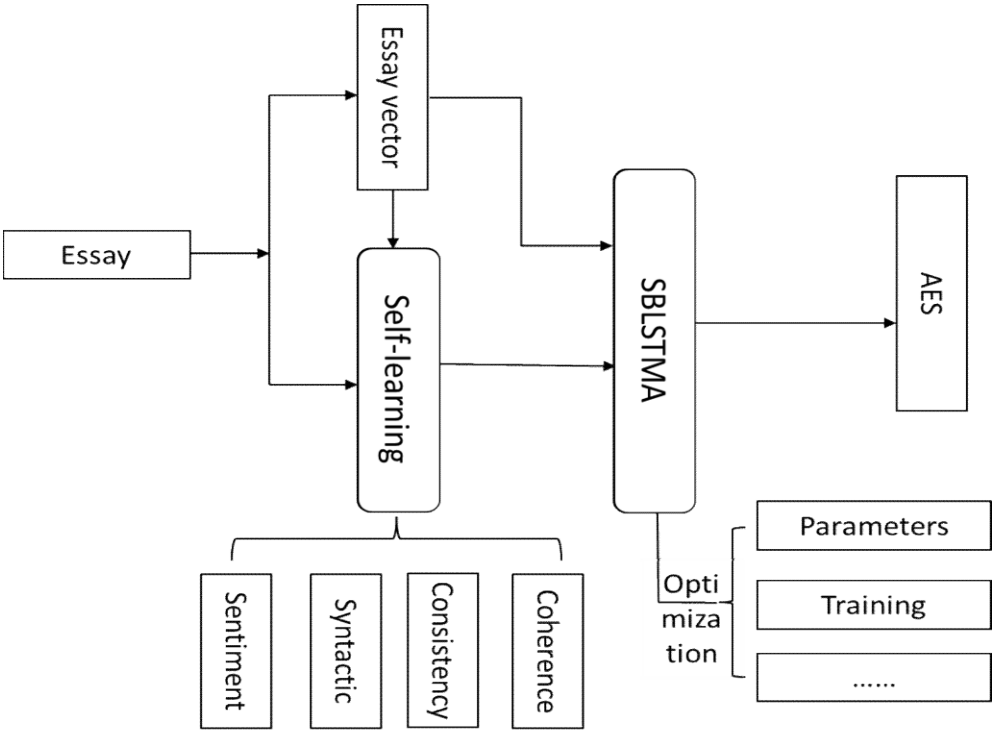

Figure 1-2. AES framework after adding self-learning mechanisms.

artificial neural network to find the scoring information better and faster, and make judgments for the scoring. The introduction of the self-learning mechanisms will also make deep learning models interpretable to a certain extent. Figure 1-1 shows the deep

learning model is changed when added with a self-learning mechanism.

1.4.2 A novel neural network architecture for AES

As mentioned above, most of the existing neural network architectures for AES only considered the essay itself without considering the rating criteria behind the essay. In this study, we introduce the self-learning mechanisms mentioned earlier and demonstrate a relatively interpretable novel neural network architecture for AES. We represent rating criteria by some sample essays or abstracts, key points, keywords, etc., which are provided by domain experts. Then, we take the input pair consisting of an essay and scoring related information as a new input. We propose a Siamese Bidirectional Long Short-Term Memory Architecture (SBLSTMA) that can accept the new input. Thus, the SBLSTMA model can capture not only the semantic features among the essay but also some specific information and the rating criteria information behind the essays. We use the SBLSTMA model for the task of AES and take the ASAP dataset as evaluation. Experimental results show that our approach is better than the previous neural network methods.

After adding self-learning, the AES framework is shown in Figure 1-2.

1.4.3 Creativity essay mining exploration

For some more creative essays, the existing automated essay scoring methods could not find them well, and usually, only gives a

"mediocre" judgment for very creative essays. This makes automated essay scoring lost much fun. Currently, there is very few researches study on creativity essay mining. Most of them focus on the area of cognitive science and machine learning and are a kind of automated essay evaluation. Some of the literature have used Generative Adversarial Networks (GANs) to research on automatic sentence generation.

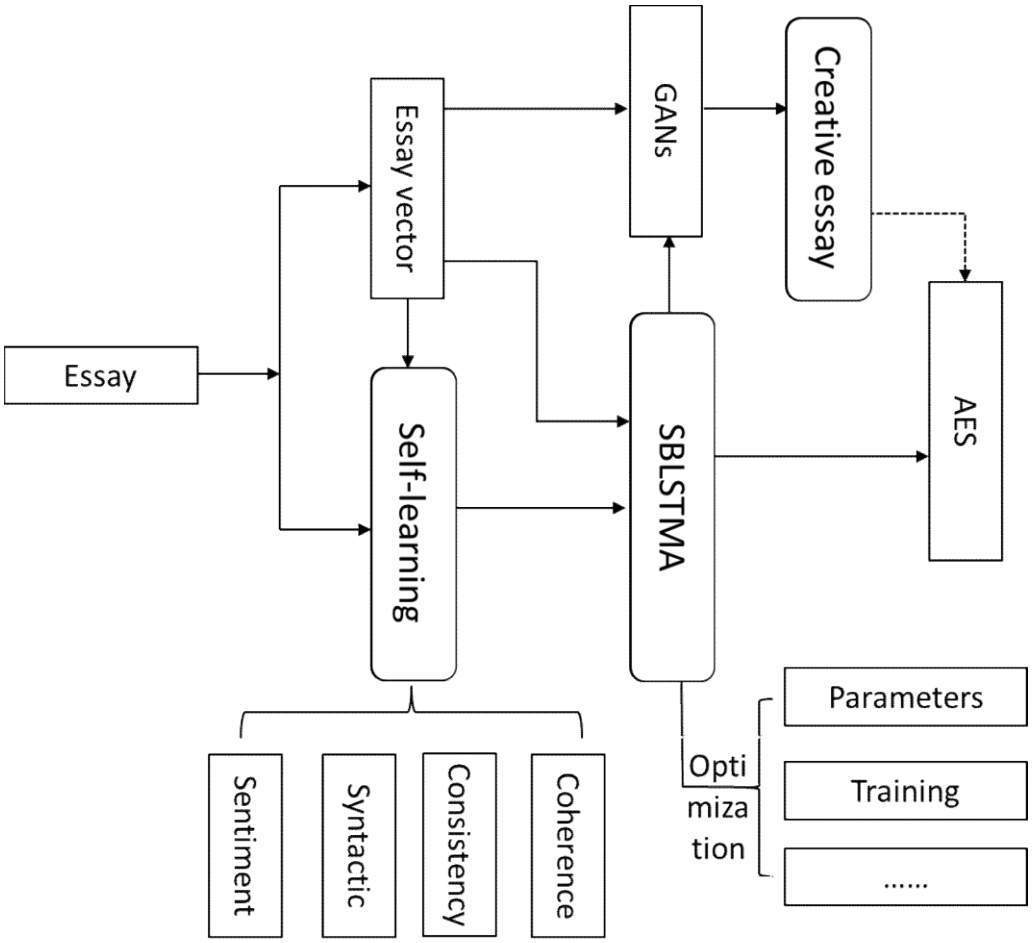

Figure 1-3 AES framework after adding self-learning mechanisms and creativity essay mining.

Inspired by this, this thesis makes exploration to use the GANs

network (Goodfellow et al., 2014) for creativity essay mining based on the above two studies. The AES framework, after adding creative essay mining, is shown in Figure 1-3.

1.5 Summary

This chapter introduces the background of deep learning and the motivation of the thesis. We state the previous work of AES in two aspects (features based and deep learning approaches) in recent decades. Among these work, we discuss the main achievements and deficiencies of the AES. Lastly, we also demonstrate the main research content and objectives of the thesis and list the main contributions of this thesis.

Chapter 2 Theoretical basics of deep learning

2.1 Introduction

Deep learning is a new research direction in machine learning based on artificial neural networks (ANN), which is a kind of representation learning. The type of learning contains unsupervised learning, semi-supervised learning, or supervised learning, etc.(Bengio et al., 2013; Schmidhuber, 2015; Bengio et al., 2015).

The main architectures of deep learning are artificial neural networks, which is a network structure that imitates human brain neurons to process information and builds a simplified model and forms different networks according to different organisation methods. Artificial neural networks have differences from biological brains. ANN is an operation model, consisting of massive number of nodes (neurons) connected. Each section is a specific output function.

Currently, deep learning neural network architectures have various types. These architectures include fully connected neural networks, recurrent neural networks, convolutional neural networks, generative adversarial networks, and deep residual learning, etc. Deep learning has successfully solved many practical problems that the traditional approaches are difficult to address in the fields of intelligent robots,

automatic control, pattern recognition, predictive estimation, medicine, biology, and economics, and has shown excellent smart characteristics. (Krizhevsky et al., 2012; Ciresan et al., 2012). From academic research to industrial applications, deep learning exists everywhere.

Also, since deep learning was introduced into AES field, various neural networks have made new progress. In this chapter, we aim to provide readers with the basic structures of deep learning used in AES, which is also used as the theoretical basis of the thesis.

2.2 Fully connected network

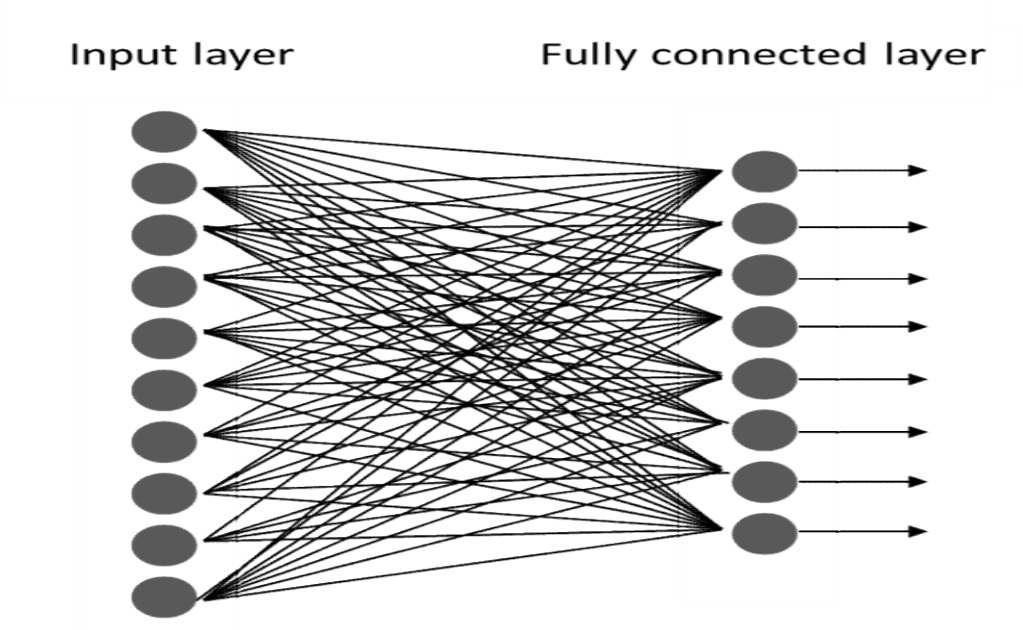

Figure 2-1. A fully connected layer.

A fully connected network (Elizondo et al., 1997) is a stack of multiple fully connected layers. A fully connected layer is a mapping from R^m to R^n. Each output layer dimension depends on each input layer dimension. Figure 2-1 shows a typical fully connected

layer as follows.

The description of a fully connected layer is as follows:

Let $x \in R^m$ denote the input to a fully connected layer, x_i be the i-th input. Let $y_i \in R$ be the i-th output from the fully connected layer. Let $w_i = [w_{1i}, w_{2i}, ..., w_{mi}] \in R^m$ be the weight vector, b_i be the bias. Then output $y_i \in R$ is computed as follows:

$$y_i = \sigma(w_{1i}x_1 + ... + w_{mi}x_m + b_i) \quad (2\text{-}1)$$

Here, σ is a activation function. The full output y is then

$$y = \sigma(W \cdot \quad x + b) \quad (2\text{-}2)$$

Where $W = [w_1, w_2, \cdots, w_n]^T$, $b = [b_1, b_2, \cdots, b_n]^T$, '$\cdot$ ' is the dot product.

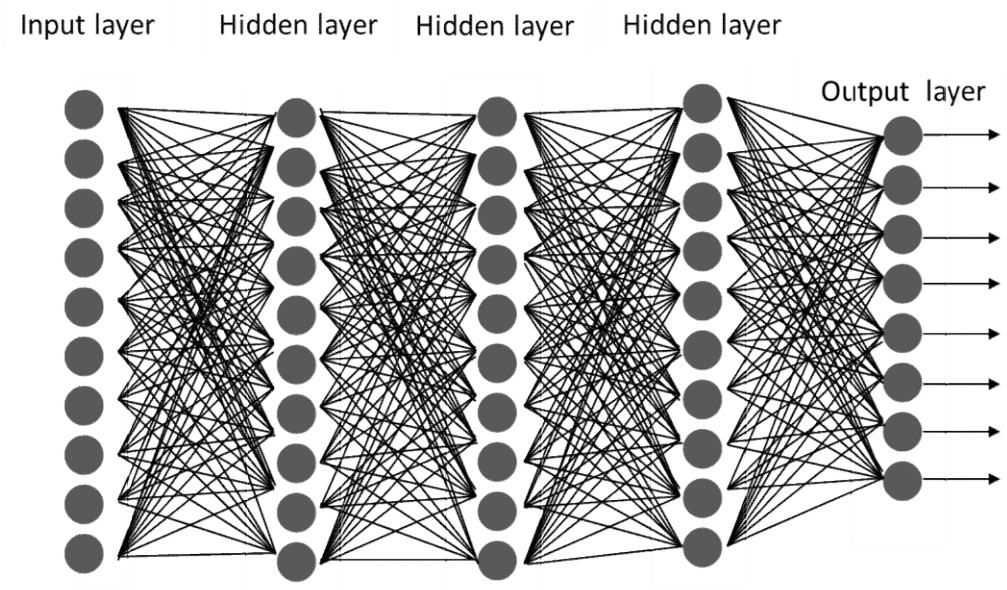

Figure 2-2. Multiple fully connected networks.

A fully connected neural network usually consists of a series of fully connected layers. An output of layer l could be the input of

layer $l+1$. A fully connected neural network is often called a "deep" network, as shown in Figure 2-2.

Fully connected networks are used for various tasks of applications. The significant advantage (or disadvantage) of fully connected networks is that they are structure agnostic (or unexplainable). We need no particular assumptions to be made about the input (hard to explain). Fully connected networks are often used at the end of other networks, such as convolutional neural networks, recurrent neural networks, etc., for classification.

2.3 Autoencoders

An Autoencoder (Hinton et al., 2006; AP et al., 2014) is a type of artificial neural network that uses semi-supervised learning or unsupervised learning. Its function is to represent the input information by using the input information as the learning target. An Autoencoder mainly includes two parts: an encoder and a decoder. An Autoencoder has the function of characterizing the learning algorithm in a general sense and is applied to dimensionality reduction and anomaly detection. Autoencoders can be used as a powerful tool for feature detection of deep neural networks. Besides, the Autoencoder can also be used for generating random data similar to the training data, which is called a generative model. For example, we can train an Autoencoder with a picture, which can generate another new image.

Autoencoder could take an unlabeled dataset and train it as a supervised learning problem tasked with outputting $\hat{x}$, here $\hat{x}$ is an

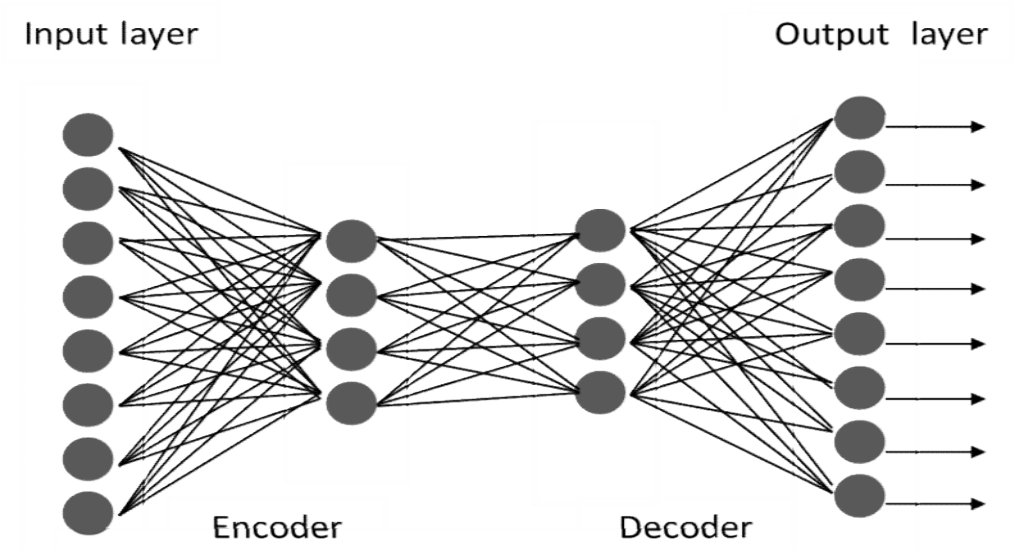

Figure 2-3. An Autoencoder.

approximation of the original input x. Autoencoder network can be trained by minimizing the reconstruction error, $L(x,\hat{x})$, which measures the differences between the original input and the approximation. The hidden layers of the Autoencoders are critical attributes of our network design, which limits the amount of information that can traverse the whole network, thus forcing the learning compression of the input data. The Autoencoders contain multiple hidden layers shown in Figure 2-3.

2.4 Convolutional neural network

Convolutional Neural Networks (CNN), a typical structure of deep learning models, proposed by LeCun et al., (1989), it is also the current research hotspots of deep learning. It is a feedforward artificial neural network with a multilayer network structure, which usually includes an input layer, a convolutional layer, an activation function, a

pooling layer, and a fully connected layer. CNN has very strong feature extraction capabilities, which can extract higher-level features. CNN is also known as shift invariant or space invariant artificial neural network (SIANN), based on their shared-weights architecture and translation invariance characteristics (Zhang et al., 1988, 1990). They are used in image and video recognition, recommender systems,(Van et al., 2013) image classification, medical image segmentation, and natural language processing (Collobert et al., 2008), etc.

A convolutional neural network usually is composed of an input layer, multiple hidden layers, and an output layer. The hidden layers of CNN consist of a series of convolution layers, which are convoluted with multiplication or other dot product. The activation function (such as Relu) is followed by additional convolutions, such as pool layer, fully connected layer, and normalized layer, which are called hidden layers because their input and output are shielded by the activation function and final convolution. Figure 2-4 shows a convolutional neural network.

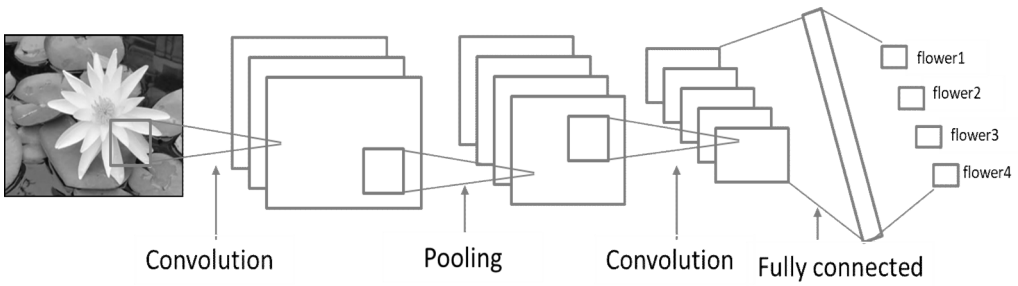

Figure 2-4. A convolutional neural network.

2.4.1 Convolutional layer

CNN is a collection of convolution kernels and convolution operations, pooling operations. The input of CNN is a four-dimensional tensor with shape image numbers × image channel × image width × image height. After passing through the convolution layer, the image will be abstracted into a feature map with shape image numbers × feature map width × feature map height × feature map channels. The convolutional layer in the neural network should have the following properties:

(1) Convolution kernels can be in different size;

(2) The number of input channels and output channels could be different;

(3) The number of the convolution filter must be equal to the number channels of the input feature map.

2.4.2 Pooling layer

The input of each node of the pooling layer is a small block of the previous layer, which usually is the convolution layer. The size of this small block is determined by the window size of the pooling core. The pooling layer does not change the depth of the node matrix. But it can change the size of the matrix. Generally speaking, for image processing, the pooling operation in the pooling layer can be understood as converting a high-resolution picture into a low-resolution picture. Common pooling operations include maximum

pooling (Krizhevsky et al., 2013) and average pooling (Ciresan et al., 2012). After passing through the convolution layer and pooling layer, the number of parameters in the network model can be further reduced.

2.4.3 Fully connected layer

As state in section 2.2. A fully connected layer connects neurons in one layer to the neurons in another layer. It is similar to the traditional multilayer perceptron neural network (MLP). By flat operation, the flattened matrix goes through a fully connected layer to classify the objects.

2.4.4 Receptive field

In the convolutional neural network, the receptive field is described as the size of the area that each pixel on the feature map output from each layer of the convolutional neural network maps on the original image. The original image here is Refers to the input image of the network, which is the image after preprocessing (such as resize, warp, crop).

The reason why neurons cannot observe all the pixels in the original image is that convolutional layers and pooling layers are usually used in convolutional neural networks, and they are all locally connected between layers.

The larger the size of the receptive field of a neuron, the larger the size of the original image it can touch, which also means that it may contain more global features with higher semantic levels; on the

contrary, the smaller the value, the features it contains More and more local and detailed. Therefore, the receptive field size can be used to roughly judge the abstract level of each layer.

2.4.5 Weights

Each neuron in the neural network uses a activation function to calculate the output value of the input value from the receptive field of the previous layer. The function used for the input value is governed by the weight and the bias vector. In neural networks, learning is performed by iteratively adjusting these deviations and weights.

The vectors of weights and bias are called filters and represent specific features of the input. A significant feature of CNN is that numbers of the neurons can share the same filter. Since a single deviation and a single weight vector are used in all receiving fields that share the filter, rather than each receiving area having its own bais and vector weight (Mittal, 2018), the memory footprint is reduced.

2.5 Recurrent neural network

Recurrent neural networks (RNN) (Zremba et al., 2014; Schmidhuber et al., 2015; Goodfellow et al., 2016, page 378) is a type of neural network with short-term memory function. The common neural networks, in which the neurons can not only receive information from other neurons but also receive their information, forming a network structure with loops. Compared with feedforward neural networks, recurrent neural networks are more in line with the

architecture of biological neural networks. The parameter learning of the recurrent neural network can be learned through the backpropagation algorithm over time (Werbos et al., 1990). As a recursive neural network, RNN takes sequence data as input and performs recursion in the evolution direction of the sequence, and all recycling units are connected in a chain. As shown in Figure 2-5, a chunk of the neural network, RNN recycles at some input x_t and outputs a value h_t. The recycle allows inputs to be passed from one step of the network to the next.

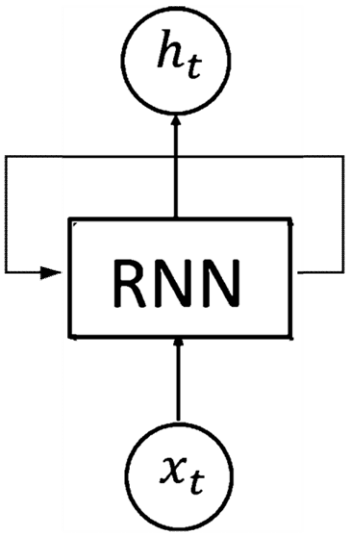

Figure 2-5. Rolled Recurrent neural network.

An RNN can be seen as multiple copies of the same network, each unit passing a message to the next one. If we unroll the recycle, shown in Figure 2-6. This chain-like nature shows that RNN is intimately related to sequences. They're the natural structure of the neural network to use for such sequences data. Recently, there has been a tremendous successful application of RNNs to a variety of

problems: speech recognition, language modeling, translation, image captioning, etc. However, the back-propagation algorithm, over time, transfers error information will gradually increase with step size. When the input sequence is relatively long, there will be the problem of gradient explosion and disappearance (Hochreiter et al., 1994; Hochreiter and Schmidhuber, 1997; Bengio et al., 2001), also known as the long-range dependence problem. Long Short Term Memory networks (LSTM) can solve this issue, it's an exceptional kind of RNN that has a memory function to reduce error accumulation and can be used for long sequence analysis. For many tasks, LSTM is much better than the standard RNN. Almost all exciting results based on RNN are achieved with LSTM.

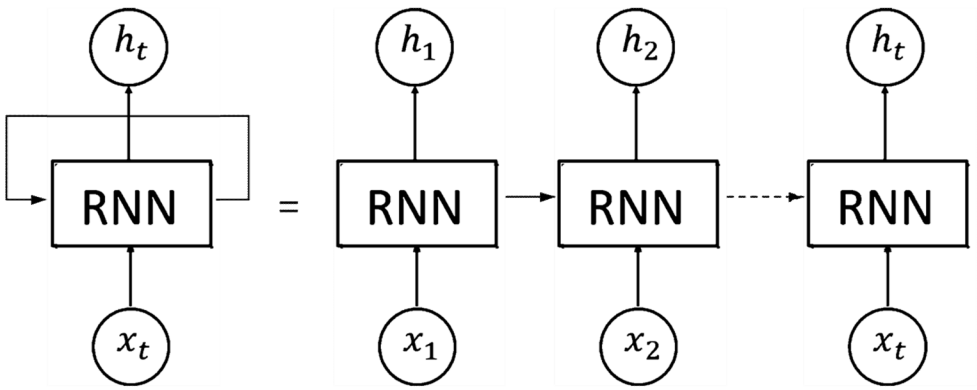

Figure 2-6. Unrolled recurrent neural network.

LSTM, proposed by Hochreiter & Schmidhuber in 1997, is a special RNN that can learn long-term dependencies. LSTM is specifically designed to avoid long-term dependency problems. They are improved and popularized by many other researchers. This thesis also uses LSTM.

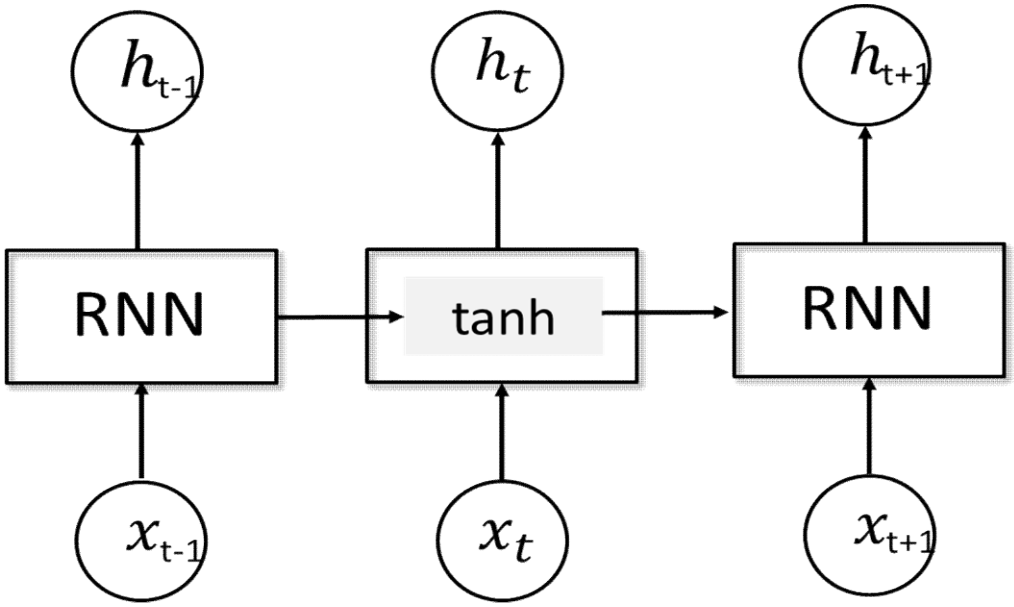

Figure 2-7. A single tanh layer.

