




Automated post scoring: evaluating posts with topics and quoted posts in online forum

Ruosong Yang¹ · Jiannong Cao¹ · Zhiyuan Wen¹ · Jiaxing Shen¹ 

Received: 7 September 2021 / Revised: 19 November 2021 / Accepted: 3 January 2022 /

Published online: 10 March 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Online forum post evaluation is an effective way for instructors to assess students' knowledge understanding and writing mechanics. Manually evaluating massive posts costs a lot of time. Automatically grading online posts could significantly alleviate instructors' burden. Similar text assessment tasks like Automated Text Scoring evaluate the writing quality of independent texts or relevance between text and prompt. And Automatic Short Answer Grading measures the semantic matching of short answers according to given problems and correct answers. Different from existing tasks, we propose a novel task, Automated Post Scoring (APS), which grades all online discussion posts in each thread of each student with given topics and quoted posts. APS evaluates not only the writing quality of posts automatically but also the relevance to topics. To measure the relevance, we model the semantic consistency between posts and topics. Supporting arguments are also extracted from quoted posts to enhance posts evaluation. Specifically, we propose a mixture model including a hierarchical text model to measure the writing quality, a semantic matching model to model topic relevance, and a semantic representation model to integrate quoted posts. We also construct a new dataset called Online Discussion Dataset containing 2,542 online posts from 694 students of a social science course. The proposed models are evaluated on the dataset with correlation and residual based evaluation metrics. Compared with measuring posts alone, experimental results demonstrate that incorporating topics and quoted posts could improve the performance of APS by a large margin, more than 9 percent on QWK.

Keywords Automated post scoring · Text regression · Text mining · Deep learning

✉ Jiaxing Shen
jiaxing.shen@connect.polyu.hk

Ruosong Yang
csryang@comp.polyu.edu.hk

Jiannong Cao
csjcao@comp.polyu.edu.hk

Zhiyuan Wen
cszwen@comp.polyu.edu.hk

¹ The Hong Kong Polytechnic University, Hung Hom, Hong Kong

1 Introduction

Online education has shown significant growth over the last decade especially under the pandemic of COVID-19 [17]. As one of the most important features of online education, the discussion forum brings various benefits including boosting learning performance, reducing dropout rates, and increasing course satisfactory [34]. So many instructors adopt online discussion to assess students' writing mechanics and knowledge understanding from discussion posts [30]. However, marking numerous posts is time-consuming and labor-intensive for instructors [24]. Mutual disagreement often occurs when multiple evaluators mark the same posts. Even for a single evaluator, grading consistency is hard to guarantee given numerous posts [22]. Therefore, there is an urgent need for automated post scoring (APS) to evaluate the writing mechanics and knowledge understanding of students by grading their posts according to given topics and quoted posts automatically. An illustration of the task is shown in Figure 1. It is difficult to assess the knowledge understanding of students directly, so we evaluate the relevance between posts and given topics instead. In addition, quoted posts are used as auxiliary features to enhance posts evaluation.

The previous two types of works focused on evaluating either the writing quality or the correctness of short answers. Automated Text/Essay Scoring (AES) [24] mainly evaluates the writing quality of independent long essays or texts. We refer to ideas in these works to measure the writing quality of posts. Prompt-relevant AES/ATS also measures the relevance between text and prompts. However, these models utilize simple statistic features which are difficult to capture deeply semantic matching. They also face difficulties to align the semantics of multiple sentences. While Automatic Short Answer Grading Task (ASAG) [15] measures the correctness of the short student's answer according to the correct answer and given question. The semantic matching between long texts (hundreds of words) is more complicated than that of short texts (one or two sentences). Unlike prompt-relevant AES/ATS, APS is required to utilize extra quoted posts to enhance the measurement of the posts. And advanced neural semantic matching methods [3] are necessary. Compared with ASAG, APS pays attention to the writing quality of long texts, the relevance between long texts, and the quoted posts. Examples from three typical datasets corresponding to three tasks are shown in Table 1. And the differences between the three tasks are summarized in Table 2.

