applied sciences

MDPI

Article

Automatic Essay Scoring Method Based on Multi-Scale Features

Feng Li [1,2,3], Xuefeng Xi [1,2,3,*], Zhiming Cui [1,2,3], Dongyang Li [1,2,3] and Wanting Zeng [1,2,3]

[1] School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou 215009, China
[2] Key Laboratory of Virtual Reality Intelligent Interaction and Application, Suzhou 215009, China
[3] Smart City Research Institute, Suzhou 215009, China
* Correspondence: xfxi@usts.edu.cn

Abstract: Essays are a pivotal component of conventional exams; accurately, efficiently, and effectively grading them is a significant challenge for educators. Automated essay scoring (AES) is a complex task that utilizes computer technology to assist teachers in scoring. Traditional AES techniques only focus on shallow linguistic features based on the grading criteria, ignoring the influence of deep semantic features. The AES model based on deep neural networks (DNN) can eliminate the need for feature engineering and achieve better accuracy. In addition, the DNN-AES model combining different scales of essays has recently achieved excellent results. However, it has the following problems: (1) It mainly extracts sentence-scale features manually and cannot be fine-tuned for specific tasks. (2) It does not consider the shallow linguistic features that the DNN-AES cannot extract. (3) It does not contain the relevance between the essay and the corresponding prompt. To solve these problems, we propose an AES method based on multi-scale features. Specifically, we utilize Sentence-BERT (SBERT) to vectorize sentences and connect them to the DNN-AES model. Furthermore, the typical shallow linguistic features and prompt-related features are integrated into the distributed features of the essay. The experimental results show that the Quadratic Weighted Kappa of our proposed method on the Kaggle ASAP competition dataset reaches 79.3%, verifying the efficacy of the extended method in the AES task.

Keywords: NLP; automated essay scoring; deep semantic features; multi-scale features

Citation: Li, F.; Xi, X.; Cui, Z.; Li, D.; Zeng, W. Automatic Essay Scoring Method Based on Multi-Scale Features. *Appl. Sci.* **2023**, *13*, 6775. https://doi.org/10.3390/app13116775

Academic Editor: Tobias Meisen

Received: 8 May 2023
Revised: 24 May 2023
Accepted: 29 May 2023
Published: 2 June 2023

1. Introduction

During the writing stage, examinees are asked to write an essay according to the prompt; then, scorers mark the essay. Since the scoring requires a lot of time and effort, it is difficult to grade a large number of essays [1]. In addition, the scorers are easily influenced by personal subjective factors in the grading process [2]. AES is a technique that utilizes NLP technology to evaluate essays, which can be analyzed and evaluated through multiple aspects (such as language, structure, and content) with an objective, fast, and accurate scoring result [3]. AES can circumvent many disadvantages of traditional scoring methods, save labor costs, and not be influenced by personal subjective factors [4], while substantially improving the fairness and accuracy of scoring. Over the years, we have designed many AES methods, which can generally be classified as feature engineering methods or DNN methods [5].

Early feature engineering-based AES methods construct shallow features based on scoring criteria, such as grammar, syntax, and chapter structure; then, they use machine learning to indirectly evaluate the essay [6]. Such technology, which is based on hand-crafted features, overlooks the potential deep information in the essay. Consequently, it cannot obtain satisfactory results for AES tasks, which require the use of the deep semantic information in the essay. Secondly, this technology requires a lot of time and cost in terms of hand-crafted feature extraction. In addition, the evaluation criteria of different AES tasks

are different, thus resulting in the weak generalization ability of the above methods in realistic scenarios.

Recently, DNN approaches have obtained outstanding performance in AES tasks. DNN-AES models are trained using large-scale datasets to score essays based on a word or sentence embedding representation [7]. These methods extract high-dimensional features in essays through the extraction, crossover, and fusion of neural units. However, the scoring process is a complex human activity that requires different levels to score essays; a single network structure cannot fully capture the semantic features of essays. As a result, the performance of a DNN-only approach is somewhat limited in the scoring process. Some researchers recently point out that the hybrid model fusing different scales of essays can significantly enhance the scoring capability [8,9]. Specifically, the document-scale feature can characterize the global information, and the sentence-scale feature can extract the local information in the essays that is lost in document-scale features.

To obtain the advantages of the feature engineering and DNN method, Dasgupta et al. [10] proposed a hybrid approach. It is a DNN-AES model with features formulated with the document-scale features and an additional recurrent neural network (RNN) for processing the sentence-scale manual features. This hybrid approach obtains advanced accuracy but has three drawbacks. First, it manually extracts sentence-scale features and cannot be fine-tuned according to the specific tasks, which results in a suboptimal performance and makes it difficult to extend to other AES tasks of different datasets. Secondly, it does not incorporate the shallow information in essays via the shallow linguistic feature extraction used in previous research. Thirdly, it does not take into account the relevance between the essay and the corresponding prompt, which will be considered in scoring.

- In order to reduce the influences of the above problems and to score essays more comprehensively, we propose the AES method based on Multi-Scale Semantic Features (MSSF). In particular, we extract multiple-scale characteristics with different modules: (1) We utilize the LSTM-MoT model to extract document-scale global semantic features of essays. (2) After the sentence vector of the essay is extracted by the SBERT, the context relevance of the local features is extracted by LSTM. Then, we utilize attention pooling to determine the contribution of the final scores and obtain sentence-scale local semantic features. (3) The relevance between the essays and their corresponding prompts is an important basis for scoring. We vectorize them by Doc2Vec and calculate their similarities to obtain their relevance features. (4) In addition, to address the shortcomings of DNN models in extracting shallow features, such as grammatical errors and text richness, we use manually extracted features with the adaptive weight to obtain the shallow linguistic features of the essays. MSSF fuses global semantic features, local semantic features, prompt relevance features, and shallow linguistic features for essays. Experiments are conducted on the Kaggle ASAP competition with our proposed model. The experimental results show that our proposed AES model with multi-scale semantic hybrid features can effectively improve the performance of the automatic scoring of essays and obtain the optimal performance compared with the baseline model. Our main contributions are as follows: we add 18 typical manual features with adaptive weights to the distributed representation of essays. They can not only extract valuable quantitative information from essays that are difficult to extract from DNN-AES, but can also adjust the weight parameters adaptively according to different AES tasks.
- We utilize SBERT as the sentence vectorization method of the essay. Compared with manually extracted sentence-level features, it can be fine-tuned according to specific tasks after pre-training the tasks and can make the final score more accurate.
- We add the relevance feature between the prompt and the essay. This feature allows the model to learn the correlation between them, rather than just utilizing the essay for scoring.

Furthermore, the features of point 2 and point 3 are easily extended to other AES models, since traditional AES models ultimately have a distributed representation layer for final scoring.

The remainder of this paper is organized as follows: we will introduce different types of existing AES methods and discuss their limitations in Section 2; Section 3 will develop the various components of the model proposed in this paper; Section 4 will present the experimental results and analysis, with the validity of the proposed method being discussed based on the experiments; finally, we will summarize the conclusions and propose further work in Section 5.

2. Related Work

The AES system is a significant tool used to assist teachers in scoring by computer technology; many research results have been achieved in related fields. According to different methods, the AES model mainly consists of four categories: AES based on shallow linguistic features, AES based on deep neural networks, pre-trained AES, and methods based on hybrid models.

