**4.0 Co-Attention Based Neural Network for Source-Dependent Essay Scoring**

**4.1 Introduction**

Because neural network models show a stronger ability to model text than feature based  
models, researchers introduced neural network models into this area. However, other models  
focus on grading essays in a general and universal way, which means the model do es not  
optimize for any single type of writing task. However, different types of writing tasks have  
their own characteristics. A one size fits all model always have a shortage. For example, the  
source-dependent essay scoring, the source article should be an essential external knowledge  
when grading the essay.  
In this chapter, we present an investigation of using a co-attention based neural network  
for source-dependent essay scoring. We use a co-attention mechanism to help the model learn  
the importance of each part of the essay more accurately. Also, this work shows that the co-attention based neural network model provides reliable score prediction of source-dependent responses. We evaluate our model on two source-dependent response corpora. Results show  
that our model outperforms the baseline on both corpora. We also show that the attention  
of the model is similar to the expert opinions with examples. Besides, we us e examples  
to show that our model can assign reasonable attention scores to different sentences in the  
essay. This work is illustrated in [112].

**4.2 Related Work**

Previous research in AES including our approach from the prior chapter needed feature  
engineering. In very early work, Page [73] developed an AES tool named Project Essay Grade (PEG) by only using linguistic surface features. A more recent well-known AES system is E-Rater [15], which employs many more natural language processing (NLP) technologies. Later, Attali and Burstein [4] released E-Rater V2, where they created a new set of features to represent linguistic characteristic related to organization and development, lexical complexity, prompt-specific vocabulary usage, etc. Similarly to [73], this system used regression equations for assessment of student essays. Such systems are also used to assess responses to English tests, for example, the Graduate Record Exam (GRE) and the Test of English as a Foreign Language(TOEFL). One limitation of all of the above models is that all need handcrafted features for training the model. In contrast, our model uses a neural network for the AES task and thus does not require feature engineering. Recently, neural network models have been introduced into AES, making the development of handcrafted features unnecessary or at least optional. Alikaniotis et al. [2] and Taghipour and Ng [96] presented AES models that used Long Short Term Memory (LSTM) networks. Differently, Dong and Zhang [33] used a Convolutional Neural Network (CNN) model for essay scoring by applying two CNN layers on both the word level and then sentence level. Later, Dong and Zhang [34] presented another work that uses attention pooling to replace the mean over time pooling after the convolutional layer in both word level and sentence levels. However, none of these neural network grading models consider the source article if it exists. In this chapter, we introduce a neural network model that takes the source article into account by using a co-attention mechanism instead of the self-attention mechanism of prior work. Although some models reached better performance on the ASAP corpus [97, 69], the same, they do not take the source article into account if it exists. Besides, those models use more complicated network structures, while our model is relatively simple. Our work not only focuses on essay assessment using a holistic score, but also evaluates a particular dimension of argument-oriented writing skills, namely use of Evidence. Louis and Higgins [58] analyze only the content of essays by detecting off-topic essays. Ong et al. [71] used argumentation mining techniques to evaluate if students use enough evidence to support their positions. However, these two prior studies are not suitable for our task because they did not measure the use of content or evidence from a source article. With respect to source-based dimensional essay analysis, Rahimi et al. [83] developed a set of rubric-based features that compared a student’s essay and a source article in terms of number of related words or paraphrases. Zhang and Litman [111] improved their model by introducing word embedding into the feature extraction process to extract relationships previously missed due to lexical errors or use of different vocabulary. However, in both of these studies, human effort was still necessary for pre-processing the source article, for example, by having experts manually create a list of important words and phrases in the article which the system would compare with features extracted from the student’s essay. In contrast, our work does not need any human effort to analyze the source article before essay grading. Although [112] investigated extracting example lists by using  
LDA [12] model, the data-driven model missed an example when there was no essay mentioning the example.

**4.3 Model**

Our network is inspired by the hierarchical neural network model [34]. In their model, they considered each essay as a sequence of sentences rather than a sequence of words. Their model has three parts. First, they used a convolutional layer and attention pooling layer to get sentence representation. Second, they used an LSTM layer and another attention pooling layer for document representation. Finally, they used a sigmoid layer for score prediction.  
Differently from their model, our model replaces the attention pooling layer for document  
representation with a bi-directional attention flow layer and an additional modeling layer  
[89]. By doing so, our model considers students’ essays associated with a source  
article and this attention mechanism captures the relationship between the essay and the  
source article. In particular, a higher attention score will be assigned to sentences that are  
mentioned in the article but less mentioned in other essays. Our model is a hierarchical  
neural network and consists of seven layers. Figure 5 shows the structure of our network.  
The layers in the dashed box were presented by [34]. The sentence level 3

Score

Esaay2Article

Dense + Sigmoid

Softmax

Output Layer

Modeling Layer

Sentence Level Co-Attention Layer

Sentence Level LSTM Layer

Word Level Convolutional Layer

Essay Article

and

Article2Essay

attentions

Article2Essay

Word level Attention Pooling Layer

Max

softmax

Word Embedding Layer

Source Article

Essay

Figure 5: The co-attention based neural network structure

co-attention layer was presented by [89].

**4.3.1 Word Embedding Layer**

This layer maps each word in sentences to a high dimension vector. We use the GloVe pre-trained word embeddings [74] to obtain the word embedding vector for each word. It was trained on 6 billion words from Wikipedia 2014 and Gigaword 5. It has 400,000 uncased vocabulary items. The dimensionality of GloVe in our model is 50 dimensions. The outputs of this layer are two matrices,for the essay and for the article, where *, , ,*  and are number of sentences of the essay and the article, length of sentences of the essay and the article, and the embedding size, respectively. A dropout is applied after the word embedding layer [34].

**4.3.2 Word Level Convolutional Layer**

In this layer, we perform 1D convolution over the word representations of both and  
, so that we can get local representation of each sentence. For each word in each  
sentence, we perform 1D convolution:

(1)

where *g* is a nonlinear activation, *k* is the kernel size, is the filter weight matrix, and  
is the bias vector. The outputs of this layer are  *∈* for the essay and  
 *∈* for the article, where and are filtered lengths of sentences of the essay  
and the article, respectively. is the number of filters of the 1D convolution layer.

**4.3.3 Word Level Attention Pooling Layer**

After the convolutional layer, a pooling layer is demanded to obtain the sentence representations. In this layer, we follow the same design presented by [34]. The  
attention pooling is defined as equations below:  
= (2)  
  
=   
*s* =

where , and are weight matrix, vector, and bias vector, respectively. and are  
attention vector and attention weight for. The outputs of this layer are  *∈*  for  
the essay and  *∈* for the article.

**4.3.4 Sentence Level LSTM Layer**

In this layer, we use a Long Short-Term Memory Network (LSTM) [44] over the sentence representations of the essay and the article to capture contextual evidence from previous sentences to refine the sentence representation. The LSTM unit is a special kind of RNN unit which has long-term dependency learning ability. LSTMs use three gates to control information flow to avoid the long-term dependency problem by forgetting or remembering information in each LSTM unit. They are an input gate, a forget gate, and an output gate. The following equations define the LSTM unit:

(5)

(6)

(7)

(8)

(9)

(10)

where and are the input sentence and the output state of time , respectively. , , and are weight matrices. , and are bias vectors. is the sigmoid function,  
and ∗is element-wise multiplication. The output of this layer are  *∈*  for the essay  
and  *∈* for the article, where is the dimensionality of the output.

**4.3.5 Sentence Level Co-Attention Layer**

The concept of this layer is presented by [89] in the part of attention flow layer. This layer links information from and , and generates a collection of article aware features vector of essay sentences. The attention is computed in two directions, from essay to article, and vice versa. Both attention scores are figured from a similarity matrix by the following equation:

(11)

where is weight matrix, and are row vector of and row vector of  
, is bias vector. ∗is element-wise multiplication. [;] is vector concatenation. After  
obtaining the similarity matrix  *∈* , we compute the attention in two directions.  
**Essay to Article Attention** measures which sentences in the article are similar to each  
sentence in students’ essays. The following equations define the essay to article attention:

(12)

(13)

where  *∈* represents the attention score of each sentence in the article associate  
with each sentence in the essay, is performed across each row. The output of this  
*∈ .*  
**Article to Essay Attention** measures which sentences in the essay have the closest  
meaning to one of the sentences in the article. The following equations define the article to  
essay attention:

(14)

(15)

where *∈* , is a maximum function performed across the column, and  *∈* .  
Because will find out which sentence in the article has the closest meaning to each  
sentence in the essay, so represents the attention score of the most important sentence in  
the essay associated with the article. After tiling times, the final output of this layer is  
 *∈* . The final output is a concatenated matrix of , , and defined by:

= [; ;  *∗* ; *∗* ] (16)

where *∗* is element-wise multiplication, and [;] is concatenation, *He* is the original representation of essay, is the essay to article attention, *∗* is the self-aware representation, and *∗*is article-aware representation. Therefore, the output of this layer is *∈* , the article-aware representation of each sentence in the essay.

**4.3.6 Modeling Layer**

is the representation of each sentence, and we need the representation of the essay.  
Therefore, we introduce another LSTM layer for modeling the essay and only use the output  
of the final LSTM unit as the output of this layer *∈* , where is the dimensionality  
of the output of LSTM units.

**4.3.7 Output Layer**

After obtaining the essay representation , a linear layer with sigmoid activation will  
predict the final output. The following equation defines the output layer:

(17)

where is weight vector, and is bias vector. is the final predicted score of the essay.

**4.4 Training**

**Loss.** [34] used mean squared error (MSE) loss, thus we use the same  
loss function. MSE evaluates the average of squared error between the predicted score and  
the gold standard. Thus it is widely used in regression tasks. The following equation defines

(18)

where is the predicted score, is the gold standard, it the total number of samples.  
**Optimization.** The optimizer we use is RMSprop [27]. The initial  
learning rate is 0.001, momentum is 0.9, and Dropout rate is 0.5 for preventing overfitting.  
These setting are the same as used by [34].