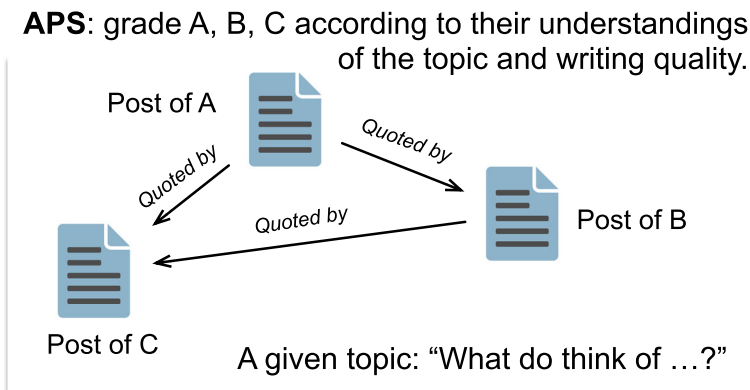


Figure 1 Illustration of APS

Table 1 Sample data of ODD, ASAP Dataset, and Semeval 2013 task 7

APS	AES/ATS	ASAG
<p>Online Discussion Dataset</p> <p>Topic: “If we could record the activity of all neurons, we could understand the brain.” Gero Miesenboeck (2010) @ TED. on the arguments presented by Based Gero Miesenboeck (TED Video: Hyperlink 1) Partha Mitra (Scientific American Letter: Hyperlink 2) would you agree with the above statement? What are your rationale(s), with reference to your textbook the psychology literature, that support your stand?”</p> <p>Post: Thanks <i>2015-1-S202</i> for taking the role to summarize our points. It's nice to see your response. I wonder which of the stance you stand for. As you point out that our discussion is mainly focusing on the feasibility of the recording method, I would like to add something to support my argument. In order to crack the neural code, understanding how single neurons and complex networks process perceptions is a vital factor to understand the brain.</p> <p>Quoted Post: What i can conclude from your discussions is that <i>2015-1-S57</i> argued that the patterns of our brains are complicated and ever-changing and the insufficiency of recording the activity of our brains obstructs the understanding of our brains. <i>2015-1-S72</i> pointed that measuring the whole brain and then decoding then is not feasible. <i>2015-1-S292</i> thought that alternative ways to understand the brains through psychology.</p> <p>Score: 15.5</p>	<p>Automated Student Assessment Prize</p> <p>Post: the essay rough road ahead: do not exceed posted speed limit describes a mans bicycle ride through california. now, california is very hot during the summer, which is when the cyclist is riding. this setting greatly affects the mans journey. it made it very difficult for him to finish his ride. he drank most of his water in the beginning of his ride so he gets very dehydrated. the text states, the water bottles contained only a few tantalizing sips. as you can see the setting makes this mans bikeride very hard.</p> <p>Score: 2.0</p>	<p>Semeval2013 Task 7</p> <p>Question: Explain why you got a voltage reading of 1.5 for terminal 1 and the positive terminal.</p> <p>Correct Answer: Terminal 1 and the positive terminal are separated by the gap.</p> <p>Student Answer: because terminal one and the positive terminal are connected</p> <p>Label: correct, <i>contradictory</i>, incorrect</p>

For page limit, only part of posts are shown in Post and Quoted Post in ODD

In this paper, we propose different methods to incorporate topics and quoted posts respectively, since they play different roles in discussion forums. During the discussion, students write posts to respond to the topics, so the relatedness between posts and topics illustrates their knowledge understanding. It is possible to use semantic matching to measure relatedness. While students quoted partial arguments of other students to support or explain their arguments. It is necessary to extract supporting arguments from quoted posts as auxiliary features.

Our vision, however, entails three challenges when applied to reality. The first challenge is how to augment topics. They are too short and abstract so that it is difficult to directly measure the relatedness between abstract concepts and detailed arguments. The second challenge is how to measure the relatedness between long posts and long topics. Posts usually introduce several arguments, and the order of the arguments may be different from that of the concepts in the topics. The last challenge is how to extract supporting arguments. Quoted posts contain many arguments, however, not each argument is useful to support or explain the students' arguments.

To tackle these challenges, we use data augmentation methods to extend the topics, and propose two different models to integrate topics and quoted posts. More specifically, we extend the topics with text contents obtained from the hyperlinks that appeared in the given topics' descriptions. To map the arguments in posts and concepts description in topics, we propose a matching model which learns sentences' representations firstly and calculates the interactions between any two sentences to extract the matching features. To extract the supporting arguments from quoted posts, we propose a representation model which uses an attention model to calculate the weighted sum of the sentences' representations of quoted posts. In addition, we also adopt hierarchical text models to learn the syntactic and semantic information of the posts. We combine these three models as a mixture model and extract features to predict the posts' scores. We conduct various experiments to verify the effectiveness of topic augmentation, and incorporating topics as well as quoted posts. More specifically, we conduct four experiments. The first experiment shows the results of all baseline models that only use the students' posts. The second experiment introduces results of the matching model or representation model that incorporates topics or quoted posts respectively. We illustrate the performance of integrating topics and quoted posts simultaneously in the third experiment. The influence of the augmented topics is shown in the last experiment. Besides, we also illustrate the effect of hyper-parameters. Our mixture model outperforms the hierarchical text model that only assesses the posts by a large margin, nearly 9 percent in Quadratic Weight Kappa.

Table 2 The difference between APS, AES/ATS and ASAG

Task	Typical Dataset	Input Data	Text Length	Labels
APS	ODD	posts, topic, quoted posts	hundreds of words	real number
AES/ATS	ASAP ¹	text	hundreds of words	real number
ASAG	Semeval-2013 ²	question, correct answer, student answer	one or two sentences	several classes

Our contribution could be summarized as follows:

1. We propose a new task called APS, which evaluates the writing quality and relevance of posts with extra topics and quoted posts.
2. To solve the task, we propose to measure the relevance by the semantic consistency between the posts and the topics, and enhance posts prediction with the related supporting arguments extracted from quoted posts.
3. Experimental results show that the measurement of relevance and writing quality can score the posts much more accurately, nearly 9 percent in QWK.

The remainder of this paper is organized as follows. In Section 2, we will briefly introduce related works. Problem definition and introduction of the dataset are shown in Section 3. Section 4 illustrates our models with more details. Experimental results and analysis are given in Section 5. Section 6 is the conclusion.