2.1. AES Based on Shallow Linguistic Features

The main shallow linguistic features (such as spelling mistakes, essay length, and average sentence length) are extracted manually by scoring criteria and the final score is obtained by regression, classification, or ranking algorithms. Project Essay Grade (PEG) [11] is one of the first AES models for essays, which analyzes the text of an essay and scores it based on factors such as grammar, vocabulary, and coherence. Mathias et al. [12] extracted features, such as content, organization, and sentence fluency, based on the essay and gave each feature an independent score; they then fused them using a random forest to determine the score. Sakaguchi et al. [13] used the N-gram and syntactic features to obtain a score with the Support Vector Machine (SVM) before using the sentence similarity feature to find the cosine similarity between sentences to obtain another score; they obtained the final score by fusing the two scores score. Cummins et al. [14] designed a method to rank all essays utilizing quantitative metrics, such as explanatory article length, and grammatical relevance, as features and then transformed the ranking into a predicted essay score. The greatest advantage of such methods is that they possess strong explanatory power, but they cannot capture the deep semantic information of essays. Nguyen et al. [15] implement an end-to-end argument mining system, which collects argumentative structures of essays and creates argumentative features from these structures, using machine learning to develop the proof-ready argument-mining-enabled AES model.

2.2. AES Based on Deep Neural Networks

Many DNN structures have been used in AES and achieved excellent performance in recent years. Compared with the traditional manual feature extraction, DNN-AES does not need manual design and feature extraction, and can automatically learn the deep semantic representation of complex essays. Essays contain intricate information that is difficult to capture by traditional feature engineering and DNN-AES models have the potential to generalize well to unseen data. In addition, DNN-AES models are highly flexible and can adapt to different types of essay prompts and writing styles. TextCNN is effective in extracting the local features of texts [16], which is often used to extract important semantic information in essay sentences. The LSTM is often used in AES systems [17], which has advantages in processing long-sequence data, effectively extracting the interdependence between words and solving long-term dependence on sentences. Dong et al. [18] proposed a stratified convolutional neural network(CNN). The first layer of the model is utilized to extract sentence-scale features, the next is used to extract article-scale content information, and the fully-connected layer is utilized to generate the final score of the essay. Taghipour et al. [19] and Nguyen et al. [20] utilized the recurrent structure in the AES system to obtain the complex relevance between the text and its score to derive the final score. Dong

et al. [21] proposed to utilize a CNN-LSTM with a double-layer network structure. The CNN is used to obtain local features, and LSTM mainly learns global context sequence features. Then, the weight relevance of each sentence vector to the final score is dynamically obtained through the attention pooling mechanism and the final score is obtained after weighted summation. Ridley et al. [22] proposed an algorithm for evaluating the total score and dimension score of an essay across topics and obtained the final essay representation through the shared layer and the private layer as input to obtain the final score.

2.3. AES Based on Pre-Trained Models

AES based on the pre-trained model has become an important part of the essay scoring task, which is a deep network structure based on large-scale unlabeled corpus training. AES based on the pre-trained model trains the parameters in advance by pre-training tasks and then fine-tuning the AES task. BERT [23] is a two-way Transform encoder that could be fine-tuned in 2018 and has achieved excellent performance in many NLP tasks. Pedroet et al. [24] proposed a sequence-to-sequence model, which extracts the semantics by BERT from both directions and features, such as the next sentence, by XLnet [25]. Wang et al. [26] used two BERT pre-trained models to extract features at multiple scales. The first BERT model was used to extract document-scale features; the second model extracted chapter-scale features through the LSTM layer after connecting multiple BERT models in parallel. The predicted scores of all scales are fused to determine the final score. The experiments of this work point out that using the pre-trained model can effectively improve the performance of AES. Yang et al. [27] used the combination of regression and sorting loss functions to finetune the BERT model for the same task. The purpose of the regression method is to try to obtain an accurate score while the sorting method is designed to obtain an accurate essay ranking.

2.4. AES Based on Hybrid Model

AES based on hybrid models has been shown to enhance the capability of scoring. Combining DNN features and hand-crafted features has been proven by many studies to obtain better scores in essays. Farag et al. [28] considered coherence features between sentences and combined them with a DNN model to further enhance the performance of scoring for AES. Cozma et al. [8] fused character-scale n-gram features and word representation features to extract semantic features and achieved better performance on the ASAP dataset. Liu et al. [29] designed a Two-Stage Learning Framework (TSLF) to extract semantic features, fluency features, and relevance features through the neural network model, fusing artificial features for scoring. Uto et al. [30] proposed a fusion method that utilizes item response theory to consider differences in scoring behavioral characteristics and integrate prediction scores from various AES models.

The methods often use a single network structure, which can no longer meet the accuracy requirements for AES. Therefore, we use extracted document-scale information and sentence-scale information of essays to characterize the deep semantic features of essays and manually extract typical shallow linguistic features to complement the features. In addition, we fuse essay and prompt relevance features to make the scoring more comprehensive.

3. Approach

In this section, we introduce the multi-scale AES method that fully considers the impact of all aspects on the final score. As shown in Figure 1, we utilize an LSTM-based model to extract document-scale global information and a hybrid model to mine essay sentence-scale local features to complement the former and extract deep semantic features of the essay through both. To mine the shallow semantic information, we design and extract the typical features. Finally, we combine these features with the prompt relevance to calculate the essay score.

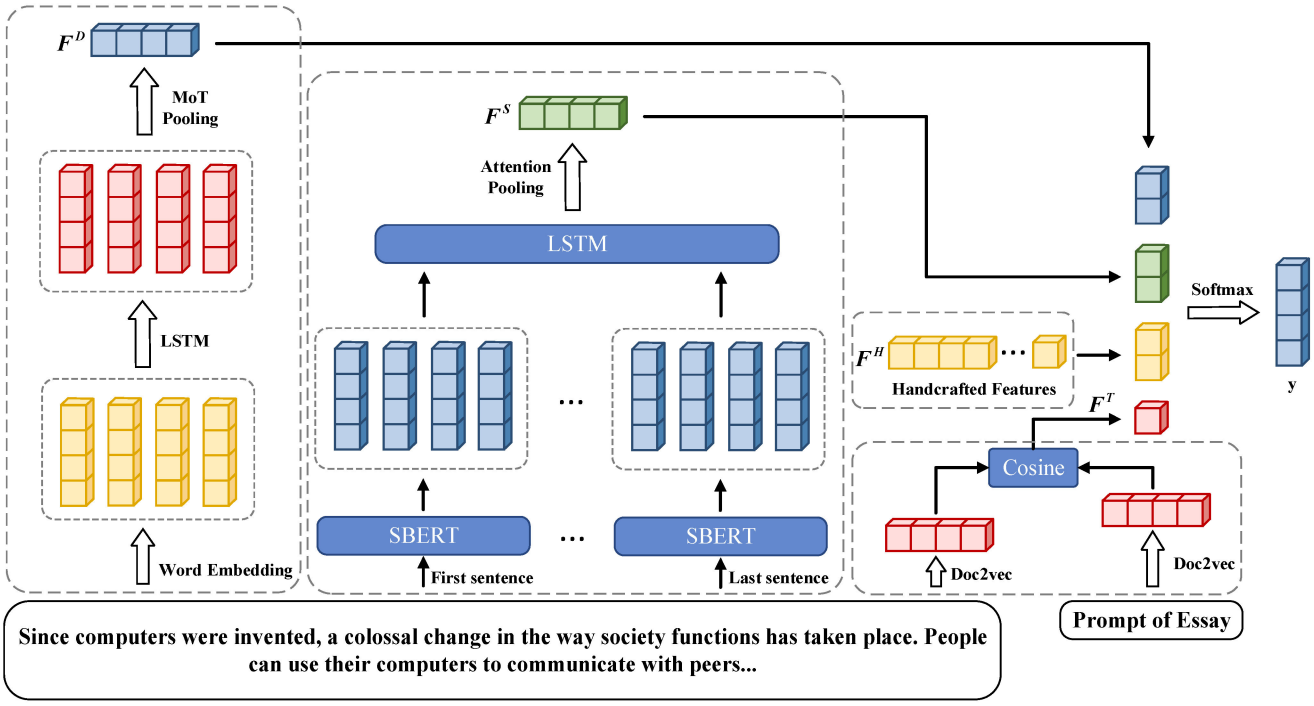

Figure 1. Automatic scoring method for essays based on multi-scale design.