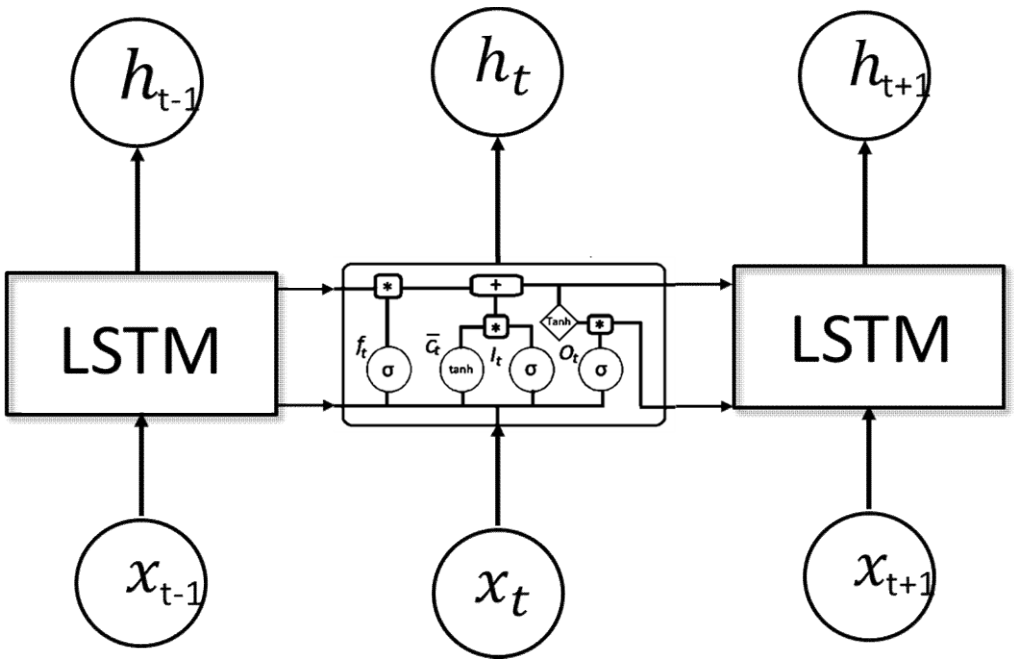

Figure 2-8. LSTM chain structure.

All recurrent neural networks have the form of a loop module chain of neural networks. In a standard RNN, this loop module will have a very simple structure, such as a single tanh layer, as shown in Figure 2-7.

LSTM also has this chain architecture, but the loop module has a different architecture. Rather than having only one neural network layer, there are four ways to interact in an extraordinary way, as shown in Figure 2-8.

In Figure 2-8, each line carries the entire vector, from the output of one node to the input of another node. Different nodes represent point-by-point operations, such as vector addition, while each box represents the learned neural network layer. The merged lines indicate concatenation, while the forked lines indicate that their contents are being copied, and the copies are in different locations.

2.6 Generative adversarial network

The Generative Adversarial Network (GAN) was proposed by Goodfellow (2014) and was called an exciting network by LeCun, which has achieved excellent results in extracting and fitting the data distribution. GAN is one of the most excellent models for unsupervised learning on complex distributions in recent years. The GAN model generates a reasonably good output through the interactive game learning of (at least) two modules in the framework: the Generative Model and the Discriminative Model. In the original GAN theory, Generative Model and Discriminative Model are not necessary to be the neural networks. On the contrary, the GAN only need to be a

function that can be generated and discriminated accordingly. However, in practice, deep neural networks are generally used as Generator and Discriminator. An excellent GAN application requires a suitable training method; otherwise, the output may not be ideal due to the freedom of the neural network model. The GAN method trains a Generator and a Discriminator separately. The Discriminator is used to discriminate whether the data comes from real training data or data generated by the Generator. Game training is used to make the Generator generate data that conforms to the distribution of real training data.

In GAN, the Generator generates fake data samples (such as images, audio, etc.) and attempts to deceive the discriminator. On the other hand, the Discriminator tries to distinguish real examples from fake samples. Both the Generator and the Discriminator are neural networks, and they compete with each other during the training phase. These steps are repeated several times. In this case, the Generator and Discriminator will work better and better after each repetition. The work can be visualized through the charts in Figures 2-9.

Here, the generative model captures the distribution of data and is trained to maximize the likelihood of the Discriminator making mistakes. On the other hand, the Discriminator is based on a model that estimates the probability of samples received from training data rather than from the generator.

GAN is formulated as a minimum and maximum game, where the Discriminator tries to minimize the reward V (D, G), and the Generator tries to minimize the discriminator's reward, in other words, to maximize the loss. It can be described mathematically by

the following formula:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim\ p_{data}(x)}[\log D(x)] + E_{z\sim\ p_{z}(z)}[\log(1-D(G(z))] \quad (2\text{-}3)$$

Where G denotes Generator, D denotes Discriminator, $P_{data}(x)$ represents distribution of real data, $P_z(z)$ is the distribution of generator, x is the sample from $P_{data}(x)$, z is the sample from $P_z(z)$, D(x) denotes the Discriminator network, and G(z) denotes Generator network.

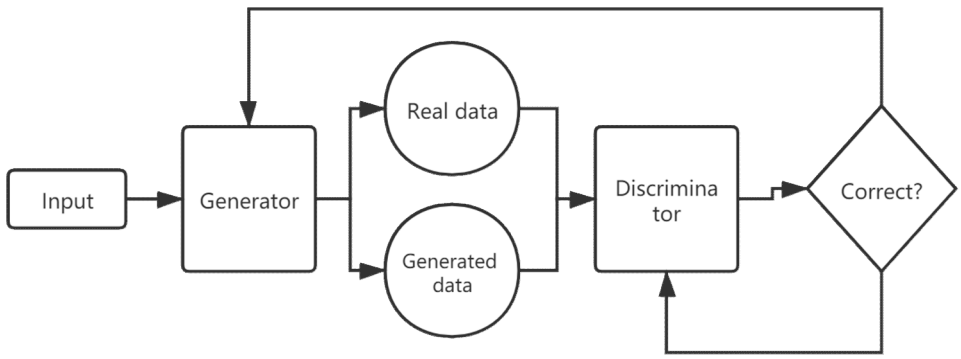

Figure 2-9. The overviews of GANs.

The GANs is trained as follows: on the one hand, the Discriminator is trained when the Generator is suspended. At this stage, the network is only forward-propagated, not back-propagated. The discriminator is trained on real data of n epochs and sees whether it can correctly predict it as real. Besides, at this stage, the Discriminator is also trained on the counterfeit data generated by the Generator to see if it can correctly predict it as fake.

On the other hand, the Generator is trained when the Discriminator is suspended. After using the fake data generated by the Generator to

teach the Discriminator, we can obtain its prediction and use the result to train the Generator and get better results from the previous state to deceive the Discriminator.

Repeat the steps of the above training, and then manually check the fake data, it seems to be true. If it looks acceptable, the training will stop. Otherwise, it can continue for a few more epochs.

2.7 Attention mechanisms

The attention mechanism (Luong et al., 2014; Dzmitry et al., 2014) is one of the latest developments in deep learning. Especially in the field of natural language processing, such as speech recognition, machine translation, text mining, dialogue generation, etc. It is a mechanism that to improve the performance of the encoder-decoder (seq2seq) RNN model. A note to solve the limitations of the encoder-decoder model is proposed. The model encodes the input sequence into a fixed-length vector and decodes the output from the vector at each time step. This problem is considered to be a problem when decoding long sequences because it makes it difficult for neural networks to deal with long sentences, especially for those data sequence that are longer than the sentences in the training corpus. Similar to the underlying encoder-decoder structure, his mechanism inserts the context vector into the gap between the encoder and the decoder. According to Figure 2-10, the upper part represents the encoder and the bottom left represents the decoder. We can see that the context vector is calculated using the output of all cells as input.

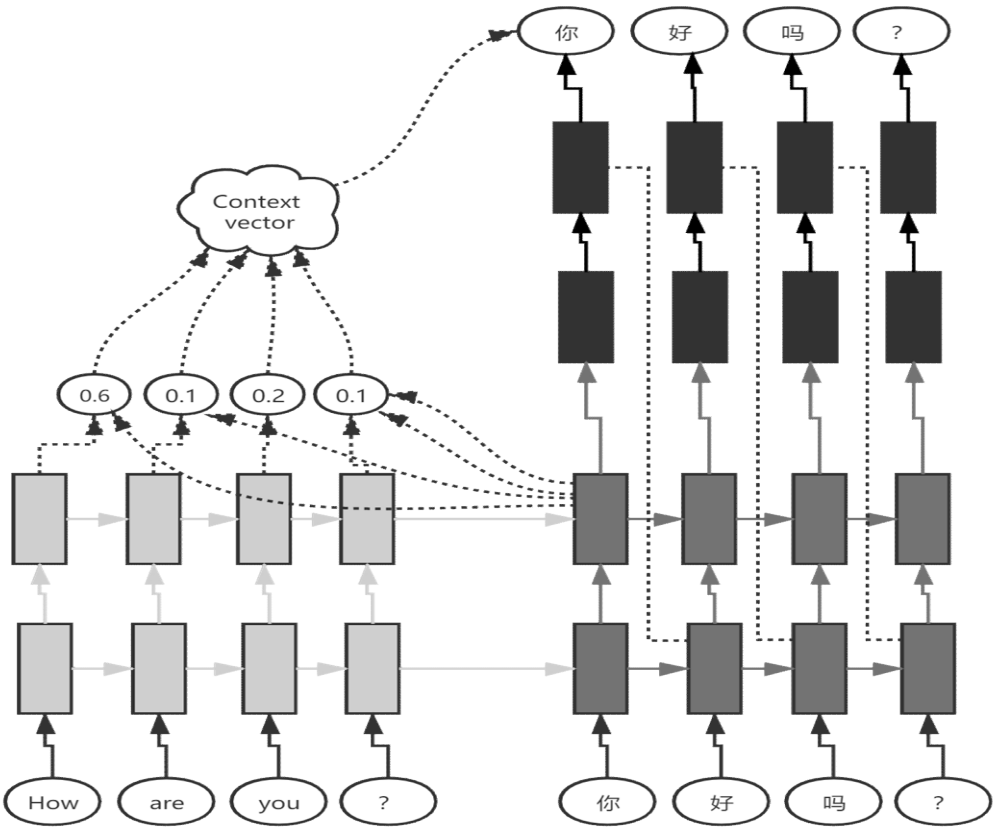

Figure 2-10. The attention mechanisms used in a seq2seq model.

The probability distribution of the source language words generated by each word decoder. By using this mechanism, the decoder can capture a certain degree of global information instead of inferring only based on the hidden state.

The calculation steps of the mechanism are as follows:

For a fixed target word, first, we traverse the state of all encoders to compare the target state and the source state to generate a score for each state in the encoder. For example, we can then use Softmax to normalize all scores to generate a probability distribution conditioned on the target state. Finally,

weights are introduced to make the context vector easy to train. The mathematical equation is as follows:

$$\alpha_{ts} = \frac{\exp(score(h_t, h_s))}{\sum_{s'=1}^{S} \exp(score(h_t, h_{s'}))} \quad (2\text{-}4)$$

$$c_t = \sum_{s}^{S} \alpha_{ts} h_s \quad (2\text{-}5)$$

$$\alpha_t = f(c_t, h_t) = \tanh(W_c[c_t : h_t]) \quad (2\text{-}6)$$

Here, equation (2-4) is for generating attention weights, equation (2-5) is for making context vector, equation (2-6) is for making attention vector.

2.8 Backprogagation algorithm

Backpropagation (Rumelhart, 1986) is usually used for supervised training , which computes the gradient in the weight space of a feedforward neural network concerning a loss function. The backpropagation algorithm is suitable for a learning algorithm of a multi-layer neuron network, which is based on a gradient descent method. The input-output relationship of the Backpropagation network is essentially a mapping relationship: The function performed by the n-input m-output neural network is a continuous mapping from n-dimensional Euclidean space to finite field in m-dimensional Euclidean space. It is highly nonlinear. Its information processing ability comes from multiple reorganizations of simple nonlinear functions, so it has a strong ability to reproduce functions. This is the basis on which the back propagation algorithm can be applied.

In the derivation of backpropagation, some intermediate quantities will be introduced if they are necessarily used. The bias terms do not need special treatment because they correspond to a fixed input weight of 1. For backpropagation, because the loss function and activation function can be adequately evaluated, the specific loss function and activation function are not necessary.

Let x be the input vector, y be the output vector, C be the loss function, L be the number of layers. Let $W^l = (w_{jk}^l)$ be the weights between layer $l-1$ and l, where w_{jk}^l is the weight between the k-th node in layer $l-1$ and the j-th node in layer l. let $f^l(\cdot\ \ )$ be the activation functions at layer l.

The entire network is a combination of functional composition and matrix multiplication:

$$g(x) = f^l(\,W^l f^{l-1}(\,W^{l-1}\cdots\ \ f^1(\,W^1 x)\cdots\,)) \tag{2-7}$$

For a supervised training data set, there will be a set of objective-label (input-output) pairs, $\{(x_i, y_i)\}$. For each objective-label pair (x_i, y_i) in the training set, the loss of the model on that pair is the cost of the difference between the predicted output $g(x_i)$ and the target output y_i:

$$C(y_i, f^l(\,W^l f^{l-1}(\,W^{l-1}\cdots\ \ f^1(\,W^1 x)\cdots\,))) \tag{2-8}$$

Backpropagation computes the gradient for a fixed objective-label pair (x_i, y_i), where $W^l = (w_{jk}^l)$ is weight matrix. Here, w_{jk}^l is the weight between the k-th node in layer $l-1$ and the j-th node in layer l. The gradient of two adjacent neurons, $\partial C/\partial w_{jk}^l$ (gradient

of the k-th node in layer $l-1$ and the j-th node in layer l) can be computed by the chain rule.

The key point is that because the only way for the weight to affect the loss in W^l is through its effect on the next layer, and it is linearly affected, so δ^l is the only data that needs to calculate the weight gradient of layer l, and then the previous one Layer δ^{l-1} can be calculated, and it needn't repeat recursively. This can avoid inefficiency in two ways.

First, it avoids repetition, because when calculating the gradient of layer l, it is not needed to recalculate all derivatives on layers $l+1$, $l+2$, etc. every time.

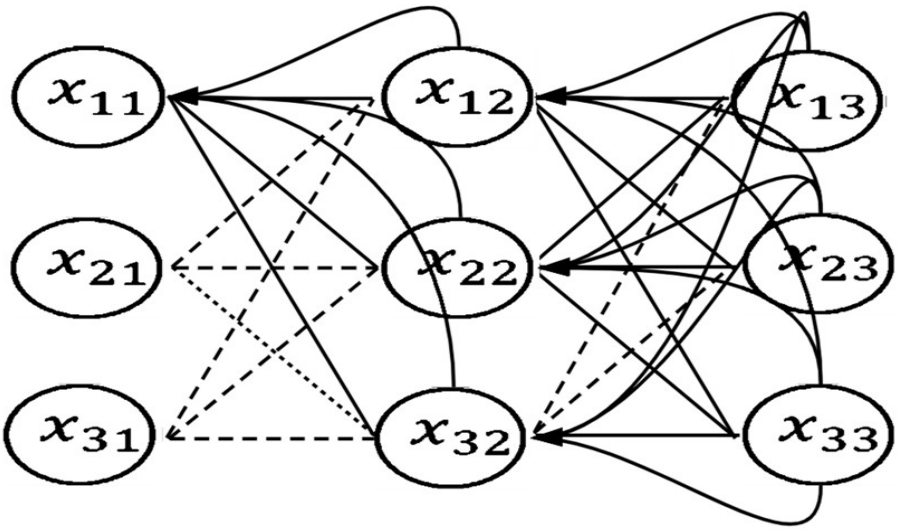

Figure 2-11 Backpropagation method.

Second, it avoids unnecessary intermediate calculations because it directly calculates the weight gradient relative to the final output (loss) at each stage, rather than unnecessarily calculating the

derivative of the hidden layer value corresponding to the weight $\partial \alpha_j^{l'} / \partial w_{jk}^l$. Backpropagation can use matrix multiplication to represent a simple feedforward network. Gradient calculation is like a chain, passing from back to front. The gradient calculation method is shown in Figure 2-11.

2.9 Summary

In chapter 2, we introduce the theoretical basics of deep learning, which are mainly the mainstream neural network structures in deep learning. We also analyze their possible applications in AES. These models are also the theoretical basis of the full thesis. The various network structures introduced in this chapter are applied to various sections of this dissertation. The fully connected network is used for classification at the end of various network structures in chapter 4 and chapter 5. Autoencoder and attention mechanisms are selected as the compared method in chapter 5. A convolutional neural network is employed for AES as an optional layer in chapter 4 and a discriminator for GANs in chapter 5. The recurrent neural network is the main tool for this thesis. The generative adversarial network is employed for creativity essay mining. Backpropagation is the basic training method.

Chapter 3 Self-learning representation mechanisms

3.1 Introduction

The deep neural network is a black box. We don't exactly know what the neural network has learned. How to make sure that what the neural network has learned is what we expect to learn? This is a difficult problem, however, to a certain extent, we can help neural networks learn by designing some mechanism. Researchers tend to add other mechanisms to various neural networks such as Autoencoders, CNNs, RNNs, GANs, ResNet, etc. These mechanisms, for instance, the attention mechanisms and SKIPFLOW mechanism (Tay et al., 2018), are useful in the neural network. The works of literature (Dong et al., 2017; Tay et al., 2018) have shown that it is more helpful to improve the accuracy of automated scoring by prior to obtain certain scoring features. Inspired by this, this thesis studies the characteristics of text semantic analysis, syntax analysis, emotion analysis, the consistency and coherence of the essay, etc., which is characterized by semantic features, syntax features, the content of the essay. We try to express these kinds of information and integrate them into the neural network so that the training of the neural network is more purposeful. We call this information representation a self-learning mechanism (the learning

mechanism). This self-learning mechanism is designed by human but the feature information is learned by the neural network itself. In this way, we can "help" the artificial neural network to find the scoring information faster and make a judgment about the score. The introduction of self-learning mechanism also makes the deep model more explainable.

3.2 Self-learning features

3.2.1 Syntactic and semantic features

The syntax is a kind of linguistic features, which is a set of rules, principles, and processes that constrain the structure of sentences in a given language. It usually includes word order. The term syntax is also used to refer to the study of such principles and processes. Many syntacticians' goal is to discover the syntactic rules common to all languages. (Definition: Syntax in English - Babylon. https://translation.babylon-software.com/english/syntax/)

Semantics, in linguistics, is the subfield that is devoted to the study of meaning, as inherent at the levels of words, phrases, sentences, and larger units of discourse (termed texts, or narratives). The study of semantics is also closely linked to the subjects of representation, reference, and denotation. The essential research of semantics is oriented to the examination of the meaning of signs, and the study of relations between different linguistic units and compounds. A critical concern is how meaning attaches to larger chunks of text, possibly as a result of the composition from smaller units of meaning.

(Definition: Semantics in English. Wikipedia. https://en.wikipedia.org/wiki/Semantics)

As described above, semantic and syntax contain the main meaning of an essay. The common approach to sequence to sequence learning maps an input sequence to a variable-length output sequence via recurrent neural networks. Facebook (Gehring et al., 2017) introduces an architecture based entirely on convolutional neural networks. Compared to recurrent models, computations over all elements can be fully parallelized during training, and optimization is more natural since the number of non-linearities is fixed and independent of the input length. Google (Vaswani et al., 2017) proposed a new simple network architecture named the Transformer. It based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Inspired by Facebook and Google, this study proposes a method to use CNN and attention mechanisms via Autoencoders neural network to represent the semantic and syntactic information. We think that extracting some critical information through CNN and attention mechanisms can make up for the shortcomings of LSTM in sequence processing, which is very necessary. Especially for those long essays, LSTM error accumulation will be severe, while the features extracted by CNN and the attention mechanisms will be very useful. This is verified by the experiments in Chapter 4. We use a vector to represent these features, as shown in Figure 3-1, the representation vector could be merged into the main scoring model in Chapter 4 that a bidirectional LSTM architecture.

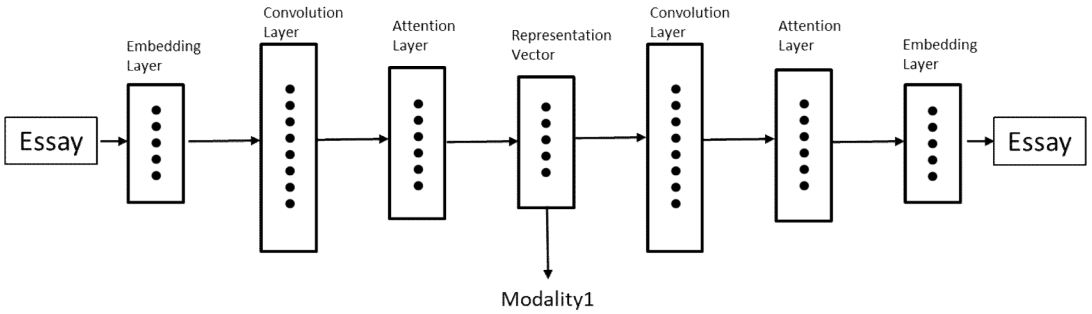

Figure 3-1. Representation vector model.

3.2.2 Consistency and coherence features

Consistency and coherence are two important indicators for measuring the quality of essay content, of which consistency reflects the facts described in the essay compared to other facts in essays are rational, while coherence reflects the semantic development.

Cosine similarity is a commonly used similarity measure for real-valued vectors, used in (among other fields) information retrieval to score the similarity of documents in the vector space model. In machine learning, many pieces of literatures use cosine similarity to measure consistency and coherence (Darwish and Mohamed 2020). Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any angle in the interval (0, π] radians. It is thus a judgment of orientation and not magnitude: two vectors with the same direction have a cosine similarity of 1, two vectors oriented at 90° relative to each other have a similarity of 0, and two vectors opposed have a similarity of -1, independent of their magnitude.

The cosine similarity is mainly used in positive space, where the outcome is neatly bounded in [0,1].

The cosine of two non-zero vectors X and Y can be derived by using the Euclidean dot product formula:

$$X \cdot \quad Y = |X||Y|\cos(\theta) \tag{3-1}$$

Given two vectors of attributes, X and Y, the cosine similarity, $\cos(\theta)$, is represented using a dot product and magnitude as

$$s(X, Y) = \cos(\theta) = \frac{X \cdot \quad Y}{|X||Y|} = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2}\sqrt{\sum_{i=1}^{n} Y_i^2}} \tag{3-2}$$

Where X_i and Y_i are components of vector X and Y respectively.

Coherence attributes are based on the hypothesis that the semantic content of a coherent essay changes gradually through its text. If we see the different parts in the essay as vectors, then we could compute the similarity of all of these vectors. An essay has a right consistency, and coherence should have a specific similarity distribution. Generally, in machine learning, the below measures can be described by this particular similarity distribution. These measures (Darwish and Mohamed 2020) are:

(1) The average distance between neighboring positions. We could calculate the similarity between sentences of a fixed average length to reflect the gradualness between sentences.

(2) The average distance between any two points. Similar to (1), we could calculate the similarity between sentences in different

positions to reflect the different changes between different sentences.

(3) The maximum difference between any two points. This is a subset of (2) is used to calculate the diameter of the area that is enclosed with points and, thus, the range of the discussed concept in the space.

(4) Clark and Evans' distance to the nearest neighbor of each point in the semantic area is vital for measuring spatial relationships.

(5) Cumulative frequency distribution of the nearest neighbors' distances.

Because of the above measures could be described by similarity. Here, we introduce how to employ cosine similarity to measure consistency and coherence. Literatures (Tay et al., 2017; Darwish and Mohamed 2020.) show that with higher similarity value the consistency and coherence wound be higher. Therefore, essays with different consistency and coherence quality would have different similarity distributions. Inspired by this, we propose the concept of a similarity matrix, which represents various complex similarities through matrices to measure the consistency and coherence. The details are as follows:

Different from the literature (Tay et al., 2017; Darwish and Mohamed 2020), the similarity here is not directly measured on the essay features but is based on the output layer representation of the LSTM neural network. Furthermore, our similarity could measure much more information, such as the similarity that measures the inner and external relationships of the essay, the spatial relationship of the word or sentence, etc.

Let $H_e \in R^{n \times d}$, $H_s \in R^{n \times d}$ be the two output layer

representation vectors, where n is the length of the essay, and d is the dimensionality of the output layer. Mark $S_{H_e} = \{H_{e_i} | i \in n\}$ as the set of the elements of H_e, analogously, $S_{H_s} = \{H_{s_i} | i \in n\}$ as the set of the elements of H_s. We define two kinds of similarities, inner-similarity (inner-s), and cross-similarity (cross-s), of which the form is the matrix.

$$inner-s = s(S_{H_e} \times S_{H_e}) = s\left(\begin{bmatrix} H_{e_1} \cdot H_{e_1} & \cdots & H_{e_1} \cdot H_{e_n} \\ \vdots & H_{e_i} \cdot H_{e_i} & \vdots \\ H_{e_n} \cdot H_{e_1} & \cdots & H_{e_n} \cdot H_{e_n} \end{bmatrix}\right)$$

$$= \begin{bmatrix} 1 & \cdots & s(H_{e_1} \cdot H_{e_n}) \\ \vdots & 1 & \vdots \\ s(H_{e_n} \cdot H_{e_1}) & \cdots & 1 \end{bmatrix} \quad (3\text{-}3)$$

Where '×' is the Cartesian product, '.' is the dot product, and

$$s(H_{e_i} \cdot H_{e_i}) = \frac{H_{e_i} \cdot H_{e_i}}{|H_{e_i}| \cdot |H_{e_i}|} = \frac{\sum_j H_{e_{ij}} \cdot H_{e_{ij}}}{\sqrt{\sum_j H_{e_{ij}}^2} \sqrt{\sum_j H_{e_{ij}}^2}} \quad (3\text{-}4)$$

Especially, because of the same vector H_{e_i} in the main diagonal of the matrix in equation (3-3), all of which have a cosine similarity of 1. Obviously, it doesn't make sense to calculate the similarity between a vector and itself, and also, it will increase the amount of computing. It's necessary to remove the value of 1 on the main

diagonal. More generally, let k>0 be the stride parameter such that H_{e_i} in equation (3-3) does not make dot product with the k-1 neighboring vectors. Then we have

$$inner-s=s\left(\begin{bmatrix} s(H_{e_1}\cdot H_{e_k}) & \cdots & s(H_{e_1}\cdot H_{e_n}) \\ \vdots & s(H_{e_i}\cdot H_{e_{i+k}}) & \vdots \\ s(H_{e_n}\cdot H_{e_1}) & \cdots & s(H_{e_{n-k}}\cdot H_{e_n}) \end{bmatrix}\right) \quad (3\text{-}5)$$

Analogously,

$$cross-s=s(S_{H_e}\times S_{H_s})=s\left(\begin{bmatrix} H_{e_1}\cdot H_{s_1} & \cdots & H_{e_1}\cdot H_{s_n} \\ \vdots & H_{e_i}\cdot H_{s_i} & \vdots \\ H_{e_n}\cdot H_{s_1} & \cdots & H_{e_n}\cdot H_{s_n} \end{bmatrix}\right)$$

$$=\begin{bmatrix} s(H_{e_1}\cdot H_{s_1}) & \cdots & s(H_{e_1}\cdot H_{s_n}) \\ \vdots & s(H_{e_i}\cdot H_{s_i}) & \vdots \\ s(H_{e_n}\cdot H_{s_1}) & \cdots & s(H_{e_n}\cdot H_{s_n}) \end{bmatrix} \quad (3\text{-}6)$$

Where $s(H_{e_i}\cdot H_{s_i})=\dfrac{H_{e_i}\cdot H_{s_i}}{|H_{e_i}|\cdot|H_{s_i}|}=\dfrac{\sum_j H_{e_{ij}}\cdot H_{s_{ij}}}{\sqrt{\sum_j H_{e_{ij}}^2}\sqrt{\sum_j H_{s_{ij}}^2}}$.

Here, we introduce a weight matrix $W\in R^{n\times m}$, where m is a constant, this weight matrix is similar to the attention function. We adjust the value of m to compress or increase the weight of similarity. The similarity matrixes adjusted by the weight matrix W are called weighted-inner-similarity (weighted-inner-s) and

weighted-cross-similarity (weighted-cross-s), respectively. We have

$$weighted-inner-s = flat(s(H_e \times H_e)W) \quad (3\text{-}7)$$

$$weighted-cross-s = flat(s(H_e \times H_s)W) \quad (3\text{-}8)$$

Where $flat(\cdot\ )$ is the flat function to flat the $n \times m$ matrix into a vector.

Lastly, the two kinds of similarities are concatenated into one vector. Shown in Figure 3-2.

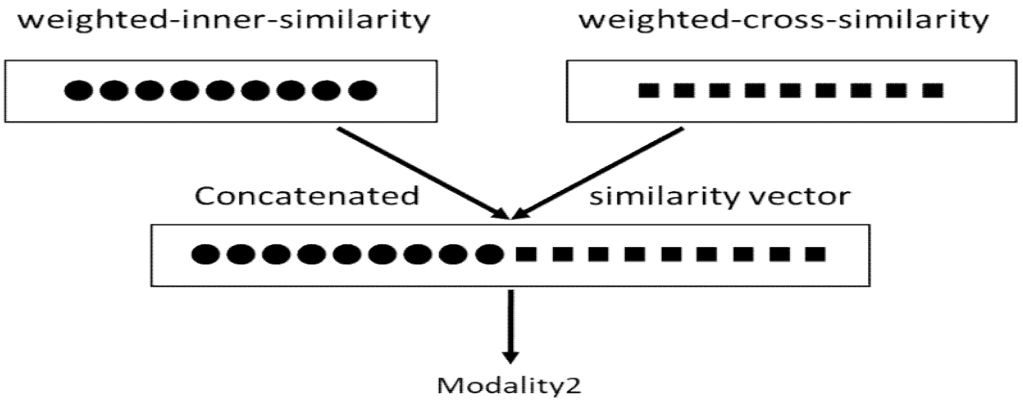

Figure 3-2. Similarity features concatenation.