2 Related work

There are two similar text assessment tasks including Automated Essay/Text Scoring and Automatic Short Answer Grading. In this section, we will briefly introduce the solutions to these two tasks.

2.1 Automated essay/text scoring

AES/ATS is a popular task in computer-assisted education, which aims to help instructors to score the writing quality of the essays automatically. An open dataset called Automated Student Assessment Prize (ASAP)¹ coming from a Kaggle competition is widely used in the research community. And the competition was organized as well as sponsored by William and Flora Hewlett Foundation (Hewlett). In general, there are three parts to solve AES problems namely feature extraction method, a score mapping function, and an objective function. Almost all works utilized linear function as the score mapping function, we will introduce these works from the two perspectives of feature extraction and objective function. For feature extraction, manual features and neural features are two popular methods. Early works focus on designing hand-crafted features [1, 7, 12, 21] such as statistic features including text length, lexical diversity, and linguistic features including coherence, organization, elaboration, sentence structure, conventions. Recently neural networks are widely adopted to learn text representations, such as the ensemble Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) model [24], the hierarchical LSTM/CNN model [6], or the adapted LSTM model [25]. With the success of large pre-trained language models, such as BERT [5], RoBERTa [13], R²BERT [31] used BERT to learn the representation of texts which could capture semantic meaning better. As for objective functions, prediction (classification or regression), ranking, and reinforcement learning based losses are widely used. For prediction, classification models attempt to classify the text into the correct category [29], regression models aim to learn the same score as that given by instructors [6, 24,

¹<https://www.kaggle.com/c/asap-aes>

25]. The ranking objective attempts to rank all the calculated scores to be in the same order as that of all the gold scores [4, 33]. R²BERT [31] combined ranking and regression objectives which performed better than every single objective. The reinforcement learning based constraint utilized a reward function to guide the score mapping function to learn more accurate scores [29]. In our experiments, LSTM models [24], hierarchical LSTM model [6], and the pre-trained language model, RoBERTa [13], are used as baseline models. In addition, [32] marks posts with their interaction relationship.

There are also some works considered prompt-relevant features to solve AES tasks. To measure the relevance of texts to the prompt, [14] utilized the number of words overlap and its variants. [2] considered word topicality. And [9] used random indexing to model semantic similarity. In these works, the prompt is similar to the given topic in the APS task.

These solutions focus on measuring the writing quality of independent texts, which can be used to evaluate the posts in our task. However, the measurement of relevance to topics stills needs more complicated models to capture information of semantic matching.

2.2 Automatic short answer grading task

ASAG is another popular task in computer-assisted assessment, which attempts to identify the correctness of the student's answer according to the correct answer as well as the given question. There is also a popular open dataset called The Joint Student Response Analysis which is the 7th task of semeval-2013.² The key step of the problem is to learn more accurate semantic matching features. Existing works mainly consider two types of features namely hand-crafted features and neural features. Hand-crafted features are proposed in early works including n-gram features [8], softcardinality text overlap features [10], graph alignment features [23], averaged word vector text similarity features [23], and other shallow lexical features [16]. Recently, neural network based features are also widely utilized such as the adapted convolutional recurrent neural network (CRNN) [19], and siamese BiLSTM with earth mover's distance pooling [11]. Besides, some works also utilized combined features, for example, sentence embedding, as well as token level hand-crafted features, are integrated [20].

In this paper, we propose a new task to evaluate both the writing quality and relevance in the discussion scenario. Compared with AES, topics are provided to evaluate the relevance, and quoted posts are also used to assess posts. And compared with ASAG, the writing quality of the post should be considered. Meanwhile, the relevance measurement of long texts is more complicated than the correctness of two short texts.

3 Problem definition and dataset

In this section, we will introduce some basic concepts in the online forum. Then we give a formal problem definition and show more details about our Online Discussion Dataset.

²<https://www.cs.york.ac.uk/semeval-2013/task7.html>

3.1 Glossary of online forum

In this section, we will explain some basic terminologies that are widely used in online forums, including thread, topic, post, and quoted post.

- Thread: A thread is a collection of posts that respond to the same topic, usually displayed from oldest to latest.
- Topic: A topic is the text description which may also contain several hyperlinks.
- Post: A post is a user-submitted text enclosed into a block containing the user’s details and the date and time it was submitted, which aims to respond to the given topic.
- Quoted Post: Quoted post is the post that was quoted by another post in the same thread.