3.1. Document-Scale Global Features

In this part, we introduce the LSTM-MoT network to extract document-scale features of essays, as shown in Figure 2, utilizing the LSTM-based AES model first proposed by Alikaniotis et al. [31]. This model predicts the score of an essay by inputting a sequence of words in the essay through a multilayer neural network. A large body of existing work shows that this structure has significant advantages in extracting document-scale features [32–36].

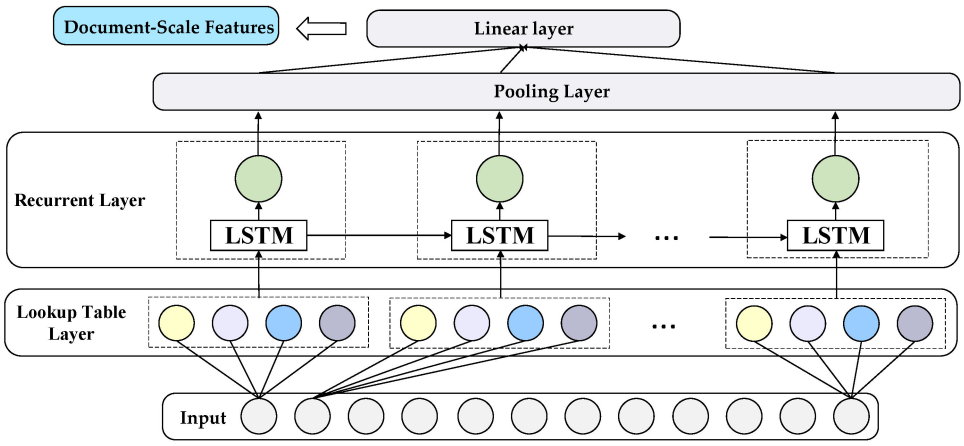

Figure 2. LSTM-MoT model for extracting Document-Scale Global Features.

Let the dictionary table of the essay be V. Define the essay in a lexical way $w_i \in V$, $i = 1, 2, \ldots, n$, where w_i denotes the *-i-th* word in the essay and n denotes the essay length and manipulate the words in the essay via the following method.

Lookup Table Layer. This layer is used to map the sequence of words to a D-dimensional hidden space to generate a vectorized representation of the words $\widetilde{w} = \{\widetilde{w}_1, \widetilde{w}_2, \ldots, \widetilde{w}_i, \ldots, \widetilde{w}_n\}$; semantically similar words have similar vectorized representations among different word vectors. Specifically, the one-hot vector w_i of essay input

is a dot product with the D × V embeddings matrix to obtain the corresponding word embedding representation x_i.

Recurrent Layer. The sequence of the word embedding is out of order, which is the output of the lookup table layer, and if we use them directly for scoring, we cannot obtain an accurate score. To provide the model with the ability to extract context, the recurrent structure is added to capture the temporal features. LSTM is an important variant of RNN that incorporates a gate mechanism into the conventional RNN architecture. This gate mechanism is specifically designed to effectively mitigate issues related to gradient explosion, gradient disappearance, and long-term dependence, which is calculated as follows:

$$f_t = \sigma\Big(W_f \cdot [h_{t-1}, x_t] + b_f\Big) \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\widetilde{C}_t = tanh\ (W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh\ (C_t) \tag{6}$$

where x_t is the input vector at time t; h_t is the output vector at each timestamp; W_f, W_i, W_C and W_o are the learnable parameter matrices; b_f, b_i, b_C and b_o are the bias terms; and σ represents the sigmoid function.

Pooling Layer. We introduce a pooling layer to convert the output dimension of the recurrent layer to the target dimension, which can help reduce the number of parameters, decrease computational complexity, and improve the model's generalization ability. Mean-over-time (MoT) [19] pooling is generally used to compute the average vectorization of each timestamp output in LSTM, as shown in the following equation:

$$M = \frac{1}{n} \sum_{t=1}^{n} h_t \tag{7}$$

Linear layer. This layer performs a linear transformation on the output of the pooling layer, followed by the application of the sigmoid function to map the output to a range of [0, 1]. This process is illustrated by the equation presented below:

$$y = \sigma(W_o \cdot M + b_o) \tag{8}$$

where W_o and b_o are both learnable parameters of weight and bias while σ is the sigmoid function. During the model training process, the scores are normalized to [0, 1] by the actual scores for scoring. For the final prediction, we rescaled the calculated scores to match the original scoring range.

3.2. Sentence-Scale Local Features

Document-scale essay scoring treats an essay as a sequence of words; during scoring with document-scale features, the influence of the sentence as a whole on the scoring of essays is ignored. In previous work, researchers have been focusing on the utilization of sentence-scale features. For example, Uto et al. [37] proposed that combining document-scale features and sentence-scale features could achieve better performance, where the document-scale version is based on the features of the embedding of the input vocabulary and the other module is based on input sentence-scale features, combining the two for

training to obtain excellent performance. Reimers et al. [38] proposed a sentence-embedding model SBERT utilizing the pre-training method on the Semantic Textual Similarity (STS) task; its aim is to quantify the degree of semantic similarity or relatedness between pairs of text snippets. In this paper, we extract sentence vectorized representations of essays by SBERT and propose a new sentence-scale feature of the AES task, as shown in Figure 3.

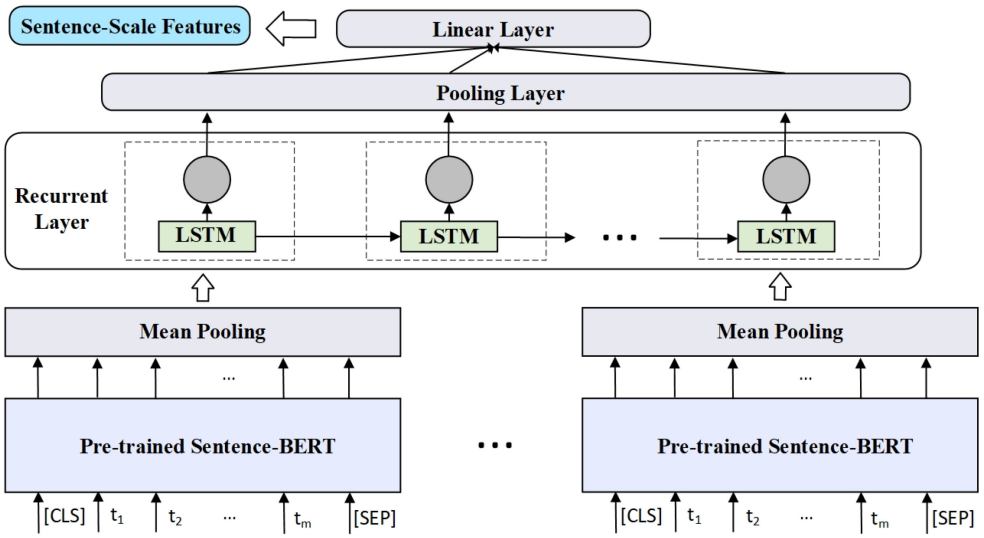

Figure 3. A hybrid neural network model for extracting Sentence-Scale Local Features.