**4.5 Experimental Setup**

We configure experiments to test three hypotheses:  
H1: the model we proposed (denoted by CO-ATTN) will outperform or at least perform  
equally well as the baseline (denoted by SELF-ATTN) [34] on four ASAP essay corpora in the holistic score prediction task.  
H2: the model we proposed will outperform or at least perform equally well as the  
baseline on two RTA corpora in the Evidence score prediction task.  
H3: the model we proposed will outperform or at least perform equally well as the  
non-neural network baselines on both corpora.  
We use NLTK [30] for text preprocessing. The vocabulary size of the data  
is limited to 4000, and all scores are scaled to the range [0, 1], following [96] and [34]. In articular, the 4000 most frequent words are preserved, with all other words treated as unknowns. The assessment scores will be converted back to their original range during evaluation. We use Quadratic Weighted Kappa (QWK) to evaluate our  
model. QWK is not only the official criteria of ASAP corpus, but also adopted as evaluation  
metric in [83] for both ASAP and RTA corpora.  
We use 5-fold cross-validation because both RTA and ASAP corpora have no released  
labeled test data. We split all corpora into 5 folds. For the ASAP corpus, the partition is  
the same as the setting presented by [96]. For the RTA corpus, since  
there is no prior work to split the corpus, we separate it into 5 folds randomly. In each fold,  
60% of the data are used for training, 20% of the data are the development set, and 20% of  
the data are used for testing.  
To select the best model, we trained each model on 100 epochs and evaluated on the  
development set after each epoch. The best model is the model with the best QWK on  
the development set. This is done five times, once for each partition in the cross-validation.  
Then the average QWK score from these five evaluations on the test set is reported. Paired  
t-tests are used for significance tests with . Table 10 shows all hyper-parameters for  
training.  
The code of SELF-ATTN are provided by [34], they used Keras [25] 1.1.1 and Theano [25] 0.8.2 as the backend. Because we are using Keras 2.1.3 and TensorFlow [1] 1.4.0 as the backend, we ran all experiments with our frameworks. Therefore, the numbers of SELF-ATTN have small differences to the numbers reported by the baseline model. For non-neural network baselines, we introduce the SVR and BLRR baselines [79] for the ASAP corpus, and SG baseline [40] for the RTA corpus. SVR and BLRR models use Enhanced AI Scoring Engine (EASE) 1 to extract four types  
of features, such as length, part of speech, prompt, and the bag of words. Then they use  
SVR and BLRR as the classifiers, respectively. We do not perform any significance test on  
both SVR and BLRR because we do not have detailed experiment data. Therefore, we only  
report the result presented in [41]. SG model extracts evidence features based on hand-crafted topic and example lists, and uses random forest tree as the classifier. We follow the same data partition. However, we only use the training set for training and the testing set for testing while ignoring the development set so that we can perform the same paired t-tests in the experiments.

|  |  |  |
| --- | --- | --- |
| **Layer** | **Parameter Name** | **Value** |
| Embedding | Embedding dimension | 50 |
| Word-CNN | Kernel size Number of filters | 5  100 |
| Sent-LSTM | Hidden units | 100 |
| Modeling | Hidden units | 100 |
| Dropout | Dropout rate | 0.5 |
| Others | Epochs  Batch size  Initial learning rate  Momentum | 100  100  0.001  0.9 |

Table 10: Hyper-parameters of training.

**4.6 Results and Discussion**

**Results for H1.** The results shown in Table 11 support this hypothesis. The COATTN model yields higher performance than the SELF-ATTN model on all ASAP prompts.  
However, the CO-ATTN model only significantly outperforms the SELF-ATTN model on  
Prompt 3.  
**Results for H2.** Again, the results shown in Table 11 support this hypothesis. The  
CO-ATTN model yields higher performance than the SELF-ATTN model, significantly on  
both of the RTA corpora.  
**Results for H3.** The results shown in Table 11 still support this hypothesis. The  
CO-ATTN model yields higher performance than all non-neural network baselines.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Prompts | SVR | BLRR | SG | SELF-ATTN | CO-ATTN |
| MVP | NA  NA  0.630  0.749  0.782  0.771 | NA  NA  0.621  0.784  0.784  0.775 | 0.653  0.632  NA  NA  NA  NA | 0.681  0.669  0.677  0.807  0.806  0.809 | **0.809**  **0.815**  **0.812** |

Table 11: The performance (QWK) of the baselines and our model. *∗* indicates that the  
model QWK is significantly better than the SELF-ATTN (*p <* 0*.*05). *†* indicates that the  
model QWK is significantly better than the SG (*p <* 0*.*05). The best results in each row  
are in bold.  
The results show that in our tasks, the neural network approaches are better than nonneural network baselines. One possible reason is the final representation of the essay from neural network contains more information. However, some of the information might be ignored by hand-crafted features. For example, the importance of different evidence in RTA task is not considered in the SG model. It treats all evidence equally. However, the neural network models capture this information automatically. Apparently, the CO-ATTN model performs better in the RTA tasks, because it always significantly outperforms the SELF-ATTN model. One possible reason is that the RTA task only considers the Evidence score. The CO-ATTN model is more suitable for the Evidence score prediction task because it can find pieces of evidence that appear in both students’ essays and the source article better. In contrast, the SELF-ATTN model only considers students’ essays associated with the scores. In this case, if a piece of evidence is  
not mentioned by students, this data-driven model cannot distinguish it. Consequently, some  
important pieces of evidence will be assigned to a lower weight. However, the CO-ATTN  
model considers not only the students’ essays but also the source article. In other words,  
if an important piece of evidence is not mentioned by too many students, but it is in the  
source article, the CO-ATTN model will assign this sentence higher attention.  
In the ASAP holistic score prediction task, although we still see a benefit in using the  
CO-ATTN model, it is reduced. In this case, the benefit we saw in the Evidence dimension  
from the CO-ATTN model becomes less significant because the model also needs to consider  
more aspects of the essay, such as organization, grammar mistakes, and so on. Our results  
suggest that the co-attention mechanism of the CO-ATTN model cannot capture these aspects significantly better than the SELF-ATTN model. Therefore, the CO-ATTN model only significantly outperforms the SELF-ATTN model on Prompt 3. In Table 12, we list 10 sentences from student *MV P* essays and their associated attention scores. Because we have a list of examples manually extracted by our experts as important evidence from the *MV P* source article, examining RTA data helps us understand the attention score assigned by our model. Bolded are examples extracted by the expert from the source article that the student includes in the essay. A lower attention score means this sentence is less important. Otherwise, the score is high. As we can see, sentences 1, 2, 3, and 4 are low attention sentences, sentences 5, 6, and 7 are mid attention sentences, and sentences 8, 9, and 10 are high attention sentences. The attention scores reflect the importance of these sentences accurately. Sentence 1 is a short and general sentence related to the source article, but it has no specific evidence from it. Sentence 2 even has no content related to the source article. Sentence 3 has many details related to the source article. However, it still has no evidence directly from the source article. Sentence 4 mentions “*The author did convince me that winning the fight against poverty is achievable in our lifetime*” which comes from both the prompt and the source article, but this statement is so general that almost every student mentions this statement in the essay which makes this statement not distinguishable. For these reasons, these four sentences receive low attention scores. Although sentence 5 is short, it mentions one piece of evidence. Sentence 6 talks about farming which is a topic from the source article. In the article, the things listed in this sentence are things the farmer needs to worry about. However, this sentence indicates “*the farmer doesn’t have to worry*” because of the MVP project. Sentence 7 also mentions conditions of hospitals nowadays. However, it mentions not only water but also electricity which is more than Sentence 5. For these reasons, these three sentences receive mid attention scores from low to high. The last three sentences receive high attention scores because they all use more pieces of evidence directly from the source article. Sentence 8 talks ab out the school, and Sentence 9 talks ab out the hospital. Sentence 10 talks ab out farming. However, sentence 10 receives the highest attention score, because it mentions evidence from both before and after the MVP project. From these sentences, we can also see that the attention score dep ends on neither the length of the sentence nor only the specificity of the sentence. It instead dep ends on how many important pieces of evidence there are in the sentence. For example, Sentence 3 is long and talks ab out some details of our modern life. Although it also talks ab out quality materials or better housing and clothing compared to people living in Kenya, it receives a  
low attention score because there is no specific evidence directly from the source article. In  
contrast, Sentence 9 is shorter than Sentence 3. However, it receives a higher attention score  
because it mentions many pieces of evidence from the source article.  
Overall, the CO-ATTN model seems to capture the importance of sentences by assigning  
reasonable attention scores based on the relevance of the sentence to the source article.

**4.7 Conclusion**

In this work, we presented a co-attention-based neural network model that outperforms a state-of-the-art attention-based neural network model for essay scoring, not only for RTA evidence assessment but also for holistic assessment of ASAP source-dependent responses. The advantages of our model are that it does not need any expert preprocessing of the source article; the input of this model is only the raw student essay and its source article. Moreover, our model somewhat captures the importance of different pieces of evidence, although it is not specifically designed for this purpose. However, quantitative experiments that can answer whether the attention scores are correlated with the importance of different pieces of evidence need to be done. Also, this leads to an interesting future investigation — the development of a neural network approach that both achieves an acceptable score prediction and can simultaneously generate evidence lists from the source article.

In Chapter 6, we are going to discuss a model that uses the intermediate output of the co-attention-based neural network for extracting Topical Components. Besides, this model only works for source-dependent essay scoring. Although it achieves better performance in this specific area, it cannot work for other situations where the source article does not exist. Therefore, in Chapter 5, we propose a hybrid model that combines a simpler neural network model and hand-crafted features to make the model work in more situations.

|  |  |  |
| --- | --- | --- |
| **No**. | **Sentences** | **Attention** |
| 1 | Life in Kenya is hard | 0.00173 |
| 2 | In this essay I will give my top 3 reasons why. | 0.00174 |
| 3 | Because like I said, we have more advanced & better & more qualified materials than them, and these days kids & adults are spoiled, we have phones stores, houses & even shoes and clothes. | 0.00243 |
| 4 | The author did **convince me that winning the fight against poverty is achievable in our lifetime** because she showed me how many people in Sauri, Kenya need our help against poverty | 0.00229 |
| 5 | **Water is connected to the hospitals.** | 0.07746 |
| 6 | Also, there were **no school fees**, and **the school now serves lunch** for the students because they **didn’t have any midday meals** to provide them with **energy they need** to help them with the rest of their days. | 0.19483 |
| 7 | The hospital also has water and electricity. | 0.07746 |
| 8 | Also, there were **no school fees**, and **the school now serves lunch** for the students because they **didn’t have any midday meals** to provide them with **energy they need** to help them with the rest of their days. | 0.19483 |
| 9 | In 2008 though, when they **checked for progress**, **the hospital had medicine**, **free of charge**, with **running water and electricity.** | 0.20177 |
| 10 | Also **farmers could not afford fertilizer and irrigation** but now **they placed irrigation** and have them **fertilizer for the crops**. | 0.25855 |

**5.0 Attention Based Neural Network for Automated Essay Scoring with Hand-crafted Features**

**5.1 Introduction**

In Chapter 4, we presented a co-attention neural network for grading source-based essays. Although it outperforms its baselines, the design of the neural network limited the usage scenario, because a source article is required for learning hand-crafted features from it.