3.2 Problem definition

In this section, we will define the problem from each student’s perspective. In the online forum, there are several threads. And for each thread, a topic $T(s_i)$ is given firstly by the instructors. Then for each students s_i , he/she is asked to submit n_i posts $P(s_i) = \{P(s_i)_1, \dots, P(s_i)_{n_i}; n_i \geq 1\}$ according to the given topic. During the discussion, the student usually quotes other students’ arguments such as “Refer to XXX’s point”, so all the posts of quoted students are used as quoted posts. For each post $P(s_i)_j$, it quotes several students’ posts $q(i, j) = \{P(s_k) : s_k \in S(i, j)\}$, where $S(i, j)$ is the students set quoted by the post. All quoted posts are $Q(s_i) = \{q(i, 1), \dots, q(i, n_i)\}$. The instructor marks all posts of each student with a numerical value $G(s_i)$. So the proposed task is to learn a mapping function mapping $T(s_i)$, $P(s_i)$, and $Q(s_i)$ to $G(s_i)$ as shown in Formula 1.

$$\min(G(s_i) - f(T(s_i), P(s_i), Q(s_i)))^2 \quad (1)$$

3.3 Dataset construction and pre-processing

This research has been approved by the ethics committee with the Reference Number HSEARS20160713001. We cooperate with the Department of Applied Social Sciences in our university and get all the online discussion posts from the course “Introduction to Psychology” in one academic year (two semesters). Based on these posts, we construct an Online Discussion Dataset, which contains 86 sub-forums (threads). In each sub-forum, the instructor gave a topic description first, which is a social science problem with several keywords and many hyperlinks. With the given topic, students were asked to write their arguments about the topic with references, which are similar to academic writing. During writing the posts, students were also encouraged to quote other students’ arguments to explain their points. Some statistics are given in Table 3. In the table, *#Students* means the total number of students in our dataset, *#Posts* means the total number of posts, and *AP* means the average number of posts for each student. Meanwhile, *APL* is the average

Table 3 Statistics of online discussion dataset

#Students	#Posts	AP	APL	ATL	#Quoted	AQ	Range
694	2542	3.66	270	41.19	405	1.21	[5.0,20.0]

number of words of each post, ATL is the average number of words of each topic, $\#Quoted$ is the total number of students who quoted other students' views, and AQ means the average number of quoted students for each student. Finally, $Range$ is the score range. Since each sub-forum is totally separated, the same students in different sub-forum will be treated as different students.

With the raw posts data, our pre-processing mainly includes two parts. Firstly, to preserve privacy, we need to mask all the students' names that appeared in the posts, and map them to the real students in the sub-forum. To identify the students' names, and ignore the names in the references, we use a regular expression to replace all the references into special tokens, and a tool of Named Entity Recognition is used to find students' names in text. Besides, fuzzy string matching is used to map identified names to real students in the forum. To make sure the complete masking, human verification is also necessary. Then, there are lots of hyperlinks and references in each post to support students' arguments, we replace various hyperlinks and references with special tokens, similar to replace the names and numbers with special tokens in AES tasks. In the dataset, each student submits several posts to show his/her arguments and quoted several students' posts. We combine all the posts of each student into one, so as the quoted posts.

Data augmentation is also adopted, since the topic description only consists of several keywords and hyperlinks. And the abstract concepts are hard to understand by the machine. We try to enrich the topic with the text provided by the hyperlinks such as the text content of the Web pages or the subtitles of the videos. For those topics without hyperlinks, we search the abstract concepts in Wikipedia³ and add the text content of the Web pages into the topics' description. With the aforementioned processing, the average topic length is extended from 41.19 words to 576.37 words. Enriching the topic description also improves the performance of post scoring as shown in Section 5.3.4. In Table 4, we show some examples of topic extension.

4 Posts assessment model

In this section, the whole framework is illustrated first. Then, we introduce the hierarchical text model first. Also, we show the cross attention model which is used in the latter two models. Finally, we introduce the matching model and the representation model respectively.

The whole model framework is shown in Figure 2. Specifically, Cross attention is used to calculate the similarity matrix in the bottom part, as well as select sentences from quoted posts in the top part. Finally, the matching feature, post representation, and post aware quoted post representation are concatenated together to calculate the student score.

4.1 Hierarchical text model

In general, there are two ways to represent a long text, the first one treats the long text as a long word sequence, and the latter one constructs a hierarchical model which regards the long text as a sentence sequence, and for each sentence, it is also a word sequence.

³https://en.wikipedia.org/wiki/Main_Page

Table 4 Examples of topic extension via recording subtitles of the video or searching the keywords in wikipedia

Original Topic	Augmented Topic
<p>Based on this discussion on MBTI types (Hyperlink), to what extent is preference towards Anime / Comic / Game (ACG) being influenced by one's personality?</p>	<p>Content of Hyperlink: According to an analytical psychologist, Carl Gustav Jung's theory of psychological types, there are 4 perceiving psychological functions: A. Introverted sensing This function is affected by hormone levels inside one's body, and one very often seeks sense of security there. Such a function forms Keirseyan temperament called Guardian, and this is commonplace in all ordinary ACG fans. The relevant MBTI types for this temper are ISFJ, ISTJ, ESFJ, ESTJ. Some may call them Pure MK because they seek order on-duty and break order off-duty. B. Extraverted Sensing This function allows for originality in production, and this facilitates conceptualization of ideas as well.</p>
<p>Evidence from psychology literature suggests that human memory is far from being very accurate and we are prone to process memory with reference to our biases. From this perspective, could we argue that human memory should be considered inferior to memory in computers and other machines?</p>	<p>Content from Wikipedia: Memory is the faculty of the brain by which data or information is encoded, stored, and retrieved when needed. It is the retention of information over time for the purpose of influencing future action. If past events could not be remembered, it would be impossible for language, relationships, or personal identity to develop. Memory loss is usually described as forgetfulness or amnesia. Memory is often understood as an informational processing system with explicit and implicit functioning that is made up of a sensory processor, short-term (or working) memory, and long-term memory. This can be related to the neuron. ...</p>