3.2.3 Scoring related information

Currently, for automated essay scoring, as far as we know, almost all deep neural networks for AES are trained directly between objects (essays) and labels (scores). Including the two self-learning mechanisms mentioned earlier, they are both information among the essay. However, when it comes to manual scoring, we humans often focus on more than the essay itself. We humans usually have some background information in advance, such as scoring criteria, or keywords to answer, etc. In conventional machine learning, researchers

usually design various manual feature extraction methods to get the features for automated scoring. Although conventional machine learning is ineffective to learn semantic information, its average accuracy is also much lower than deep learning. But apparently, this handcrafted feature extraction method, the same as human background knowledge, is a kind of prior knowledge, which helps to improve the average accuracy of automated essay scoring. If we can design such a mechanism to help the neural network learn this kind of knowledge, the average accuracy of automated essay scoring should be improved to some extent. Based on this assumption, here, this thesis studies how to input scoring information beyond the essay into the neural network to improve the average accuracy further.

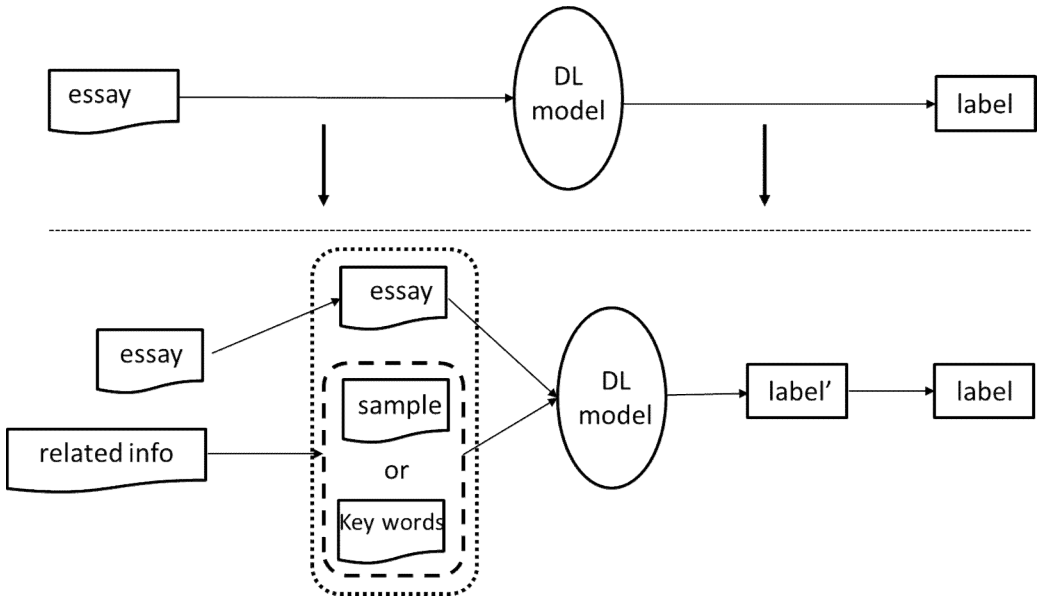

Figure 3-3. Scoring related information added deep learning model.

As shown in Figure 3-3, after introducing external scoring information, the general deep learning model (above section) is

transformed into the below section. Here, the external information we introduce is a sample text provided by an expert or a summary, key points, or keywords to answer the question. The same to the essay to be graded, samples, summary, key points, or keywords are also with different scores. Different input combinations (red boxes in Figure 3-3) will change the corresponding training labels. Therefore, we need to construct a new mapping to handle the relationship between new inputs and new labels and the relationship between new labels and original labels.

Let $E \in R^{n \times d}$, $S \in R^{n \times d}$ be the essay embedding vector, let $K \in R^{m \times d}$ be the sequence vector, where n is the number of the essay, and m is the number of sequences, and d is the dimensionality of the word embedding. $E_i \in R^d$, $S_i \in R^d$, $K_i \in R^d$ are the i-th word embedding of the essay or sequence.

For the essay embedding and sample vectors, the new input combination is defined as

$$input(E, S, \times) = E + E \times S \tag{3-9}$$

where '+' is the concatenation operation that makes the vector E concatenate to $E \times S$, '×'is an operator parameter that has two operations, one is '.' operation, and the other is '-' operation, we have

$$input(E, S, \cdot\ \ ) = E + E \cdot\ \ S = E + \sum_i E_i S_i \tag{3-10}$$

where '.' is a dot product.

$$input(E, S, -) = E + (E - S) = E + \sum_i (E_i - S_j) \tag{3-11}$$

where '-' is a minus.

For the essay embedding vector and keywords vector, we first make an abstract extraction on the essay embedding through a convolutional network and then do operations with keywords. The new input combination is defined as

$$input(E,K,\times) = E + cov(E) \times K \quad (3\text{-}12)$$

where $cov(\cdot\ )$ is convocational neural network output, $cov(E) \in R^{m \times d}$.

Analogously, we have,

$$input(E,K,\cdot\ ) = E + cov(E) \cdot\ K = E + \sum_i cov(E)_i K_i \quad (3\text{-}13)$$

$$input(E,K,-) = E + (cov(E) - K) = E + \sum_i (cov(E)_i - K_i) \quad (3\text{-}14)$$

So far, we have got the new inputs, shown in Figure 3-4. Here, we use the original input and label to construct a new mapping to map labels to the new inputs.

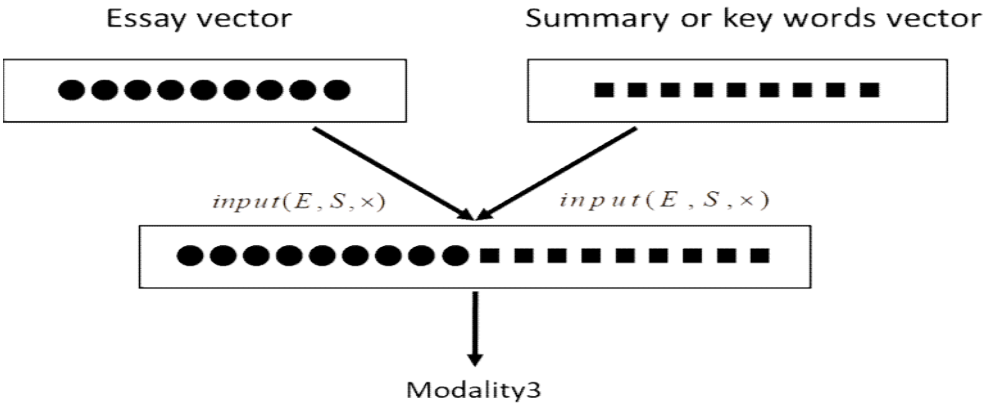

Figure 3-4. Scoring related information concatenation.

Let $f(\cdot\ )$ be the score function, $f(E) = F_E$ denotes the essay E has a score of F_E , Mark the score of $input(E,S,\times)$ as

$f(input(E,S,\times)) = \psi(F_E, F_S)$, where ψ is the new mapping. We do not define $\psi(\cdot\ )$ specifically, because different definitions may be related to the training of deep models. But it needs to meet the following conditions:

1) The function $\psi(\cdot\ )$ is continuous;

2) The function $\psi(\cdot\ )$ is strictly monotonous;

3) The function $\psi(\cdot\ )$ is differentiable.

Thus, function $\psi(\cdot\ )$ is invertible, we have

$$F_E = \psi^{-1}(\psi(F_E, F_S), F_S) \tag{3-15}$$

In deep learning training, if the input $input(E,S,\times)$ has a predicted score of $\widetilde{\psi}(F_E, F_S)$, then, we have the score of E,

$$F_E = \psi^{-1}(\widetilde{\psi}(F_E, F_S), F_S) \tag{3-16}$$

At this point, we have defined a new input, and the mapping of the new input and its label has been completed. The predicted value of the new input can also be used to calculate the label corresponding to the original input. In practical applications, we only need to make the function $\psi(\cdot\ )$ be specifically defined. Chapter 4 has made a specific definition of $\psi(\cdot\ )$.

3.2.4 Other features

Also, we could use some preprocess technology for AES. For example, Named-entity recognition and Sentiment analysis, etc.

Named entity recognition (NER, also known as entity identification, entity chunking and entity extraction, retrieved from Wikipedia) is a

sub-task of information extraction that aims to locate and classify named entities mentioned in unstructured text into predefined categories, Such as person's name, organization, location, medical code, time expression, quantity, monetary value, percentage, etc.

Sentiment analysis (also known as opinion mining or sentiment AI, retrieved from Wikipedia) refers to the use of natural language processing, text analysis, computational linguistics, and biometric recognition to systematically identify, extract, quantify, and study emotional states and personal information. Sentiment analysis has been widely used in the voice of customer materials, such as comments and survey responses, online and social media, and healthcare materials, and its applications range from marketing to customer service to clinical medicine.

3.3 Mechanisms application

For the contents of the above self-learning mechanism, we can choose one or more to integrate into a neural network for training, as shown in Figure 3-5. The experiment shows that mechanism consistency and coherence (modality2) and external scoring information (modality3) are the most useful, syntactic, and semantic (modality1) would increase the calculation cost, and the preprocess technology is very necessary.

When using the mechanisms, the critical point is, for syntactic and semantic, it needs to design a suitable Autoencoder neural network using CNN and attention mechanisms. For consistency and coherence, the weight matrix needs to be set as a rational parameter m. For

External scoring information, it needs to construct a function to map the new inputs and new labels.

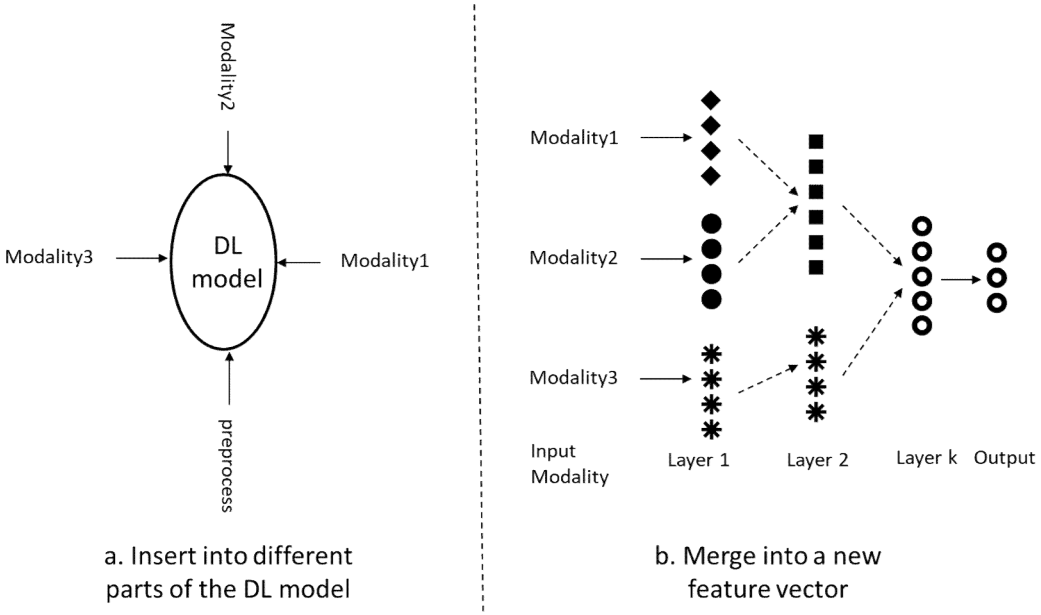

Figure 3-5. Different ways to use the self-learning mechanism.

3.4 Summary

In this chapter, we put forward the idea of self-learning representation mechanisms, and use the mechanism to help the deep model to learn specific knowledge and external knowledge, to improve the learning ability of the deep model and present a general representation of the mechanism. We consider the syntactic and semantic features, consistency, and coherence features, in which we define a similarity matrix for extensive space similarity calculation and the scoring related information. We also think some preprocess technology maybe impact on AES. The self-learning mechanisms make deep learning model has a way to incorporate prior knowledge.

Chapter 4 A novel neural network architecture for AES

4.1 Introduction

Manual scoring has a large workload and sometimes is subjective according to different experts. The goal of automated essay scoring (AES) is to enable computers to score students' essays automatically, thereby reducing the subjectivity of manual ratings and the workload of teachers and speeding up the feedback in the learning process. Currently, there are some AES systems, such as Project Essay Grade (PEG) (Ellis et al., 1966), Intelligent Essay Assessor (IEA) (Foltz et al., 1999), E-rater (Attali et al. 2004), and Besty that are applied to educational practice, but these systems are not promising in the future. AES is quite complicated; it depends on how much the machine could understand the language, such as spelling, grammar, semantics and other grading information. Traditional AES approaches were regarded as a machine learning approach, such as classification (Larkey, 1998; Lawrence and Liang, 2002), regression (Attali and Burstein, 2004; Foltz et al., 1999), or ranking classification problems (Yannakoudakis et al., 2011). These approaches make use of various features, such as the length of the essay, Term Frequency-Inverse Document Frequency (TF-IDF), etc., to achieve AES. One drawback of this kind of feature

extraction is that it is often time-consuming, and the regulation for feature extraction is often sparse, instantiated by discrete pattern-matching, and it being hard to generalize.

The neural network and distributed representation (Santos and Gatti, 2014) have provided the tremendous potential for natural language processing. A neural network can train an essay represented by distributed representation and producing a single dense vector that represents the whole essay. Furthermore, the single dense vector and the score are trained by the neural network to form a one-to-one correspondence. Without any other handcrafted features, a nonlinear neural network model has been shown its particular advantages—that it's much more robust than the traditional statistical models across different domains.

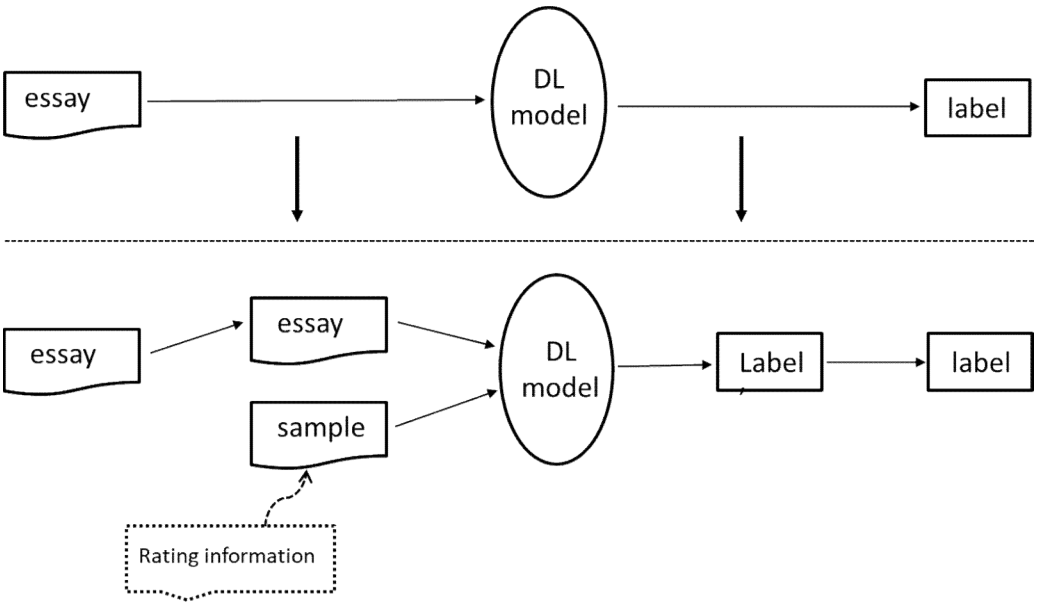

Figure 4-1. The overall framework of the approach.

Recently, many researchers have studied AES using neural

networks (Alikaniotis et al., 2016; Taghipour et al., 2016; Dong et al., 2017; Tay et al., 2018; Bahdanau et al., 2014; Lee et al. 2014) and made quite good progress. These researchers mainly focus on convolutional neural networks (CNN) (Santos et al., 2014; Yin et al., 2016; Zhang et al., 2015), recurrent neural networks (Lipton et al., 2015) (RNN, the most widely used RNN is long short-term memory (LSTM) (Hochreiter et al., 1997)), the combination of CNN and RNN (LSTM), attention mechanisms, and some special internal features representation, such as coherence feature among sentences (Yin et al., 2016). CNN has a useful application in the image (Zhang et al., 2018; Wang et al., 2018), and it can also be applied to sequence models. RNN is very advantageous for sequence modeling. Google applied the attention module to the language mode directly (Vaswani et al., 2017; Dehghani et al., 2018).

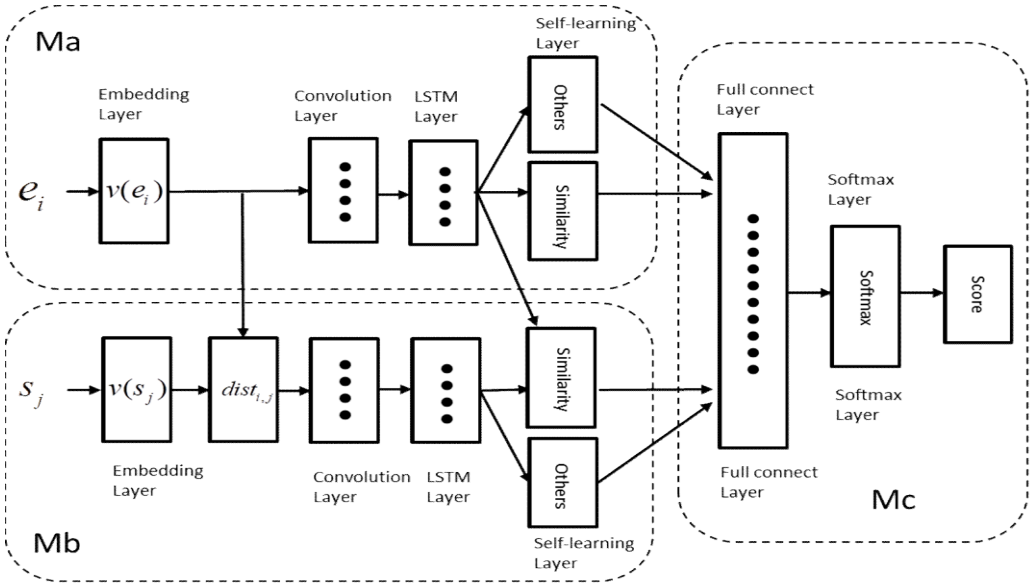

Figure 4-2. Siamese bidirectional long short-term memory architecture model architecture.

However, at present, the researchers applied all kinds of models to AES, only considering the essay itself while neglecting the rating criteria behind the essay. In this thesis, we propose the self-learning mechanism in chapter three. We consider this kind of information and gave an interpretable novel end-to-end neural network AES approach. By representing the rating criteria by introducing some sample essays (the following short as the sample) with different ranks, which were provided by domain experts (if not, manually get an average one from the dataset instead). Thereby, we get some essay pairs as new inputs to AES. Each pair consists of an essay itself and a sample. We propose a Siamese Bidirectional Long Short-Term Memory Architecture (SBLSTMA) to receive the new input to achieve AES. Because the rating information was also involved, our SBLSTMA model can capture not only the semantic information in the essays but also the information beyond the dataset—rating criteria. We explored the SBLSTMA model for the task of AES and used the Automated Student Assessment Prize (ASAP) dataset (ASAP, https://www.kaggle.com/c/asap-aes/data) as evaluation. The results show that our model empirically outperforms the previous neural network AES methods.

Figure 4-1 shows the overall framework of the approach. Different from the previous approaches that train or predict the dataset directly (the above of Figure 4-1), we added rating criteria as a part of input (the bottom of Figure 4-1). Experience tells that human raters give scores not only by essays themselves but also by rating criteria (We use samples instead). Our model is to imitate this behavior of human

raters. We believe that essays don't have all the rating information, and some of that is beyond the essays. Therefore, to take this kind of information as a part of the input is a benefit for scoring. We briefly describe how to make use of this sample first. We simply mark $v(\cdot\ )$ as the distribution represention function; then, $v(e)$ and $v(s)$ are the word embeddings of essay e and sample s, respectively. The difference between the essay vector $v(e)$ and the sample vector $v(s)$ is defined as the distance information of these two. Mark $dist = v(e) - v(s)$ as the distance information, subsequently, as shown in Figure 4-2, $dist$ and $v(e)$ are fed into the model together. We mark pair $(v(e), v(s))$ as the new input, and we can also construct a map to represent the label of pair $(v(e), v(s))$. The detail description of input was described in Section 4.2.

The prime contributions of this chapter are as follows:

• For the first time, we introduce some samples to represent the rating criteria to increase the rating information and construct a pair consisting of an essay and a sample as the new input. We can understand it as how similar is the essay and sample or how close is the essay and sample. This, to a certain extent, is similar to semantic similarity (Mueller et al., 2016) and question-answer matches (Tay et al., 2018). We introduce it to AES.

• We provide a self-feature mechanism at the LSTM output layer. We compute two kinds of similarities: the similarity between sentences in the essay and the similarity between essay and sample. The experiment shows that it is a benefit for the essays, which are long

and complicated. This idea is inspired by the SKIPFLOW (Tay et al., 2018) approach, but we make an extension of it.

• We propose a Siamese Bidirectional Long Short-Term Memory Architecture (SBLSTMA); this is a Siamese neural network architecture that can receive the essay and sample on each side. We use the ASAP dataset as an evaluation. The results show that our model empirically outperforms the previous neural network AES approaches.

4.2 Model architecture for AES

In this section, we define the input, evaluation metric, and the SBLSTMA model architecture.

4.2.1 Input definition

Our input contains the essay and sample. We need to determine a new label (score) for each new input, and after training, we also need to be able to compute the original essay score in the new input. The overall new input and the new label is shown in Figure 4-1 .

According to section 3.2.3 in chapter 3, we define the new mapping officially as follows:

Let G be the score set, $i \in G$ is a score, $|G| = K$, $i \in [0, K]$; Let E be the essays set, $e_i \in E$ is the an essay , $|E| = N$, $i \in [1, N]$; Let S be the sample set, $s_j \in S$ is the a sample , $|S| = C$, $j \in [0, K]$, C is the number of samples set with different score. Usually, C is less than or equals to K.

Let $v(\cdot\ )$ be the word embedding function, we simply mark $v(x)$ as the word embedding of essay x. $dist_{i,j} = v(e_i) - v(s_j)$ was marked as the distance information between e_i and s_j. Let f be the score function, For the essay e_i whose score is $j, j \in [0, K]$, we mark $f(e_i) = j$; similarly, for the sample s_i whose score is $j, j \in [0, K]$, we mark $f(s_i) = j$.

Mark $p_{i,j} = (e_i, s_j)$ is an input, $e_i \in E$, $s_j \in S$, then set $P = \{p_{i,j} | i \in [1, N], j \in [0, K]\}$ is our input dataset. Compared with the original essay dataset E, the new data set P was expanded by $|S|$ times, which S is the samples set.

We use score function $\psi(\cdot\ )$ to denote the score of input $p_{i,j}$, that is, to say, the score of $p_{i,j}$ is $\psi(p_{i,j})$. We define $\psi(p_{i,j})$ (specifically define the function $\psi(\cdot\ )$ mentioned in section 3.2.3 in chapter 3) as :

$$\psi(p_{i,j}) = Cf(e_i) + (C-1)f(s_j), i \in [1, N], j \in [0, K] \quad (4\text{-}1)$$

Where $C = |S|$ is the number of the sample set. Equation (4-1) is a monotone function that meets the condition mentioned in section 3.2.3 in chapter 3 , which was used to initialize the input data's label. Especially, when $C = 1$ equation (4-1) will degenerate into equation (4-2):

$$\psi(p_{i,j}) = f(e_i), i \in [1, N] \quad (4\text{-}2)$$

From equation (4-1) we have

$$f(e_i) = \frac{\psi(p_{i,j}) - (C-1)f(s_j)}{C}, i \in [1, N], j \in [0, K] \quad (4\text{-}3)$$

From equation (4-3) and (4-1), we know $f(e_i)$ is independent of $f(e_j)$, while if we use $\widetilde{\psi}(p_{i,j})$ to denote the prediction value of $\psi(p_{i,j})$, then $f(e_i)$ will be changed. We use $\widetilde{f}(e_i)$ instead of the prediction value of $f(e_i)$ shown in equation (4-3).

$$\widetilde{f}(e_i) = \frac{1}{C}\sum_{s_j \in S} \frac{\widetilde{\psi}(p_{i,j}) - (C-1)f(s_j)}{C}, i \in [1,N], j \in [0,K] \quad (4\text{-}4)$$

Equation (4-3) and (4-4) were used to evaluate the test results of the model. Especially, when $C=1$ equation (4-4) will degenerate into equation (5):

$$\widetilde{f}(e_i) = \widetilde{\psi}(p_{i,j}), i \in [1,N], j \in [0,K] \quad (4\text{-}5)$$

Equation (4-5) and equation (4-2) are consistent in form. Here, we get the new input and their scores (labels). In the actual training, we can gradually increase the number of the sample set. Empirical results show that, usually, $C \leqq 5$ can we get a good result, in rare cases, we need a further discussion at the circumstance of $C > 5$. (according to section 4.4 experiment based on the dataset ASAP in chapter 4)

Now we just use the sample as a part of the input. In 2 ways, can we get the sample. The one is the experts can provide us some samples with different ranks. The other one, also used in this paper, is to use the average value of the vector representation of all the essays that have the same rank to denote the sample. The specific process we get the samples according to equation (4-6).

Assume that M is the number of all the essays with the same

score j, e_i is one of them, then the sample s_j was given by equation (4-6)

$$s_j = \frac{1}{M}\sum_{i=1}^{M} v(e_i) \tag{4-6}$$

Where v is the word embedding function, we defined earlier in this section. For the different score j, we can easily get the sample s_j. The experiment shows that such a way to get the sample is feasible.

4.2.2 Evaluation

Essay score predictions are evaluated using objective criteria. Quadratic Weight Kappa (QWK) measures the agreement between two raters. Different from Kappa, QWK considers quadratic weights by a quadratic weight matrix. This metric typically varies from 0 (only random agreement between raters) to 1 (complete agreement between raters). If there is less agreement between the raters than expected by chance, this metric may go below 0. The QWK is calculated between the automated scores for the essays and the resolved score for human raters on each set of essays. The official evaluation metric of ASAP Kaggle competition is QWK. Moreover, many follow-up researchers who use ASAP datasets to study AES take QWK as an evaluation metric. In this paper, our experiment dataset is the ASAP dataset as well. To make a better comparison with the relevant research, we adopt QWK as an evaluation metric too. The QWK is defined as follows:

$$W_{i,j} = \frac{(i-j)^2}{(N-1)^2} \tag{4-7}$$

Where i and j are the rating and machine rating, respectively, N is the number of possible ratings.

A matrix O is constructed over the essay ratings, such that $O_{i,j}$ corresponds to the number of essays that received a rating i by human and a rating j by machine.

A histogram matrix of expected ratings, E, is calculated, assuming that there is no correlation between rating scores. This is calculated as the outer product between each rater's histogram vector of ratings, normalized such that E and O have the same sum.

From these three matrices W, E and O, the quadratic weighted kappa is calculated:

$$\kappa = 1 - \frac{\sum_{i,j} W_{i,j} O_{i,j}}{\sum_{i,j} W_{i,j} E_{i,j}} \tag{4-8}$$

4.2.3 Model architecture

At present, the researchers applied all kinds of models to AES, only considering the essay itself while neglecting the rating criteria behind the essay. In this study, we consider this kind of information and gave an interpretable novel end-to-end neural network AES approach. We represent rating criteria by introducing some sample essays with different ranks, which were provided by domain experts

(if not, manually get an average one from the dataset instead). Thereby, we get some essay pairs as new inputs to AES. Each pair consists of an essay itself and a sample. We propose a Siamese Bidirectional Long Short-Term Memory Architecture (SBLSTMA) to receive the new input to achieve AES. Because the rating information was also involved, our SBLSTMA model can capture not only the semantic information in the essays but also the information beyond the dataset--rating criteria.

Figure 4-2 shows the SBLSTMA model. As shown in Figure 4-2, the SBLSTMA model consists of three modules: Ma, Mb, and Mc. The different module combinations receive different inputs. The combination of module Ma and Mc receives the essay only; Mb and Mc receive distance information; Ma, Mb, and Mc receive both essay and sample. The results of the three combinations are different. Usually, the third one is the best, the first one is the worst and the second is in the middle of the two. This confirms our previous hypothesis that the more input of scoring information, the better the scoring results.