In this chapter, we propose a hybrid model that builds on an attention-based neural network model for AES [42], in order to be able to combine hand-crafted features on the sentence and on the word level as well as on the essay level. While enabling the use of hand-crafted features as a side input, our approach offers the neural network the ability to model the hand-crafted features. We hypothesize that the strong modeling ability of neural networks will be able to learn useful knowledge from the hand-crafted features. Within-prompt experiments show that our proposed hybrid model outperforms a neural baseline model, supporting our hypothesis. We also conduct cross-prompt experiments to show the usefulness of our model in a more difficult scenario typical of classroom AES usage.

**5.2 Related Work**

Historically, most AES research has used feature-based models [106, 37], which typically require carefully designed hand-crafted features for essay representation and off-the-shelf learning algorithms for model training. Surprisingly, even though neural network models currently dominate most natural language processing research areas, feature-based models still have a role to play in the AES community. For example, the model of Cozma et al. [51] combines bag-of-super-word-embeddings [52] with a string kernel and outperforms most neural network models. Nevertheless, much AES research now uses neural network models because they generally demonstrate state-of-the-art performance compared to feature-based models [96, 2, 33, 34, 97, 79, 45, 112, 69]. The most significant difference between either existing feature-based or neural models and our model is that we propose a hybrid model combining a neural network with hand-crafted features. In our hybrid, rather than playing the leading role in training, the hand-crafted features provide guidance for training the neural network model.

With respect to other hybrid approaches combining a neural network model with hand crafted features, the model of Liu et al. [57], and Uto et al. [100] provide state-of-the-are performance on the Automated Student Assessment Prize (ASAP) corpus. However, all combined features are essay-level features (e.g., the vocabulary size of the essay), which means the model only concatenates all hand-crafted features with highly abstracted essay-level information. In contrast, our model provides the possibility to combine hand-crafted features from a smaller linguistic unit, such as sentence level or word level. For example, a sentence-level feature could be the topic distribution of each sentence, and a word-level feature could be the POS tag of each word. In addition, our model takes lower level input as a sequence and models the sequence further. Dasgupta et al. [30] presented a hybrid model that uses LSTM Layer to model word-level feature sequence and combines pooling layer output on the essay level. This model is more similar to our model, but the same, our model provides flexibility to incorporate hand-crafted features from all linguistic units. Besides, we use a more complex way to combine hand-crafted features, which is an attention layer. This layer potentially provides the model with a stronger learning ability. Especially for low level hand-crafted features, the attention layer could distinguish the part of the sequence which is more important than others.

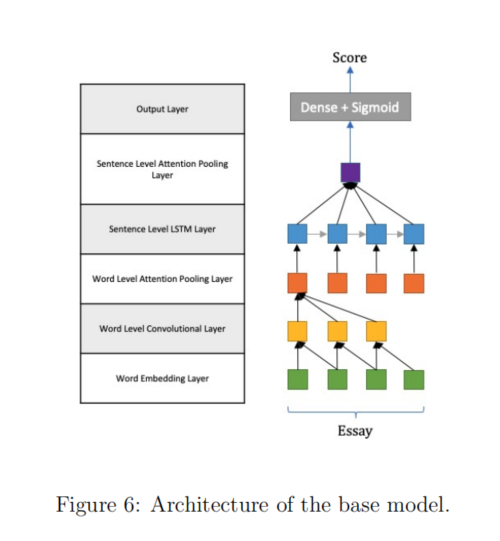
Beyond AES, Ko et al., [53] present a hybrid model for specificity prediction. However, their model still just concatenates hand-crafted features and a neural network without encoding the hand-crafted features. In contrast, Chen et al., [22] propose a hybrid model for automated speech scoring which has similarities with our model. In particular, they use a pure neural network model to encode the lexical aspect and a hybrid model to model acoustic cues. However, while they use a neural network to encode handcrafted features, they only concatenate the outputs of both neural networks. Consequently, the final prediction model treats features from both sides equally. In contrast, our hybrid model uses an attention layer to combine two models. Our model focuses on the pure neural network output and uses hand-crafted features as side input. Also, our model is a hierarchical model, which provides the possibility to combine hand-crafted features at different levels.

Recently, the natural language processing research community has been dominated by deep transformer architectures such as BERT [32]. However, such a deep and complex model might not be suitable for AES tasks, because AES tasks typically have relatively small amounts of training data [63]. Considering the costinefficiency of using such complex architectures, AES tasks often tend to prioritize simpler models. Therefore, in this work, we also start from a relatively simple neural model [34]. The performance of this attention-based, hierarchical neural network model on the ASAP corpus is 0.760, while the result of a more complex state-of-the-art neural network model is 0.773 [57]. Given the performance similarity, we build on the simpler neural model to demonstrate the utility of our hybrid approach.

**5.3 Base Model**

The main contribution of this work is to test a new hybrid neural network model, which can learn from both student essays and hand-crafted features. We mainly focus on exploring what hand-crafted features can be combined with the neural network, and how they can be combined. Therefore, we need to select a base neural network model and combine hand crafted features with it. Since the state-of-the-art deep transformer architectures are not as helpful for the AES task as for other tasks in the NLP area [63], we use a relatively simple model as our base model.

The base neural model [34] is a hierarchical neural network. In this model, each essay is considered to be a sequence of sentences rather than a sequence of words. Coherence between words and sentences can thus be learned in two steps, rather than in only one mixture step. The model uses a CNN layer [54] and a self-attention layer for word-level modeling to get sentence representation, and an LSTM layer [44] and another self-attention layer for sentence-level modeling. Figure 6 shows the architecture of the base model.



**Word Embedding Layer.** This layer maps each word in sentences to a high dimension vector. Currently, we are using Glove pre-trained word embeddings [74] to obtain the word embedding vector for each word. Following the same setting from previous work, the pre-trained embedding in the network was trained on 6 billion words from Wikipedia 2014 and Gigaword 5. It has 400,000 uncased vocabulary items, and the dimensionality of the GloVe model is 50 dimensions. As in [34], a dropout layer is applied after the word embedding layer to prevent the neural network from overfitting [94].

**Word Level Convolutional Layer.** This layer performs 1D convolution over the word representation. The output of this layer is the local representation of each sentence.

**Word Level Attention Pooling Layer.** This pooling layer is applied over the convolutional layer and is designed to obtain the sentence representation by calculating the weighted sum of each sliding window. The output of this layer is the sentence representations of each essay.

**Sentence Level LSTM Layer.** We apply a Long Short-Term Memory Network (LSTM) [44] over the sentence representations to capture contextual evidence from previous sentences to refine the sentence representation.

**Sentence Level Attention Pooling Layer.** Same as the Word Level Attention Pooling Layer, this pooling layer is applied over the LSTM layer and is designed to capture the essay representation by calculating the weighted sum of each sentence. The output of this layer is the essay representation, which will be passed to the final output layer.

**Output Layer.** After obtaining the essay representation, a linear layer with sigmoid activation predicts the final output. Note that the model treats AES as a regression problem. This setting provides the flexibility to grade essays with continuous or discrete scores, and with different score ranges.

**Loss Function.** Mean Squared Error (MSE) loss is the loss function. The MSE evaluates the average of squared error between the predicted score and the gold standard. Therefore, it is widely used in regression tasks.

**Optimization.** The optimizer of the base model is RMSprop [31]. Following Dong et al., [34], the initial learning rate is 0.001, momentum is 0.9. The dropout rate is 0.5.

**5.4 Proposed Hybrid Model**

We extend the base model from a pure neural network to a hybrid model that learns from both student essays and hand-crafted features. In this section, we introduce the hybrid model and the hand-crafted features to be tested.

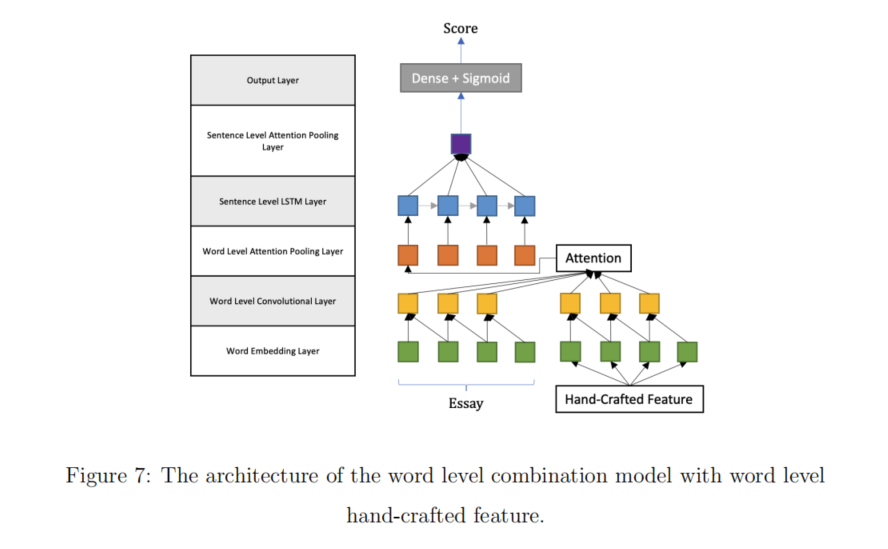
**5.4.1 Combination Models**

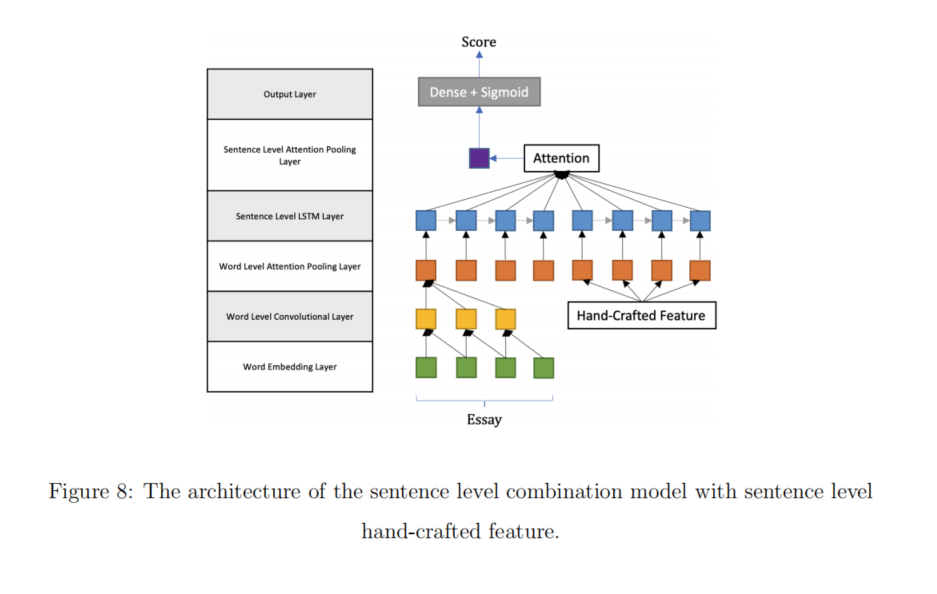
Different hand-crafted features could be extracted from different linguistic levels, such as word-level, sentence-level, and essay-level. For example, a word-level feature could be the POS tag of each word, a sentence-level feature could be the length of each sentence, and an essay-level feature could be the topic distribution of the essay.