In our scenario, students usually write the first post to introduce their main argument, and the rest post to explain and add extra references. So we combine all the posts from the same student into one, and each combined post may have nearly 1000 words on average. In general, there are three sequence models including Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and Transformer. LSTM always faces some difficulties when coping with very long sequences for limited memorization ability. Coping with long sequence, efficiency is always the most significant problem for transformer, such as BERT, a transformer-based pre-train language model, sets the maximum sequence length to 512. CNN model will lose word order information, which is essential to learn text semantic representation. Since the text is too long, it is unreasonable to adopt the first representation method. In this paper, we adopt the second method, the hierarchical text model, to learn the post representation.

In the hierarchical model, a long text is tokenized into several sentences, and each sentence is tokenized into several words. For efficiency reasons, we use LSTM or CNN as the semantic composition model to learn the sentence representations from the word representations. Besides, similar semantic composition models are used to learn the text representations from the sentence representations. In the rest, we will mainly introduce how to learn the sentence representation, since the method to learn the text representation is the same.

To learn the semantic representation of each sentence $p_i = \{w_{i,1}, \dots, w_{i,m_i}\}$, where m_i is the number of words, a word embedding matrix E is constructed first. Then each word $w_{i,j}$ is mapped into a vector $E(w_{i,j})$ via the embedding matrix. With obtained word vectors, LSTM or CNN is used to combine word semantics into the sentence representation.

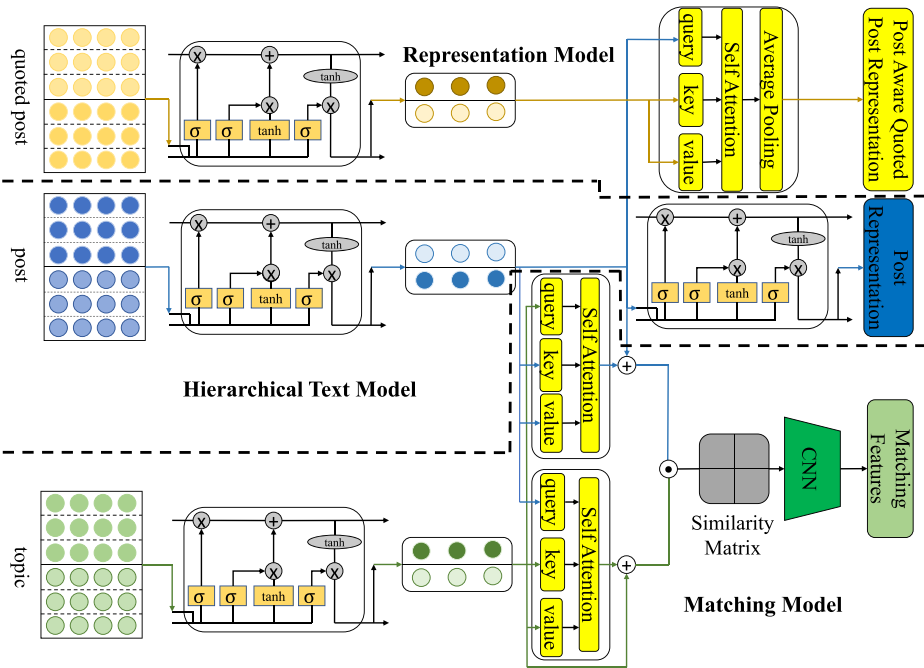


Figure 2 Framework of our mixture model. The hierarchical text model is in the middle of the picture which provides the representations of sentences from posts to the representation model and matching model. The bottom part is the matching model capturing the relevance of the topic and post at the sentence level. And a representation model integrating quoted posts is on the top. All input examples are two sentences with 3 words in each, LSTM is used as an example to learn sentence representations and post representations

LSTM could capture the word order information to learn the syntactic information. Meanwhile, compared with long text, each sentence only has a pretty small number of words, and the gate mechanism in LSTM is powerful enough to cope with these sequences with limited length. To calculate the sentence representation $R_s(p_i)$ via LSTM, we obtain all the hidden states $H(p_i) = [h_0, h_1, \dots, h_{m_i}]$ as shown in Formula 2, then average pooling or attention mechanism is used to combine these hidden states.

$$H(p_i) = \text{LSTM}(\{E(w_{i,1}), E(w_{i,2}), \dots, E(w_{i,m_i})\}) \tag{2}$$

The advantage of average pooling is to capture more early information in each sequence, which will gradually decrease in RNN based models. However, not each word contributes equally to the sentence semantic. Unlike average pooling, attention is proposed to learn different weights for each word. The calculation of the attention weights is shown in Formula 3, where W_1 and W_2 are projection matrices. The sentence representation could be computed by Formula 4.

$$Wt = \text{softmax}(W_2 \cdot \tanh(W_1 \cdot H(p_i)^T)) \tag{3}$$

$$R_s(p_i) = Wt \cdot H(p_i)^T \tag{4}$$

CNN can capture phrase information which is useful to identify abstract concepts in posts. With obtained word embedding sequence $E(p_i)$, the convolutional layer is computed

as Formula 5, where W and b are the parameters and shared across all windows in the sequence. Relu is used as the non-linear activation function and maximum pooling is added to compress the learned features as shown in Formula 6.