• Embedding layer

Our model accepts a pair as a training instance each time. Each pair contains an essay e_i and a sample s_j as shown in Figure 4-2. The essay was represented as a fixed-length sequence in which we pad all sequences to the maximum length. Subsequently, each sequence is converted into a sequence of low dimensional vectors via the embedding layer. For the convenience of description, we use the function v to represent the process of word embedding. $v(e_i) \in R^{|V| \times D}$ and $v(s_j) \in R^{|V| \times D}$ are the word embedding outputs,

where $|V|$ is the size of the vocabulary and D is the dimensionality of the word embedding.

After word embedding, we use $dist_{i,j} = v(e_i) - v(s_j)$ represent the distance information between essay e_i and s_j. For instance, sample s_0 has a score of 0 and s_1 has a score of 1, $dist_{i,0} = v(e_i) - v(s_0)$ and $dist_{i,1} = v(e_i) - v(s_1)$ are different, and they have different distance information. We think that the distance information can be trained in the model, and it makes the model easier to converge, especially for those data sets with smaller data volumes.

• Convolution Layer

This layer is optional. We use it for long essays. Once the dense representation of the long input sequence is calculated, it is fed into the LSTM layer of the network. However, it might be beneficial for the network to extract local features from the sequence before applying the recurrent operation. Especially for those essays that are very long. This optional characteristic can be achieved by applying a convolution layer on the output of the embedding layer.

• LSTM Layer

The sequence of word embeddings obtained from the embedding layer (or convolution layer) is then passed into a long short-term memory (LSTM) network (Hochreiter and chmidhuber, 1997).

$$h_t = LSTM(h_{t-1}, x_t) \tag{4-9}$$

where x_t and h_{t-1} are the input vectors at time t. The LSTM model is parameterized by output, input, and forget gates,

controlling the information flow within the recursive operation. The following equations formally describe the LSTM function:

$$i_t = \sigma(W_i \cdot \quad x_t + U_i \cdot \quad h_{t-1} + b_i) \tag{4-10}$$

$$f_t = \sigma(W_f \cdot \quad x_t + U_f \cdot \quad h_{t-1} + b_f) \tag{4-11}$$

$$\tilde{c_t} = \tanh(W_c \cdot \quad x_t + U_c \cdot \quad h_{t-1} + b_c) \tag{4-12}$$

$$c_t = i_t \circ \quad \tilde{c_t} + f_t \circ \quad c_{t-1} \tag{4-13}$$

$$o_t = \sigma(W_O \cdot \quad x_t + U_o \cdot \quad h_{t-1} + b_o) \tag{4-14}$$

$$h_t = o_t \circ \quad \tanh(c_t) \tag{4-15}$$

At every time step t, LSTM outputs a hidden vector h_t that reflects the semantic representation of the essay at position t. The final representation of the essay is again feature-extracted in the self-information layer.

In this thesis, we employ bidirectional LSTM and attention mechanisms in the LSTM layer.

• Self-learning representation mechanism Layer

In this layer, we describe how to use the self-learning mechanism proposed in chapter 3. In particular, we select the similarity as the self-learning information in this layer. The self-learning representation information derives from the vectors obtained from the LSTM layer. We think that the hidden layer representation of essay vector $v(e_i)$ and distance information vector $dist_{i,j}$ should have some external relationships, and the adjacent sentences in the essay should have some internal relationships.

Let h_e be the essay hidden layer, h_{e_t} denotes the vector at

position t of h_e; let h_d be the distance information hidden layer, h_{d_t} denotes the vector at position t of h_d. Here, h_e and h_d correspond to H_e and H_s in section 3.3.2 chapter 3 respectively. Let δ be the length of the sentence (we assume the lengths are the same in different sentences). Then we compute the similarity of vector h_{e_t} at position t and $t+\delta$, here δ has the same meaning to the stride k mentioned in section 3.2.3 chapter 3. We call this similarity inner-similarity (inners-s):

$$inner-s = \frac{h_{e_t} \cdot \quad h_{e_{t+\delta}}}{|h_{e_t}||h_{e_{t+\delta}}|} \tag{4-16}$$

And also, we can compute the similarity in the same position t of vector h_{e_t} and h_{d_t}, we call this similarity cross- similarity (cross-s):

$$cross-s = \frac{h_{e_t} \cdot \quad h_{d_t}}{|h_{e_t}||h_{d_t}|} \tag{4-17}$$

The symbol '.' in equation (4-16) and (4-17) is the dot product. Then let weight matrix $W \in R^{n \times m} (m = 1)$ make a dot product with inner-similarity and cross- similarity, Then we get weighted-inner-similarity and weighted-cross-similarity, which are concatenated into a vector (we also named as inner- similarity and cross- similarity directly) respectively and output to the next layer.

Besides inner-similarity and cross-similarity, we have other 2 main outputs: essay hidden layer and distance information hidden layer. We can do two kinds of processing for these two layers. One way is

to take vector at the last position of h_e and h_d directly; the other way is to take the mean vector over time. We name these 2 vectors as he-vector, and hd-vector. As Figure 4-2 shows, 4 vectors are output to the full connect layer.

• Fully-Connected Layer

Subsequently, we get 4 vectors obtained from the self-information layer: he-vector, hd-vector, inner-feature and cross-feature. We can concatenate these 4 vectors into one, and also we can select several important vectors to concatenate into one according to different essays. Then we output the concatenated one to the softmax layer.

• Softmax Layer

This layer is to classify the output of the fully connected layer. Its classification is achieved by equation (4-18)

$$s(x) = sigmoid(W \cdot \quad x + b) \tag{4-18}$$

Where x is the input vector (the output of fully-connected layer), W is the weight vector, and b is the bias.

4.3 Training

The optimization algorithm we adopt is the Adaptive Gradient Algorithm (Duchi et al., 2011), and the loss function we use is cross-entropy loss function. It's defined as equation (4-19)

$$H(y, \widetilde{y}) = -\sum_{i}^{N}(p(y_i)\log q(\widetilde{y_i}) + (1 - p(y_i))\log(1 - q(\widetilde{y_i}))) \tag{4-19}$$

Where N is the number of training essays, y, $\widetilde{y}$ are the true

label and predicted label of the training essays respectively, p, q are the probability.

In addition, we use the dropout mechanism to avoid training overfitting. Our training method is to train a fixed number of epochs, and each epoch it's trained, the QWK value is tested with the validation data, then the parameters of the best QWK value are saved and used for the model predicting on the test dataset.

The specific training hyper-parameters are listed in table 4-1.

Table 4-1. Training hyper-parameters.

Layer	Parameter Name	Parameter Value
Embedding Layer	Pretrained embedding	GloVe 50-dimensional
Convolution Layer	Window size Filters	5 20
LSTM Layer	Layers Hidden units Dropout	1 64 0.75
Self-learning Layer	Attention length	50
	Epochs Batch size Learning rate	100-300 100-200 0.01

4.4 Experiment

In this section, we describe the procedure of the experiment,

including setup, baseline, results and discussion.

We explored the SBLSTMA model for the task of AES and used the Automated Student Assessment Prize (ASAP) dataset (ASAP, https://www.kaggle.com/c/asap-aes/data) as evaluation. The results show that our model empirically outperforms the previous neural network AES methods.

4.4.1 Setup

The dataset we used is ASAP, a Kaggle competition dataset sponsored by the William and Flora Hewlett Foundation (Hewlett Foundation) in 2012. Many researchers have done the AES study on this dataset; choosing this dataset will help us to compare it with the previous experimental results. It contains eight prompts, each of which is a different genre. It was described in Table 4-2.

We take Stanford's publicly available GloVe 50-dimensional embedding (Pennington et al., 2014) as pre-trained word embedding instead of training it ourselves. Because we think that using the third party pre-trained word embedding makes the model more generally and more open, the data is tokenized with a Natural Language Toolkit (NLTK, http://www.nltk.org/) tokenizer. For those words that can't be found in pre-trained word embedding, we replace them with UNKNOW. In addition, we adopt QWK mentioned in Section 4.2 to measure the output results and use 5-fold cross-validation to evaluate our model.

The software environment in the experimental program run is under Windows 10, Python 3.6, TensorFlow-gpu 1.4, and hardware is CPU: Intel(R) Xeon(R) L5640 @2.27GHz 2.26GHz; RAM: 16G;

HDD:100G; GPU:GTX1080i.

Table 4-2. Statistics of ASAP dataset.

Prompt	Essay #	Avg Length	Scores
1	1783	350	2-12
2	1800	350	1-6
3	1726	150	0-3
4	1772	150	0-3
5	1805	150	0-4
6	1800	150	0-4
7	1569	250	0-30
8	723	650	0-60

4.4.2 Baseline

To evaluate the performance of our model, we take two models, which are the best two as our baselines. One is called SKIPFLOW (Tay et al., 2017), which demonstrates state-of-the-art performances on the benchmark ASAP dataset. The other one is also based on ASAP called attention based recurrent convolutional neural network (LSTM-CNN-att) (Dong et al., 2017), which incorporates the latest neural algorithms such as attention mechanisms, CNN, LSTM, etc. The two models both adopt 5-fold cross-validation to evaluate, and the measured metric is QWK.

SKIPFLOW model considered the neural coherence features within the context, and also, this model has a performance optimization that alleviates and eases the burden of the recurrent model by implicit

access to hidden representations over time.

LSTM-CNN-att model adopts a hierarchical neural network structure. It considers the distributed representation of essays from two levels: sentence and text. The model learns text representation with LSTMs, which could model the coherence and coherence among the sequence of sentences. And also, attention pooling is used to capture more relevant words and sentences that contribute to the final quality of essays.

The results of the two baseline models are listed in Table 4-3.

Table 4-3. The Quadratic Weight Kappa (QWK) value compared with the baseline model.

Model	Prompts								
	1	2	3	4	5	6	7	8	Avg
LSTM-CNN-att	0.822	0.682	0.672	0.814	0.803	0.811	0.801	0.705	0.764
SKIPFLOW	0.832	0.684	0.695	0.788	0.815	0.810	0.800	0.679	0.764

4.4.3 Result and discussion

The results are listed in Table 4-4. Our model SBLSTMA outperforms the baseline model LSTM-CNN-att and SKIPFLOW by approximately 5% on average QWK (Quadratic Weighted Kappa). The results are statistically significant with $p < 0.05$ by 2-tailed $t-test$.

From Table 4-4, we know that the empirical results have been

significantly improved. We think that this is because the knowledge of the rating criteria-distance information plays a very significant role. To explain it, we further decompose the model SBLSTMA to another two submodels. As described in Section 4.3, the model SBLSTMA consists of modules Ma, Mb, and Mc, in which we can get three combined models: Ma + Mc, Mb + Mc, and Ma + Mb + Mc. Ma + Mc means the model receives the essay only without receiving the rating criteria information, and, during the training, it also computes the inner-feature information in the essay. Mb + Mc receives the distance information; during the training, it calculates the inner-feature information in the distance information. Ma + Mb + Mc receives an essay and sample during the training; it computes inner-feature information and cross-feature information. We give the experimental results in Table 4, where the sample sets used were listed in Table 4-5.

The information distance is based on the sample set described in Section 4.2.1. It is directly related to the quality of the experimental results. We need to find the samples that could reflect the rating criteria as accurately as possible. The maximum element of sample set depends on the range of essay's score, but we can't select all the different score essays as the samples, especially for the essays with a large score range; if so, the training will be very time-consuming, and the results are not necessarily good. Empirical results show that, usually, for the dataset that has a narrow score range, we can take all the samples with different scores as a sample set, such as prompts 3, 4, 5, and 6; for the dataset that has a large score range, we can make

some of the samples as a sample set, such as prompts 1, 2, 7, and 8.

The way we get a sample set for the dataset that has a large score range according to the steps as follows:

1. According to Equation (4-6), we compute all the samples s_j of each prompt.

2. For each s_j in a prompt, make a pre-training under Mb + Mc and gives a sort, of which the order is sorted by the quality of Kappa value of the training results.

3. Take the first sample in the sort gives in step 2 as the initial sample set. If the training results are less than the threshold (the result expectation was initialized before), then continue to add the second sample in the sort into the samples set, … , until the results are greater than the threshold or all the samples are added into the sample set.

Take prompt 4, for example, the scores are 0, 1, 2, 3, and the corresponding samples are s_0, s_1, s_2, s_3. By pre-training, we get a sort of $[s_2, s_1, s_3, s_0]$, which means that the training result of s_2 is the best one, s_1 is the second one, and so on. Then, we first take sample set $\{s_2\}$ as the initial sample set, $\{s_2, s_1\}$ as the second one, and so on. Table 4-5 shows the samples that we used in the experiment.

The results of each decomposed sub-model listed in Table 4-4 shows that the Kappa value under model Mb + Mc is better than Ma + Mc. It means that the distance information as input is useful for training. Such an input based on rating criteria contains more rating

information, and it does reflect a certain distance between the essay and sample. For a more intuitive explanation, we provide the Kappa value diagrams of the first 100 epochs of all eight prompts under Ma + Mc and Mb + Mc shown in Figure 4-3.

Table 4-4. The Kappa value under different module combinations.

	Prompts								
Module	1	2	3	4	5	6	7	8	Avg
Ma+Mc	0.521	0.486	0.546	0.685	0.800	0.704	0.469	0.425	0.560
Mb+Mc	0.727	0.670	0.724	0.797	0.817	0.816	0.795	0.658	0.757
Ma+Mb +Mc	0.861	0.731	0.780	0.818	0.842	0.820	0.810	0.746	0.801
SBLST MA	0.861	0.731	0.780	0.818	0.842	0.820	0.810	0.746	0.801

Figure 4-4 intuitively shows that the Kappa value under Mb + Mc is better than the value under Ma + Mc. Furthermore, Table 4-6 shows the mean value and standard deviation value under Ma + Mc, Mb + Mc, and Ma + Mb + Mc. The mean value reflects how good the training results are, while standard deviation indicates the size of training space and the training stability. It is obvious that, based on greater mean value, the greater the standard deviation, the better the results.

From Table 4-6, we can conclude that the training under Mb + Mc is better than the training under Ma + Mc, and the training under Ma + Mb + Mc is much more stable than the other two. Table 4-6 also tells the mean value, and standard deviation of

prompt 8 are relatively worse for the first 100 epochs. We consider this due to the fewest number of essays and the longest essay length and the largest size of the score range of prompt 8. For the other prompts, we can increase the number of samples set to improve the training effect, but, for prompt 8, we are not able to do this. When increasing the number of the sample set of prompt 8, the training process is not stable and is hard to converge. Therefore, in the experiment, the sample set number of prompt 8 is the smallest one.

Table 4-5. The sample set was used in the experiment.

#	Sample set	#	Sample set
1	$\{s_3, s_7, s_9\}$	5	$\{s_0, s_1, s_2, s_3, s_4\}$
2	$\{s_1, s_3, s_4, s_5\}$	6	$\{s_0, s_1, s_2, s_3, s_4\}$
3	$\{s_0, s_1, s_2, s_3\}$	7	$\{s_6, s_{10}, s_{16}, s_{22}, s_{24}\}$
4	$\{s_0, s_1, s_2, s_3\}$	8	$\{s_{29}, s_{46}\}$

Furthermore, from Table 4-6, we know that the results under Ma + Mb + Mc are the best. The average Kappa value of Ma + Mb + Mc is 0.44 greater than that of Mb + Mc. In particular, prompt 2 and prompt 3, which have the worst Kappa value in baseline models was improved obviously in our model. We think that the input under this model contains more information: essay, distance information, and self-feature mechanism, which are suitable for rating.

The value of parameter δ, which denotes the length of the

sentence defined in Section 3.3.4, was fed as 10. To explain it clearly, we take prompt 2 and prompt 3, for example. We give these first 100 epochs under Ma + Mc, Mb + Mc, and Ma + Mb + Mc showed in Figure 4-6. From the figure, we can easily see that model Ma + Mb + Mc made a further improvement than model Ma + Mc and Mb + Mc. Table 6. The mean value and standard deviation of each prompt's Kappa value at the first 100 epochs under Ma + Mc, Mb + Mc, and Ma + Mb + Mc.

Table 4-6. The mean value and standard deviation of each prompt's Kappa value at the first 100 epochs under Ma + Mc, Mb + Mc, and Ma + Mb + Mc. The figure of mean value and standard deviation are shown in Figure 4-4. and 4-5. (M:Mean, S:Std.Deviation)

#		1	2	3	4	5	6	7	8	Avg
M	Ma+Mc	0.366	0.367	0.477	0.606	0.759	0.613	0.24	0.26	0.461
	Mb+Mc	0.614	0.493	0.542	0.711	0.694	0.691	0.26	0.313	0.54
	Ma+Mb +Mc	0.751	0.621	0.681	0.754	0.739	0.727	0.576	0.373	0.653
S	Ma+Mc	0.052	0.069	0.058	0.083	0.048	0.103	0.134	0.066	0.077
	Mb+Mc	0.139	0.148	0.111	0.119	0.192	0.209	0.218	0.090	0.153
	Ma+Mb +Mc	0.037	0.094	0.103	0.033	0.096	0.055	0.137	0.172	0.091

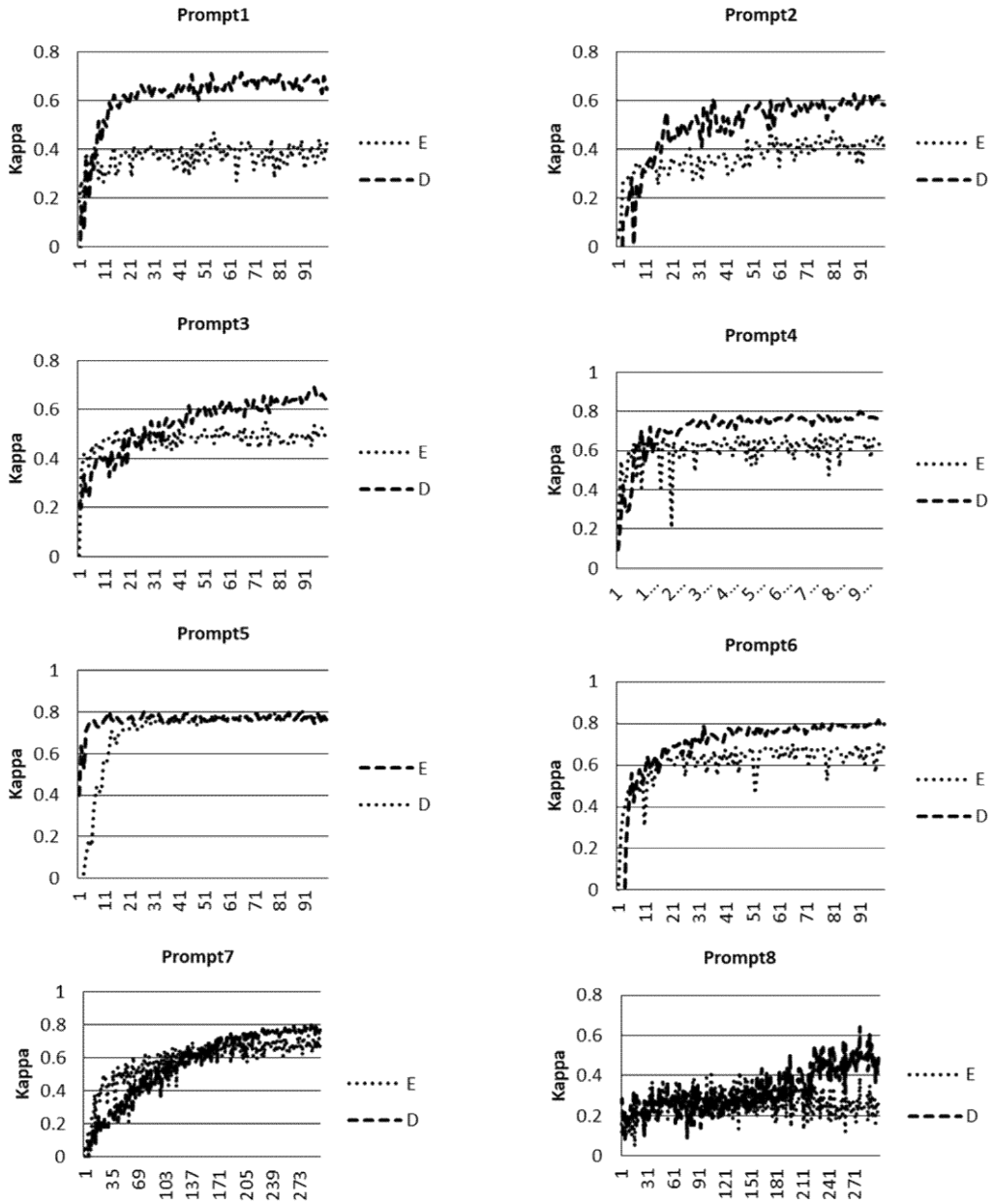

Figure 4-3. Each prompt's Kappa value comparison under Ma + Mc and Mb + Mc at the first 100 epochs (prompt7 and prompt8 are 300epochs), where E denotes the output under Ma + Mc, D denotes the output under Mb + Mc. The X-axis and Y-axis denote epochs and Kappa values, respectively.

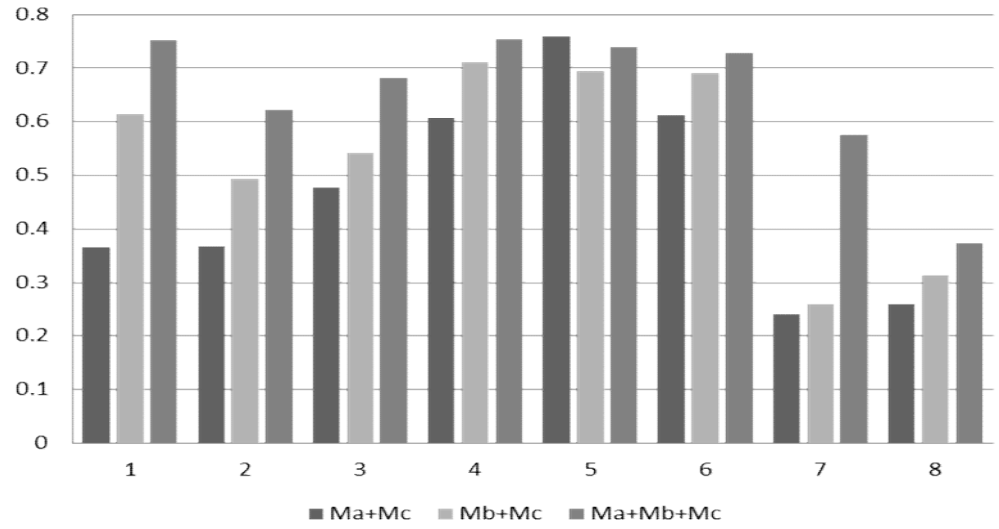

Figure 4-4. The mean value of each prompt's Kappa value at the first 100 epochs under Ma + Mc, Mb + Mc, and Ma + Mb + Mc.

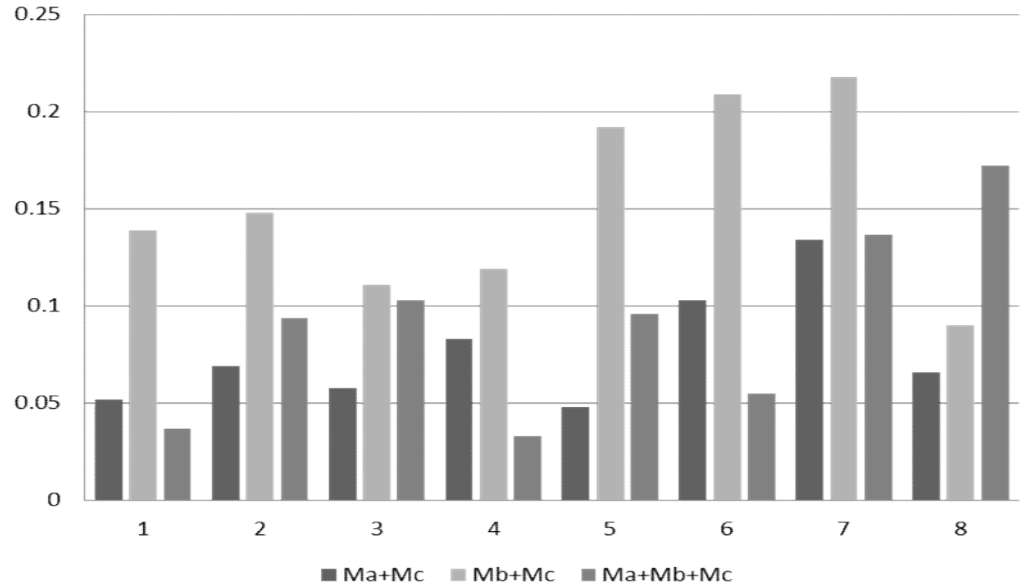

Figure 4-5. The standard deviation of each prompt's Kappa value at the first 100 epochs under Ma + Mc, Mb + Mc, and Ma + Mb + Mc.

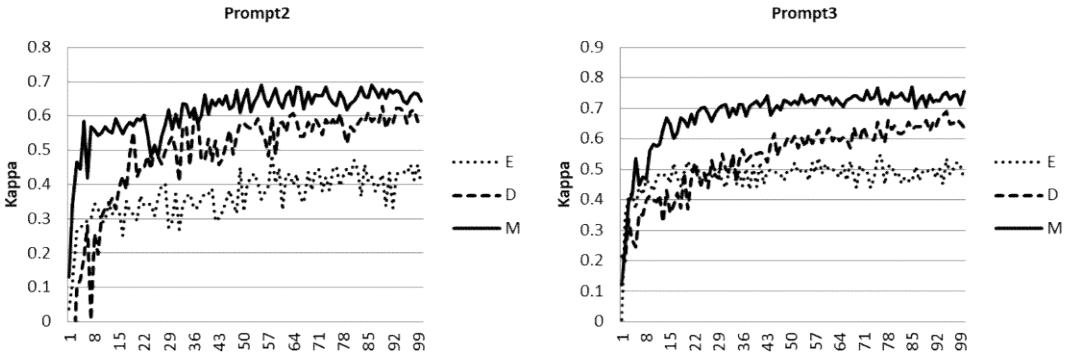

Figure 4-6. Kappa value comparison under Ma + Mc, Mb + Mc, and Ma + Mb + Mc (prompt2 and prompt3), where E denotes the output under Ma + Mc, D denotes the output under Mb + Mc, M denotes the output under Ma + Mb + Mc. The X-axis and Y-axis denote epochs and Kappa values, respectively.

4.5 Summary

In this chapter, we represent the scoring related information--rating criteria behind the essay by some samples and take it as a part of the input. Meanwhile, a self-feature mechanism at the LSTM output layer was provided as well. Then, we propose a novel model, a Siamese Bidirectional Long Short-Term Memory Architecture (SBLSTMA), to learn the text semantics and grade essays automatically. Our approach outperforms the baseline by approximately 5%. By decomposing the model, we find that the model with distance information input is much better than the one without distance information. It means that it is feasible to represent rating criteria from samples. We also hypothesize that distance information

derived from the difference between the examples and the mean example benefits all the other supervised learning methods. We will try using this approach in other fields in the coming future to check whether the hypothesis is right or not. Besides, we will also consider applying data augmentation technology to enhance the essay dataset, of which the example is relatively small.

Chapter 5 Exploration of creativity essay mining

5.1 Introduction

Creativity is defined as the ability to produce original and unusual ideas or to make something new or imaginative from the Cambridge Advanced Learner's Dictionary & Thesaurus. According to the definition, we know that creativity is a subjective concept, which is quite challenging to evaluate. However, many researchers are still working on the evaluation of creativity from cognitive science to machine learning. All these works provide a specific theoretical basis for the creative mining of essay.

In the field of cognitive science, early in 1947, Guilford et al. started to find out ways of recognizing creative people. By the middle of the 1960s, the Guilford Alternate Uses test (Guilford, 1967) was widely used for evaluating creativity around the world. Sternberg et al. (1992) believe that creativity can be seen as a function of six variates: knowledge, intelligence, personality, motivation, way of thinking, and environment. These variates may fluctuate daily due to changes in people's internal and external environments, causing different subscales (such as painting or writing) to shift in different ways. Psychological methods adapt to these multiple factors by

managing a large number of short-term tests to cover all aspects of creativity. Most tests require people to generate or manipulate a large number of ideas. Guilford et al. (1966) provided fifty seven tasks, asking participants to do something, such as grouping and regrouping objects based on common attributes, listing the results of unlikely occurrences, and the "use of objects" task. Runco & Pritzker (1999) believe that originality and practicality are the two most essential components of creativity. Creativity depends on generating various ideas and subsequent trimming. Runco and Pritzker (1999) also proposed that improving and perfecting a specific idea can enhance the quality of the idea so that a well-designed level may be an indicator of quality. This may also indicate that the concept is more practical or applicable. One requirement for a detailed description is that the idea must first be specified only if there is a tangible link between the object and its use. Since the response will not be seen or edited, there is also the danger that some ideas may not be used legally at all. A highly detailed statement may correspond to a more appropriate response. In cognitive science, the main objects of creativity evaluation are people, and the research content is to design reasonable methods to evaluate creativity.