Sentence vectorization layer. Let the words in the essay sentences be defined as $\{t_1, t_2, \ldots, t_m\}$, where m denotes the word number of one sentence. We use the pre-trained SBERT model to extract the deep semantic information of the words and obtain the vectorized representation of the sentences in the essay by mean pooling. The separate BERT model cannot be directly used for the calculation of sentence vectors without the pooling layer. The mean pooling applies the same weight to each word in the sentence and it can reduce over-reliance on specific words while preserving the overall semantic information.

Recurrent layer. After converting essays into sentence vectors, there is no sequential relationship between the vectors. In order to obtain the original temporal order relationship of the sentences in the essay, we utilize the LSTM model to mine the temporal order features. Specifically, the sequence of sentence vectors $\{s_1, s_2, \ldots, s_N\}$ obtained from the output of the sentence vectorization layer is added to the LSTM model to obtain the sequence of output vectors $\{h_1, h_2, \ldots, h_N\}$ to capture the dependencies of the sentences, where N is the number of sentences in the essay.

Pooling Layer. This layer generally uses attention-pooling to compute each time output of the LSTM model at each time step, converting it to a fixed-length vector. Dong et al. [21] compared MoT pooling and attention-pooling in a global essay scoring task, where MoT pooling can equally treat each word in an essay and can better extract the words' semantic information. During the process of sentence-level semantic extraction, attention-pooling is employed to emphasize the impact of important sentences on essay scoring. Additionally, the sentence sequences are much smaller in comparison to the extra-long sequences of global features. This causes attention to be diminished during attention-pooling for extra-long sequences. Taghipour et al. [19] focused on words on a single-layer LSTM but did not exceed the MoT baseline model. Attention pooling obtains the weight of the contribution of the words in the score to the final score in the essay, as shown in the following equation.

$$m_i = \tanh(W_m \cdot h_i + b_m) \tag{9}$$

$$u_i = \frac{e^{W_u \cdot h_i}}{\sum e^{W_u \cdot h_j}} \tag{10}$$

$$x = \sum u_i h_i \tag{11}$$

where W_m and W_u are the weight parameters, b_m is the bias vector, h_i is the LSTM intermediate hidden layer state at different time steps, m_i and u_i are the attention vector and attention weight at different time steps, and x is the final text representation.

Linear layer. This layer applies a linear transformation to the output of the pooling layer and normalizes the output using the sigmoid function. This layer then calculates the final score. However, during the prediction process, the predicted scores are rescaled to match the original scoring range.

3.3. Prompt Relevance Features

A prevalent pattern in essay writing is the prompt essay, where writers receive a prompt asking them to write about a particular topic essay. Evaluators consider the conformity of the essay to the prompt as a crucial criterion during the grading process. Therefore, measuring the semantic relevance between the prompt and the essay is essential in the essay grading process. During the essay scoring process, determining whether the essay aligns with the given prompt becomes an important criterion for scoring essays. Extracting the semantic relevance between the essay and the corresponding prompt will directly affect the performance of AES. Figure 4 shows the prompt relevance model which is proposed in this paper.

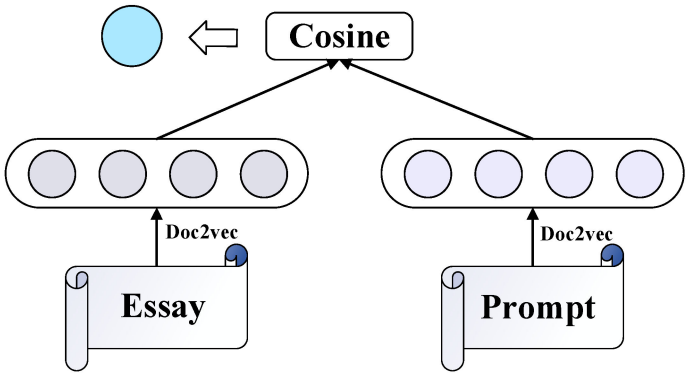

Figure 4. Models for computing prompt relevance.

In the process of traditional prompt model extraction, word co-occurrence and LDA prompt models are commonly used extraction methods [39], which usually extract only the statistical features of essays and cannot capture the degree of relevance between the essay and prompt from the semantic scale.

Since essays are long texts, transformer-based models cannot provide a better-vectorized representation of long texts. Doc2Vec [40] can represent the whole essay using a single low-dimensional dense vector; its structure overcomes the shortcomings of traditional prompt models and transformer-based models and is more suitable for performing essay-prompt similarity degree analyses.

We utilize the Doc2Vec to capture the vectorized representation $E = \{e_1, e_2, \ldots, e_N\}$ of the essay and the prompt of essay $P = \{p_1, p_2, \ldots, p_N\}$ before introducing the cosine similarity function to calculate the semantic similarity relationship F^T between them, which can be formulated as follows.

$$F^T = \frac{\sum_{i=1}^{N} p_i \times e_i}{\sqrt{\sum_{i=1}^{N} (p_i)^2} \sqrt{\sum_{i=1}^{N} (e_i)^2}} \tag{12}$$

3.4. Shallow Linguistic Features

Scoring essays is a complex act and a single-structured model has constraints on scoring performance. Compared to traditional technologies, DNN-AES models have shown superior ability in extracting essay features from the semantic layer of neural networks. However, the detection of spelling and grammatical errors, which are critical in scoring, remains challenging. To address this limitation, the combination of artificial features and deep learning has been proposed to enhance the performance of AES. This hybrid approach takes advantage of both feature-engineering-based and neural-network-based approaches, with their relevance being complementary rather than competitive.

The selection of appropriate hand-crafted features has a significant impact on performance. On the one hand, too many hand-crafted features are time-consuming, labor-intensive, and cause computational redundancy, resulting in sub-optimal performance. On the other hand, the features extracted by DNN-AES can hardly leverage too few hand-crafted features as a useful complement, thus leading to sub-optimal performance. Here, we mainly select the linguistic richness features that are difficult to mine via DNN models and reflect the hand-crafted features of essay quality from various aspects; Table 1 shows the specific hand-crafted features.

Table 1. Representative shallow linguistic features.

Feature Type	No.	Detailed Description
Word-scale features	1	Number of characters
	2	Number of words
	3	Number of punctuation symbols
	4	Number of nouns
	5	Number of verbs
	6	Number of adverbs
	7	Number of adjectives
	8	Number of conjunctions
	9	Number of distinct words
	10	Number of misspellings
	11	Mean of word length
Sentence-scale features	12	Number of sentences
	13	The average length of clauses
	14	The average sentence length
	15	The variance of sentence length
	16	The average depth of the syntax tree of each sentence
	17	the average depth of each leaf node of the syntax tree
Prompt-relevant features	18	Number of words in the essay that appears in the prompt

We design 18 typical hand-crafted features as shallow semantics to reflect the shallow information of the essay. The hand-crafted features are expressed as $\{F_1, \ldots\ldots, F_{18}\}$. Different hand-crafted features contribute differently to the final score. We add weight parameters to each hand-crafted feature to obtain the hand-crafted features with the weight $F^w = \{w_1F_1, \ldots\ldots, w_{18}F_{18}\}$. We obtained the final shallow linguistic features via a linear transformation, as shown in the following equation.

$$F^H = \sigma(W_o \cdot F^w + b_o) \tag{13}$$

3.5. Essay Scoring

In order to improve the scoring efficiency, we propose a method of blending multi-scale features. The document-scale global features, sentence-scale local features, manually

extracted shallow features, and prompt relevance features are fused to obtain the overall feature of the final essay F which is shown as follows:

$$F = \left[F^H, F^D, F^s, F^T\right] \tag{14}$$

We take F as the input of the fully connected layer and finally generate the output as the score through the sigmoid activation function. Its calculation formula is as follows:

$$Score = \sigma(W \cdot F + b) \tag{15}$$

where W is the weight matrix, b is the bias, σ represents the sigmoid function, and Score is the predicted essay score. We normalize all gold standard scores to [0, 1] and use them to train the network. However, we rescale the output to the original score and use the rescaled scores to evaluate the system during testing.