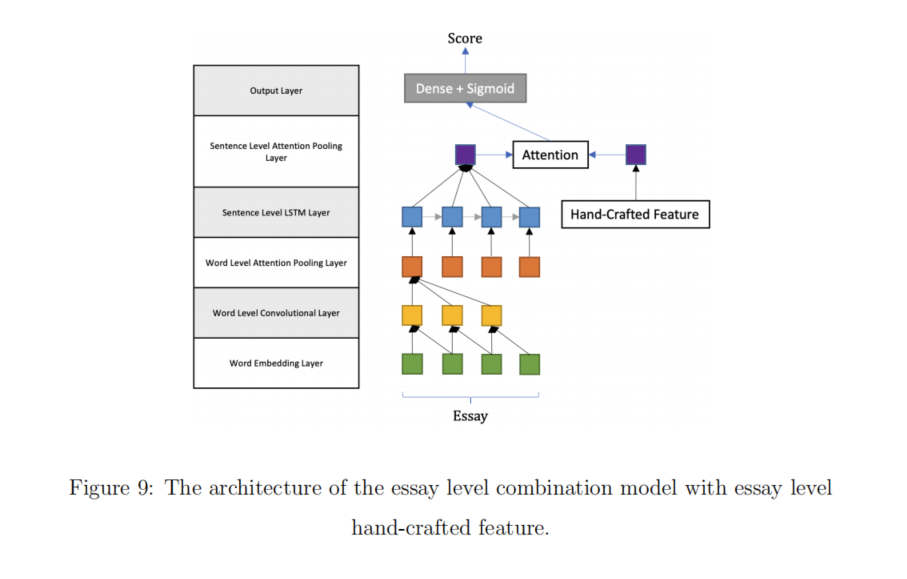
Depending on what hand-crafted features to combine, we might have to combine them on the same or a higher model level, such as sentence-level or even essay-level. In the combination model, we use an attention mechanism [6] to learn the relation between essays and their hand-crafted features. The attention mechanism calculates a feature-aware essay representation.

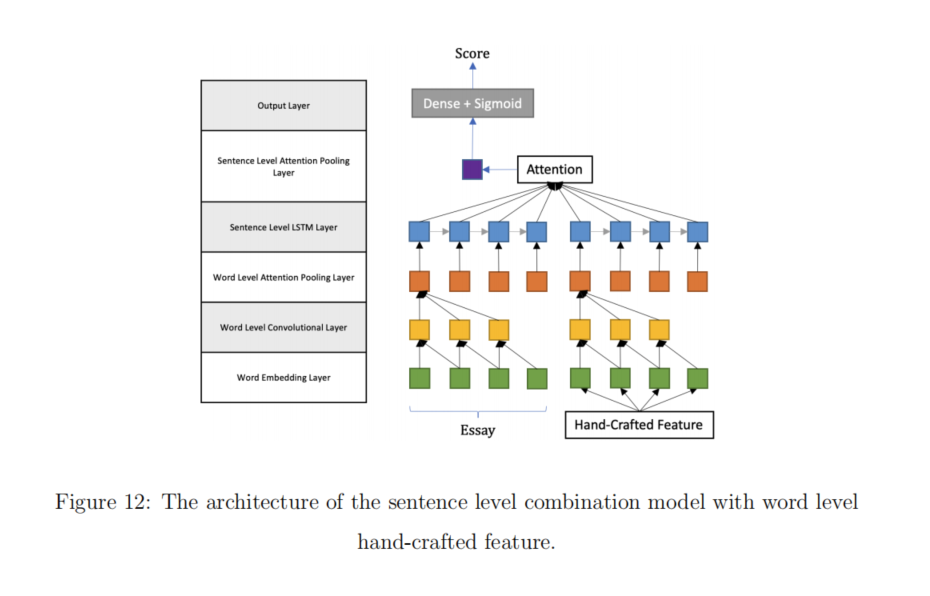
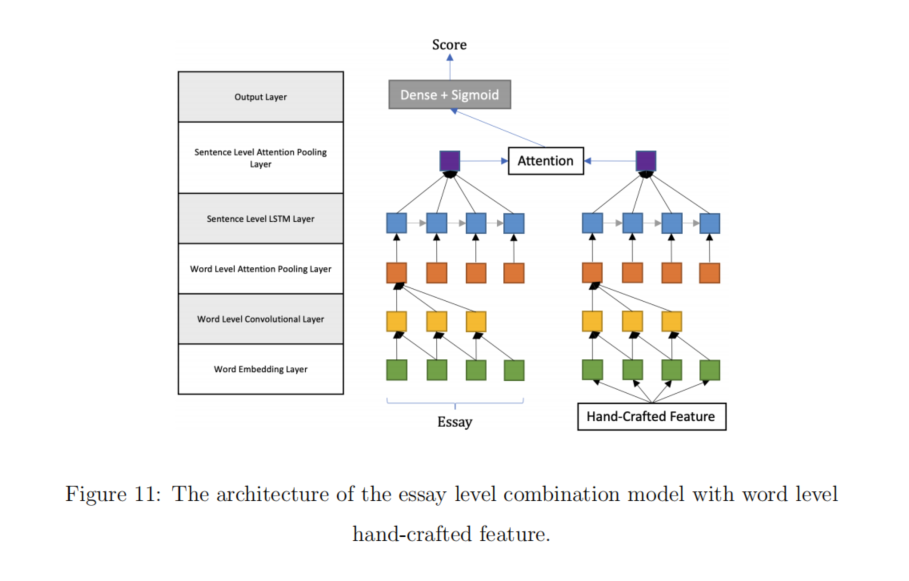
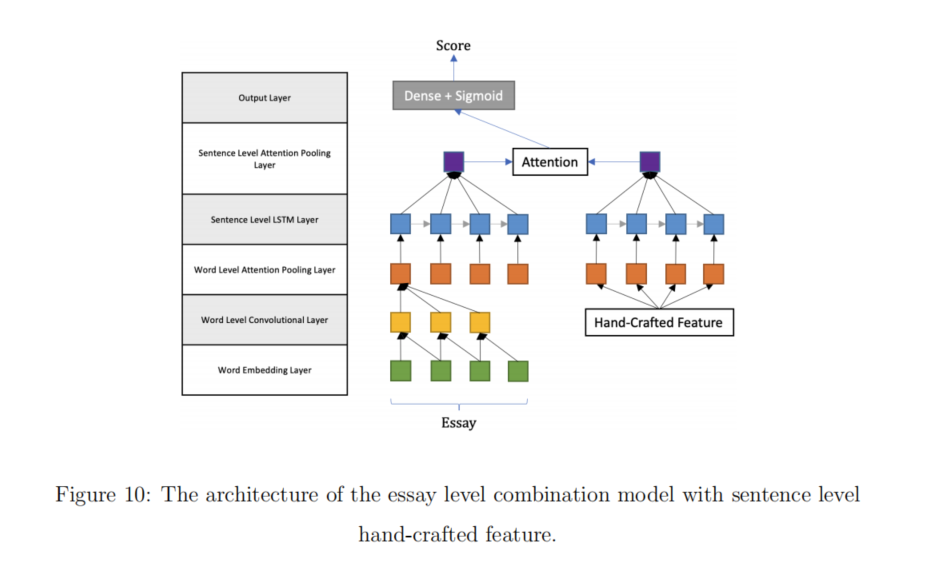
Figure 7 and Figure 8 show the architecture of the word-level combination model with a word-level hand-crafted feature, and the architecture of the sentence-level combination model with a sentence-level hand-crafted feature, respectively. In Figure 7, the combination model combines essay-side representation and hand-crafted feature representation before the word level attention pooling layer, while the combination model in Figure 8 combines essay-side representation and hand-crafted feature representation before the sentence level attention pooling layer. By combining the hand-crafted features at different levels, we preserve the information from hand-crafted features from different abstract levels. However, a handcrafted feature can only be combined on the same or a higher level. For example, an essaylevel feature can be combined on the model essay level, but not the sentence level, because the architecture of the base model does not allow this unpacking operation. Figure 9, Figure 10, Figure 11, and Figure 12 show all other possible model architectures.

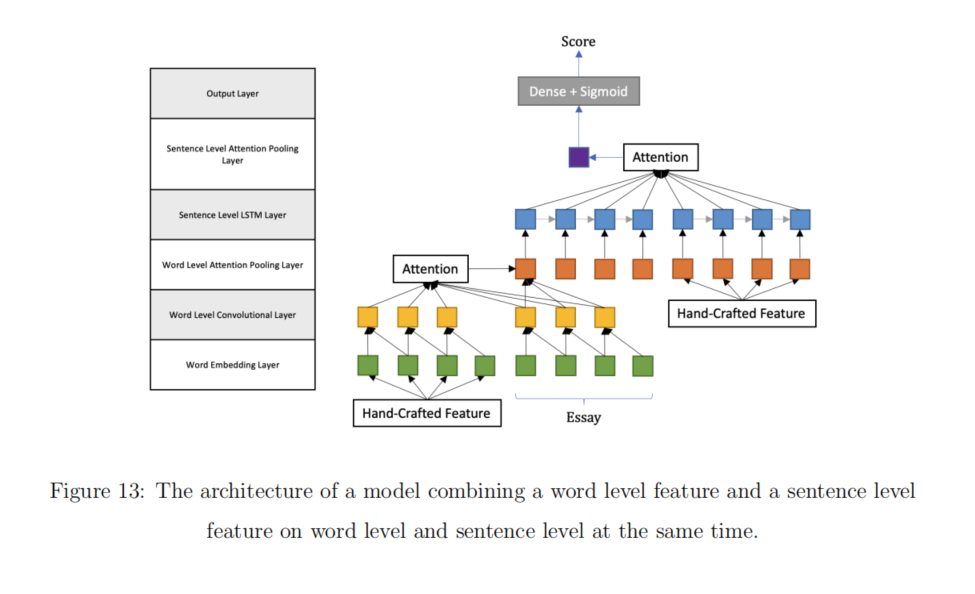
Since an individual feature may only provide limited extra information for learning, we might need to combine multiple features. Figure 13 shows an example model architecture that combines a word level feature on the model word level, and a sentence level feature on the model sentence level at the same time. One of the advantages of this model design is that the hand-crafted feature combination is modular. This provides flexibility to combine multiple features at one time. We only need to calculate attended representation for each feature, and eventually concatenate them together.