In machine learning, new learning research proves that creativity is an essential issue in the field of education. The best way to assess learning performance and student creativity is to compose questions. Forster and Dunbar (2009) proposed a new calculation method for scoring creativity: Latent Semantic Analysis (LSA), a tool for measuring the semantic distance between words. 33 participants

provided creative uses for 20 individual objects. They compared the scores of human judges and LSAs and found that the LSA method can better reflect the potential semantic originality of the response than traditional methods. Zhu et al. (2009) explored the measurement of creativity from the perspective of computer science and cognitive psychology. Darwish and Mohamed (2020) proposed a system that uses latent semantic analysis (LSA) and fuzzy ontology to evaluate papers, where LSA will be responsible for checking semantics. The fuzzy ontology is used to test the consistency and consistency of the article, because this is the best way to overcome the ambiguity of the language, and the system will also provide students with scored feedback. Guan et al. (2019) proposed a Chinese paper scoring method based on lexical features. Amplayo, etc. (2019) Assess the research novelty in the paper. The author evaluates novelty based on the time the paper was published and the impact of the paper. Since most of the work used to detect the novelty of papers usually use Autoencoder neural networks. The author also compared their method with the Autoencoder. In this chapter, an Autoencoder is also used to compare with the proposed method.

Despite the above research works, there is still something more about creativity essay mining that need us to explore. Assessing creativity is a highly subjective activity, as far as we know, there are a few kinds of research on the evaluation of essay creativity directly using traditional machine learning. As for deep learning, as mentioned above, many are used for automated essay scoring. However, the paper that explicitly uses deep learning to evaluate essay creativity still has

not been seen.

Here, based on the AES approach proposed, we further explore creativity essay mining. Our overall thought is that a creative essay should first be the essay with a higher score. Therefore, before creativity essay mining, we could make an AES rating first, then, we select those essays with higher scores as the objects of creativity essay mining from the scoring results. Under this premise, we look for those essays that are non-mediocre and special. According to the concept of creativity mentioned earlier, we believe that such essays are more creative. The critical point is how we could find out these non-mediocre and special essays, which is also the work to be explored in this chapter.

Contextual word representations have recently been used to perform state of the art performance over a series of language understanding tasks (Gehring et al., 2017; Devlin et al., 2018; Radford et al., 2018; Peters et al., 2018). These representations are obtained by optimizing a language modeling (or similar) objective on massive numbers of text. The essential architecture may be convocation, as convolutional sequence to sequence learning (Gehring et al., 2017), recurrent, as in ELMo (Peters et al., 2018), multi-head self-attention, as in OpenAI's GPT (Radford et al., 2018) and BERT language model (Devlin et al., 2018), which are based on the Transformer (Vaswani et al., 2017). Recently, the GPT-2 model (Radford et al., 2019) exceeded other language models in large margin, again based on self-attention. Among BERT (Devlin et al., 2018), Devlin et al. came up with the idea of first hiding parts of the text, then to predict them.

Bowman et al. (2015) proposed a method to generating sentences from a continuous space. Fedus et al. proposed MASKGAN, a kind of Generative Adversarial Networks (GANs), in which authors introduced an actor-critic conditional GAN that fills in missing text conditioned on the surrounding context. Inspired by these two papers, we think that if there is a smart enough generator, and also, we consider to mask the part of the given essay and then let the generator generate the hidden part. If the generated essay is very similar to the original one, we consider this essay is an ordinary one. On the contrary, if the generated essay is different from the original essay largely, then we think this essay is more creative. Because of that, we have reason to believe that a creative essay is more difficult to predict.

Based on the above ideas, we exploratively propose an unsupervised creativity essay mining method, as shown in figure 5-1. The mining process of creativity essay is described as follows:

Firstly, to ensure the generation effect of the generator, we can only mask a part of the essay at a time, meanwhile, to make sure that the masked part of the essay can cover the full text to make the mining of creativity more comprehensive. It needs several times to mask the essay. This is the idea of K-fold cross-validation, which is equivalent to turning the verification part of K-fold cross-validation into masked text, and the training part is used to generate or predict text.

Secondary, we build a network of Generative Adversarial Networks for training a smart enough generator to predict masked parts of the text. Referring to the state of the art NLP technology, we use a

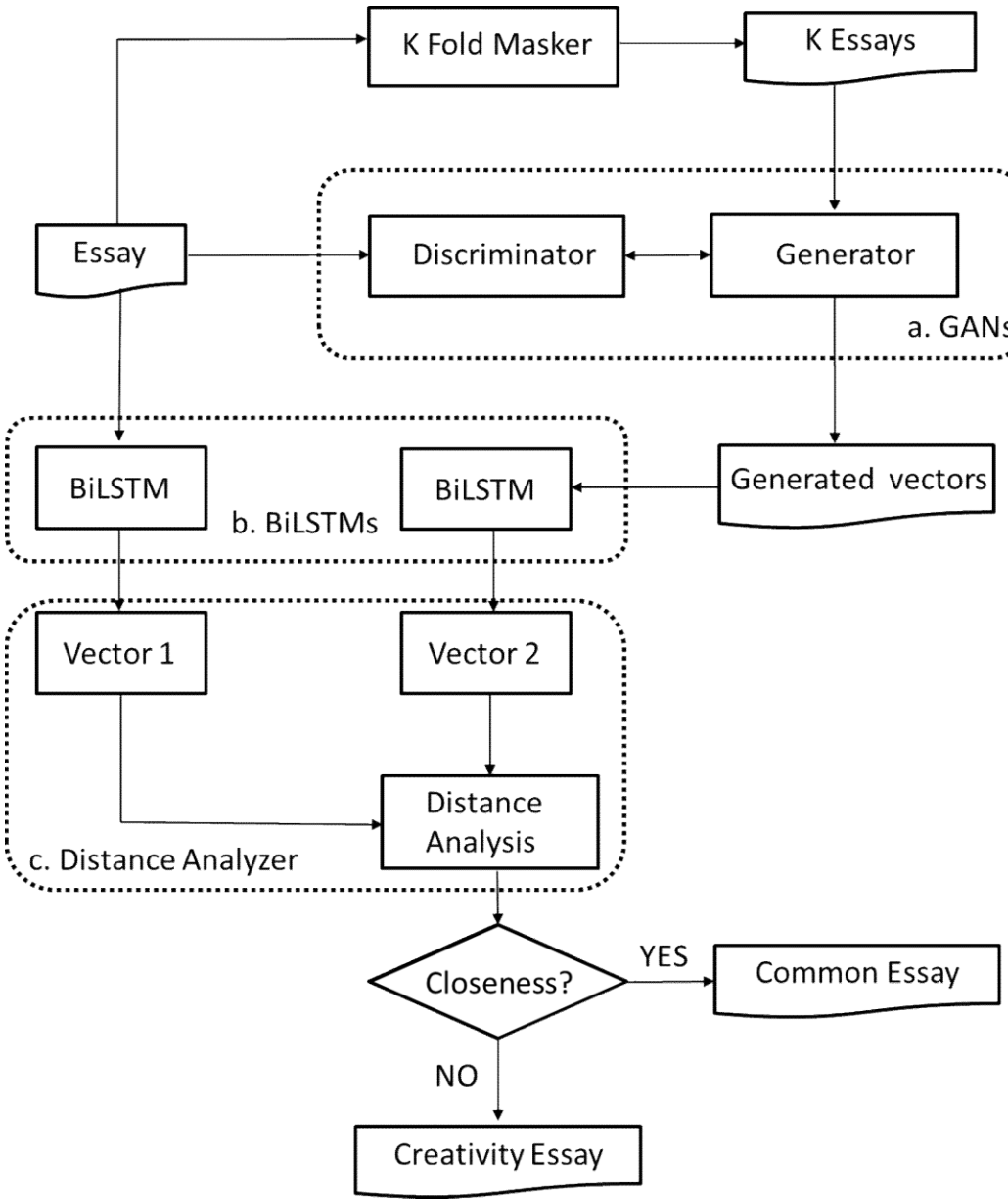

Figure 5-1. Overall of creativity essay mining.

convolutional network as a discriminator and an LSMT-based encoder-decoder neural network as the generator. The convolutional network is used to summarize the essay and predict

the essay and determine the difference between the two. The LSMT-based encoder-decoder is used to generate a predicted essay.

Thirdly, we build a BiLSTM network for representing the essay and generated essays as vectors, which are sent to the distance analyzer for analysis and make a judgment of whether they are creativity essays or not.

The prime contributions of this chapter are as follows:

• A K-fold mask method is proposed. We introduce the K-fold cross-validation method into a text mask to make sure that the masked part of the essay can cover the full text. Also, the masked part of the essay can be distributed evenly on the full text, which could make the mining of creativity more comprehensive.

• We integrate the state of the art text generation technology for creative text mining. Referring to TextGAN (Zhang et al., 2016) and MASKGAN (Fedus et al., 2018), we build a stable Generative Adversarial Networks with high prediction accuracy.

• The proposed creative text mining method is an unsupervised method, which reduces the amount of work required for data annotation.

• We developed a small scale dataset for creativity essay measure.

5.2 Model architecture for creativity essay mining

In this section, we describe the mask method, evaluation metric, and model architecture.

5.2.1 K-fold mask

Text is a discrete sequence. It is tough to train a neural network to generate text or predict text. Because of that, the text generating process is easy to accumulate errors. The prediction of a wrong word will have a great impact on subsequent sentences and even produce completely opposite sentences. Many researchers use the method of hiding part of the text to train the generator to improve the prediction accuracy of the generator (Zhang et al., 2016; Fedus et al., 2018; Devlin et al., 2018). In such a way, we also could reduce the accumulation of errors generated. Inspired by this, here, we propose a K fold mask method for creativity essay mining.

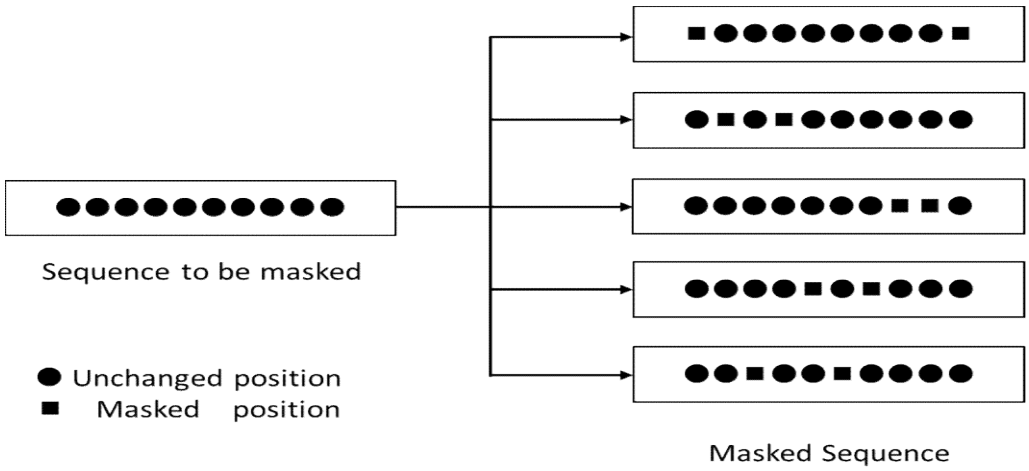

Figure 5-2. A K-fold mask example of sequence with 10 positions, K=5.

Creativity is an ability to produce original and unusual ideas or to make something new or imaginative. This ability is manifested in the text as unusual words, sentences, semantics, etc. which may exist in

any position of the essay. If we mask these creative positions, the creativity of the essay will become lower. Then we compare the masked essay with the original essay to see if the creativity has decreased. So we should mask each possible position in the essay, But we can't mask them all at once, we could only mask them multiple times. Therefore, we introduce the K-fold cross-validation idea for the essay mask. We call it K fold mask. The mask method is as follows:

Let $E=(w_1, w_2, \cdots, w_T)$ be an essay, w_t is the t-th word in the essay. Let $P=(p_1, p_2, \cdots, p_T)$ be a position sequence corresponds to E that with the same length of E. Let $M_j=(m_{j_1}, m_{j_2}, \cdots, m_{j_T})$ be the j-th random sequence of P, $m_{j_t} \in \{0,1\}$. A mask is generated stochastically on M_j, in which $m_{j_t}=0$ means the word at position t, e_t is then replaced with a special mask token <m>, if $m_{j_t}=1$ remains unchanged, then

$$m_{j_t}=\begin{cases}0 & t \in [(j-1)\times floor(\frac{T}{K}), j\times floor(\frac{T}{K})], \quad j<K \\ 1 & otherwise\end{cases} \quad (5\text{-}1)$$

Where $K>0$ is the number of mask times of essay, $floor(\cdot\ )$ denotes the rounding function. Figure 5-2 shows a mask example of sequence with 10 positions, K=5.

5.2.2 Evaluation

The evaluation of creativity essay is highly subjective. Currently,

as far as we know, there is no literature study on how to measure creativity essay in machine learning. As exploration research on creativity essay mining, we use a combination measure of quantitative and qualitative analysis to evaluate creativity essay. For qualitative analysis, we compare the proposed model to compared methods for evaluating whether the different methods have consistent or not. And also, we demonstrate the specific example to analyze the content of the essay to evaluate whether the output of the model is reasonable.

For quantitative analysis, we further decompose the distance indicator to 4 sub-indicators and analyze the different distance indicators between different essays to evaluate the creativity level among different essays. The 4 distance indicators are described in detail in section 5.4.3.

We introduce the F1 score as an evaluation metric quantitative analysis. The F1 score (also F-score or F-measure) is a measure of a test's accuracy. It takes into account the precision p and the recall r of the test to compute the score: p is the correct number of positive results divided by the number of all positive results returned by the classifier. We use precision to judge the accuracy of the recommended creativity essay. We want this indicator to be higher because we don't want to recommend some common essay. r denotes the correct number of positive results, which is divided by the number of all relevant examples . Generally, we do not require the model to recognize all creativity essays so that this indicator could be lower. The F1 score is the average of precision and recall, which reaches the best value at 1 (the best precision

and recall) and reaches the worst value at 0.

Table 5-1. Confusion matrix

	Actual positive	Actual negative
Predicted positive	True positive	False positive Type I error
Predicted negative	False negative Type II error	True negative

According to table 5-1, we have,

$$p = \frac{\sum True positive}{\sum True positive + \sum False positive} \tag{5-2}$$

$$r = \frac{\sum True positive}{\sum True positive + \sum False \neg ative} \tag{5-3}$$

$$F_1 - score = 2\frac{p \cdot \quad r}{p + r} \tag{5-4}$$

In addition, we introduce the T-test and Person and Spearman correlation coefficient to evaluate the significant difference and correlation of the output.

5.2.3 Model architecture

We have shown the Overall creativity essay mining in the introduction. As shown in Figure 5-1, there are three key parts of the model. They are GANs, in which a generator and a discriminator are included, two BiLSTMs, and a distance analyzer. We describe the model in four parts: generator, discriminator, BiLSTMs, distance

analyzer.

• Generator

Let $e = (w_1, w_2, \cdots, w_T)$ be the essay vector, w_t is the t-th word vector in essay e. We employ BiLSMT encoder-decoder to build the generative networks. First, the masked essay vector $e' = (w'_1, w'_2, \cdots, w'_T)$ is encoded into latent vector $z = (a_1, a_2, \cdots, a_T)$ by encoder BiLSTM, then z is decoded into generated essay vector $\tilde{e} = (\widetilde{w_1}, \widetilde{w_2}, \cdots, \widetilde{w_T})$. The process is shown in Figure 5-3.

Here, we detail the BiLSTM decoder that translates a latent vector z into the predicted essay vector $\tilde{e} = (\widetilde{w_1}, \widetilde{w_2}, \cdots, \widetilde{w_T})$. $\tilde{e}$ is generated by the following equation:

$$p(\tilde{e}|z) = p(\widetilde{w_1}|z)\prod_{t=2}^{T} p(\widetilde{w_t}|\widetilde{w_{<t}}, z) \tag{5-5}$$

We generate the first word $\widetilde{w_1}$ from z, with $p(\widetilde{w_1}|z) = \arg max(Vh_1)$ where $h_1 = \tanh(Cz) + b_1$. b_1 is the Bias. All other words in the essay are then sequentially generated using the LSTM with Equation (5-5) until the end symbol is generated. Each condition $p(\widetilde{w_t}|\widetilde{w_{<t}}, z)$ is specified as $\arg max(Vh_t)$, where h_t is the hidden units, are recursively updated through

$$h_t = H(y_{t-1}, h_{t-1}, z) \tag{5-6}$$

where the transition function $H(\cdot)$ is implemented with an LSTM. The specific computational equations are the same as

equation (4-10) to (4-15). For readability, again, we list the equation without equation tags as follows:

$$i_t = \sigma(W_i \cdot \quad x_t + U_i \cdot \quad h_{t-1} + b_i)$$

$$f_t = \sigma(W_f \cdot \quad x_t + U_f \cdot \quad h_{t-1} + b_f)$$

$$\tilde{c_t} = \tanh(W_c \cdot \quad x_t + U_c \cdot \quad h_{t-1} + b_c)$$

$$c_t = i_t \circ \quad \tilde{c_t} + f_t \circ \quad c_{t-1}$$

$$o_t = \sigma(W_O \cdot \quad x_t + U_o \cdot \quad h_{t-1} + b_o)$$

$$h_t = o_t \circ \quad \tanh(c_t)$$

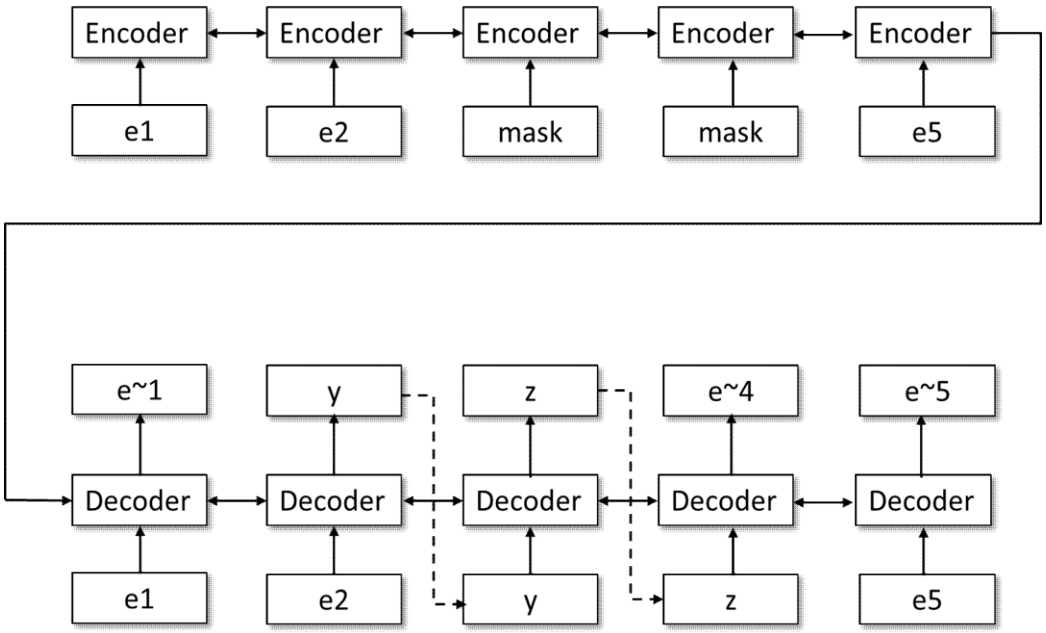

Figure 5-3. The framework of LSTM generator

• **Discriminator**

The CNN architecture are extensively used for sentence encoding and summary abstracting (Kim., 2014; Collobert et al., 2011), which consists of convolution layers and pooling layers over the entire essay

or each feature map. For an essay $e=(w_1, w_2, \cdots, w_T)$ with the length of T (padded where necessary) is represented as a matrix $e \in R^{k \times T}$, the t-th column of e is w_t. Then the convolution operation is as follows:

A convolution operation involves a filter $W_{c_i} \in R^{k \times h}$, applied to a window of h words to produce a new feature. Generally, for a feature map

$$c_i = f(e * W_{c_i} + b_i) \in R^{T-h+1} \quad (5\text{-}7)$$

where $f(\cdot)$ is a nonlinear activation function such as the ReLU function or sigmoid function, $b_i \in R^{T-h+1}$ is a bias vector and $*$ denotes the convolutional operator, W_{c_i} could be multiple filters with varying window sizes. Different filters can be seen as different linguistic features detector that learns different semantic features. Each position in the essay is extracted features independently by W_{c_i}. We then apply a max-over-time pooling operation to the feature map, by which the pooling scheme tries to capture the most important feature. In such a way, for each feature map, the pooling is effectively filtering out less informative compositions of words. And simultaneously, the pooling operation also greatly reduces the complexity of the next layer's convolutional operation.

Assume we have m window sizes $\{h_1, h_2, ..., h_m\}$, and for each window size, we use d filters; then, we obtain a $d(T-h_i+1)$

dimensional vector for c_i to represent an essay. We concatenate all these $c_1, c_2, \cdots c_m$ into a $\sum_{i=1}^{m} d(T-h_i+1)$ dimensional feature vector layer, then using a Softmax layer to map the input essay to an output distribution $D(e) \in [0,1]$. We use this distribution to represent the input essay e. Similarly, the generated essay generated by the generator is also represented as $[0,1]$ distribution. The discriminator outputs the discriminant result by comparing the two distributions. The framework of the CNN discriminator is shown in Figure 5-4.

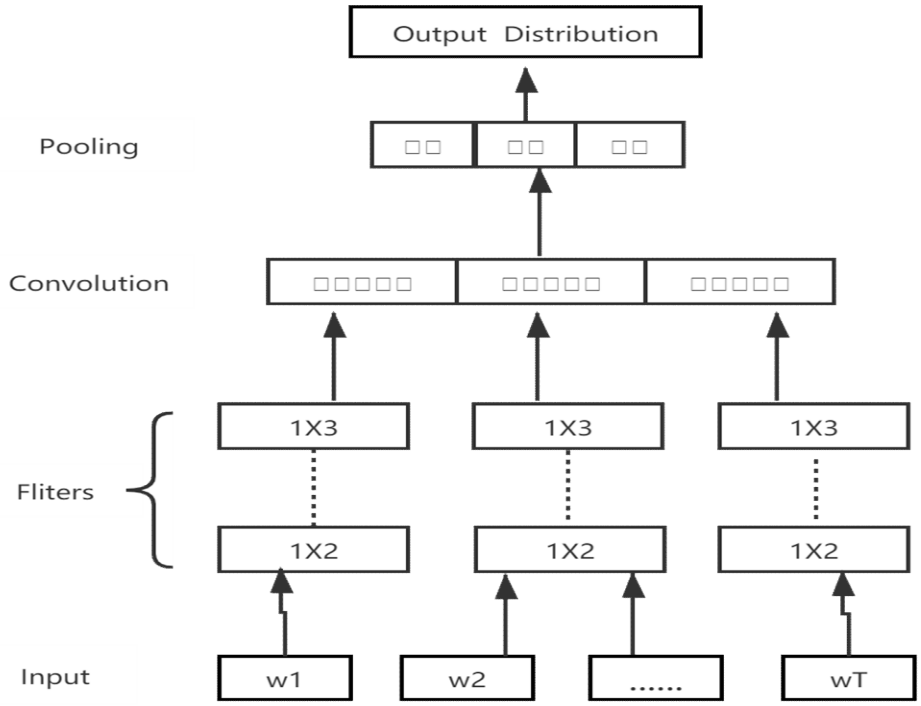

Figure 5-4. The framework of CNN discriminator.

• BiLSTM

Different from the BiLSTM in the generator is used for training and generating essays, the BiLSTM used here is mainly used to represent the essays as output feature vectors for the distance analyzer processing. We use the LSTM output layer of the SBLSTMA model proposed in Chapter 4 to represent the essay vector, as shown in Figure 5-5. We removed Mc part and self-learning mechanism layer, remaining the LSTM output layer and using the weight parameters of the experimental training in Chapter 4 as model parameters.

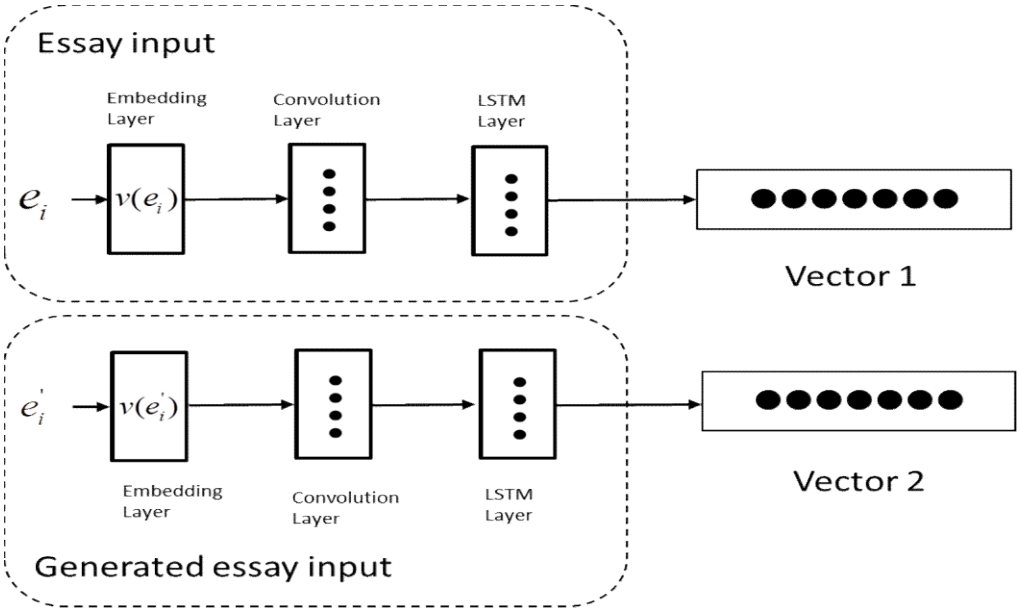

Figure 5-5. The framework of essay vector representation.

• Distance analyzer

We calculate the distance between the represented essay vector (vector $1, v_1$) and the represented generated vector (vector $2, v_2$) by

BiLSTMs, and set a threshold, by comparing the distance and threshold to determine whether an essay is creativity essay. As shown in Figure 5-1.

Using distance to measure creativity between essays is an unsupervised way. According to equation (5-8), we calculate the distance of essay vector (vector $1, v_1$) and generated vector (vector $2, v_2$)

$$d(v_1, v_2) = \sqrt{\sum_i (v_{1_i} - v_{2_i})^2} \tag{5-8}$$

Let $h > 0$ be the threshold, we say v_1 is a creativity essay if $d(v_1, v_2) > h$. Also, we could compare the level of creativity of the two essays. Let $v_1^{'}$ and $v_2^{'}$ be another essay vector and its generated vector. We say the essay represented by vector v_1 is more creative than the essay represented by vector $v_1^{'}$ if $d(v_1, v_2) > d(v_1^{'}, v_2^{'})$.

5.3 Training

According to the above model architecture, the training of this model needs to solve two issues: design a reasonable loss function and training method. We integrate the techniques (Zhang et al., 2016; Tim et al., 2016; Fedus et al., 2018; Goodfellow et al., 2014) for the GANs training.

We design the loss function first. Zhang et al. (2016) propose a method for text generating. Here, we further optimize this method for

small scale dataset and apply it to training creativity text mining. The main idea of optimization is to use the eigenvalues of the matrix to weaken the similar conditions in the loss function.

Given the essay set $S=\{e_1,e_2,\cdots,e_n\}$, instead of directly minimizing the objective function from standard GAN (Goodfellow et al., 2014), Zhang et al. adopted an approach similar to feature matching (Tim et al., 2016). The optimization training schemes consists of two steps:

minimizing:

$$L_D=-E_{s\sim S}\log D(s)-E_{z\sim p_z(z)}\log[1-D(G(z))] \quad (5\text{-}9)$$

$$L_G=tr(\Sigma_s^{-1}\Sigma_r+\Sigma_r^{-1}\Sigma_s)+(\mu_s-\mu_r)^T(\Sigma_s^{-1}+\Sigma_r^{-1})(\mu_s-\mu_r) \quad (5\text{-}10)$$

where Σ_s,Σ_r represents the covariance matrices of essay and generated essay feature vector f_s,f_r, respectively. μ_s,μ_r denote the mean vector of f_s,f_r, respectively. Equation (5-10) is the Jensen-Shannon divergence between two multivariate Gaussian distribution $N(\mu_r,\Sigma_r)$ and $N(\mu_s,\Sigma_s)$, which is usually used to calculate the similarity for two distribution. $\Sigma_s,\Sigma_r,\mu_s,\mu_r$ should be initialized first. Set $\Sigma_s=\Sigma_r=I$. Instead of capturing the first moment similarity, the author proposes stricter criteria to match the feature covariance of real and synthetic data. Although the feature vectors are not necessarily Gaussian distributed, empirically, this loss Equation (5-10) works well. Intuitively, this technique provides a stronger signal for modifying the generator to make the synthesized data more realistic.