For the training of the model, the Mean Squared Error (MSE) between the predicted and actual scores is typically employed as the loss function, which can be expressed mathematically as follows:

$$loss = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{16}$$

where y_i denotes the actual score of the i-th essay, while $\hat{y}_i$ denotes the predicted score for the i-th essay, and N denotes the essay number. Many of the modules and features in our proposed method can easily expand the existing DNN-AES model because we finally output the distributed representation of the essays.

4. Experiment

4.1. Dataset

We utilize a publicly available dataset published in the 2012 Kaggle Automated Student Assessment Prize (ASAP) competition (https://www.kaggle.com/c/asap-aes/data accessed on 23 May 2023), which is widely used in the field of AES. The essays comprising the ASAP competition dataset are authored by students in grades 7 through 10 and are categorized into 8 groups based on the essay themes, as listed in Table 2. Each group of essays consists of an essay prompt document, which includes multiple essays related to a particular prompt, and each article has an overall rating. The time, place, person, organization, and other information appearing in the essay have been desensitized.

Table 2. Statistics of the ASAP dataset. For genre, ARG indicates argumentative essays, RES corresponds to response essays, and NAR refers to narrative essays.

Prompt	# of Essays	Genre	Average Essay Length	Score Range
1	1783	ARG	350	2~12
2	1800	ARG	350	1~6
3	1726	RES	150	0~3
4	1772	RES	150	0~3
5	1805	RES	150	0~4
6	1800	RES	150	0~4
7	1569	NAR	250	0~30
8	723	NAR	650	0~60

4.2. Evaluation Metric

The Quadratic Weighted Kappa (QWK) coefficient is commonly used as an evaluation method for AES in existing methods. This is primarily due to its strong sensitivity to incorrect predictions. The penalty mechanism becomes increasingly stronger as the difference between the actual score and the predicted result increases, which can better measure the consistency of the model score. This indicator is also officially designated by Kaggle as the

evaluation standard for ASAP competition. QWK is improved by the Kappa coefficient and the quadratic weight matrix in QWK is defined as follows:

$$w_{i,j} = \frac{(i-j)^2}{(N-1)^2} \tag{17}$$

Among them, i and j represent the scores given by the actual manual scoring and the AES system and N is the number of possible ratings. The final QWK coefficient formula is as follows:

$$k = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}} \tag{18}$$

where O is the observation matrix and $O_{i,j}$ represents the number of essays that were manually rated as the i-th category by the AES system and misjudged as the j-th category. The expected count matrix E is obtained by taking the outer product of the reference and hypotheses-rating histograms. Finally, the QWK coefficients are calculated through the three matrices of W, O, and E.

4.3. Experimental Configuration

To evaluate the proposed model, a 5-fold cross-validation approach is utilized in this study. Specifically, each fold comprises 60% training data, 20% validation data, and 20% testing data.

We adopt the 50-dimensional Glove [41] as the embedding matrix in the document-scale DNN-AES model. The basic model of SBERT uses the public pre-trained RoBERT-large (https://public.ukp.informatik.tu-darmstadt.de/reimers/sentence-transformers/ (accessed on 23 May 2023)). The LSTM hidden vector dimension used in the hybrid method is 300, the batch size is 16, and the maximum epoch is set to 60. We use dropout with a probability of 0.5 to avoid overfitting. We introduce Adam as the optimization algorithm and also construct an early stopping setting to prevent overfitting problems.

4.4. Comparative Experiment

To validate the efficacy of the proposed multi-scale feature-based AES method, we conducted a comparison with the baseline methods listed in Table 3.

Table 3. Comparing the performance of the present models with that of the state-of-the-art models on ASAP. The best performance for each prompt is highlighted in bold.

Models	**Prompt**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Ave.**
EASE(SVR)	0.781	0.621	0.630	0.749	0.782	0.771	0.727	0.534	0.699
LSTM-MoT	0.775	0.687	0.683	0.795	0.818	0.813	0.805	0.594	0.746
CNN(10runs) + LSTM(10runs)	0.821	0.688	0.694	0.805	0.807	0.819	0.808	0.644	0.761
CNN-LSTM-ATT	0.822	0.682	0.672	0.814	0.803	0.811	0.801	0.705	0.764
SkipFlow LSTM	0.832	0.684	0.695	0.788	0.815	0.810	0.800	0.697	0.764
TSLF	**0.852**	0.736	0.731	0.801	0.823	0.792	0.762	0.684	0.773
Tran-BERT-MS-ML-R	0.834	0.716	0.714	0.812	0.813	**0.836**	**0.839**	**0.766**	0.791
MSSF	0.846	**0.748**	**0.737**	**0.820**	**0.826**	0.825	0.820	0.725	**0.793**

1. **EASE (SVR)** [42]: Enhanced AI Scoring Engine (EASE) is a public machine-learning-based classification scoring engine (https://github.com/openedx-unsupported/ease (accessed on 23 May 2023)). As EASE relies on hand-engineered features and regression techniques, we adopt the Support Vector Regression (SVR) model as the baseline approach for comparison purposes in this paper.

2. **LSTM-MoT** [19]: Treating the essay text as a word sequence, we use the LSTM network to extract the temporal relationship and average the output of all time step states for the final scoring.
3. **CNN(10runs) + LSTM(10runs)** [19]: By employing an integrated learning approach, ten CNN and ten LSTM models were utilized for prediction, with the resulting predictions being averaged to yield the final prediction outcome.
4. **CNN-LSTM-ATT** [21]: We use CNN and the attention mechanism to extract sentence-scale features of the essay; then, we input the result into the LSTM model and perform the final scoring through the attention mechanism.
5. **SkipFlow LSTM** [43]: To enhance the performance of essay scoring, the model incorporates the SkipFlow mechanism into the LSTM network. This mechanism leverages the semantic relationships between the hidden layers of the LSTM as auxiliary features.
6. **TSLF** [29]: We use LSTM to obtain semantic features, consistency features, and semantic relevance features, combined with features, such as grammatical errors. We then use XGBoost to score the essay.
7. **Tran-BERT-MS-ML-R** [26]: Two BERT models are used to extract token-scale, document-scale, and segment-scale features; multiple losses are used for essay scoring.

The above table shows the proposed multi-scale essay scoring model and the baseline model for comparison in this paper and the experimental results show that:

(1) The prediction of the final score by manually extracting features and then utilizing traditional machine learning algorithms is not effective, indicating that relying on shallow linguistic features alone to characterize the whole essay will miss the deep semantic meaning of the essay, which is very fatal in the process of scoring essays. Secondly, parameters such as the kernel function and penalty parameters of SVR have a significant impact on performance; however, it is difficult to determine the optimal value and it is more difficult to expand the future than the DNN-AES method. LSTM-MoT and CNN(10runs) + LSTM(10runs) utilize the document-scale features of the essay to extract deep semantic information, which has significantly improved the effect compared to the manually extracted features; however, the individual document-scale DNN structure does not extract the semantic features of longer texts well. Furthermore, the whole essay is split into words, with the relevance between the words in the same sentence and the relevance between different sentences of the essay being lost. The training of multiple individual networks using integrated learning has slightly improved the effect but the scoring performance is not good.