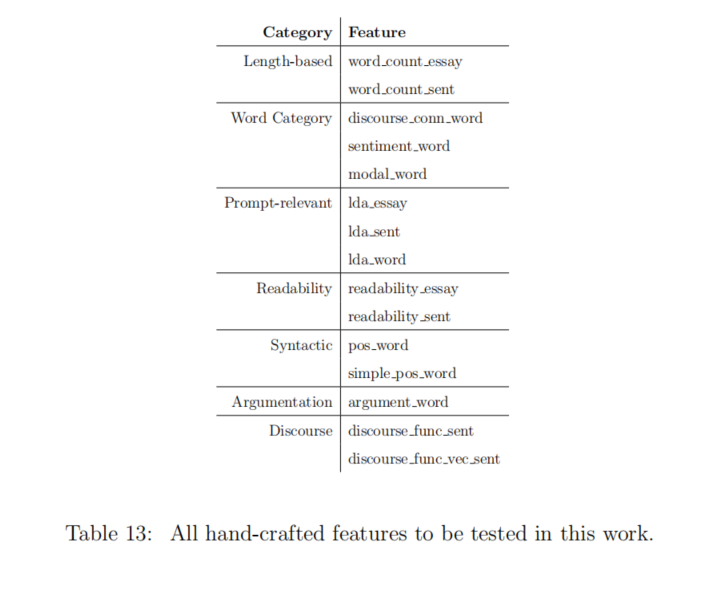




**5.4.2 Hand-crafted Features**

Hand-crafted features are widely used for AES, and each of them falls into a feature category [46], e.g., length-based features, lexical features, embeddings, word category features, prompt-relevant features, readability features, syntactic features, argument features, semantic features, and discourse features. We test a few features from each category in this work, except lexical features, embeddings, and semantic features. Lexical features are usually n-grams features. However, if we combine n-grams with a neural network model, the embedding layer is necessary to model each word. Therefore, in this work, we consider the lexical features are similar to embeddings, eventually. Although embeddings and semantic features are powerful for AES, we believe that the neural network model has modeled essays with embedding in a semantic way. Therefore, we consider it is redundant to combine similar features again, especially as the neural network shows a strong ability to model student essays.

Note that there is likely no one-size-fits-all feature that could be used in all writing tasks. For example, a general prompt such as write about “a time that you failed and learned something useful” would likely have little related topics from an LDA perspective. However, in this section, we introduce all features that we could potentially combine into the base model. We present two feature selection strategies later. The category and feature columns of Table 13 show all features that will be tested in this work.

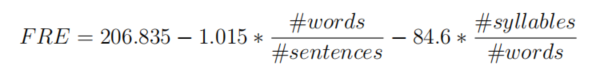


**Length-based Features.** These features are widely used for AES, since length is highly positively correlated with essay scores [5, 21, 72, 79, 110]. We include two features based on word count: essay length (denoted by ***word count essay***), and sentence length (***word count sent***).

**Word Category Features.** An essay should demonstrate the writer’s ability for word usage. This feature could be computed based on an external wordlist or dictionary. However, the wordlist or the dictionary could contain a variant of categories of words such as lexical, syntactic, and semantic [106, 64, 3]. This feature is useful when the size of the training data is small because these features help generalize word n-gram features [46]. Therefore, this feature is potentially helpful for AES because essay corpora are often small. To be more specific, we select discourse connectives [80] (denoted by ***discourse conn word***), sentiment words [9] (***sentiment word***) and modals (***modal word***). We implement this set of features as word-level features, with each word labeled as to whether it belongs to the word category.

**Prompt-relevant Features.** A good essay should be highly related to the prompt. Thus, this feature represents the relatedness between the essay and the prompt. A variety of similarity measures have often been used to compute this feature, such as word overlap, word topicality, and semantic similarity. In this category, we will use the LDA model [12] to compute this feature. Since this is a data-driven method, we assume most essays talk about the same topic, and that they are related to the prompt. Then the LDA model helps us find out if an essay is off topic. With the LDA model, we can know the topic distribution of the essay or a single sentence. Therefore, this feature could be either an essaylevel feature (denoted by ***lda essay***) or a sentence-level feature (***lda sent***). Each essay or sentence will be represented by its topic distribution. Since we can also know the word-topic distribution from the LDA model, we can also use the distribution as word representation. Therefore, this feature could also be a word-level feature (***lda word***).

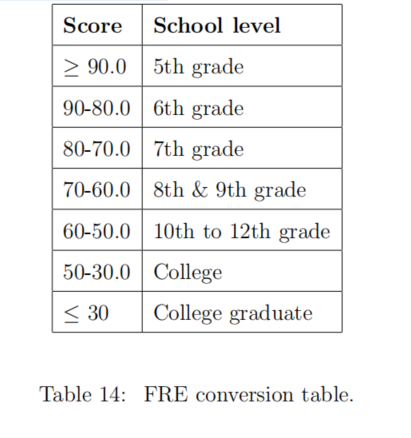
**Readability Features.** A good essay should be easy to read by a specific group of people, which means the word choice should be neither too difficult nor too easy to read. A good writer should demonstrate vocabulary that matches their school level [110]. A widely used readability metric is the Flesch Reading Ease Test (FRE) [49]:



The score range of FRE is from *−∞* to 121.22, the lower the number, the harder to read.

Table 14 shows the conversion table from FRE score to grade level. Although the FRE

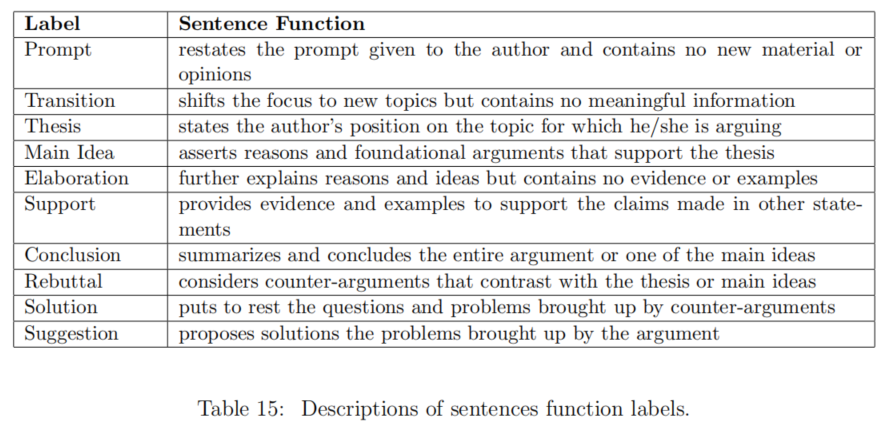
measures the essay level readability, we could also calculate FRE for independent sentences. Therefore, this feature could be either an essay level (denoted by ***readability essay***) or sentence level feature (***readability sent***).



**Syntactic Features.** This feature encodes the syntactic information about the essay, and it demonstrates the writer’s style [110]. This feature is a word-level feature, and we label each word with its part-of-speech tag. We use two tagsets. The first one is the most commonly used Penn Treebank Tagset [98] (denoted by ***pos word***). However, this tagset is too comprehensive and contains 36 different tags without special symbols. We doubt whether a comprehensive tagset is suitable for the AES task because the size of the training data is usually small. Thus, we also use a simple tagset that only contains “adjective”, “noun”, “adverb”, and “verb” (***simple pos word***).

**Argumentation Features.** Using argumentative structures to score persuasive essays has drawn increasing attention [78, 41, 70]. We label each word with IOB-formatted labels for argument units (premises, claims) with TARGER [24]. Thus, this sequence tagging feature is a word-level feature (***argument word***).

**Discourse Features.** Typically, there are four widely used discourse features: entity grids [8], rhetorical structure theory trees [62], lexical chains [68], and discourse function labels [75]. In this work, we plan to test the discourse function label because it provides a label for each sentence, which makes this feature a sentence level feature. Table 15 show the full possible discourse function labels and explanations for sentence. Possible labels are “Prompt”, “Transition”, “Thesis”, “Main Idea”, “Elaboration”, “Support”, “Conclusion”, “Rebuttal”, “Solution”, and “Suggestion”. A heuristic algorithm for labeling each sentence was developed [75] 1 . For each sentence, this algorithm calculates a score for each label and assigns the label with the highest score to the sentence. Therefore, we created 2 forms of this feature. The first is the label for each sentence (denoted by ***discourse func sent***), while the second is the score vector (***discourse func vec sent***).



Besides, this is our initial exploration of this hybrid model. Therefore we listed widely used hand-crafted features as many as possible, and we only select one or two features from each category. We thought features were meant to be exhaustive over types but illustrative within each type. Thus, features like LDA features or discourse function label features might not be the most optimal feature over all AES tasks. However, we still observe that the LDA model is one of the best features at the essay level.

**5.5 Experimental Setup**

We use the Automated Student Assessment Prize (ASAP) corpus and the Response-toText Assessment (RTA) corpus to evaluate our hybrid model. We configure experiments to test four hypotheses:

H1: The hand-crafted features work better when combined on the sentence-level or word-level of the neural model, compared to the essay-level.

H2: The hybrid model that combines features extracted from the essay at the sentencelevel or word-level works better, compared to essay-level features.

H3: The hybrid model will outperform or at least perform as well as the base neural model when trained and tested on the same prompt.

H4: The hybrid model will generalize better across different prompts because handcrafted features generalize better over prompts.

For within-prompt experiments, we use 5-fold cross-validation as in prior work [34], to split the data for each prompt into 5 folds. In each fold, 60% of data are used for training, 20% are used for development, and 20% are used for testing. Note that we are using a different deep learning framework to implement the base model compared to that used in [34]. The original paper used Keras 1.1.1 and Theano 0.8.2, while we use TensorFlow 2.2.0. Thus, the base model AES results reported in this work have small differences compared to the numbers reported in the original paper.