However, there is an implicit condition in Equation (5-10) that the matrices Σ_s and Σ_r are non-singular matrices. This condition will not be satisfied in the following cases:

1. If the subset size is less than the feature dimensionality, covariance matrix Σ_s and Σ_r. would result in singular matrices. To remedy this issue, Zhang et al. use a sliding window of most recent m mini-batches for estimating both Σ_s and Σ_r.

2. Due to lack of constraints, both matrices Σ_s and Σ_r may still become singular matrices during training.

Here, we give weaker similar conditions that assume matrices Σ_s and Σ_r have the eigenvalues $\lambda_s = [\lambda_{s1}, \lambda_{s2}, \cdots, \lambda_{sn}]$ and $\lambda_r = [\lambda_{r1}, \lambda_{r2}, \cdots, \lambda_{rn}]$ respectively. If the number of one of the matrices' eigenvalues less than n, we padding with 0. Then computing the cosine similarity equation (5-11) instead of Equation (5-10).

$$L_G = \frac{\lambda_r \cdot \lambda_s}{|\lambda_r||\lambda_s|} \tag{5-11}$$

So far, we have the loss function for training the GANs. The second issue is to find a training method. If we use the traditional back-propagation algorithm to train the model, the word vector generated may do not have a specific word embedding corresponding to it. Because of that, the text word vectors are discrete. We think that it's not necessary to use specific word embeddings corresponding to words to train the model. The experiment shows that whether to use

word embeddings to train or not, the experimental results are not significantly changed. On the contrary, it will increase the complexity of model training. But if we want to explore what words the generator does generate, which is also attractive. Here we conclude that if we needn't generate the text words, then we can use the traditional back-propagation algorithms, such as AdagradOptimizer, AdagradDAOptimizer, RMSPropOptimizer, and so on (here we use AdagradOptimizer). If not, it's necessary to employ a specific training algorithm that is suitable for discrete distribution. Here, we demonstrate the text generation method proposed by Fedus et al. (2018) for discrete distribution.

For the text generating issue, because of that, the traditional back-propagation algorithm is not applicable. We need to find a method suitable for the discrete variables to train the generative adversary network. The actor-critic architecture (Fedus, et al., 2018) is good be used for generating text.

According to Fedus et al. (2018), they gave a reward and punishment mechanism. The generator tries to maximize the cumulative total reward $R = \sum_{t=1}^{T} R_t$. That means to optimize the parameters of the generator, θ, by performing gradient ascent on $E_{G(\theta)}[R]$. Using one kind of the reinforce family of algorithms, they find an unbiased estimator of this as $\nabla_{\theta} E_G[R_t] = R_t \nabla_{\theta} \log G_{\theta}(\widetilde{w_t})$. The variance of this gradient estimator could be reduced by employing the learned value function as a baseline $b_t = V^G(w_{1:t})$, which is

provide by the experts. This results in the generator gradient contribution for a single token $\widetilde{w_t}$

$$\nabla_\theta E_G[R_t] = (R_t - b_t)\nabla_\theta \log G_\theta(\widetilde{w_t}) \quad (5\text{-}12)$$

Where $R_t = \sum_{s=t}^{T} \gamma^s \log P(\widetilde{w_t} = w_t | \widetilde{w_{0:T}}, m(x))$, γ is the discount factor at each position in the sequence. In the nomenclature of Reinforcement Learning, the quantity $(R_t - b_t)$ can be explained as an estimate of the advantage $A(a_t, s_t) = Q(a_t, s_t) - V(s_t)$. Here, the action a_t is the token chosen by the generator $a_t = \widetilde{w_t}$ and the state s_t are the current tokens produced up to that point $s_t = \widetilde{w_1}, \cdots, \widetilde{w_{t-1}}$. This approach is an actor-critic architecture where G determines the policy $\pi(s_t)$ and the baseline b_t is the critic (Sutton & Barto, 1998; Degris et al., 2012).

Fedus, et al. design rewards for individual sequences at each time step to help with credit allocation (Li et al., 2017). As a result, the tokens generated at the time step t will affect the rewards received at that time step and subsequent time steps. The gradient of the generator will include the contribution of each filled token to maximize the discounted total return $R = \Sigma_{t=1}^{T} R_t$. The full generator gradient is given by Equation (5-13).

$$\nabla_\theta E[r] = E_{\widetilde{w_t} \sim G}\left[\sum_{t=1}^{T} (R_t - b_t)\nabla_\theta \log(G_\theta(\widetilde{w_t}))\right]$$

$$= E_{\widetilde{w}_t \sim G}\left[\sum_{t=1}^{T}\left(\sum_{s=t}^{T}\gamma^s \log P(\widetilde{w}_t = w_t | \widetilde{w}_{0:T}, m(x)) - b_t\right) \nabla_\theta \log(G_\theta(\widetilde{w}_t))\right] \tag{5-13}$$

Equation (5-13). shows that the gradient of the generator associated with the generation of $\widetilde{w}_t$ will depend on all discounted future rewards allocated by the discriminator ($s \geqq t$). For a non-zero λ discount factor, the generator will be punished for greedily choosing tokens that only receive high rewards at that time step. Then for a complete sequence, we add all the generated words $\widetilde{w}_t$, where $t = 1 : T$.

Finally, as in conventional GAN training, The discriminator will be updated according to the gradient

$$\nabla_\phi \frac{1}{m}\sum_{i=1}^{m}\left[\log D(w_i) + \log(1 - D(G(z_i)))\right] \tag{5-14}$$

Here, in the practical training, the dataset for training does not need to be divided into two parts: the training set and the test set. We only need to calculate the above four indicators during the training when the model converges steadily or the number of training epoch reaches a threshold.

5.4 Experiment

In this section, we describe the procedure of the experiment, including setup, compared methods, results, and discussion.

5.4.1 Setup

We develop a small scale dataset from the ASAP dataset for the experiment, in which we choose 100 essays from each prompt, most of which are relative with high scores (To better train the GANs network, 20 essays with a low score in each prompt are also included). For the 80 high-score essays from each prompt, we invited three experts to label half of these essays as creativity essays. For one essay, if two or three experts think that the essay is creative, then the essay is labeled as a creativity essay. In this way, after manually fine-tuning, we finally get about 40 essays of creativity in each prompt. We calculate the F1 score of the three experts and the final labels' evaluation labels. We get an F1 score of 0.75. Thus, we think the opinions of the three experts are basically consistent.

In the dataset, we omit those essays with a too-short length. Because of that, the essay with a short length may not make a K-fold mask. For a fixed K-fold, the too-long essays are also omitted, because of that, the long length of essay would be masked too many words and do no good for words generating. For solving this issue, we could set a bigger K value for the K-fold mask. The parameter K-fold will be discussed in detail in section 5.4.3.

The overall of the selected essays in each prompt is shown in Table 5-2. All the essays in table 5-2 are preprocessed with the same procedure to the experiment in chapter 4. The partial training hyper-parameters used are shown in table 5-3.

The software environment in the experimental program run is

under Ubuntu 18.04, Python 3.6, TensorFlow-gpu 1.15, and hardware is CPU: Intel(R) Xeon(R) L5640 @2.27GHz 2.26GHz; RAM: 16G; HDD:100G; GPU:GTX1080i.

Table 5-2. Statistics of selected essays from ASAP dataset.

#	Essay #	Avg Length	Scores
1	100	300	4-8
2	100	300	1-3
3	100	150	1-3
4	100	150	1-3
5	100	150	1-4
6	100	150	1-4
7	100	250	6-20
8	100	300	15-40

Table 5-3. Training hyper-parameters.

	Parameter Name	Parameter Value
K-fold mask	K	5
Discriminator	Window size Convolution Filters	5 20
BiLSTMs	Layers Hidden units Dropout	1 64 0.75
	Batch size Learning rate	100 0.01

5.4.2 Compared methods

Amplayo et al., (2019) proposed a novelty detective method among the papers, in which the authors evaluated the novelty based on the time the paper published and the impact of the paper. Although the concept of Novelty is different from Creativity, to some extent, the 2 concepts have some similarities. The paper compared their proposed method to an Autoencoder model. The paper stated that the majority of work for detecting novelty of papers usually uses Autoencoders and neural networks. Refer to this paper. The Autoencoder model is also adopted as a compared method to compare to the proposed method in this chapter. Here the proposed method is a kind of neural network consists of CNN and RNN. For wider comparison, in addition, we add the sole attention mechanisms as the compared method to the proposed method. The model we adopt is the Transformer Encoder (Vaswani et al., 2017), a total attention mechanisms without RNN or CNN, which got a state of the art performance in translation is a competitive model for comparison.

• **Autoencoder-based method (AEBM)**

As shown in section 2.3, Autoencoders encode the input x into encodings and decode an output x' such that x and x' as similar as possible. An Autoencoder is used for creativity essay mining in the following way. First, we train the Autoencoder using an unchanged essay x_e. After training, the masked essay x_g is inputted into the Autoencoder, and the output is represented as the

generated essays x_g'. We calculate the distance $d(x_e, x_g')$ of x_e and x_g', which is calculated as the root of the sum of the squared difference of x_e and x_g', the equation is shown in equation (5-8). The creativity rating of the essay depends on the distance. If x_e and x_g' are nearly identical, thus the essay is not creative. Otherwise, the masked essay may lose some creative information and hence is considered a creativity essay.

• **Attention based method**

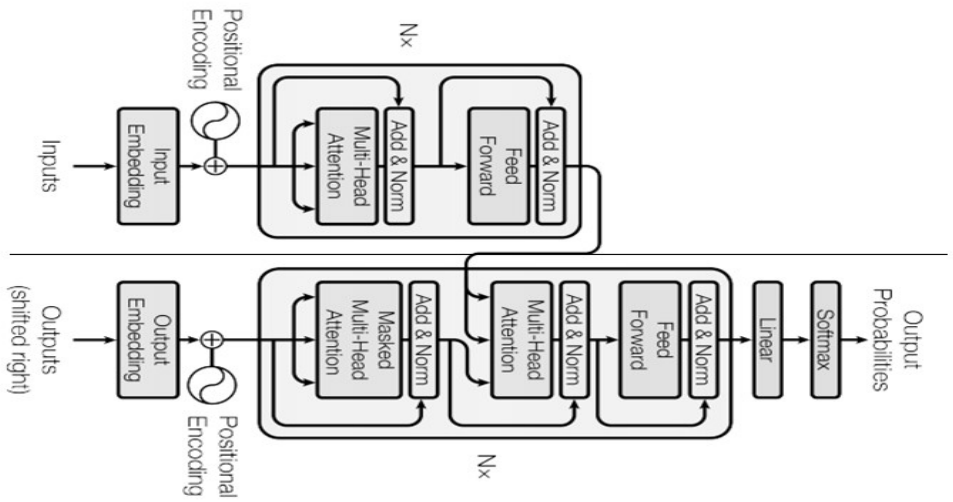

Figure 5-6. Transformer (Vaswani et al., 2017, "Attention is all you need")

The majority of sequence transduction models are primarily based on complicated recurrent or convolutional neural networks that encompass an encoder and a decoder. The best performing models usually additionally join the encoder and decoder via an attention mechanism. Vaswani et al. (2017) proposed new simple network architecture, the Transformer, primarily based entirely on

attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks exhibit these models to be most advantageous in first-class while being more parallelizable and requiring significantly much less time to train. The model is shown in figure 5-7.

The Transformer model mainly consists of an Encoder (the above part of figure 5-6) and a Decoder, in which the model structure is the stack of attention mechanism and fully connected layers.

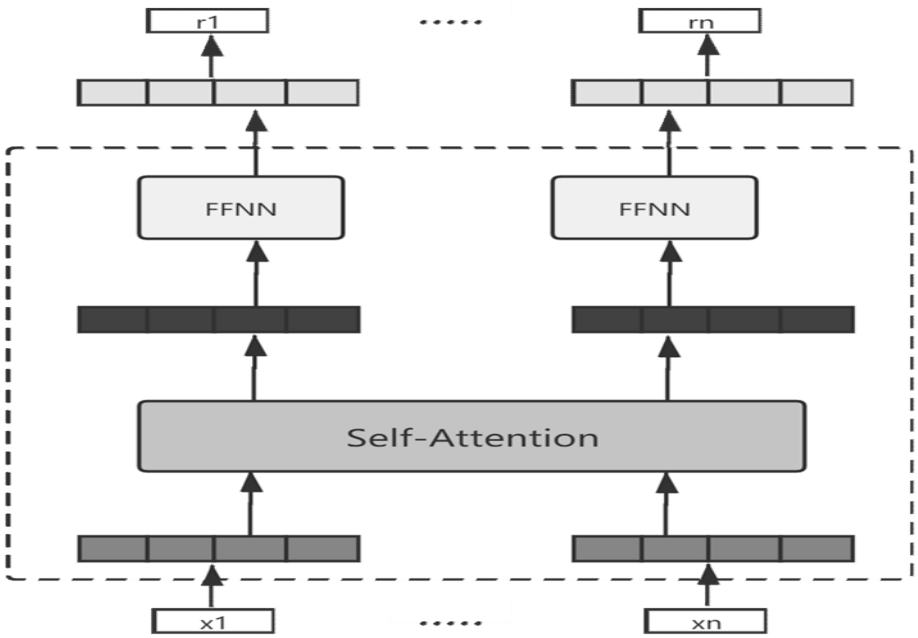

Figure 5-7. Transformer-Encoder

Here, we only use Encoder to represent the text vector. First, we use essays to train Encoder, then input the masked essay to represent and them as the generated essay vectors. The output of the Encoder has the same shape with input, for instance, as shown

in figure 5-7, the input is $x = [x_1, ..., x_n]$ and the output is $r = [r_1, ..., r_n]$. The same to Autoencoders, we calculate the distance $d(x,r)$ of x and r. The creativity rating of the essay depends on the distance. If x and r are nearly identical, thus the essay is not creative. Otherwise, the masked essay may lose some creative information and hence is considered creative.

5.4.3 Result and discussion

As mentioned in section evaluation. The evaluation of creativity essay is highly subjective. Currently, as far as we know, there is no literature study on how to measure creativity essay. In this chapter, we use both quantitative and qualitative methods to the analysis and evaluate creativity essays. Quantitative analysis based on the distance data and some indicators we defined. Qualitative analysis will discuss the semantic content of different essays under various distance data.

Here, we define 4 indicators for evaluation. For an essay, after K-fold mask, we have K generated essays. Calculating the distance between the essay vector (v) and each generated essay vector $(v_i^{'})$. We get K distance values $\left\{d(v,v_1^{'}),\cdots,d(v,v_K^{'})\right\}$. considering the indicators:

$$Avg = \frac{1}{K}\sum_{i=1}^{K} d(v,v_i^{'}) \tag{5-17}$$

$$Max = \max\left\{d(v,v_1^{'}),\cdots,d(v,v_K^{'})\right\} \tag{5-15}$$

$$Min = \min\left\{d(v,v_1^{'}),\cdots,d(v,v_K^{'})\right\} \tag{5-16}$$

$$Span = \max - \min \tag{5-18}$$

In equation (5-15) to (5-18), Avg means the average distance among the essay and its masked essays. The average distance reflects 2 facts: for the entire dataset, the less Avg value indicates that the essays generated by the model on the dataset are most similar to the original essays, the less Avg the better model's learning performance. For a single essay and its masked essays, the greater the Avg value indicates the essay generated by the model is the most dissimilar to the original essay, which means the essay may be more creative so that the model is difficult to generate an essay that is similar to the original one. For the entire dataset, the greater average Avg means average creativity of the dataset is greater. Thus, the greater Avg the better for measuring the creativity essay.

Max means the masked positions in the essay are hard to generate where are maybe the most creative positions. Min means the masked positions in the essay are essay to generate where are maybe the least creative positions. While $Span$ reflects the various degree of language changing of the essay.

Now, we discuss about the parameter K-fold. As we mentioned in section 5.1 that we train a smart enough generator to generate an essay, which is partly masked. We assume that it should be difficult to fill the masked part for a creativity essay, while is relatively easy to fill a common essay. So, the parameter K-fold is very critical for the generator to generate the essay. We use different K values for creative essay mining and apply the K that is the most consistent with

the experimental results and expert results as the optimal parameter for the experiment. Here, the candidate value of K is set to {2, 5, 10, 15, 20}. We compare the experimental results and expert results under the three methods, and calculate the F1 score between them. we get the table 5-4.

Table 5-4. Average F1 score under proposed method with different K value.

#\K	2	5	10	15	20
1	0.53	0.71	0.61	0.64	0.68
2	0.56	0.65	0.56	0.56	0.56
3	0.72	0.79	0.64	0.68	0.6
4	0.62	0.62	0.61	0.57	0.54
5	0.65	0.76	0.73	0.68	0.67
6	0.53	0.75	0.64	0.62	0.55
7	0.52	0.66	0.56	0.65	0.63
8	0.56	0.68	0.6	0.54	0.56
AVG	0.586	0.702	0.619	0.617	0.598

We conducted a T-test for table 5-4 on each two K values, and set the hypothesis that H0: The two k values have no significant difference; H1: The two K values have a significant difference. The p-value is set as 0.05. We use Scipy Python package to calculate the p-value of each two K values and get table 5-5.

Combine table 5-4 and table 5-5, we see that the F1 score of K=5 is the highest and it is significant different from others. Besides K=5, each two of the others has no significant difference. Especially,

K=10 and K=15 almost have the same distribution. Intuitively, we explain that, for K=10, only 10% words are masked, and 90% remain unchanged, which means the masked essay and original one are at least 90% the same. After training, theses two essays are even more similar, and the distance between the two essays are so close that it is difficult to distinguish the creativity among them. So, we can see that the F1 scores under k>=10 have no significant difference, and their value is lower than K=5. While K=2, we explain that to many of the words are masked, and the generating effectiveness of the generator is not good. Besides, referring to the BERT language model (Devlin et al., 2018), in which the authors masked 15% of the words for training. Therefore, combining the data in Table 5-4 and BERT model, K=5 is a good candidate parameter for the experiment.

Table 5-5. p-value under different combination of K values (p-value=0.05).

K	2	5	10	15	20
2	\	0.0018	0.1762	0.2495	0.7017
5	0.0018	\	0.0011	0.0007	0.0025
10	0.1762	0.0011	\	0.9473	0.3876
15	0.2495	0.0007	0.9473	\	0.2390
20	0.7017	0.0025	0.3876	0.2390	\

On the other hand, we try to employ the parameter K={2, 5, 10, 15, 20} to AEBM and ABM methods to find an optimal K for training. But we encounter some issues. For AEBM, we find that when K>=10 the distance be calculated between essay and masked almost goes to

NONE, which shows that AEBM is difficult to detect the difference between almost the same essays. For ABM, there also is no significant difference. Lastly, by K-fold mask, the data generated for training will increase K times. The bigger K, the more data generated for training that is quite time-consuming. In summary, we choose K=5 as the experimental parameter.

Now, we use K=5 for the next experiment. To compare the performance of three methods, we list the four indicators' experimental results of the 8 prompts under the proposed method and compared methods.

Table 5-6. Average indicators of each prompt under Autoencoders.

#	1	2	3	4	5	6	7	8
Max	0.9215	0.9939	1	0.9558	0.8967	0.9714	0.9412	0.903
Min	0.5941	0.648	0.6825	0.6254	0.5881	0.6244	0.6371	0.5824
Span	0.3271	0.3457	0.3173	0.3301	0.3084	0.3466	0.3038	0.3202
Avg	0.754	0.8119	0.8363	0.7975	0.7372	0.79	0.7837	0.7395

Table 5-7. Average indicators of each prompt under Attention.

#	1	2	3	4	5	6	7	8
Max	0.0138	0.0139	0.0148	0.0145	0.0139	0.0148	0.0153	0.0143
Min	0.0134	0.0136	0.0144	0.0142	0.0136	0.0145	0.015	0.014
Span	9E-05	9E-05	0.0002	9E-05	9E-05	0	0.0002	9E-05
Avg	0.0136	0.0137	0.0146	0.0143	0.0134	0.0147	0.0152	0.0141

For a more intuitive comparison, Table 5-6 - 5-8 show the distance of the four indicators that are normalized in [0,1]. We see that the average Avg under Autoencoders is the biggest, the value under the proposed method is the smallest, and the Attention is the middle one, which means the learning performance of the proposed model is the best, Attention is second, and Autoencoders is the worst. However, from the perspective of the time complexity in the experiment, it is just the opposite result. the proposed method is the most time-consuming, Attention is in a balance of time and performance, Autoencoders is the both worst. In order to better explain the model performance, we demonstrate the distance convergence graphs of the 8 prompts under the three methods in detail, as shown in Figure 5-8.

Table 5-8. Average indicators of each prompt under proposed method.

#	1	2	3	4	5	6	7	8
Max	0.0047	0.0052	0.006	0.0065	0.0054	0.0057	0.0062	0.0059
Min	0.0042	0.0046	0.0053	0.0058	0.0048	0.0051	0.0055	0.0051
Span	0.0004	0.0004	0.0005	0.0006	0.0003	0.0003	0.0005	0.0005
Avg	0.0045	0.0049	0.0057	0.0063	0.0052	0.0054	0.0059	0.0055

Also, from table 5-6 - 5-8, we rank all the prompts based on the average Avg value under three methods, from big to small, we have table 5-9. From the table, we think this has something to do with the writing topic and requirements of each prompt. The

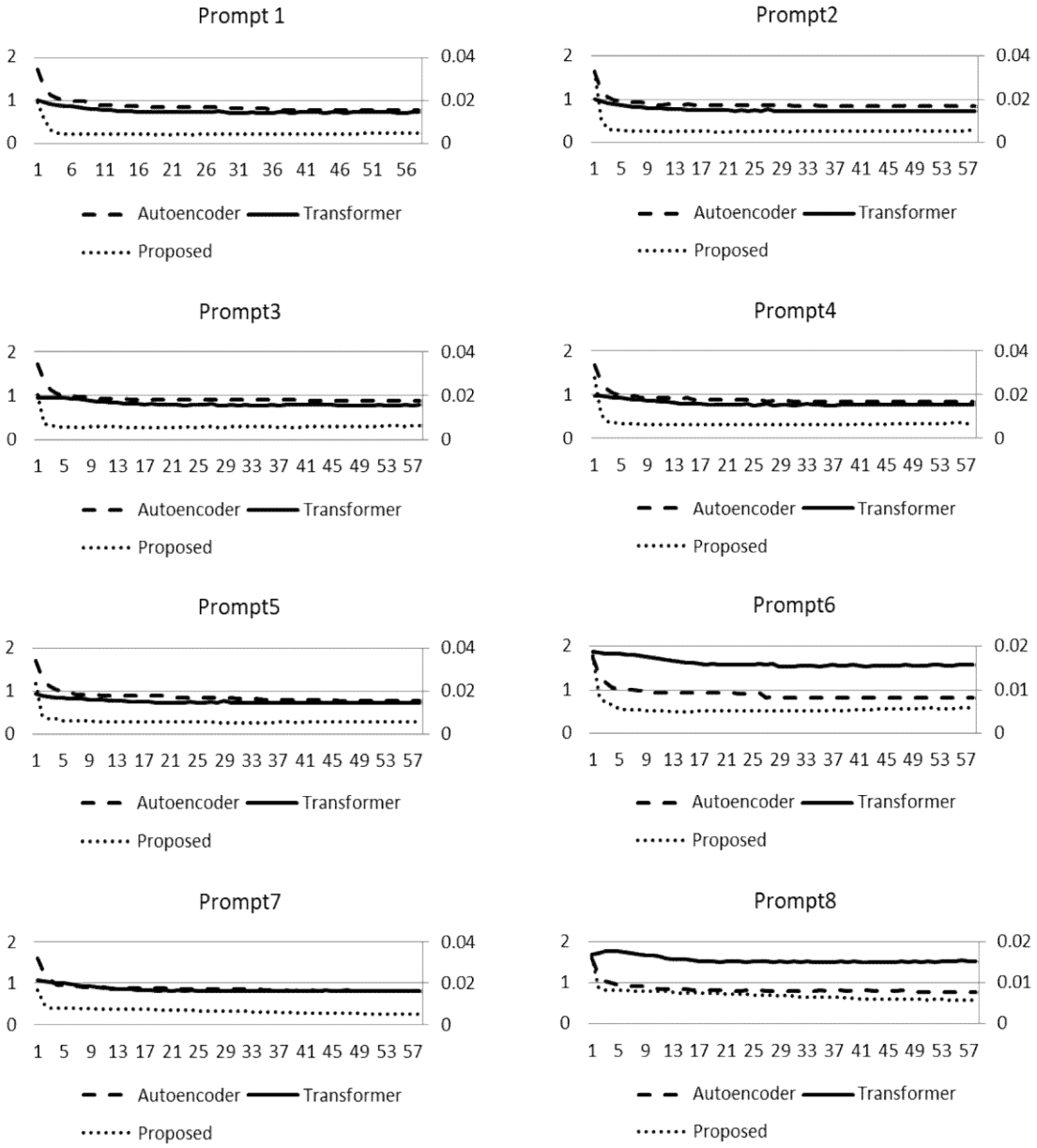

Figure 5-8. Distance convergence of the 8 prompts under the three methods. Note that Autoencoder is under the left scale, Attention and Proposed model are under the right scale.

writing topic and requirements in the prompt are as follows:

Prompt 1: "Write a letter to your local newspaper in which you state your opinion on the effects computers have on people. Persuade the readers to agree with you."

Prompt 2: "Write a persuasive essay to a newspaper reflecting your vies on censorship in libraries. "

Prompt 3: "Write a response that explains how the features of the setting affect the cyclist. In your response, include examples from the essay that support your conclusion."

Prompt 4: "Write a response that explains why the author concludes the story with this paragraph. In your response, include details and examples from
the story that support your ideas."

Prompt 5: "Describe the mood created by the author in the memoir. Support your answer with relevant and specific information from the memoir."

Prompt 6: "Based on the excerpt, describe the obstacles the builders of the Empire State Building faced in attempting to allow dirigibles to dock there.
Support your answer with relevant and specific information from the excerpt."

Prompt 7: " write a story about a time when you were patient OR write a story about a time when someone you know was patient OR write a story in your own way about patience."

Prompt 8: "Tell a true story in which laughter was one element or part."

We think that under different themes and writing requirements, the

creativity of the essays written is different. The more common and clear the topic requirements, the more common essay will be. The more specific and the creative topic is, the more creative essay will be. Prompt 1 is the lowest, while prompt 7 is the highest. Prompt 1 demands students to write an opinion on the computer. Most students write two aspects, positive and negative, which are really common opinion. While prompt 7 is a relatively open and characteristic theme that demands students to write about someone you know was patient or wrote a story in your own way about patience. Obviously, whether it is someone you know was patient or a story in your own way about patience is a very unclear subject. The content that students can write is very different.

We believe that this reason causes the average indicators of the dataset to be relatively high.

From table 5-9, we found that the average distance ranking of the three methods is consistent overall. For the Attention and Proposed model, the value of the Person and Spearman correlation coefficient is around 0.65. Besides prompts 4 and 6, the rest of the prompts almost the same. As to Autoencoder, besides prompt 2 and 7, it is consistent with the other to models overall. The consistency of the three models on the indicators further shows that it is reasonable to use distance to measure the creativity of the essay. It also proves that paper (Amplayo et al., 2019) adopts the distance as the measure for the novelty of the paper.

To further compare the performance of the three models in the evaluation of creativity essays, The results of the three methods

compared to the human rating are shown in table 5-10.

Table 5-9. The ranked the prompts based on Avg.

Models	Prompts ranking							
Autoencoder	3	2	4	6	7	1	8	5
Attention	7	6	3	4	8	5	2	1
Proposed model	4	7	3	8	6	5	2	1

Also, we conducted a T-test of the F1 score in table 5-8 on the three methods, and set the hypothesis that H0: The two methods have no significant difference; H1: The two methods have a significant difference. The p-value is set as 0.05. We put the hypothesis under the proposed method and the AEBM, the proposed method and the ABM, the AEBM and the ABM, respectively, to make T-tests. We use the Scipy Python package to calculate the p-value and get the p-values as 0.00012, 0.00032, 0.12, respectively. The results show that H0 is rejected under the proposed method and the AEBM, the proposed method and the ABM; H0 is accepted under the AEBM and the ABM. That means the proposed method is significantly different from the AEBM and the ABM, and there is no significant difference between the AEBM and the ABM. Also, it means that the performance of the proposed model is the best, and it is much better than the other two methods. From Table 5, we see that the ABM is a bit better than AEBM; however, they are no significant difference. Table 5-8 shows that the performance of the proposed model is the best, and it is much better than the other two methods. Attention is a bit better than Autoencoder. However, they are both around the average

accuracy. The performance of the three means is also consistent with table 5-4, 5-4, 5-6. We get the following conclusions based on the experimental results.

(1) The three models' results are quite different, and Atuoencoder and Attention are much worse than the proposed model. We think that the word order between words and sentences plays a critical role in the formation of creativity evaluation features. Here, we employ LSTM to generate sentences. When GANs are used to adversarial generate training and use LSTM to generate feature vectors, the model has learned some sequence information among the essays. On the contrary, Autoencoder does not use the sequence information, or the sequence information learned is rarely. The transformer is based on a large-scale data set and has achieved the state of the art results in machine translation. We only apply the Encoder to the feature representation of the essay. This is a self-attention mechanism, which takes into account the relationship between words and sentences. When a word is masked, the learning ability of the model will be weakened rapidly, especially the data set is not extensive.