(2) Incorporating the SkipFlow mechanism into the LSTM to model the semantic relevance between hidden layers leads to a slight improvement in performance. However, it can only rely on word-level co-occurrence patterns rather than capturing the overall meaning and coherence of the essay. Concretely, in comparison to the conventional LSTM, the performance improvement is only significant for the P1 and P8 subsets, whereas the performance enhancement is comparatively similar for the remaining subsets. The CNN-LSTM-ATT model uses sentence-scale features as the basis for scoring, which is better than document-scale scoring; however, the performance improvement of the model is limited by scoring only at the sentence semantic scale. Furthermore, its accuracy in scoring largely depends on the sentence vectorization method.

(3) Combining manually extracted features and deep semantic features extracted by DNN models can effectively improve the scoring performance of essays. The TSLF model with character features improves by 2.7% over LSTM-MoT. The experimental results show that the hand-crafted features and deep learning features are complementary and are effective complements to the semantic features; additionally, fusing multi-scale features can effectively improve the performance of essay scoring. However, it relies too much on document-level LSTM feature extraction capabilities and does not use additional methods to extract other scales features.

(4) The Tran-BERT-MS-ML-R model achieves a QWK coefficient of 79.1% on the ASAP dataset, indicating the effectiveness and superiority of the pre-trained model in the AES task. This also indicated that scoring based on features of different scales can effectively improve the performance but would cause more computational load. Moreover, our proposed MSSF method outperforms the Tran-BERT-MS-ML-R model and achieves the highest performance scores on four subsets because MSSF extracts document-scale global features and sentence-scale local features from deep semantic features, proposes the similarity relevance between the essay and prompt from the topic scale, and extracts linguistic features from shallow information for scoring. Thus, the proposed model in this paper exhibits the best overall performance when compared to the other models and without excessive computational load.

4.5. Feature Performance Analysis

To verify the effect of features at different scales on the final essay scores, we perform ablation experiments on each scale module of the model. We use SLF to denote the shallow linguistic feature module; DOC to denote the global semantic module based on document scale; SEN to denote the local semantic module based on sentence scale; DOC-SEN to denote the deep feature module combining global semantics and local semantics; DOC-SEN-SLF to denote the simultaneous use of local, global, and shallow semantic features; and MSSF to denote the paper's proposed essay scoring method based on all features. Table 4 lists the effects of different levels of semantic features on the model; the experimental results show that:

(1) The hybrid DOC-SEN model makes a significant contribution to the overall enhancement of the AES performance. Compared with DOC and SEN, the hybrid model has significantly higher performance improvement on a subset of both, with an overall performance improvement of 2.2% and 1.3%, respectively. The experimental results show that combining document-scale global semantic features and sentence-scale local semantic features can better characterize essays and improve essay scoring performance.
(2) Utilizing shallow linguistic features alone is insufficient for essay scoring. However, incorporating shallow linguistic features into the model alongside deep semantic features improves the performance by 2%. Experimental results indicate that since DNN models cannot capture shallow features, such as grammatical errors and linguistic richness, artificial features can effectively complement deep semantic features to improve performance.
(3) Prompt relevance has an impact on the performance of the model. Compared with deep and shallow semantic features, the addition of prompt-relevance features improves the scores on all subsets. However, the improvement is small, with an overall performance improvement of 0.5%. The experimental results demonstrate that incorporating essay and prompt relevance features can enhance the performance of essay scoring. However, the current approach yields limited performance improvement for the model.

To confirm the validity of the hand-crafted shallow linguistic features, we experimentally verified their validity. Figure 5 shows the heat map of the feature weights of the shallow linguistic features as they perform on each data set, with higher weights implying a greater impact on score prediction. We undertake further analysis and find that the weights of the shallow features varied widely between different groups. We believe there are four main causes of this issue: (a) The process of scoring by scorers not only refers to shallow manual features, such as the number of words and sentences, but means they will pay more attention to the deep semantics of the essay, which leads to inaccurate weights on shallow features. (b) Different scorers focus differently on shallow manual features. For example, some scorers care a lot about the number of sentences and some scorers care a lot about the number of nouns. (c) Even if scorers pay attention to the same shallow features, they will not quantitatively analyze the specific value of each shallow feature during the scoring

process, which leads to bias in the final weights. (d) Different essay types and different prompts lead to a different emphasis on shallow feature weights for each essay. Therefore, it is difficult to guarantee the consistency of weights in different essays.

Table 4. Ablation Experiment Results. The bold number is the best performance for each prompt.

Models	Prompt								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Ave.**
SLF	0.802	0.652	0.665	0.768	0.774	0.691	0.732	0.612	0.712
DOC	0.775	0.687	0.683	0.788	0.810	0.807	0.805	0.614	0.746
SEN	0.796	0.695	0.701	0.779	0.801	0.793	0.792	0.678	0.755
DOC-SEN	0.808	0.709	0.705	0.798	0.819	0.814	0.807	0.682	0.768
DOC-SEN-SLF	0.839	0.742	0.732	0.812	0.821	0.817	0.815	0.721	0.788
MSSF	**0.846**	**0.748**	**0.737**	**0.820**	**0.826**	**0.825**	**0.820**	**0.725**	**0.793**