For cross-prompt experiments, we extend the 4 single direction pairs of essay prompts used in prior work [79] to 5 bi-direction pairs. More specifically, these 5 pairs of essay prompts were picked based on the similarity in their genres, score ranges, and median scores. The essay set pairs are 1 *↔* 2, 3 *↔* 4, 5 *↔* 6, 7 *↔* 8, and *M ↔ S*, where the pair 1 *↔* 2 denotes using prompt 1 (or prompt 2) as the source prompt and prompt 2 (or prompt 1) as the target prompt. We use all essays from the source prompt for training. Target prompt data are randomly divided into 5 folds (same as the within-prompt experiment), where one fold is used as test data, and one fold is used as the development set. We do not include data from the target prompt for training in order to test the ability of prompt adaptation of our model. We use the development set from the target prompt, but only to determine early stopping. All other hyper-parameters are not selected based on the development set. Consequently, our approach is not zero-shot but instead assumes a small amount of data from the target prompt.

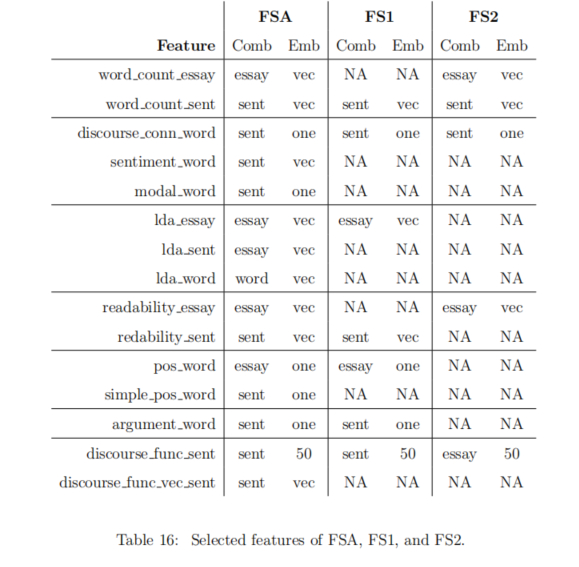
Besides, we also include results of four other models reported in prior work [79, 27, 57, 19] for pairs that were previously studied, denoted by ML-*ρ*, SKWE, TSLF, and HA, respectively. Note that although these results are numbers from the original papers, the size of the training and testing data are the same as in our experiments. Since we do not obtain model performance for each fold, we cannot perform significance tests between our model and these models.

Since some hand-crafted features used by our model are categorical, we need to change their representations to serve as the input of the neural network. We will test two forms, either one-hot representation or embedding representation.

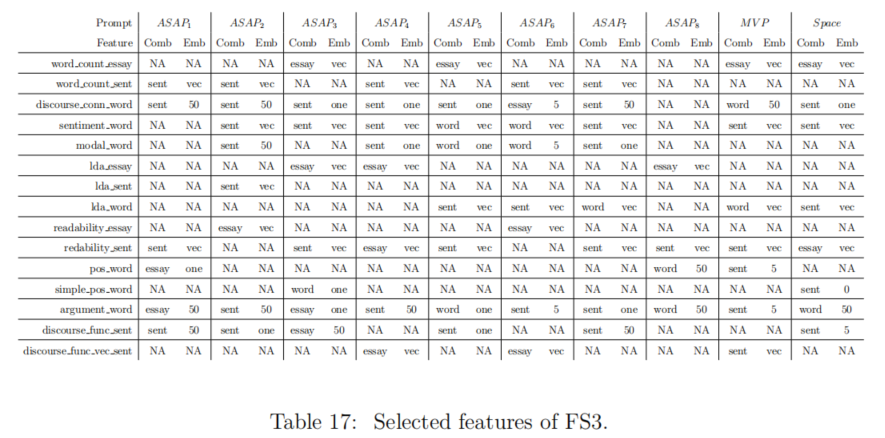
Since our experiments combine multiple features at a time, we want to perform feature selection to select the best level of combination for each feature. The level of combination is the level that we combine the essay representation and feature representation, either word level, sentence level, or essay level. First, we combine one feature at a time, so that we know the best representation and combination level of each feature. Next, we need to figure out which features to combine. We adopt three strategies. First, we simply combine the best variant of each feature (denoted by FSA, where A stands for all features). Therefore, FSA shows the best representation and combination level of each feature (possible values described below). Second, we select one subset of features that works for all prompts. Specifically, we select features that improve the base model on the development set for at least 9 (out of 10) prompts (denoted by FS1). We also select features that **significantly** improve the base model on the development set for at least 6 prompts (denoted by FS2). The intuition is that we want to combine features that improve the base model on as many

prompts as possible, while preserving a reasonable number of features. Third, we select a set

of features separately for each prompt. We select a feature as long as using it in the hybrid model improves over the base model when evaluated on the development set for the prompt (denoted by FS3). Table 16 shows the selected features of FSA, FS1, and FS2. For each feature set, the “Comb” column indicates the best combination level, and “Emb” indicates the best feature representation: “vec” means the feature is a vector, “one” means one-hot representation, and number means the number of dimensions of embedding representation. Table 17 shows the selected features of FS3. Note that all feature selection is made on the development set with an within-prompt experimental setting.



Following Dong et al., [34], the vocabulary size of the data is limited to 4000, all scores are scaled to the range [0, 1], all hyper-parameters for training shown in Table 18, and use Quadratic Weighted Kappa (QWK) for evaluation.

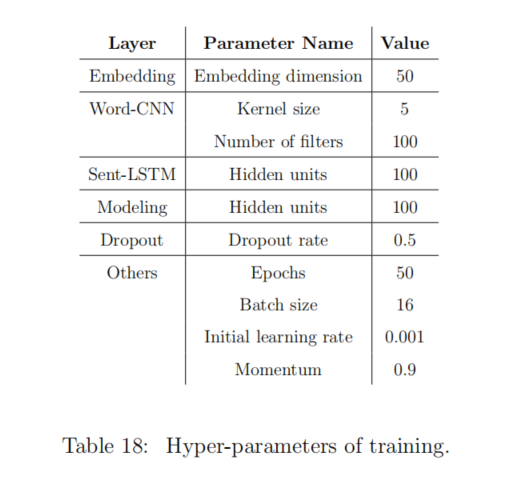


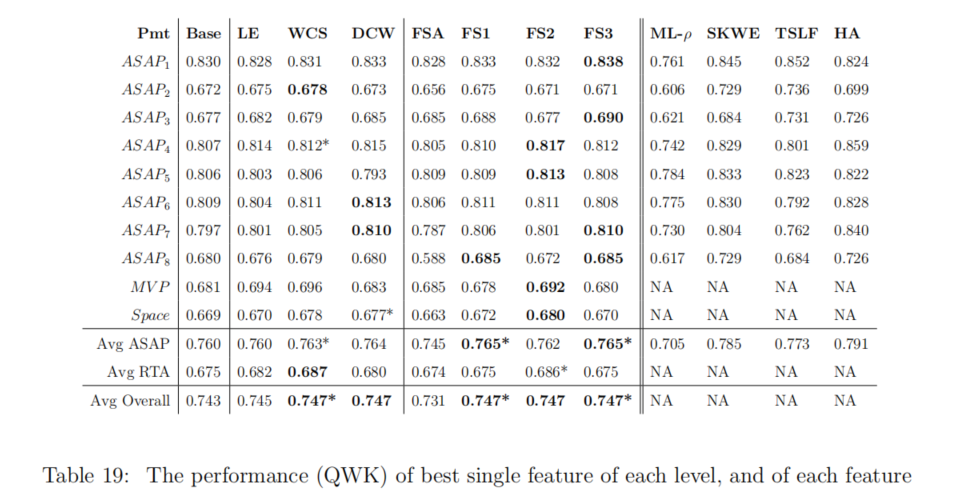
**5.6 Results and Discussion**

**Results for H1.** The feature list of FSA in Table 16 supports H1. We observe that the hand-crafted features work better when combined on sentence-level, compared to wordlevel and essay-level. Overall, we have 7 word-level features, 5 sentence-level features, and 3 essay-level features. Since essay-level features can only be combined on essay-level, we only focus on word-level features and sentence-level features. There are 5 (out of 7) word-level features that perform the best when combined on the sentence level. There are 4 (out of 5) sentence-level features that perform the best when combined on the sentence level. However, only 1 word-level feature and sentence-level feature perform the best when combined on the essay level.

One possible reason for the utility of sentence-level combination is that the word-level representation contains too much detailed information, including noise explicitly, while the sentence-level representation abstracts word-level representations and reduces explicit noise. Essay-level representation, which is even more abstracted, may in turn lose too much detailed

information of the essay.



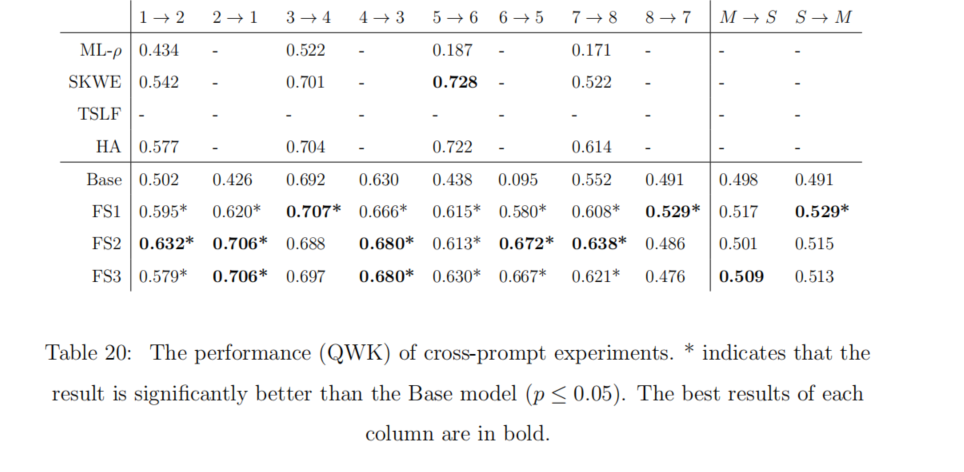
Table 19: The performance (QWK) of best single feature of each level, and of each feature selection set for within-prompt experiments. \* indicates that the result is significantly

better than the baseline (*p ≤* 0*.*05). The best results within the base model and proposed

model of each row are in bold.