(2) The performance of the same model on different prompts is quite different. We think that the creativity essay evaluation is so subjective that there are too many uncertain factors. There are also some inconsistencies in the analysis given by experts. All of these make the experimental results vary significantly on different prompts. We look forward to a more massive data set evaluated by authoritative experts for creativity essay mining.

(3) According to the proposed model, we find that some of the

essays with higher distance are usually the creativity essay, which indicates that the recognition accuracy of the creative essay is higher than the essay with a lower distance. This makes us think that the model is a good choice as a creative essay recommendation method. Especially under the premise of AES, we do creativity essay mining on those high-score essay data set, and then recommend the identified essay as the creative essay to the teacher for confirmation, or share it with other students to learn. In such a way, AES will be more intelligent and user-friendly. Therefore, the proposed model is available and acceptable. Based on this analysis, a few essays will be demonstrated for discussing below.

Table 5-10. F1 score under three methods with K=5.(P: Precision, R: Recall)

#	AEBM			ABM			Proposed		
	P	R	F1	P	R	F1	P	R	F1
1	0.5385	0.5122	0.525	0.575	0.561	0.5679	0.775	0.7045	0.7162
2	0.4762	0.4651	0.4706	0.575	0.5349	0.5542	0.65	0.65	0.6455
3	0.5	0.5	0.5	0.625	0.625	0.625	0.777	0.80	0.7916
4	0.550	0.5366	0.5432	0.561	0.561	0.561	0.6060	0.625	0.6211
5	0.5641	0.55	0.557	0.525	0.525	0.525	0.7631	0.7631	0.7631
6	0.5641	0.55	0.557	0.5897	0.575	0.5823	0.6756	0.7575	0.7182
7	0.5897	0.5111	0.5476	0.5385	0.4667	0.5	0.6060	0.6896	0.6608
8	0.4	0.4324	0.4156	0.6	0.6486	0.6234	0.6944	0.6756	0.6805
AVG	0.5223	0.5071	0.5145	0.5736	0.5621	0.5673	0.6913	0.6975	0.7025

So far, experimental results show that the assumption on how to recognize creative essays, to a certain extent, is reasonable. We further discuss the proposed model in recognition of creativity essay by some specific essay cases. Fore each prompt, we demonstrate a creative essay and a common essay. Given that the score of essays has an impact on the evaluation of creativity essays (We mentioned earlier that the creativity essay should first be a high score essay). By combining the level of average indicators and the topic prompt, we take prompt 5 for a more detailed analysis, in which we divide the essay's score into three groups: high, medium, and low. We list the above indicators of prompt 5 and the corresponding essays (6 pairs of essays) to the tables and demonstrate the discussion under each pair of essays, including some opinions from the English professors. For other prompts, we list the most creative essays and common essays that with higher score for readers' reference.

In prompt 5, The essays correspond to Table 5-11 are as follows. The judgment from the proposed model is that the first essay is more creative than the second essay in each group.

Table 5-11. Average indicators of prompt 5.

Groups	High		Medium		Low	
	Higher	Lower	Higher	Lower	Higher	Lower
Max	0.0072	0.0049	0.0068	0.0047	0.0051	0.0049
Min	0.0065	0.0042	0.0063	0.0040	0.0043	0.0042
Span	0.0006	0.0006	0.0004	0.0057	0.0005	0.0005
Avg	0.0068	0.0045	0.0066	0.0044	0.0044	0.0047

High group: The higher AVG value essay in prompt 5 (ID: 12964)

The aurther talks about home and family which would be a calm, caring, and happy mood. The author created these moods by talking about memories and how her parents moved from cuba to give there child a better life and how they take in family mebers in need until they get back on there feet. She said her parents both shared cooking duties and unwittingly passed on to me their rich culinary skills and a love for cooking that is still me today. Passionate Cuban music filled the air, mixing with the aromas of the kitchen. Which is kinda like a happy mood and " Here, the innocence of childhood the congregation of family and friends, and endless celebrations that encompassed both, formed the backdrop to life in our warm home. Which is definatly a calm and happy mood.

High group: The lower AVG value essay in prompt 5 (ID: 13197)

The mood created by the author In the memoir is a good mood @CAPS1 knows that her parents moved from Cuba for her, to give her a better life. "My young parents created our traditional Cuban home." There trying to give her the life they would have had in Cuba, only better. And @CAPS1 had unselfish family because they moved for her and let her grow up in a good community. And when they say "All these cultures came together in great solidarity it just makes you feel good that segrigation stoped and people can hang out with others, from different places. Family is always first no matter what even if you dont understand, its always first which is why @CAPS1 creating a

happy and good mood in the people reading this because of family.

Comments: Through plain language, the essay 12964 describes the importance of warm family for growth. Although the words used in the essay are plain, the sentence structure is clear, and the meaning expressed is powerful. In essay 12964, the author has more opinions about the material that the happy mood is described with specific examples. For instance, the sentence "Passionate Cuban music filled the air, mixing with the aromas of the kitchen. Which is kinda like a happy mood ". Also, the essay has a better sentence structure than essay 13197, in which it has more complex sentences with a progressive relationship, and with changeable words which is full of change. Essay 13197 is relatively common, and the narrative is relatively bland. Words in essay 12964, such as "Passionate" and "endless", "encompassed", etc. are full of changeable. The four indicators' value in essay 12964 is much higher than the value in essay 13197. It indicates that the essay 12964 exists more positions with creative words. Therefore the machine's judgment that essay 12964 is more creative than essay 13197 is reasonable.

Medium group: The higher AVG value essay in prompt 5 (ID: 13318)

In the memoir, @PERSON1, the author creates a mood that inspires us to try our hardest and that mood is perseverance. The mood perseverance is expressed in the memoir in many ways. One way it is a stress is when @ORGANIZATION1 and @PERSON1 moved to the United States and lived in a one bedroom apartment. They finally saved up enough money to be able to move into a @NUM1

bedroom apartment. In this neighborhood the family raise as much money as possible to buy food for themselves and to help the people that were in need of anything from, water, shelter, and clothing. The reason they move from Cuba was to begin a better life in the United States which they never gave up until the day that they finally had enough money to live here.

Medium group: The lower AVG value essay in prompt 5 (ID: 13611)

The mood that the author in the memoir created was good by his love, his blood relative, and also by courage. To start with her love is like part of her life. He love his parents. For example In the story say's "I will always be grateful to my parents for their love and sacrifice." @CAPS1 this mean she loves he parent for every little single thing they did. Also he learn how to love people. As will as her blood relative meaning that he @CAPS2 got to do with nothing of this. She is from cuba, and came to the united state on 1956. Also born to this a simple house. His that build a traditional home. This is how the author describe the mood, by his love, his blood relative, and also by courage.

Comments: The essay 13318 describes the "hardest" and "perseverance" truthfully. This seems to be a process of making money. The author said that the protagonists first lived in a one-bedroom apartment, and then through efforts, move into a @NUM1 bedroom apartment. Such a description of the essay lacks a spiritual level, which makes the article very common. In essay 13611, the first word that comes into mind is "love", indicating that the author

has made a conclusion. The important role of "love" is also described later. The essay 13611 has a higher mood than essay 13318. Besides, essay 13611 has a clear, logical structure compared to the essay 13318, and also it has more abundant words and good content coherence. However, for machine learning, essay 13318 may be recognized some changeable words so that the machine gives the result that essay 13318 is more creative. In particular, for this pair of essay, we asked five experts for further analysis, three experts thought the essay 13611 is more creative than essay 13318. Therefore, we think that the judgment of the model is unacceptable. The essay 13611 is a bit more creative than essay 13318.

Low group: The higher AVG value essay score in prompt 5 (ID: 12580)

In the memoir there were actually two main moods. The first mood and most prominent mood was joy and happiness. Because through the memoir the reader learns about the writers family and how grateful the writer was for her family. An example of this is when the writer wrote: "I will always be grateful to my parents for their love and sacrifice." But joy is not the only mood in the memoir, because though the memoir is mostly about positive things, the reader still has to remember that this is a memoir. Meaning this was only written because Narciso Rodriguez passed away. Therefor sadness is also a mood in this memoir. In the end, the joyful mood sort of covers the depressing one, but to really get a sense of where the writer is coming from the reader most consider both these moods.

Low group: The lower AVG value essay in prompt 5 (ID: 12789)

The mood in this story @PERSON1. Moving can be hard for some people others it is an @CAPS1. Most people hate to leave their @CAPS2, work, and friends behind. Other people it's a new opportunities to meet new people a better job, or go to a better @CAPS2 for your education. I understand it is hard to move from cuba to the united States. Thats across the world not a town over or a street down. But he made a friend and his mom and dad love him very much. They always have food on the table so at lest they dont starve. It was a step down moving to united States going from a good jobs to poopy. The mood in this story is very loving between the family and all there friends.

Comments: Although the score of essay 12580 is low, it is well organized, with a clear point of view and logic. It describes two kinds of moods and also includes thinking about the author. In essay 12789, the author discusses the two kinds of people's understanding of moods and then describes his understanding, which is also a bit logical. However, in experts' opinion, essay 12580 is a bit more creative than the other. The computer results show that the relevant indicators of these two essays are quite a closeness that two essays are very similar. We think that the judgment of the model is acceptable.

Here, we compare six essays on prompt 5 from low scores to high scores. Through the analysis above, we found that the proposed model is feasible for creative essay mining . Especially for high-scoring essays, the effect is the best, middle-scoring essays are somewhat ambiguous, and low-scoring essays are difficult to distinguish. This is also in line with the actual situation that the low-scoring essay is

already poorly expressed, and the creativity is far to meet. In essence, this way of finding creativity essays is how we find some strange essays or outliers through distance, so for low-scoring essays, this kind of outlier is likely to be a wrong expression, not creative, because that the content expression is incomplete. However, for high-scoring essays, the phrase usually is well organized. These outliers should be good essays or creative essays.

Table 5-12. Most creative and most common essays ID in other prompts.

	1	2	3	4	6	7	8
Creativity Essay	563	3463	6335	10473	15524	19548	20775
Common Essay	639	3359	6121	9011	15347	17878	21596

Also, we list the most creative essays and common essays in other prompts for readers' reference. Table 5-12 shows the most creative and most common essays in each prompt.

(**Prompt 1: #563**, Max:0.0063, Min:0.0057, Span:0.0007, Avg:0.0061) *Dear Newspaper, I think Computers are good for many things and are useful for anyhting. I believe this because every year new and exciting @CAPS1 and websites are created and make life easier. For example, "@CAPS2" is a way to communicate with friends and family. For me I have family out-of-state, and I am happy when I go on @CAPS2 and speak to them. Or I can also talk to friends after a long day of school. Another example is "youtube." On youtube They have how to videos that show you step-by-step instructions on anything, even on news and music entertrainment. Other @CAPS1 like*

word, help you set up writing essays and @CAPS4 stories. On powerpoint it lets you make slide show videos for school projects to presentations. Even excel let's you make graphs and any char t imaginable. And let's not forget @ORGANIZATION1. @ORGANIZATION1 helps you when you need info on a person, place, or thing. And by useing @ORGANIZATION1 it helps you when you need information for other @CAPS1 like word and more. See, computers help everyday people wiht everyday life.

(**Prompt 1: #639**, Max:0.0037, Min:0.0035, Span:0.0002, Avg:0.0036) *Dear @CAPS1, @CAPS2 has come to my attention that people are spending to much time on computers, and I agree because people spend so much time on computers that @CAPS2 gets addicting, people don't get enough exersize, and eyesite gots worse. So please keep reading my letter and I will tell you the side effects of the computer. Most of all @CAPS2 gets really adicting for a lot of people that they dont spend time with thie familys or sibleings and some kid get so adicted that they come from school and start useing the computer all day and dont study for a test or do home work. Now people spend so much time on the computer that they dont exersize or play sports with thier friends and just sit all day and get lazy and fat just because of the computer. We just need the computer just to chek @CAPS3 and facebook like I do but I just check my facebook and the go play basket ball. One of the biggist problem that I think is when you sit on the computer to long and get a head ache and ruins your ey site. So thank you for reading my letter and hope you agree with me.*

Comments: These two essays are with opposite views. Essay 563 with a positive view while essay 639 with a negative opinion. Essay 563 shows that the computer is beneficial through many examples. The author gave examples of entertainment (Youtube), study, and work (PowerPoint, Excel), which brought convenience in entertainment, learning or work. Essay 639 believes that people spend too much time on the computer and thus has less exercise time. The description of essay 639 is relatively bland, while the essay has many examples. We think that the description of the abundant examples in essay 563 makes the value of four indicators higher than essay 639. The essay 563 is more creative than essay 639.

(**Prompt 2: #3463**, Max:0.0083, Min:0.0074, Span:0.0008, Avg:0.0078) *I think they should'nt remove material from libraries if it's found offensive. People find different things offensive that others do not. If you started takeing these things out, thers a good possibility that you'll loose everything, books and your people. One example I say this is because, some people are offensive bout movies that have violent scense in them. Others like the violence in those shows. The librarie might be the only way someone can get ahold of something they like. Another example is, magazines can contain material you don't want your kids round or seeing. That same magazine could help a student with a assignment at school. Kids might need the material that is offensive, it could help them through life. In conclusion I do not think we should take movies and books out if someone finds it offnesive. Some of the offensive material could be used to help students in there education. Other offensive material some adults enjoy*

reading or watching. Just because one person finds something offensive does not mean the next person that comes in will.

(**Prompt 2: #3359**, Max:0.0048, Min:0.0041, Span:0.0007, Avg:0.0044) *I think that if u feel like they are sayin bad thing that you should not read the books or magazines and dont lisen to the music and dont look at the moves. If you feel offensive about it just dont look at it because a porson might not find it offensive.It is port of life to look at something offensive so why do it now. I think that if they do that they might just take some @CAPS1.Vs shows off the air becuas there are a lot off shows out there that are offensive and some do not like that but they dont take it off the air so why take books music, movies, and magazines off the shelves. i dont believe that they should do that at all becuas if i dont like it i dont look at it so i dont get mad at something somebody say bout my believe. A porson might not like the same music u like i might not like the same music he or she likes but i dont wont to take it off the shelves they should just stop lookin and lisning to all this stuff a porson says about there believe.*

Comments: The essay 3463 is evident in logic, with appropriate examples, closely related to the topic, smooth expression of sentences, and bright ideas. The expression of 3359 is more colloquial, compared with 3463, the logic is not sound, the sentence structure is simple, and the overall persuasion is not as good as 3463. The essay 3463 is more creative than the essay 3359.

(**Prompt 3: #6335**, Max:0.0076, Min:0.0071, Span:0.0004, Avg:0.0073) *In the story "Do Not Exceed Posted Speed Limit"many*

features of the setting challenge the cyclists. The cyclist accepts information from foreign old people which made his first mistake. After realizing this he,"I had been hitting my water bottles pretty regularly, and i was traveling through the high deserts of California,in June"the lack of water plus the heat in California, in june would have turned to heatstroke for most people.Even further into the journey,even with all those things against him it states,"flat rode was replaced by short,rolling hills,"@CAPS1 not only did he have excessive heat,and no water,but @CAPS1 he was handed troublesome rodes too.Although the cyclist must have been near to complete exhaustion,he continued through.Fighting all adds,and elements,many features in the setting of this story affected the cyclist,but he overcame them honorably.

(**Prompt 3: #6121**, Max:0.0053, Min:0.0046, Span:0.0006, Avg:0.0049) *The features of the setting greatly effected the cyclist. He was riding along on a route he had little confidence would end up anywhere. That being the first time ever being on that rode and having only not of date knowledge about it made the cyclist rework. The temperature was very hot, where was little shade, the sun was beating down on him. The route was very discouraging .all the cyclist wanted wanted was water and the route kept touenting him with false hope. At the first town he noticed a water pump, but all he could get of it was sludge and water that tasted like battery acid. Next he came to a deserted old building that approved to be an old bottling factory for Welch's grape juice. The cyclist would appear to be closing in on something to drink, but be left with nothing.*

Comments: The requirements demand that "Write a response that

explains how the features of the setting affect the cyclist. In your response, include examples from the essay that support your conclusion." The essay 6335 discussed with specific examples, the content is clearly expressed, and it meets the writing requirements. Compared to essay 6121, essay 6335's wording is rich and varied. The essay 6121 lack of specific examples. Overall, the essay 6335 is more creative than essay 6121.

(**Prompt 4: #10473**, Max:0.007, Min:0.0063, Span:0.0007, Avg:0.0066) *In the end of "Winter Hibiscus" by Minfong Ho, the narrator describes her desire to take and this time pass the driver's test she failed earlier in the day, she "vowed silently to herself" that she would succeed at something she had failed at in the past. This is an important concluding message, because it shows that the author will overcome obstacles in her future. Most of the story is about the sadness that the narrator experiences because of longing for her homeland, Saeng misses her grandmother and the plants of Vietnam, and she even buys a hibiscus plant to remind her of her homeland. She is also not enjoying @LOCATION2, because she has already experienced failure at her new home. However, the end of the story provides contrast do this, for Saeng vows to succeed in the future.*

(**Prompt 4: #9011**, Max:0.0066, Min:0.006, Span:0.0005, Avg:0.0063) *In the story '' Winter Hibiscus,'' the author concludes the story with this paragraph for many reasons. This paragraph shows that Saeng will not give up, she will wait for next time to come around to take the test. It also means she will be well prepared and will have a list of confidence in passing this test. It gives the*

reader an idea that Saeng will not give up and she will complete the test. Her mother will be proud of her for it and it shows Saeng will never give up. Just like when she said "let's plant it, right now,'' T his example shows that even though Saeng had failed the the test and disapointed her mother,she will never give up and she still has hope in the plant just like how her mother has hope for Saeng.

Comments: The essay 10473's expression is more artistic than the other, and described the expectations for the future. The author, combined with the sentence "she vowed silently to herself", started his thinking. The author's thinking changes from low to high, describes the power of belief support. While essay 9011 just focus on the "test", and lack of expectations for future life. The essay 10473 is more creative than the essay 9011.

(**Prompt 6: #15524**, Max:0.0087, Min:0.0082, Span:0.0004, Avg:0.0083) *The builders faced many problems in their attempts to allow a dirigible to land at the Empire State Building. One of the main reasons was safety, "most dirigibles outside the United States used hydrogen rather than helium, and hydrogen is highly flammable." I would have been to large of a risk to allow such a dirigible over New York. Another problem was the air currents, "The winds on top of the building were constantly shifting due to violent air currents. Even if the dirigible were tethered to the mooring mast, the back of the ship would swivel around and around the mooring mast." A final problem was a law "against airships flying too low over urban areas." This law would forever prohibit any airship to dock with the mast, or even approach the city to attempt to dock.*

(**Prompt 6: #15347**, Max:0.0048, Min:0.0045, Span:0.0003, Avg:0.0047) *While designing and building the mooring mast atop the Empire State building, the engineers seem to have ignored some seemingly useful information. As with any vehicle in the air or at sea, wind is either your enemy, or your friend. The engineers should have taken into greater consideration that changing wind speeds and direction 1,250 feet in the air are huge threats. Your dirigible will have a hard time getting close enough to the mast to moor, never mind to stay steady enough for passengers to exit, and board safely. Also if the law itsself prevented you from floating your blimp at such low altitudes, then its game over. Why even bother if its illegal? Although it seems like a futuristic sci-fi, and not to mention downright cool way to land a blimp, to many things prevented the idea from prevailing.*

Comments: The essay 11524 described four obstacles the builders of the Empire State Building faced, each obstacle explains the reason in detail. The author's description is comprehensive and organized well. The essay 15347 only proposed two obstacles that is a part of essay 11524. Essay 11524 is more suitable for the theme. Therefore, the essay 11524 is more creative than essay 15347.

(**Prompt 7: #19548**, Max:0.0097, Min:0.0087, Span:0.001, Avg:0.0093) *On a fine @DATE1 day I was heading toward the @CAPS1 office. I wasn't pleased. I didn't want the son to get there. But it did. When we got inside, my mom went to the checkin lane I just sat down. When my mom got done she sat right next to me and we waited. And we waited and waited. We waited for about @NUM1*

minutes until the @CAPS1 came and called us in. Right when we get into his office a nurse came in and checked my temperature, heat beat, ex. When she was done and left we waited and waited. We sat there for @NUM2 minutes before the @CAPS1 came in. when he was done we checked out and left and @CAPS2 don't have to go there again.

(**Prompt 7: #17878**, Max:0.0076, Min:0.0064, Span:0.0012, Avg:0.0071) *Patience. Being patient means that you needs to make and dor one' thing constantly to make one????understand. That is patience. My mom is patient. When I dart understand a problem, she will explain @CAPS1 to me, never in rush. She don't yell a t me when I don't understand, unlike my dad, who is very impatient. He would yell, and possiblely cursed at me in chinease. That happens nearly every time I don't understand something. But my mom will contune explain to me until I understand what @CAPS1 means. Some people @MONTH1 be impatient because of triats or because that he or she has high blood pressure. Patience is something my people have, but @CAPS1 seems like even move people lacks @CAPS1.*

Comments: The essay 19548 describes a story about patience encountered by an author himself. The essay does not mention patience, but the author's mood is described as impatience through several details: the son is sick and needs to be checked in the hospital, and the author's mood is already terrible. It takes a few minutes to enter an office, and again, it takes another a few minutes to enter another office. The essay fully describes the details of patience. It is a creative essay. The essay 17878 describes a story about the author's parents. The essay lacks a detailed description of

patience. Therefore, the essay 19548 is more creative than essay 17878.

(**Prompt 8: #20775**, Max:0.0107, Min:0.0096, Span:0.001, Avg:0.01) *One time when i was spending the night at @ORGANIZATION1's she wanted to sleep oustide in a tent. @CAPS2 was @DATE1 and we always tryed seeing each other anytime possible because she had to go back to boarding school alot. So her mom set up the tent for us on the deck, and after we ate dinner, we grabbed our things and went ouside. She was on the phone with her boyfriend and i was on the phone with mine, and both of them were best friends. @CAPS2 was like two best friends dating two bestfriends, and we were closer then ever. My boyfriend had to get off the phone because he had work in the @TIME1 so @CAPS2 was just @LOCATION1 on the phone. I started saying really funny things and she just started laughing so hard that she snorted and at the time she was still on the phone with her boyfriend. At this point i was just cracking up with laughfter, @ORGANIZATION1 started laughing even harder because she saw me laughing and pee'd her self while she was on the phone with her boyfriend. For some reason she forgot that she was on the phone with him and started laughing even more after that happend, so all of this was going on at around one in the @TIME1 and i was surprised her mom didn't come out and yell at us for being loud. When i looked over and saw that she had pee'd her self i yelled out "@CAPS1 my god, @CAPS3 pee'd your self @ORGANIZATION1!" @CAPS2 was the most funniest thing, i dont think i have ever laughed so hard in my entire life. So her boyfriend finally got her attention*

and said "@CAPS3 should probably go take care of that," and then they got off the phone. She was really embarassed after that, i mean i would be too if that happend to me.

(**Prompt 8: #21596**, Max:0.0068, Min:0.0059, Span:0.0009, Avg:0.0064) *I woke up just like any other day happy yet lacking sleep. As i got out of bed i would have never known that to day would be the funniest day of my life. I got ready for school after getting out of bed. When i got to school every thing seemed like our normal homecoming tell there was a announcement on the intercoms that had told every body out of no where there was a dance tonight. So after school was done me and my friends were going to head over to our house's to get dressed for the dance. After we were all dressed @PERSON1 picked us all up and we headed to the dance looking fly. When we got there every body was looking dressed to dance except one guy, he was wearing corduroy pants with a red tucked in flannel and some brown worn out work boots. We look at him from head to toe and we thought to our self are we in a messed up hillbilly dream? That was just the beginning of what was yet to come. As every body started to get in grove of the beat we soon all started dancing to the music the music was good and every body was having a good time even the kid with the flannel. But just as every thing was going good a song came on that was called cotton eyed @CAPS1 when the flanneled kid heard this song he almost jumped out of his corduroy pants he soon stared kicking and swinging his feet and arms like if they had no bone or joints in them. Every body started to form a circular around the kid and every body was laughing and copying the*

kids movement even us. He didn't rely care he just kept dancing and singing to the song. The funnest thing about this was that the dance was a formal one and yet this kid manged to pull off wearing a flannel, some boots, and a pair ofcorduroy pants this kid was out of his mind in fact we still laugh and talk about it tell this day.

Comments: Both essays describe little stories about laughter. The essay 20775 describes the details of a laughter more, and the plot is more volatile, and there are more surprises in detail. The essay 21596 has more descriptions of the process of things happening, this part is relatively bland, and the description about laughter does not give readers too many surprises. Therefore, the essay 19548 is more creative than essay 17878.

Besides the above examples, we made more comparisons in the experiment and found that the exploratory work we did in the essay mining of creativity, to a certain extent, is consistent with the real situation and has practical significance.

5.5 Summary

In this chapter, We studied creativity essay mining. This work is performed after AES. Based on the assumption that creative essays are relatively difficult to generate, we propose an unsupervised generative adversarial network architecture that contains 4 parts: generator, discriminator, BiLSTM, and distance analyzer. Through distance analysis of essays expressed as vectors to find some characteristic essays. The experiment shows the creative essays from the high-scoring essays are more in line with the actual situation. It

is very meaningful for the machine to find some characteristic essays from a large number of high-scoring essays as creative essays or to provide candidates for experts. We think that this work increases human fun to the boring essay scoring job.

Chapter 6 Conclusion and future work

6.1 Conclusion

In this dissertation, we reviewed the research history of AES, the current research progress on AES in deep learning, introduce the theoretical basics of deep learning, and propose a novel neural network AES model. Based on AES, we further explore the creativity essay mining.

In chapter 1, we introduce the background of deep learning, the potentially huge market demand for AES, and state the main work of AES in recent decades. We discuss the main achievements and deficiencies of the current AES work. We also identified the main research content and objectives of the thesis and listed the main contributions.

In chapter 2, we introduce the theoretical basics of deep learning, which are mainly the mainstream neural network structures in deep learning. We also analyze their possible applications in AES. These models are also the theoretical basis of the full thesis. The various network structures introduced in this chapter are applied to various sections of this dissertation. A fully connected network is used for classification at the end of various network structures in chapter 4 and

chapter 5. Autoencoder and attention mechanisms are selected as the compared method in chapter 5. A convolutional neural network is employed for AES as an optional layer in chapter 4 and a discriminator for GANs in chapter 5. A recurrent neural network is the main tool for this thesis. The generative adversarial network is employed for creativity essay mining. Backpropagation is the basic training method

In chapter 2, we discussed the basic neural network structures in deep learning and analysis their applications in AES. These models are also the basic research of this thesis

In chapter 3, we put forward the idea of self-learning mechanisms, and use the mechanism to help the deep model to learn specific knowledge and external knowledge, to improve the learning ability of the deep model and present a general representation of the mechanism. We consider the syntactic and semantic features, consistency, and coherence features, in which we define a similarity matrix for wide space similarity calculation and the scoring related information. We also think some preprocess technology maybe impact on AES. The self-learning mechanisms make deep learning model has a way to incorporate prior knowledge.

In chapter 4, we represent the rating criteria behind the essay by some samples and take it as a part of the input. Meanwhile, a self-feature mechanism at the LSTM output layer was provided as well. Then, we propose a novel model, a Siamese Bidirectional Long Short-Term Memory Architecture (SBLSTMA), to learn the text semantics and grade essays automatically. Our approach outperforms the baseline by approximately 5%. By decomposing the model, we find

that the model with distance information input is much better than the one without distance information. It means that it is feasible to represent rating criteria from samples. We also hypothesize that distance information derived from the difference between the examples and the mean example benefits all the other supervised learning methods. We will try using this approach in other fields in the coming future to check whether the hypothesis is right or not. Besides, we will also consider applying data augmentation technology to enhance the essay dataset, of which the example is relatively small.

In chapter 5, We studied creativity essay mining. This work is performed after AES. Based on the assumption that creative essays are relatively difficult to generate, we propose an unsupervised generative adversarial network architecture that contains 4 parts: generator, discriminator, BiLSTM, and distance analyzer. Through distance analysis of essays expressed as vectors to find some characteristic essays. The experiment shows the creative essays from the high-scoring essays are more in line with the actual situation. It is very meaningful for the machine to find some characteristic essays from a large number of high-scoring essays as creative essays or to provide candidates for experts. We think that this work increases human fun to the boring essay scoring job.

6.2 Future work

This thesis studied on AES and creativity essay mining. Nevertheless, whether it is AES or creativity essay mining, it is challenging research. We think that there is still a lot of work to start.