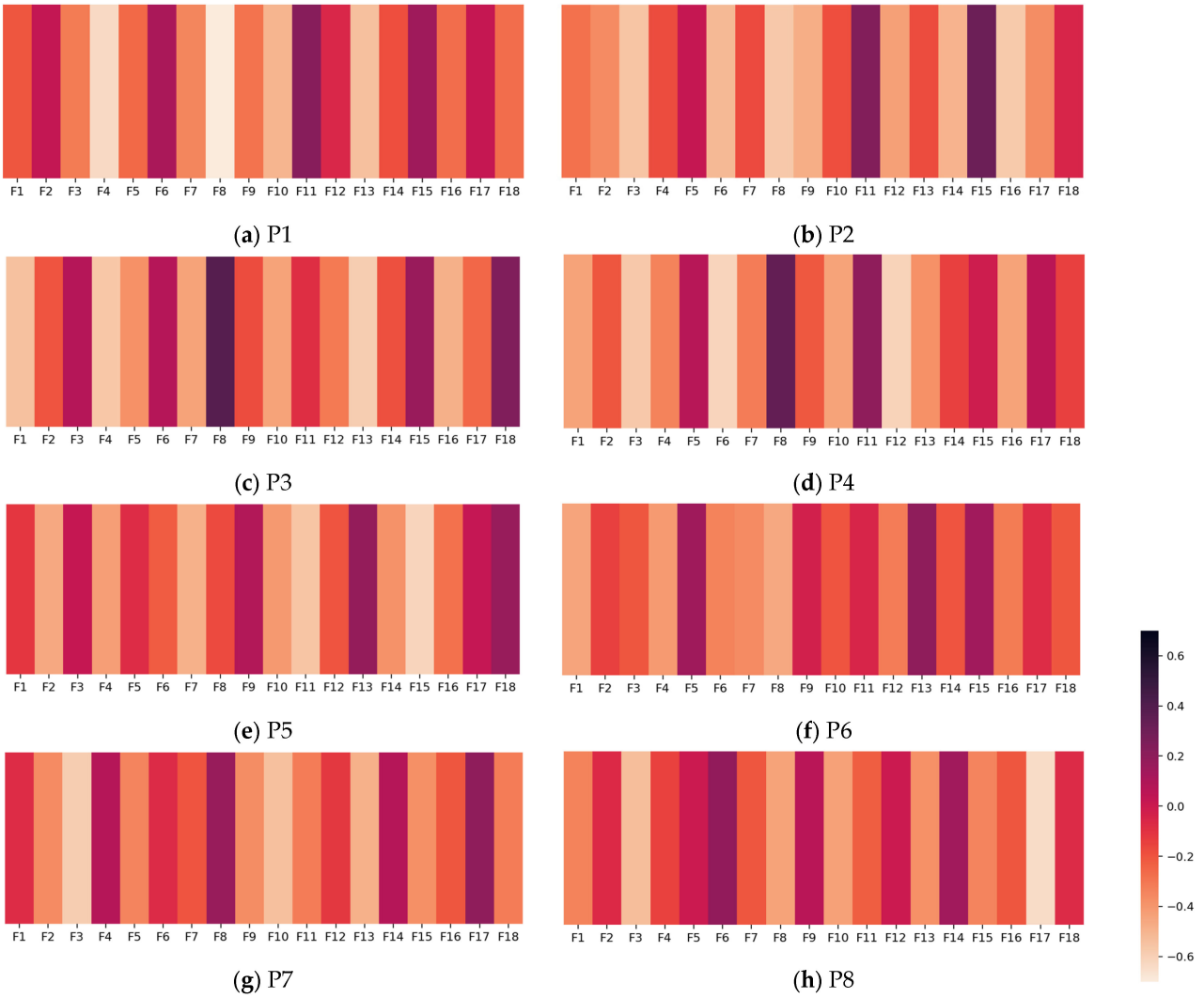

Figure 5. Effectiveness analysis. Feature weights of shallow linguistic features on each dataset. The brighter the color, the greater the influence on the final score.

In order to verify the influence of the sentence vectorization method on the final score during the extraction of sentence-scale local features, we used different sentence vectorization methods for the essay to the one proposed in this paper. The experimental results are shown in Figure 6, which shows that utilizing Avg. Glove Embedding to obtain

sentence vectors by obtaining word vectors and then averaging them did not yield excellent results. Since the SBERT-based model obtained excellent performance by pre-training in the STS task and by fine-tuning the mechanism, its result is substantially better than InferSent and Universal Sentence Encoder (USE). We analyzed the following four reasons that make MSSF work best:

- It generates contextually aware sentence embeddings. This allows SBERT to capture the contextual meaning of sentences and include word order and dependencies, rather than just averaging out the word vectors in the sentence, resulting in more accurate embeddings.
- It benefits from the large-scale pre-training of BERT models. It is typically pre-trained on vast amounts of text data, allowing it to learn a broad range of language patterns. This extensive pre-training contributes to the model's ability to generalize well across different tasks and domains.
- It allows fine-tuning on specific downstream tasks. This fine-tuning process adapts the pre-trained model to the specific task, resulting in improved performance. On the other hand, InferSent and USE are not designed for easy fine-tuning and lack this adaptability.
- It can be used to calculate the semantic similarity between sentences. The more accurate measure of similarity can be obtained by sentences embedding in a high-dimension and computing the cosine similarity between them. It can better understand the semantics of sentences and improve the capability of sentence vectorization.

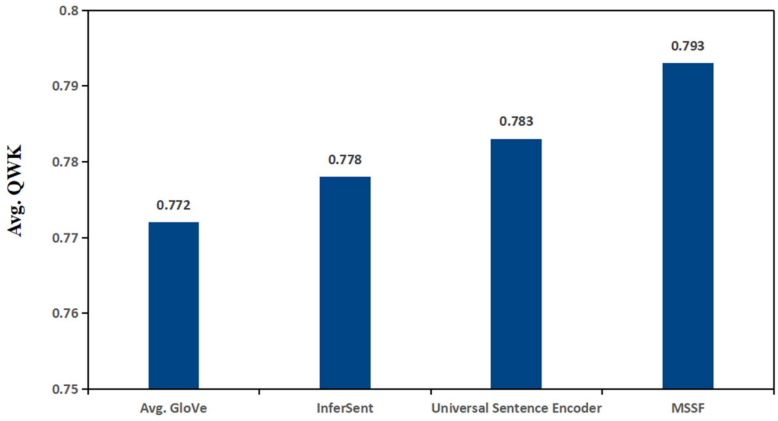

Figure 6. The influence of sentence vectorization on the final score.

4.6. Time Complexity

We have many different-scale features in our proposed method. In order to verify the efficiency of the method, we performed the time complexity analysis experiment to study its computational load.

In the document-scale feature, since the Glove dimension is smaller than the hidden layer dimension of the LSTM, the main time complexity of the model is produced by LSTM, which is $O\left(KE(D_1)^2\right)$ where K is the number of LSTM layers; meanwhile, E is the sequence length, which is the length of the essay, and D_1 is the number of the hidden layer units in LSTM.

In the sentence-scale feature, we utilize the pre-trained SBERT model; additionally, the number of hidden layer units is greater than the length of the input sequence. Its main time complexity is produced by SBERT, which is $\left(PLS(D_2)^2\right)$ where P is the number of sentences in the essay, L is the number of layers in SBERT, S is the length of the sentence, and D_2 is the hidden dimensions of the model in BERT.

In the prompt-related features, Doc2vec has been trained in advance and its time complexity is similar to the lookup table layer. The time complexity after cosine operation is $O(D_3)$ where D_3 is the dimension of the essay and the prompt vectorization.

In the shallow linguistic features, the main time complexity is $O(D_4 D_5)$ where D_4 is the input dimension that is the number of manual features and D_5 is the dimension of the output.

On the whole, since the size of D_2 is much larger than D_1, the sentence-scale feature has the largest time complexity and the document-scale features come next. The shallow linguistic features and prompt-related features are very small. In the sentence-scale features, BERT has a large count of parameters which has more layers and a big size of hidden layer nodes. Secondly, LSTM cannot be calculated in parallel and requires loop calculations of E sentences. Therefore, the extraction of sentence-level features requires a large time complexity. On the other hand, compared with other pre-trained models, our sentence-level feature extraction structure is already lighter.

It can be shown from the time complexity analysis that the time complexity of the shallow linguistic features and prompt-related features is very small and they are easily extended to the distributed representations of other DNN-AES models; therefore, they will achieve a great performance boost with little cost, which shows the effectiveness of these two parts features.

4.7. Threats to Validity

Our proposed method based on multi-scale features shows excellent results on the ASAP dataset. However, there are many potential limitations or threats to the validity of the research:

- It does not have high content validity, which is the degree to which it can cover the content areas it intends to measure. Specifically, AES systems are typically trained on the specific dataset of essays, which may not fully represent the entire scope of writing styles. Consequently, AES may struggle to assess essays that fall outside the scope of the training data, leading to potential biases and incomplete evaluations.
- It can be susceptible to language and cultural biases. The difference in language usage, dialects, or cultural references can impact the accuracy of AES, particularly for non-native English speakers or individuals from diverse linguistic backgrounds. In addition, scoring varies greatly between different languages and the model cannot be transferred well between different languages.
- It often relies on surface-level information, such as word, sentence, or grammar, which is important. However, it may not capture the deeper aspects of writing quality, such as coherence, organization, or argumentation.

5. Conclusions

In this paper, we propose an AES method based on multi-scale semantic features, which extracts essay features from multiple perspectives to address the complexity of essay scoring. We take document-scale as input to extract the global features of essays by the LSTM-MoT structure; we then utilize a hybrid neural network structure to extract sentence-level local semantic features after sentence vectorization of the essay via SBERT, so as to obtain deep semantic features of essays. For the shortage of DNN-AES to extract shallow linguistic features, we construct shallow features with the weight manually. We also add prompt relevance features to enhance the scoring effect of the model. These two-part features are easily extended to distributed representations of other AES models without significant time complexity. Finally, the final scoring is performed by semantic feature fusion. We conducted experiments on the Kaggle ASAP dataset. The results demonstrate that our proposed multi-scale AES method is highly effective in extracting diverse semantic features and outperforms the baseline model in terms of performance. We also conducted ablation experiments and a time complexity analysis to verify the effectiveness of our method. One of the limitations of the current approach is that the hand-crafted features

utilized in this paper are designed in advance and computed offline. Furthermore, the selection and construction of the shallow features is still a challenge to be faced in the future. In addition, due to differences in languages, methods via which to score the essay of different languages is one of the important research directions in the future. Due to the uninterpretable characteristic of DNN models, we cannot well know the potential error information in various scale features. We will conduct research and analysis about potential errors in the future and adjust the structures of the model according to them.