**Results for H2.** Table 19 (columns 3-5) shows the best feature of each linguistic level. Obviously, the *word count sent* (WCS) feature and the *discourse conn word* (DCW) feature improve the base model more than the *lda essay* (LE) feature. Besides, on average, all essay-level features improve 6.33 prompts, while sentence-level features and word-level features improve 6.60 and 6.14 prompts, respectively. All results support H2.

**Results for H3.** The results in Table 19 generally support H3. When using multiple features chosen via feature selection, FS1 and FS3 yield significantly higher performance than the base model on average. Although FS2 does not yield significant improvement on average, it significantly outperforms the base model on the RTA corpus. Even when the hybrid model uses only one feature, *lda essay* (LE), *word count sent* (WCS) or *discourse conn word* (DCW) can also outperform the base model on average. In contrast, when using all handcrafted features (FSA), the hybrid model performs worse than the base model. One possible reason is that size of the ASAP training set is relatively small. If we combine too many features, the hybrid model might become too complicated for the AES task.



**Results for H4.** Since FSA does not perform well in within-prompt experiments, we exclude FSA from cross-prompt experiments. The results in Table 20 support H4. For each prompt pair, we can always find the best result from our proposed model, except prompt pair 5 *→* 6. Although our model does not outperform SKWE and HA for prompt pair 5 *→* 6, it still outperforms the base model, which indicates that our approach has a positive contribution to the base model.

Comparing to within-prompt experiments, the hybrid model shows a more considerable improvement over the base model. One possible reason is that the features we combined are general features that may be sharing the same behavior over prompts. For example, if the average lengths of source prompt and target prompt essays are similar, length-based features should still be highly predictive over prompts. A similar reason might hold for the readability feature, as long as the age groups are similar. The discourse-related features encode the discourse structure of an essay, and reflect the organization of an essay. Since the quality of the organization is content independent, thus, the discourse-related features generalize over prompts as well.

Bibliography

[1] [Abadi et al., 2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[2] [Alikaniotis et al., 2016] Alikaniotis, D., Yannakoudakis, H., and Rei, M. (2016). Automatic text scoring using neural networks. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 715–725.

[3] [Amorim et al., 2018] Amorim, E., Cancado, M., and Veloso, A. (2018). Automated essay scoring in the presence of biased ratings. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 229–237.

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

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

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

[7] [Baroni et al., 2014] Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In ACL (1), pages 238–247.

[8] [Barzilay and Lapata, 2008] Barzilay, R. and Lapata, M. (2008). Modeling local coherence: An entity-based approach. Computational Linguistics, 34(1):1–34.

[9] [Bird et al., 2009] Bird, S., Klein, E., and Loper, E. (2009). Natural language processing with Python: analyzing text with the natural language toolkit. O’Reilly Media, Inc.

[10] [Blanchard et al., 2013] Blanchard, D., Tetreault, J., Higgins, D., Cahill, A., and Chodorow, M. (2013). Toefl11: A corpus of non-native english. ETS Research Report Series, 2013(2):i–15.

[11] [Blei and Lafferty, 2009] Blei, D. M. and Lafferty, J. D. (2009). Visualizing topics with multi-word expressions. arXiv preprint arXiv:0907.1013.

[12] [Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. Journal of Machine Learning Research, 3(Jan):993–1022.

[13] [Burstein et al., 2004] Burstein, J., Chodorow, M., and Leacock, C. (2004). Automated essay evaluation: The criterion online writing service. AI Magazine, 25(3):27–27.

[14] [Burstein et al., 2001] Burstein, J., Kukich, K., Wolff, S., Lu, C., and Chodorow, M. (2001). Enriching automated essay scoring using discourse marking.

[15] [Burstein et al., 1998] Burstein, J., Kukich, K., Wolff, S., Lu, C., Chodorow, M., Braden-Harder, L., and Harris, M. D. (1998). Automated scoring using a hybrid feature identification technique. In Proceedings of the 17th International Conference on Computational Linguistics-Volume 1, pages 206–210. Association for Computational Linguistics.

[16] [Burstein et al., 2010] Burstein, J., Tetreault, J., and Andreyev, S. (2010). Using entity-based features to model coherence in student essays. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 681–684.

[17] [Butnaru and Ionescu, 2017] Butnaru, A. M. and Ionescu, R. T. (2017). From image to text classification: A novel approach based on clustering word embeddings. Procedia Computer Science, 112:1783–1792.

[18] [Cao et al., 2015] Cao, C., Liu, X., Yang, Y., Yu, Y., Wang, J., Wang, Z., Huang, Y., Wang, L., Huang, C., Xu, W., et al. (2015). Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In Proceedings of the IEEE International Conference on Computer Vision, pages 2956–2964.

[19] [Cao et al., 2020] Cao, Y., Jin, H., Wan, X., and Yu, Z. (2020). Domain-adaptive neural automated essay scoring. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 1011–1020.

[20] [Chawla et al., 2002] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16:321–357.

[21] [Chen and He, 2013] Chen, H. and He, B. (2013). Automated essay scoring by maximizing human-machine agreement. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1741–1752.

[22] [Chen et al., 2018] Chen, L., Tao, J., Ghaffarzadegan, S., and Qian, Y. (2018). End-to-end neural network based automated speech scoring. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6234–6238. IEEE.

[23] [Chen et al., 2010] Chen, Y.-Y., Liu, C.-L., Lee, C.-H., Chang, T.-H., et al. (2010). An unsupervised automated essay-scoring system. IEEE Intelligent Systems, 25(5):61–67.

[24] [Chernodub et al., 2019] Chernodub, A., Oliynyk, O., Heidenreich, P., Bondarenko, A., Hagen, M., Biemann, C., and Panchenko, A. (2019). Targer: Neural argument mining at your fingertips. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 195–200.

[25] [Chollet et al., 2015] Chollet, F. et al. (2015). Keras. https://keras.io.

[26] [Correnti et al., 2013] Correnti, R., Matsumura, L. C., Hamilton, L., and Wang, E. (2013). Assessing students’ skills at writing analytically in response to texts. The Elementary School Journal, 114(2):142–177.

[27] [Cozma et al., 2018] Cozma, M., Butnaru, A., and Ionescu, R. T. (2018). Automated essay scoring with string kernels and word embeddings. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 503–509.

[28] [Crossley and McNamara, 2016] Crossley, S. A. and McNamara, D. S. (2016). Adaptive educational technologies for literacy instruction. Routledge.

[29] [Cummins et al., 2016] Cummins, R., Zhang, M., and Briscoe, E. J. (2016). Constrained multi-task learning for automated essay scoring. Association for Computational Linguistics.

[30] [Dasgupta et al., 2018] Dasgupta, T., Naskar, A., Dey, L., and Saha, R. (2018). 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, pages 93–102.

[31] [Dauphin et al., 2015] Dauphin, Y., de Vries, H., and Bengio, Y. (2015). Equilibrated adaptive learning rates for non-convex optimization. In Advances in Neural Information Processing Systems, pages 1504–1512.

[32] [Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

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

[34] [Dong et al., 2017] Dong, F., Zhang, Y., and Yang, J. (2017). Attention-based recurrent convolutional neural network for automatic essay scoring. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pages 153–162.

[35] [Ebrahimpour et al., 2019] Ebrahimpour, M. K., Li, J., Yu, Y.-Y., Reesee, J., Moghtaderi, A., Yang, M.-H., and Noelle, D. C. (2019). Ventral-dorsal neural networks: Object detection via selective attention. In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 986–994. IEEE.

[36] [Farag et al., 2018] Farag, Y., Yannakoudakis, H., and Briscoe, T. (2018). Neural automated essay scoring and coherence modeling for adversarially crafted input. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 263–271.

[37] [Farra et al., 2015] Farra, N., Somasundaran, S., and Burstein, J. (2015). Scoring persuasive essays using opinions and their targets. In Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 64–74.

[38] [Florescu and Caragea, 2017] Florescu, C. and Caragea, C. (2017). PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1105–1115.

[39] [Florescu and Jin, 2018] Florescu, C. and Jin, W. (2018). Learning feature representations for keyphrase extraction. In Thirty-Second AAAI Conference on Artificial Intelligence.

[40] [Foltz and Rosenstein, 2015] Foltz, P. W. and Rosenstein, M. (2015). Analysis of a large-scale formative writing assessment system with automated feedback. In Proceedings of the Second (2015) ACM Conference on Learning@ Scale, pages 339–342. ACM.

[41] [Ghosh et al., 2016] Ghosh, D., Khanam, A., Han, Y., and Muresan, S. (2016). Coarse-grained argumentation features for scoring persuasive essays. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 549–554.

[42] [Gierl et al., 2014] Gierl, M. J., Latifi, S., Lai, H., Boulais, A.-P., and De Champlain, A. (2014). Automated essay scoring and the future of educational assessment in medical education. Medical Education, 48(10):950–962.

[43] [Granger et al., 2009] Granger, S., Dagneaux, E., Meunier, F., and Paquot, M. (2009). International Corpus of Learner English. Version 2. Louvain-la-Neuve: Presses universitaires de Louvain.

[44] [Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8):1735–1780.

[45] [Jin et al., 2018] Jin, C., He, B., Hui, K., and Sun, L. (2018). TDNN: A two-stage deep neural network for prompt-independent automated essay scoring. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1088–1097.

[46] [Ke and Ng, 2019] Ke, Z. and Ng, V. (2019). Automated essay scoring: A survey of the state of the art. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, pages 6300–6308. AAAI Press.

[47] [Kenter and de Rijke, 2015] Kenter, T. and de Rijke, M. (2015). Short text similarity with word embeddings. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pages 1411–1420. ACM.

[48] [Kiela et al., 2015] Kiela, D., Hill, F., and Clark, S. (2015). Specializing word embeddings for similarity or relatedness. In EMNLP, pages 2044–2048.

[49] [Kincaid et al., 1975] Kincaid, J. P., Fishburne Jr, R. P., Rogers, R. L., and Chissom, B. S. (1975). Derivation of new readability formulas (automated readability index, fog count and Flesch reading ease formula) for navy enlisted personnel.

[50] [Klebanov and Flor, 2013] Klebanov, B. B. and Flor, M. (2013). Word association profiles and their use for automated scoring of essays. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1148–1158.

[51] [Klebanov et al., 2016] Klebanov, B. B., Flor, M., and Gyawali, B. (2016). Topicality-based indices for essay scoring. In Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications, pages 63–72.

[52] [Klebanov et al., 2014] Klebanov, B. B., Madnani, N., Burstein, J., and Somasundaran, S. (2014). Content importance models for scoring writing from sources. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), volume 2, pages 247–252.