$$\text{Conv}(E(p_i)) = W \cdot E(p_i) + b \quad (5)$$

$$R_s(p_i) = \text{MaxPooling}(\text{Relu}(\text{Conv}(E(p_i)))) \quad (6)$$

With learned sentence representations of each post $R_s(p) = \{R_s(p_1), \dots, R_s(p_m)\}$, where m is the number of sentences, Formula 4 and Formula 6 are also used to learn the post representation $R(p)$ by replacing the words embedding sequence $E(p_i)$ with the sentences representations sequence $R_s(p)$ as shown in Formula 7 and Formula 8. Attention mechanism and LSTM could be used together to learn the importance of different sentences.

$$R(p) = \text{AvgPooling}(H(R_s(p))^T) \quad (7)$$

$$R(p) = \text{MaxPooling}(\text{Relu}(\text{Conv}(R_s(p_i)))) \quad (8)$$

In this section, we introduced the whole process that how to implement the hierarchical text model by various combinations of word representations composition and sentence representations composition.

4.2 Cross attention model

Referring to previous work [35], cross attention is a key approach to model the word-level interaction between two sentences or sentence-level interaction between two texts. Cross attention is a variation version of self-attention proposed in Transformer [28].

Cross attention also has three input sequences, namely $Q = [R_s(p_i)]_{i=0}^{n_Q-1}$, $K = [R_s(p_j)]_{j=0}^{n_K-1}$, and $V = [R_s(p_k)]_{k=0}^{n_V-1}$, where n_Q , n_K and n_V denote the number of sentences in each long text, and $R_s(\cdot)$ stands for the sentence representation model, n_K is equal to n_V . The model first takes each sentence in the query text to attend to sentences in the key text via Scaled Dot-Product Attention [28]. Then those attention results were applied upon the value text, which is defined as:

$$\begin{aligned} \text{Att}(Q, K) &= [\text{softmax}(\frac{Q[i] \cdot K^T}{\sqrt{d}})]_{i=0}^{n_Q-1} \\ V_{att}(Q, K, V) &= \text{Att}(Q, K) \cdot V \in R^{n_Q \times d} \end{aligned} \quad (9)$$

where $Q[i]$ is the i_{th} sentence representation in the query text Q and d is the representation size. Each row of V_{att} , denoted as $V_{att}[i]$, stores the fused semantic information of sentences in the value text that possibly have dependencies to the i_{th} sentence in the query text. For each i , $V_{att}[i]$ and $Q[i]$ are then added up together, compositing them into a new representation that contains their joint meanings.

With the detailed implementation of cross attention, the matching model and representation model will be introduced respectively.

4.3 Matching model

Instructors evaluate students' posts by estimating the relevance between the post and the topic. In this paper, the matching model attempts to calculate the semantic matching features

to capture the semantic consistency between the post and the given topic. More specifically, with learned sentences' representations, a cross attention model calculates the interactions between any two sentences to obtain the similarity matrix. Then, CNN is used to extract the matching features.

We utilize cross attention to calculate a similarity matrix $Sim_{p,t} \in \mathbb{R}^{n_p \times n_t}$ between post sentences $R_s(p) = R_s(P(s_i))$ and topic sentences $R_s(t) = R_s(T(s_i))$ via Formula 10. Similar to Formula 6, a two-dimensional convolutional layer and a maximum pooling layer are used to extract the matching feature as shown in Formula 11.

$$\begin{aligned} a &= V_{att}(R_s(p), R_s(t), R_s(t)) \\ b &= V_{att}(R_s(t), R_s(p), R_s(p)) \\ a &= R_s(p) + a \\ b &= R_s(t) + b \\ Sim_{p,t} &= a \cdot b^T \end{aligned} \quad (10)$$

$$\begin{aligned} \text{Conv2D}(Sim_{p,t}) &= \mathbf{W} \cdot Sim_{p,t} + \mathbf{b} \\ R(p, t) &= \text{MaxPooling2D}(\text{Relu}(\text{Conv2D}(Sim_{p,t}))) \end{aligned} \quad (11)$$

In summary, to measure the semantic consistency between posts and given topics, the cross attention model is used to calculate the semantic matching matrix, then CNN is used to extract the matching features.

4.4 Representation model

Quoted posts are also important parts to measure students' posts. Since the quoted arguments in quoted posts reveal the student's understanding of the given topics. The key idea to incorporate quoted posts is similar to the attention mechanism, we select the most relevant sentences from quoted posts as the auxiliary post aware quoted post representation. More specifically, making quoted posts and posts attend to each other, it is significant to capture dependencies between those latently matched segment pairs, which can provide complementary information for post representation. In addition, the cross attention model will not consider the sentence order, we combine all the quoted posts into one post.

With calculated sentence representation sequence for the post $R_s(p) = R_s(P(s_i))$ and that for the combined quoted post $R_s(q) = R_s(Q(s_i))$, post aware quoted post representation $R(q|p)$ is calculated by the cross attention model as shown in Formula 12.

$$R(q|p) = \text{AvgPooling}(V_{att}(R_s(p), R_s(q), R_s(q))) \quad (12)$$

In this section, the cross attention model is used to select sentences from quoted posts referring to posts.