How to evaluate the long length articles, etc. The common issue such as the generalization performance of AES. These models still need to be improved, and the average accuracy has room for improvement. The new NLP technology will bring new help to AES, etc. Many applications, such as applying AES to MOOC learning, IELTS test evaluation, studying other specific language's AES, like Chinese-based AES. There is even more work to be done in the discovery of creativity essays, such as carrying out new research approaches on creative essay mining. How to develop a common and effective evaluation of creative essays. Also, large scale creativity dataset development, application of creativity essay mining, and so on are expected.

References

Ahamad, A. (2018). Generating Text through Adversarial Training using Skip-Thought Vectors. arXiv preprint arXiv:1808.08703.

Alikaniotis, D., Yannakoudakis, H., & Rei, M. (2016, August). Automatic Text Scoring Using Neural Networks. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 715-725).

Akmal Haidar, M., Rezagholizadeh, M., Do-Omri, A., & Rashid, A. (2019). Latent Code and Text-based Generative Adversarial Networks for Soft-text Generation. arXiv preprint arXiv:1904.07293.

Amplayo, R. K., Hwang, S. W., & Song, M. (2019, November). Evaluating Research Novelty Detection: Counterfactual Approaches. In Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13) (pp. 124-133).

AP, S. C., Lauly, S., Larochelle, H., Khapra, M., Ravindran, B., Raykar, V. C., & Saha, A. (2014). An autoencoder approach to learning bilingual word representations. In Advances in neural information processing systems (pp. 1853-1861).

Atapattu, T., Falkner, K., & Falkner, N. (2017). A comprehensive text analysis of lecture slides to generate concept maps. Computers & Education, 115, 96-113.

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

Assendorp, J. P. (2016). Project Report: Deep Learning for Text Classification in Digital Journalism.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Balfour, S. P. (2013). Assessing Writing in MOOCs: Automated Essay Scoring and Calibrated Peer Review™. Research & Practice in Assessment, 8, 40-48.

Barrett, C. M. (2015). Automated essay evaluation and the computational paradigm: Machine scoring enters the classroom.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2), 157-166.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8), 1798-1828.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

Bengio, Y., Lee, D. H., Bornschein, J., Mesnard, T., & Lin, Z. (2015). Towards biologically plausible deep learning. arXiv preprint arXiv:1502.04156.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., & Bengio, S. (2015). Generating sentences from a continuous space. arXiv preprint arXiv:1511.06349.

Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096.

Cai, Z., & Vasconcelos, N. (2018). Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6154-6162).

Cambria, E., Schuller, B., Xia, Y., & Havasi, C. (2013). New avenues in opinion mining and sentiment analysis. IEEE Intelligent systems, 28(2), 15-21.

Cao, Z., Long, M., Wang, J., & Jordan, M. I. (2018). Partial transfer learning with selective adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2724-2732).

Chen, H., & He, B. (2013, October). Automated essay scoring by maximizing human-machine agreement. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (pp. 1741-1752).

Chiru, C. G. (2013). Creativity detection in texts. In Proceedings of the 8th International Conference on Internet and Web Applications and Services (ICIW2013) (pp. 174-180).

Cho, K., Courville, A., & Bengio, Y. (2015). Describing multimedia content using attention-based encoder-decoder networks. IEEE Transactions on Multimedia, 17(11), 1875-1886.

Choi, Y., Choi, M., Kim, M., Ha, J. W., Kim, S., & Choo, J. (2018). Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8789-8797).

Ciregan, D., Meier, U., & Schmidhuber, J. (2012, June). Multi-column deep neural networks for image classification. In 2012 IEEE conference on computer vision and pattern recognition (pp. 3642-3649). IEEE.

Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011, June). Flexible, high performance convolutional neural networks for image classification. In Twenty-Second International Joint Conference on Artificial Intelligence.

Clary, R. M., Brzuszek, R. F., & Fulford, C. T. (2011). Measuring creativity: A case study probing rubric effectiveness for evaluation of project-based learning solutions. Creative Education, 2(04), 333.

Cohen, T., & Widdows, D. (2017). Embedding of semantic predications. Journal of biomedical informatics, 68, 150-166.

Collobert, R., & Weston, J. (2008, July). A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning (pp. 160-167).

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011).

Natural language processing (almost) from scratch. Journal of machine learning research, 12(Aug), 2493-2537.

Cui, G., Xu, J., Zeng, W., Lan, Y., Guo, J., & Cheng, X. (2018, September). MQGrad: Reinforcement Learning of Gradient Quantization in Parameter Server. In Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval (pp. 83-90).

Darwish, S. M., & Mohamed, S. K. (2019, March). Automated Essay Evaluation Based on Fusion of Fuzzy Ontology and Latent Semantic Analysis. In International Conference on Advanced Machine Learning Technologies and Applications (pp. 566-575). Springer, Cham.

Dauphin, Y. N., Fan, A., Auli, M., & Grangier, D. (2017, August). Language modeling with gated convolutional networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 933-941). JMLR. org.

Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., & Kaiser, Ł. (2018). Universal transformers. arXiv preprint arXiv:1807.03819.

Dennis, S., Landauer, T., Kintsch, W., & Quesada, J. (2003). Introduction to latent semantic analysis. In Slides from the tutorial given at the 25th Annual Meeting of the Cognitive Science Society, Boston.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

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

Dong, F., Zhang, Y., & Yang, J. (2017, August). 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).

Dos Santos, C., & Gatti, M. (2014, August). Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers (pp. 69-78).

Drolia, S., Rupani, S., Agarwal, P., & Singh, A. (2017). Automated essay rater using natural language processing. International Journal of Computer Applications, 163(10), 44-46.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research, 12(Jul), 2121-2159.

Dundar, A., Jin, J., & Culurciello, E. (2015). Convolutional clustering for unsupervised learning. arXiv preprint arXiv:1511.06241.

Dzmitry B., Kyunghyun C., & Yoshua B., (2014). Neural Machine Translation by Jointly Learning to Align and Translate. arXiv preprint. arXiv:1409.0473

Elbayad, M., Besacier, L., & Verbeek, J. (2018). Pervasive attention: 2d convolutional

neural networks for sequence-to-sequence prediction. arXiv preprint arXiv:1808.03867.

Elizondo, D., & Fiesler, E. (1997). A survey of partially connected neural networks. International journal of neural systems, 8(05n06), 535-558.

Page, E. B. (1967). Grading essays by computer: Progress report. In Proceedings of the invitational Conference on Testing Problems.

Fauzi, M. A., Utomo, D. C., Setiawan, B. D., & Pramukantoro, E. S. (2017, August). Automatic essay scoring system using N-gram and cosine similarity for gamification based E-learning. In Proceedings of the International Conference on Advances in Image Processing (pp. 151-155).

Fedus, W., Goodfellow, I., & Dai, A. M. (2018). MaskGAN: better text generation via filling in the_. arXiv preprint arXiv:1801.07736.

Foltz, P. W., Laham, D., & Landauer, T. K. (1999). Automated essay scoring: Applications to educational technology. In EdMedia+ innovate learning (pp. 939-944). Association for the Advancement of Computing in Education (AACE).

Forster, E. A., & Dunbar, K. N. (2009, July). Creativity evaluation through latent semantic analysis. In Proceedings of the Annual Conference of the Cognitive Science Society (Vol. 2009, pp. 602-607).

Fryer, M. (2012). Some key issues in creativity research and evaluation as seen from a psychological perspective. Creativity Research Journal, 24(1), 21-28.

Gabora, L. (2016). Honing theory: A complex systems framework for creativity. arXiv preprint arXiv:1610.02484.

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017, August). Convolutional sequence to sequence learning. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 1243-1252). JMLR. org.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

Guan, Y., Xie, Y., Liu, X., Sun, Y., & Gong, B. (2019, August). Understanding Lexical Features for Chinese Essay Grading. In CCF Conference on Computer Supported Cooperative Work and Social Computing (pp. 645-657). Springer, Singapore.

Graves, A., Fernández, S., & Schmidhuber, J. (2005, September). Bidirectional LSTM networks for improved phoneme classification and recognition. In International Conference on Artificial Neural Networks (pp. 799-804). Springer, Berlin, Heidelberg.

Hinton, G. E. (1986, August). Learning distributed representations of concepts. In Proceedings of the eighth annual conference of the cognitive science society (Vol. 1, p. 12).

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural computation, 18(7), 1527-1554.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with

neural networks. science, 313(5786), 504-507.
Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.
Hong, Y., Zhou, W., Zhang, J., Zhou, G., & Zhu, Q. (2018, July). Self-regulation: Employing a generative adversarial network to improve event detection. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 515-526).
Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146.
Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., & Xing, E. P. (2017, August). Toward controlled generation of text. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 1587-1596). JMLR. org.
Huang, C. Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Hawthorne, C., Dai, A. M., ... & Eck, D. (2018). An improved relative self-attention mechanism for transformer with application to music generation. arXiv preprint arXiv:1809.04281.
Huang, D. A., Nair, S., Xu, D., Zhu, Y., Garg, A., Fei-Fei, L., ... & Niebles, J. C. (2019). Neural task graphs: Generalizing to unseen tasks from a single video demonstration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8565-8574).
Japkowicz, N., Myers, C., & Gluck, M. (1995, August). A novelty detection approach to classification. In IJCAI (Vol. 1, pp. 518-523).
Jawahar, G., Sagot, B., & Seddah, D. (2019, July). What does BERT learn about the structure of language?.
Jordanous, A. (2012). A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. Cognitive Computation, 4(3), 246-279.
Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4401-4410).
Karampiperis, P., Koukourikos, A., & Panagopoulos, G. (2016). From computational creativity metrics to the principal components of human creativity. In Knowledge, Information and Creativity Support Systems (pp. 447-456). Springer, Cham.
Ke, P., Huang, F., Huang, M., & Zhu, X. (2019). ARAML: A Stable Adversarial Training Framework for Text Generation. arXiv preprint arXiv:1908.07195.
Ke, Z., & Ng, V. (2019, August). Automated essay scoring: a survey of the state of the art. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (pp. 6300-6308). AAAI Press.
Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.
Kumar, S., Chakrabarti, S., & Roy, S. (2017, August). Earth Mover's Distance Pooling over Siamese LSTMs for Automatic Short Answer Grading. In IJCAI (pp.

2046-2052).
Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
Krizhevsky, Alex , (2013). ImageNet Classification with Deep Convolutional Neural Networks. Retrieved 17 November 2013.
Lample, G., & Conneau, A. (2019). Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291.
Lample, G., Conneau, A., Denoyer, L., & Ranzato, M. A. (2017). Unsupervised machine translation using monolingual corpora only. arXiv preprint arXiv:1711.00043.
Larkey, L. S. (1998, August). Automatic essay grading using text categorization techniques. In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (pp. 90-95).
Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. Discourse processes, 25(2-3), 259-284.
LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), 541-551.
LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.
Lee, K., Han, S., & Myaeng, S. H. (2018). A discourse-aware neural network-based text model for document-level text classification. Journal of Information Science, 44(6), 715-735.
Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019.
Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In Mining text data (pp. 415-463). Springer, Boston, MA.
Liu, C., Yang, D., Xia, X., Yan, M., & Zhang, X. (2019). A two-phase transfer learning model for cross-project defect prediction. Information and Software Technology, 107, 125-136.
Liu, J., Cohen, S. B., & Lapata, M. (2018, July). Discourse representation structure parsing. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 429-439).
Liu, J., Xu, Y., & Zhao, L. (2019). Automated essay scoring based on two-stage learning. arXiv preprint arXiv:1901.07744.
Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025.
Ma, W. D. K., Lewis, J. P., & Kleijn, W. B. (2019). The HSIC Bottleneck: Deep Learning without Back-Propagation. arXiv preprint arXiv:1908.01580.
Mahana, M., Johns, M., & Apte, A. (2012). Automated essay grading using machine learning. Mach. Learn. Session, Stanford University.
Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial autoencoders. arXiv preprint arXiv:1511.05644.

Marblestone, A. H., Wayne, G., & Kording, K. P. (2016). Toward an integration of deep learning and neuroscience. Frontiers in computational neuroscience, 10, 94.
Markou, M., & Singh, S. (2003). Novelty detection: a review—part 1: statistical approaches. Signal processing, 83(12), 2481-2497.
Markou, M., & Singh, S. (2003). Novelty detection: a review—part 2:: neural network based approaches. Signal processing, 83(12), 2499-2521.
McNamara, D. S., Crossley, S. A., Roscoe, R. D., Allen, L. K., & Dai, J. (2015). A hierarchical classification approach to automated essay scoring. Assessing Writing, 23, 35-59.
Mehmood, A., On, B. W., Lee, I., & Choi, G. S. (2017). Prognosis essay scoring and article relevancy using multi-text features and machine learning. Symmetry, 9(1), 11.
Meng, Y., & Rumshisky, A. (2018, July). Context-aware neural model for temporal information extraction. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 527-536).
Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.
Misra, D. (2019). Mish: A Self Regularized Non-Monotonic Neural Activation Function. arXiv preprint arXiv:1908.08681.
Mittal, S. (2018). A survey of FPGA-based accelerators for convolutional neural networks. Neural computing and applications, 1-31.
Mohiuddin, T., Joty, S., & Nguyen, D. T. (2018). Coherence modeling of asynchronous conversations: A neural entity grid approach. arXiv preprint arXiv:1805.02275.
Montahaei, E., Alihosseini, D., & Baghshah, M. S. (2019). Jointly measuring diversity and quality in text generation models. arXiv preprint arXiv:1904.03971.
Mueller, J., & Thyagarajan, A. (2016, March). Siamese recurrent architectures for learning sentence similarity. In thirtieth AAAI conference on artificial intelligence.
Nadeem, F., Nguyen, H., Liu, Y., & Ostendorf, M. (2019, August). Automated Essay Scoring with Discourse-Aware Neural Models. In Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications (pp. 484-493).
Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, 381(6583), 607-609.
Panagopoulos, G., Karampiperis, P., Koukourikos, A., & Konstantinidis, S. (2015). Creativity Profiling Server: Modelling the Principal Components of Human Creativity over Texts. In UMAP Workshops.
Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

Pimentel, M. A., Clifton, D. A., Clifton, L., & Tarassenko, L. (2014). A review of novelty detection. Signal Processing, 99, 215-249.

Phandi, P., Chai, K. M. A., & Ng, H. T. (2015, September). 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).

Qiao, T., Zhang, J., Xu, D., & Tao, D. (2019). Mirrorgan: Learning text-to-image generation by redescription. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1505-1514).

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog, 1(8), 9.

Ramos-Soto, A., Bugarín, A., & Barro, S. (2016). On the role of linguistic descriptions of data in the building of natural language generation systems. Fuzzy Sets and Systems, 285, 31-51.

Rajeswar, S., Subramanian, S., Dutil, F., Pal, C., & Courville, A. (2017). Adversarial generation of natural language. arXiv preprint arXiv:1705.10929.

Reddy, S., Chen, D., & Manning, C. D. (2019). Coqa: A conversational question answering challenge. Transactions of the Association for Computational Linguistics, 7, 249-266.

Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A Primer in BERTology: What we know about how BERT works. arXiv preprint arXiv:2002.12327.

Rudner, L. M., & Liang, T. (2002). Automated essay scoring using Bayes' theorem. The Journal of Technology, Learning and Assessment, 1(2).

Runco, M. A., Pritzker, M. A., Pritzker, S. R., & Pritzker, S. (Eds.). (1999). Encyclopedia of creativity (Vol. 2). Elsevier.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. In Advances in neural information processing systems (pp. 2234-2242).

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE transactions on Signal Processing, 45(11), 2673-2681.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.

Shermis, M. D., & Burstein, J. (Eds.). (2013). Handbook of automated essay evaluation: Current applications and new directions. Routledge.

Song, M., & Ding, Y. (2014). Topic modeling: Measuring scholarly impact using a topical lens. In Measuring Scholarly Impact (pp. 235-257). Springer, Cham.
Spinks, G., & Moens, M. F. (2018). Generating continuous representations of medical texts. arXiv preprint arXiv:1805.05691.
Stephen, T. C. (2019). Using Automated Essay Scoring to Assess Higher-Level Thinking Skills in Nursing Education.
Sternberg, R. J. (2019). Evaluation of Creativity Is Always Local. In The Palgrave Handbook of Social Creativity Research (pp. 393-405). Palgrave Macmillan, Cham.
Sun, R. (2019). Optimization for deep learning: theory and algorithms. arXiv preprint arXiv:1912.08957.
Taghipour, K., & Ng, H. T. (2016, November). A neural approach to automated essay scoring. In Proceedings of the 2016 conference on empirical methods in natural language processing (pp. 1882-1891).
Talo, M., Baloglu, U. B., Yıldırım, Ö., & Acharya, U. R. (2019). Application of deep transfer learning for automated brain abnormality classification using MR images. Cognitive Systems Research, 54, 176-188.
Tang, J., Yang, Y., Carton, S., Zhang, M., & Mei, Q. (2016). Context-aware natural language generation with recurrent neural networks. arXiv preprint arXiv:1611.09900.
Tandalla, L. (2012). Scoring short answer essays. ASAP short answer scoring competition-Luis Tandalla's approach. ASAP Short Answer Scoring Competition-Luis Tandalla's Approach, 9.(accessed on 14 November 2018).
Tamaazousti, Y., Le Borgne, H., Hudelot, C., Seddik, M. E. A., & Tamaazousti, M. (2019). Learning more universal representations for transfer-learning. IEEE transactions on pattern analysis and machine intelligence.
Tay, Y., Phan, M. C., Tuan, L. A., & Hui, S. C. (2018, April). SkipFlow: Incorporating neural coherence features for end-to-end automatic text scoring. In Thirty-Second AAAI Conference on Artificial Intelligence.
Tevet, G., Habib, G., Shwartz, V., & Berant, J. (2018). Evaluating text gans as language models. arXiv preprint arXiv:1810.12686.
Thomson, C., Reiter, E., & Sripada, S. (2018). Comprehension driven document planning in natural language generation systems. In Proceedings of The 11th International Natural Language Generation Conference. Association for Computational Linguistics (ACL).
Ullman, S. (2019). Using neuroscience to develop artificial intelligence. Science, 363(6428), 692-693.
Van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. In Advances in neural information processing systems (pp. 2643-2651).
van der Maas, H. L., & Molenaar, P. C. (1994). The empirical detection of creativity. Behavioral and Brain Sciences, 17(3), 555-555.
Vanni, L., Ducoffe, M., Aguilar, C., Precioso, F., & Mayaffre, D. (2018, July). Textual Deconvolution Saliency (TDS): a deep tool box for linguistic analysis. In

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 548-557).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

Vig, J., & Belinkov, Y. (2019). Analyzing the structure of attention in a transformer language model. arXiv preprint arXiv:1906.04284.

Vinodhini, G., & Chandrasekaran, R. M. (2012). Sentiment analysis and opinion mining: a survey. International Journal, 2(6), 282-292.

Wang, H. C., Chang, C. Y., & Li, T. Y. (2005, December). Automated Scoring for Creative Problem Solving Ability with Ideation-Explanation Modeling. In ICCE (pp. 524-531).

Wang, H. C., Chang, C. Y., & Li, T. Y. (2008). Assessing creative problem-solving with automated text grading. Computers & Education, 51(4), 1450-1466.

Wang, J., & Brown, M. S. (2007). Automated Essay Scoring versus Human Scoring: A Comparative Study. Journal of Technology, Learning, and Assessment, 6(2), n2.

Wang, S. H., Lv, Y. D., Sui, Y., Liu, S., Wang, S. J., & Zhang, Y. D. (2018). Alcoholism detection by data augmentation and convolutional neural network with stochastic pooling. Journal of medical systems, 42(1), 2.

Wang, T. C., Liu, M. Y., Zhu, J. Y., Tao, A., Kautz, J., & Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8798-8807).

Wang, Y., Wei, Z., Zhou, Y., & Huang, X. J. (2018). Automatic essay scoring incorporating rating schema via reinforcement learning. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 791-797).

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., & Ma, Y. (2008). Robust face recognition via sparse representation. IEEE transactions on pattern analysis and machine intelligence, 31(2), 210-227.

Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., & He, X. (2018). Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1316-1324).

Xu, Z. J. (2018). Understanding training and generalization in deep learning by fourier analysis. arXiv preprint arXiv:1808.04295.

Xu, Z. Q. J., Zhang, Y., & Xiao, Y. (2019, December). Training behavior of deep neural network in frequency domain. In International Conference on Neural Information Processing (pp. 264-274). Springer, Cham.

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.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016, June). Hierarchical

attention networks for document classification. In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies (pp. 1480-1489).

Yin, W., Ebert, S., & Schütze, H. (2016). Attention-based convolutional neural network for machine comprehension. arXiv preprint arXiv:1602.04341.

Yin, W., Schütze, H., Xiang, B., & Zhou, B. (2016). Abcnn: Attention-based convolutional neural network for modeling sentence pairs. Transactions of the Association for Computational Linguistics, 4, 259-272.

You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., ... & Hsieh, C. J. (2019, September). Large batch optimization for deep learning: Training bert in 76 minutes. In International Conference on Learning Representations.

Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent neural network regularization. arXiv preprint arXiv:1409.2329.

Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2018). Self-attention generative adversarial networks. arXiv preprint arXiv:1805.08318.

Zhang, Y., Gan, Z., & Carin, L. (2016). Generating text via adversarial training. In NIPS workshop on Adversarial Training (Vol. 21).

Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820.

Zhang, Y. D., Muhammad, K., & Tang, C. (2018). Twelve-layer deep convolutional neural network with stochastic pooling for tea category classification on GPU platform. Multimedia Tools and Applications, 77(17), 22821-22839.

Zhang, S., Zhao, H., Wu, Y., Zhang, Z., Zhou, X., & Zhou, X. (2019). Dual co-matching network for multi-choice reading comprehension. arXiv preprint arXiv:1901.09381.

Zhang, W. (1988, September). Shift-invariant pattern recognition neural network and its optical architecture. In Proceedings of annual conference of the Japan Society of Applied Physics.

Zhang, W., Itoh, K., Tanida, J., & Ichioka, Y. (1990). Parallel distributed processing model with local space-invariant interconnections and its optical architecture. Applied optics, 29(32), 4790-4797.

Zhao, Y., Zhang, L., & Tu, K. (2018). Gaussian mixture latent vector grammars. arXiv preprint arXiv:1805.04688.

Zhou, H., Huang, M., Zhang, T., Zhu, X., & Liu, B. (2018, April). Emotional chatting machine: Emotional conversation generation with internal and external memory. In Thirty-Second AAAI Conference on Artificial Intelligence.

Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., & Zhao, T. (2018). Neural document summarization by jointly learning to score and select sentences. arXiv preprint arXiv:1807.02305.

Zhu, X., Xu, Z., & Khot, T. (2009, June). How Creative is Your Writing?. In Proceedings of the workshop on computational approaches to linguistic creativity (pp. 87-93).

Zupanc, K., & Bosnić, Z. (2017). Automated essay evaluation with semantic analysis. Knowledge-Based Systems, 120, 118-132.

에세이 자동 스코어링을 위한 딥러닝

리앙 궈시

군산대학교 대학원 글로벌창업학과

글로벌창업전공

(지도교수 온 병 원)

인공지능에서 기계가 텍스트에 점수를 매기는 것은 오랜 시간동안 도전적이면서 흥미로운 과제이다. 최근 신경망 모델은 AES (Automated essay Score) 작업에 적용되고 있으며 엄청난 잠재력을 보이다. 기존에 수작업으로 특징 추출하는 방안과 비교해보면 딥러닝에 일반적 AES 방법의 정확도가 크게 향상되었다. 그러나 대부분의 딥러닝에 모델은 텍스트 자체의 등급 정보 만 배울 수 있으며 정확도는 여전히 개선의 여지가 있다. 이 논문은 평점 평균 정확도를 높이 는 문제 에 대해 창조 적 인 논문 발굴을 모색 했다. 1) 정보를 채점하기 전에 딥 네트워크를 통해 더 많은 정보를 얻는 방법. 2) AES를 위한 새롭고 효과적인 딥 네트워크를 구축하는 방법. 3) AES를 수행 할 때 창의적인 에세이를 찾는 방법. 저희는 딥러닝 모델로 하여 금 평점 정보를 더 많이 배우게 하고 더 좋은 딥러닝 네트워크 구조를 디자인 하는 것 이 자동 평점 효과를 높이 는 데 도움 이 된다고 생각한다. 한편, 창의 적인 논문의 발견은 AES 더욱 매력 적 이고 스마트 하 게 만들 것 이라고 생각한다.

본 논문의 연구 중심은 AES의 정확성과 창조적 논문에 대한 발굴이다. 먼저, 자체 학습 표시 체제와 새로운 신경망 구조를 제시 하여 AES의 정

밀도를 높인다. 전통적 인핸드 메이드 모델 보다 단아한 신경망 모델은 언어 현상을 풍부하게 배우는 데 효과적이다. 이는 일반 신경망에 비해 연결체 구조를 가진 심층 신경망 네트워크도 성능 이 좋다. 손으로 뽑은 것과 같은 자체 학습 메커니즘은 신경망에서 더 많은 정보를 배우는 데 적합하다. 우리가 제시 한 방법을 ASAP 임무에 활용 하여 기존의 방법 보다 더 좋은 성능을 거두었다.

이를 바탕으로 AES 기반의 창의적 인 글 발굴을 모색 했다. 우리는 창의 적인 글을 추천하는 감독 없는 방식으로 교육을 받을 수 있는 텍스트 인검스를 제안 했다. 이 모델 이 추천 하는 창의적 인 글은 받아들일 수 있는 것으로 나타났다. 우리는 이 작업이 인력의 양을 줄이고 미래의 온라인 학습을 가속화하는 데 도움이 될 것 이라고 굳게 믿는다.

이 논문은 6개의 챕터로 구성된다. 1장에서는 AES의 연구 배경, 관련 연구에 대해 소개한다. 2장에서는 현재 딥러닝 위한 주요 신경망 모델을 설명하며 이 논문의 기초적 이론이기도한다. 3장에서는 신경망이 사전 정보를 배우고 표현 방법을 제공 할 수 있는 자가 학습 메커니즘이라는 방법을 제안한다. 4장에서는 SBLSTMA (Siamese Bidirectional Long Short-Term Memory Architecture) 라고하는 AES 용 Siamese 신경망을 제안한다. 5 장에서는 창의성 에세이를 찾는 방법에 대한 새로운 연구를 살펴본다. 창의성 에세이 마이닝을 위해 최첨단 언어 모델과 GAN 신경망을 사용한다. 마지막으로 6장에서는 결론 및 향후 연구에 대해 논의한다.

ABSTRACT

Deep Learning for Automated Essay Scoring

Liang Guoxi

Department of Global Company Start-Up

Kunsan National University

Gunsan, Korea

(Supervised by professor **On Byung-Won**)

[1] Teaching machines to learn how to score text is one of the most fantastic tasks and long-standing challenges in Artificial Intelligence. The neural network model has recently been applied to the task of automated essay scoring (AES) and demonstrates tremendous potential. Compared with the conventional handcrafted feature extraction approaches, the AES method's average accuracy based on deep learning has been greatly improved. However, improving

---

*A thesis submitted to the Committee of the Graduate School, Kunsan National University in partial fulfillment of the requirements for degree of doctor of philosophy in August 2020.

the scoring accuracy of AES and creativity evaluation is still our primary goal. This thesis tackles the problem of improving the average accuracy of the score and makes exploration of creativity essay mining: 1. How to make a deep network to learn more prior to scoring information 2. How to build a novel, more effective deep network for AES. 3. How to find out creative essays when doing AES. On the one hand, we think that let the deep model learn more rating information, and design a better deep network architecture helps improve the automatic scoring effect. On the other hand, we think that the discovery of creativity essays will make AES more attractive and intelligent.

In this thesis, we focus on improving the accuracy of AES and the creativity essay mining. First, we propose a self-learning representation mechanism and a novel neural network architecture for improving the accuracy of AES. Compared to traditional handcrafted feature-based models, this kind of end-to-end neural model has proven to be more effective in learning-rich linguistic phenomena. Compared with general neural networks, this deep neural network with a siamese architecture also has better performance. The self-learning mechanisms that are similar to handcrafted feature extraction are suitable for neural networks learning more information. We apply the proposed approach to the task of ASAP and get better performance than the previous methods.

Furthermore, we make an exploration of creativity essay mining based on AES. We propose a text GANs, which can be trained in an unsupervised way to recommend the creativity essay. The experimental

results show that the creativity essay recommended by the proposed model is acceptable. We firmly believe that this work will help reduce manual workloads and speed up online learning in the future.

This thesis consists of six chapters. Chapter 1 introduces the research background of AES, related work, research motivation, and the main contributions of this thesis. Chapter 2 states the main neural network models for current deep learning, which are also the theoretical basics of this thesis. Chapter 3 proposes a method called self-learning mechanism to help neural networks learn prior information and give a representation method on it. Chapter 4 proposes a Siamese neural network for AES, called Siamese Bidirectional Long Short-Term Memory Architecture (SBLSTMA), by which the self-learning mechanism was also involved. In chapter 5, we explore new work on how to find out creativity essays. We employ the state of the art language model and GANs neural network for creativity essay mining. Lastly, in chapter 6, we summarize the existing work and look forward to future development trends.