[53] [Ko et al., 2019] Ko, W.-J., Durrett, G., and Li, J. J. (2019). Domain agnostic real-valued specificity prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 6610–6617.

[54] [LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324.

[55] [Lei et al., 2016] Lei, T., Barzilay, R., and Jaakkola, T. (2016). Rationalizing neural predictions. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 107–117.

[56] [Li et al., 2018] Li, X., Chen, M., Nie, J., Liu, Z., Feng, Z., and Cai, Y. (2018). Coherence-based automated essay scoring using self-attention. In Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, pages 386–397. Springer.

[57] [Liu et al., 2019] Liu, J., Xu, Y., and Zhao, L. (2019). Automated essay scoring based on two-stage learning. arXiv preprint arXiv:1901.07744.

[58] [Louis and Higgins, 2010] Louis, A. and Higgins, D. (2010). Off-topic essay detection using short prompt texts. In Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications, pages 92–95. Association for Computational Linguistics.

[59] [Louis and Nenkova, 2013] Louis, A. and Nenkova, A. (2013). Automatically assessing machine summary content without a gold standard. Computational Linguistics, 39(2):267–300.

[60] [Madnani et al., 2018] Madnani, N., Burstein, J., Elliot, N., Klebanov, B. B., Napolitano, D., Andreyev, S., and Schwartz, M. (2018). Writing mentor: Self-regulated writing feedback for struggling writers. In Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations, pages 113–117.

[61] [Mahata et al., 2018] Mahata, D., Kuriakose, J., Shah, R. R., and Zimmermann, R. (2018). Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 634–639.

[62] [Mann and Thompson, 1988] Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. Text-Interdisciplinary Journal for the Study of Discourse, 8(3):243–281.

[63] [Mayfield and Black, 2020] Mayfield, E. and Black, A. W. (2020). Should you fine-tune BERT for automated essay scoring? In Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 151–162.

[64] [McNamara et al., 2015] McNamara, D. S., Crossley, S. A., Roscoe, R. D., Allen, L. K., and Dai, J. (2015). A hierarchical classification approach to automated essay scoring. Assessing Writing, 23:35–59.

[65] [Meng et al., 2017] Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., and Chi, Y. (2017). Deep keyphrase generation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 582–592.

[66] [Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[67] [Mikolov et al., 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems, pages 3111–3119.

[68] [Morris and Hirst, 1991] Morris, J. and Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational Linguistics, 17(1):21–48.

[69] [Nadeem et al., 2019] Nadeem, F., Nguyen, H., Liu, Y., and Ostendorf, M. (2019). Automated essay scoring with discourse-aware neural models. In Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 484–493.

[70] [Nguyen and Litman, 2018] Nguyen, H. V. and Litman, D. J. (2018). Argument mining for improving the automated scoring of persuasive essays. In Thirty-Second AAAI Conference on Artificial Intelligence.

[71] [Ong et al., 2014] Ong, N., Litman, D., and Brusilovsky, A. (2014). Ontology-based argument mining and automatic essay scoring. In Proceedings of the First Workshop on Argumentation Mining, pages 24–28.

[72] [Ostling et al., 2013] Ostling, R., Smolentzov, A., Hinnerich, B. T., and Höglin, E. (2013). Automated essay scoring for Swedish. In Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, pages 42–47.

[73] [Page, 1968] Page, E. B. (1968). The use of the computer in analyzing student essays. International Review of Education, 14(2):210–225.

[74] [Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543.

[75] [Persing et al., 2010] Persing, I., Davis, A., and Ng, V. (2010). Modeling organization in student essays. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 229–239. Association for Computational Linguistics.

[76] [Persing and Ng, 2013] Persing, I. and Ng, V. (2013). Modeling thesis clarity in student essays. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 260–269.

[77] [Persing and Ng, 2014] Persing, I. and Ng, V. (2014). Modeling prompt adherence in student essays. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 1534–1543.

[78] [Persing and Ng, 2015] Persing, I. and Ng, V. (2015). Modeling argument strength in student essays. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 543–552.

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

[80] [Pitler and Nenkova, 2009] Pitler, E. and Nenkova, A. (2009). Using syntax to disambiguate explicit discourse connectives in text. In Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 13–16.

[81] [Rahimi and Litman, 2016] Rahimi, Z. and Litman, D. (2016). Automatically extracting topical components for a response-to-text writing assessment. In Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications, pages 277–282.

[82] [Rahimi et al., 2017] Rahimi, Z., Litman, D., Correnti, R., Wang, E., and Matsumura, L. C. (2017). Assessing students’ use of evidence and organization in response-to-text writing: Using natural language processing for rubric-based automated scoring. International Journal of Artificial Intelligence in Education, pages 1–35.

[83] [Rahimi et al., 2014] Rahimi, Z., Litman, D. J., Correnti, R., Matsumura, L. C., Wang, E., and Kisa, Z. (2014). Automatic scoring of an analytical response-to-text assessment. In International Conference on Intelligent Tutoring Systems, pages 601–610. Springer.

[84] [Ramachandran et al., 2015] Ramachandran, L., Cheng, J., and Foltz, P. (2015). Identifying patterns for short answer scoring using graph-based lexico-semantic text matching. In Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 97–106.

[85] [Rei and Cummins, 2016] Rei, M. and Cummins, R. (2016). Sentence similarity measures for fine-grained estimation of topical relevance in learner essays. arXiv preprint arXiv:1606.03144.

[86] [Röder et al., 2015] Röder, M., Both, A., and Hinneburg, A. (2015). Exploring the space of topic coherence measures. In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pages 399–408.

[87] [Roscoe et al., 2014] Roscoe, R. D., Allen, L. K., Weston, J. L., Crossley, S. A., and McNamara, D. S. (2014). The Writing Pal intelligent tutoring system: Usability testing and development. Computers and Composition, 34:39–59.

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

[89] [Seo et al., 2017] Seo, M., Kembhavi, A., Farhadi, A., and Hajishirzi, H. (2017). Bidirectional attention flow for machine comprehension. In Proceedings of the International Conference on Learning Representations (ICLR).

[90] [Shermis and Burstein, 2003] Shermis, M. D. and Burstein, J. C. (2003). Automated essay scoring: A cross-disciplinary perspective. Routledge.

[91] [Shibani et al., 2019] Shibani, A., Knight, S., and Shum, S. B. (2019). Contextualizable learning analytics design: A generic model and writing analytics evaluations. In Proceedings of the 9th International Conference on Learning Analytics & Knowledge, pages 210–219. ACM.

[92] [Somasundaran et al., 2014] Somasundaran, S., Burstein, J., and Chodorow, M. (2014). Lexical chaining for measuring discourse coherence quality in test-taker essays. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 950–961.

[93] [Song et al., 2017] Song, W., Wang, D., Fu, R., Liu, L., Liu, T., and Hu, G. (2017). Discourse mode identification in essays. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 112–122.

[94] [Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1):1929–1958.

[95] [Stab and Gurevych, 2014] Stab, C. and Gurevych, I. (2014). Annotating argument components and relations in persuasive essays. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 1501–1510.

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

[97] [Tay et al., 2018] Tay, Y., Phan, M. C., Tuan, L. A., and Hui, S. C. (2018). SkipFlow: Incorporating neural coherence features for end-to-end automatic text scoring. In Thirty-Second AAAI Conference on Artificial Intelligence.

[98] [Taylor et al., 2003] Taylor, A., Marcus, M., and Santorini, B. (2003). The Penn Treebank: An overview. In Treebanks, pages 5–22. Springer.

[99] [Theano Development Team, 2016] Theano Development Team (2016). Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints, abs/1605.02688.

[100] [Uto et al., 2020] Uto, M., Xie, Y., and Ueno, M. (2020). Neural automated essay scoring incorporating handcrafted features. In Proceedings of the 28th International Conference on Computational Linguistics, pages 6077–6088.

[101] [Vajjala, 2018] Vajjala, S. (2018). Automated assessment of non-native learner essays: Investigating the role of linguistic features. International Journal of Artificial Intelligence in Education, 28(1):79–105.

[102] [Witten et al., 2016] Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

[103] [Woods et al., 2017] Woods, B., Adamson, D., Miel, S., and Mayfield, E. (2017). Formative essay feedback using predictive scoring models. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 2071–2080. ACM.

[104] [Xie et al., 2012] Xie, S., Evanini, K., and Zechner, K. (2012). Exploring content features for automated speech scoring. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 103–111. Association for Computational Linguistics.

[105] [Yamamoto et al., 2019] Yamamoto, M., Umemura, N., and Kawano, H. (2019). Proposal of Japanese vocabulary difficulty level dictionaries for automated essay scoring support system using rubric. Journal of the Operations Research Society of China, pages 1–17.

[106] [Yannakoudakis and Briscoe, 2012] Yannakoudakis, H. and Briscoe, T. (2012). Modeling coherence in ESOL learner texts. In Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, pages 33–43. Association for Computational Linguistics.

[107] [Yannakoudakis et al., 2011] Yannakoudakis, H., Briscoe, T., and Medlock, B. (2011). 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, pages 180–189. Association for Computational Linguistics.

[108] [Yuxin et al., 2018] Yuxin, P., Xiangteng, H., and Junjie, Z. (2018). Object-part attention model for fine-grained image classification. IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society, 27(3):1487–1500.

[109] [Zesch et al., 2015a] Zesch, T., Heilman, M., and Cahill, A. (2015a). Reducing annotation efforts in supervised short answer scoring. In Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 124–132.

[110] [Zesch et al., 2015b] Zesch, T., Wojatzki, M., and Scholten-Akoun, D. (2015b). Task-independent features for automated essay grading. In Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 224–232.

[111] [Zhang and Litman, 2017] Zhang, H. and Litman, D. (2017). Word embedding for response-to-text assessment of evidence. In Proceedings of ACL 2017, Student Research Workshop, pages 75–81.

[112] [Zhang and Litman, 2018] Zhang, H. and Litman, D. (2018). Co-attention based neural network for source-dependent essay scoring. In Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 399–409.

[113] [Zhang and Litman, 2020] Zhang, H. and Litman, D. (2020). Automated topical component extraction using neural network attention scores from source-based essay scoring. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8569–8584.

[114] [Zhang et al., 2019] Zhang, H., Magooda, A., Litman, D., Correnti, R., Wang, E., Matsmura, L., Howe, E., and Quintana, R. (2019). eRevise: Using natural language processing to provide formative feedback on text evidence usage in student writing. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 9619–9625.