4.5 Scoring function

With learned post representation $R(p)$, matching feature $R(p, t)$, and post aware quoted post representation $R(q|p)$, we get the final representation $R_f(p) = [R(p), R(p, t), R(q|p)]$ via combining the post representation and additional features.

Then a fully connected neural network $FCNN(\cdot)$ is used as the score mapping function. In addition, $\sigma = \text{Sigmoid}(\cdot)$ activation function is used to normalize the score into $[0,1]$ as shown in Formula 13. More specifically, $FCNN$ is a linear function, \mathbf{W} is the weight matrix and \mathbf{b} is the bias. To learn better parameters, the mean score of all students in the training set is used to initialize the bias \mathbf{b} .

$$G(s_i)' = \sigma(\mathbf{W} \cdot R_f(p) + \mathbf{b}) \quad (13)$$

For regression problems, the mean square error is used as the loss of the neural networks. With all the gold scores $\{G(s_i)|i \in [1, N]\}$ and calculated scores $\{G(s_i)'|i \in [1, N]\}$ for each student, the objective function is Formula 14.

$$L = \text{MSE}(G(s_i), G(s_i)') = \frac{1}{N} \sum_{i=1}^N (G(s_i) - G(s_i)')^2 \quad (14)$$

5 Experiment

5.1 Experiment setting

In this paper, we utilize 5-fold cross-validation to evaluate all models with a 60/20/20 split for train, validation, and test sets. All models are trained for 100 epochs and the best model based on the performance on the validation set is selected to evaluate the performance on the test set. We tokenize the text into sentences and words using NLTK,⁴ and normalize all scores range to $[0,1]$. The scores are rescaled back to the original scale for calculating scores of Quadratic Weighted Kappa, Spearman Correlation Coefficient, Pearson Correlation Coefficient, and Rooted Mean Square Error. GloVe⁵ [18] is used to initialize the word embedding matrix, and the dimension is set to 300. Besides, the learning rate is set to 0.00015, and the batch size is set to 16.

5.2 Evaluation metrics

We employ two types of evaluation metrics, namely correlation based measurements including Quadratic Weight Kappa (QWK), Spearman Correlation Coefficient (SCC), and Pearson Correlation Coefficient (PCC), as well as residuals based measurements, Rooted Mean Square Error (RMSE).

QWK measures the agreement between human raters and machine raters quadratically. As the calculation shown in [27], a weight matrix W is constructed firstly as shown in Formula 15, where i and j are the reference rating (the gold ones assigned by human annotator) and the hypothesis rating (calculated by the machine system) respectively, and N is the number of possible ratings.

$$W_{i,j} = \frac{(i - j)^2}{(N - 1)^2} \quad (15)$$

⁴<https://www.nltk.org>

⁵<https://nlp.stanford.edu/projects/glove/>

In addition, a matrix O is also calculated such that $O_{i,j}$ denotes the number of students who obtained a rating i by the human annotator and a rating j by our model. Another matrix E with the expected count is calculated as the outer product of histogram vectors of the two ratings. The matrix E is then normalized such that the sum of elements in E is the same as that of elements in O . Finally, with given matrices O and E , the QWK score is calculated according to Formula 16.

$$\kappa = 1 - \frac{\sum_{i,j} W_{i,j} O_{i,j}}{\sum_{i,j} W_{i,j} E_{i,j}} \quad (16)$$

SCC and PCC are two popular correlation measurements to compare the orders imposed by gold and system scores over all students [26]. More specifically, the Pearson coefficient, commonly denoted by ρ , is defined as the covariance of the two variables divided by the product of their respective standard deviations as shown in Formula 17. $\text{cov}(X, Y)$ refers to the covariance of the two variables, and σ means the standard deviation.

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \quad (17)$$

The Spearman coefficient is calculated by applying the Pearson coefficient to rank transformed data. Both are unaffected by linear transformations of the data. Given vectors x and y , respectively sampling X and Y and each of length n , the sample Pearson coefficient $r_{x,y}$ is obtained by estimating the population covariance and standard deviations from the samples, as defined in Formula 18. Here \bar{x} and \bar{y} denote the sample means.

$$r_{x,y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (18)$$

RMSE is the root of the MSE loss we used, which evaluates the residuals between gold scores and calculated scores as shown in Formula 19.

$$\text{RMSE}(y, y') = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2} \quad (19)$$

5.3 Experiment results and analysis

In this section, we conduct three experiments, the first experiment shows the results of different text representation models that only utilizing the posts' information. The second experiment shows the effectiveness of integrating given topics or quoted posts. The last experiment illustrates the results of combining given topics and quoted posts. Furthermore, we extend the topic description during pre-processing, we also verify the effectiveness of the extension. Since APS is a new application that considers given topics and quoted posts during marking students' online posts, in our experiment, we only verify the effectiveness of our model to integrate given topics and quoted posts.