Author Contributions: Conceptualization, X.X.; methodology, F.L. and X.X.; formal analysis, F.L. and X.X.; investigation, F.L.; resources, F.L.; data curation, F.L.; writing—original draft preparation, F.L.; writing—review and editing, F.L., D.L. and W.Z.; supervision, X.X. and Z.C.; funding acquisition X.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been supported by the National Natural Science Foundation of China under grants 62176175; the Innovative Team of Jiangsu Province under grant XYDXX-086; the Science and Technology Development Project of Suzhou under grants SGC2021078.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: https://www.kaggle.com/c/asap-aes/data (accessed on 23 May 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hussein, M.A.; Hassan, H.; Nassef, M. Automated language essay scoring systems: A literature review. *PeerJ Comput. Sci.* **2019**, *5*, e208. [CrossRef] [PubMed]
2. Hua, C.; Wind, S.A. Exploring the psychometric properties of the mind-map scoring rubric. *Behaviormetrika* **2019**, *46*, 73–99. [CrossRef]
3. McNamara, D.S.; Louwerse, M.M.; Graesser, A.C. *Coh-Metrix: Automated Cohesion and Coherence Scores to Predict Text Readability and Facilitate Comprehension*; Technical Report; Institute for Intelligent Systems, University of Memphis: Memphis, TN, USA, 2002.
4. Landauer, T.K. Automatic essay assessment. *Assess. Educ. Princ. Policy Pract.* **2003**, *10*, 295–308. [CrossRef]
5. Ke, Z.; Ng, V. Automated Essay Scoring: A Survey of the State of the Art. *IJCAI* **2019**, *195*, 6300–6308.
6. Borade, J.G.; Netak, L.D. Automated grading of essays: A review. In Proceedings of the Intelligent Human Computer Interaction: 12th International Conference, IHCI 2020, Daegu, Republic of Korea, 24–26 November 2020; pp. 238–249.
7. Uto, M. A review of deep-neural automated essay scoring models. *Behaviormetrika* **2021**, *48*, 459–484. [CrossRef]
8. Cozma, M.; Butnaru, A.M.; Ionescu, R.T. Automated essay scoring with string kernels and word embeddings. *arXiv* **2018**, arXiv:1804.07954.
9. Butnaru, A.M.; Ionescu, R.T. From image to text classification: A novel approach based on clustering word embeddings. *Procedia Comput. Sci.* **2017**, *112*, 1783–1792. [CrossRef]
10. Dasgupta, T.; Naskar, A.; Dey, L.; Saha, R. Augmenting textual qualitative features in deep convolution recurrent neural network for automatic essay scoring. In Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications, Melbourne, Australia, 19 July 2018; pp. 93–102.
11. Page, E.B. Grading essays by computer: Progress report. In Proceedings of the Invitational Conference on Testing Problems, Princeton, NJ, USA, 28 October 1967.
12. Mathias, S.; Bhattacharyya, P. ASAP++: Enriching the ASAP automated essay grading dataset with essay attribute scores. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018.
13. Sakaguchi, K.; Heilman, M.; Madnani, N. Effective feature integration for automated short answer scoring. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, 31 May–5 June 2015; pp. 1049–1054.
14. Cummins, R.; Zhang, M.; Briscoe, T. Constrained multi-task learning for automated essay scoring. *Assoc. Comput. Linguist.* **2016**, *1*, 789–799. [CrossRef]
15. Nguyen, H.; Litman, D. Argument mining for improving the automated scoring of persuasive essays. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
16. Kim, Y. Convolutional Neural Networks for Sentence Classification. *arXiv* **2014**, arXiv:1408.5882.
17. Graves, A.; Graves, A. Long short-term memory. In *Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 37–45.

18. Dong, F.; Zhang, Y. Automatic features for essay scoring—An empirical study. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–4 November 2016; pp. 1072–1077.
19. Taghipour, K.; Ng, H.T. A neural approach to automated essay scoring. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–4 November 2016; pp. 1882–1891.
20. Nguyen, H.; Dery, L. Neural networks for automated essay grading. *CS224d Stanf. Rep.* **2016**, 1–11. Available online: https://cs224d.stanford.edu/reports/huyenn.pdf (accessed on 1 June 2023).
21. Dong, F.; Zhang, Y.; Yang, J. Attention-based recurrent convolutional neural network for automatic essay scoring. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 153–162.
22. Ridley, R.; He, L.; Dai, X.-y.; Huang, S.; Chen, J. Automated cross-prompt scoring of essay traits. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 2–9 February 2021; pp. 13745–13753.
23. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
24. Rodriguez, P.U.; Jafari, A.; Ormerod, C.M. Language models and automated essay scoring. *arXiv* **2019**, arXiv:1909.09482.
25. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
26. Wang, Y.; Wang, C.; Li, R.; Lin, H. On the use of bert for automated essay scoring: Joint learning of multi-scale essay representation. *arXiv* **2022**, arXiv:2205.0383.
27. Yang, R.; Cao, J.; Wen, Z.; Wu, Y.; He, X. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 1560–1569.
28. Farag, Y.; Yannakoudakis, H.; Briscoe, T. Neural automated essay scoring and coherence modeling for adversarially crafted input. *arXiv* **2018**, arXiv:1804.0689.
29. Liu, J.; Xu, Y.; Zhu, Y. Automated essay scoring based on two-stage learning. *arXiv* **2019**, arXiv:1901.0774.
30. Uto, M.; Aomi, I.; Tsutsumi, E.; Ueno, M. Integration of Prediction Scores from Various Automated Essay Scoring Models Using Item Response Theory. *IEEE Trans. Learn. Technol.* **2023**, 1–18. [CrossRef]
31. Alikaniotis, D.; Yannakoudakis, H.; Rei, M. Automatic text scoring using neural networks. *arXiv* **2016**, arXiv:1606.0428.
32. Wang, Y.; Wei, Z.; Zhou, Y.; Huang, X.-J. Automatic essay scoring incorporating rating schema via reinforcement learning. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 791–797.
33. Mesgar, M.; Strube, M. A neural local coherence model for text quality assessment. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 4328–4339.
34. Nadeem, F.; Nguyen, H.; Liu, Y.; Ostendorf, M. Automated essay scoring with discourse-aware neural models. In Proceedings of the Fourteenth Workshop on Innovative Use of Nlp for Building Educational Applications, Florence, Italy, 2 August 2019; pp. 484–493.
35. Uto, M.; Okano, M. Robust neural automated essay scoring using item response theory. In Proceedings of the Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, 6–10 July 2020; Proceedings, Part I 21. pp. 549–561.
36. Mim, F.S.; Inoue, N.; Reisert, P.; Ouchi, H.; Inui, K. Unsupervised learning of discourse-aware text representation for essay scoring. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, Florence, Italy, 28 July–2 August 2019; pp. 378–385.
37. Uto, M.; Xie, Y.; Ueno, M. Neural automated essay scoring incorporating handcrafted features. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 6077–6088.
38. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084.
39. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
40. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1188–1196.
41. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
42. Phandi, P.; Chai, K.M.A.; Ng, H.T. Flexible domain adaptation for automated essay scoring using correlated linear regression. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 431–439.
43. Tay, Y.; Phan, M.; Tuan, L.A.; Hui, S.C. Skipflow: Incorporating neural coherence features for end-to-end automatic text scoring. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.