In general, we have three types of models including hierarchical text models, models integrating topics or quoted posts respectively, and models incorporating topics and quoted posts simultaneously. Names of all models are in the form of **A-B** for hierarchical text

models, and **A-B-D** for the last two types of models. **A** refers to the word composition model and the sentence composition model including **L** means **LSTM**, **LL** means **LSTM** for word composition and sentence composition, **LC** refers to **LSTM** for word composition and **CNN** for sentence composition, **CL** means **CNN** for word composition, and **LSTM** for sentence composition. **B** represents the methods that combine the hidden states calculated by **LSTM**. **AP** means average pooling, and **Attn** means attention methods. **D** shows the representation model or/and matching model used by given topics or/and quoted posts. **T** means integrating **Topics** and **Q** means incorporating **Quoted posts**. **R** refers to the **Representation** model and **M** refers to the **Matching** model. For example, **RQ** uses the **Representation** model to integrate **Quoted posts**, **MT** utilizes the **Matching** model to incorporate given **Topics**, **RQT** means that using the **Representation** model to integrate **Quoted posts** and **Topics**, and **MTRQ** refers to incorporating given **Topics** and **Quoted posts** by the **Matching** model and **Representation** model respectively.

5.3.1 Baseline models

In this subsection, we introduce a basic text representation model, four hierarchical text representation models, and one pre-trained model that only using the posts. The basic text representation model treats each student's posts as a long word sequence, **LSTM** as well as average pooling is used to learn the text representation. The four hierarchical text models utilize **CNN**, **LSTM** to compose word sequences or sentence sequences respectively. Meanwhile, average pooling and attention are used to obtain the sentence or text representations from the hidden states, the output of the **LSTM**. In addition, we also adopt a widely used pre-trained model, **RoBERTa** [13], to learn text representations. All these models will be adopted as baseline models.

- **L-AP LSTM** is used to learn the whole text representation, to alleviate the poor memorization ability, average pooling is used on all the hidden states. More specifically, a two-layer unidirectional **LSTM** is adopted, and the dimension of the hidden state is set to 300.
- **LL-AP** Two two-layer unidirectional **LSTM** models are used to learn sentence representations and post representations. Also, average pooling is utilized to combine the hidden states from the output of word sequences or that of sentence sequences. And 300 is the dimension size of hidden states.
- **LL-Attn** Unlike the previous model, the attention mechanism is used to combine all the hidden states of the word representations. We utilize the additive attention mechanism with an additional linear transformation layer following calculated attention weights, which has been proved to be useful in representation learning.
- **CL-AP CNN** is used to learn sentence representations, and **LSTM** is utilized to learn text representations via average pooling. For the **CNN** model, 100 filters are used and 2 is the filter size.
- **LC-AP LSTM** is used to learn sentence representations via average pooling, and **CNN** is utilized to learn text representations. The 1-D convolutional network is used to combine adjacent sentences. We set the filter size to 2, 3, 4 and also use 100 filters.
- **RoBERTa** **RoBERTa_{BASE}** is used to learn the semantic representation of each student's posts. Since the maximum length of the input sequence is 512 tokens, we combine all the posts of each student into one and truncate the first 510 tokens. More specifically, we set the batch size to 8, and the learning rate to $3E-5$. We fine-tune the model for 30 epochs.

The results of these six models are shown in Table 5. Firstly, the hierarchical model LL-AP performs comparatively even better than the pure text model L-AP. The results show that although the average pooling can alleviate the poor memorization ability, the hierarchical model can largely avoid the shortage of LSTM coping with long sequences. CL-AP achieves the best performance on two evaluation metrics namely SCC and RMSE, and outperforms the other three hierarchical models by a large margin. More specifically, at least 3 percent in QWK, nearly 6 percent in SCC, and approximately 5 percent in PCC. With the analysis of the posts data, students usually utilize various terminology to explain and support his/her points, CNN performs better to learn the phrase information. Meanwhile, the posts are required to be organized logically, LSTM is good at capturing the order information. Pre-trained models have gained great success in natural language understanding and generation tasks, since they can capture the deeply semantic meaning of the input sequences. With the only first 510 tokens of each student's posts, RoBERTa also achieves much higher performance on QWK and PCC, which outperforms the best hierarchical model, CL-AP, more than 3 point on QWK, and near 2 point on PCC. The results verified the significant effectiveness of the pre-trained model on representation learning..

5.3.2 Models using topics and quoted posts respectively

In this subsection, we mainly introduce how the representation model and the matching model utilize given topics and quoted posts respectively. And conducting experiments on these models to verify the effectiveness of incorporating extra texts such as given topics or quoted posts. Also, we verify that the matching model is more suitable to use the topic information, and the representation model takes advantage of integrating quoted posts.

In these models, the hierarchical models are used to learn the representation of posts. And we only use LSTM to learn the sentence representations of given topics and quoted posts. With given hierarchical text models, we show how to incorporate given topics and quoted posts by the representation model and matching model respectively. Based on the three hierarchical models, LL-AP, CL-AP, LC-AP, there are three representation models to use given topics namely **LL-AP-RT**, **CL-AP-RT**, **LC-AP-RT**, and three representation models to utilize quoted posts including **LL-AP-RQ**, **CL-AP-RQ**, **LC-AP-RQ**. Since the LL-Attn model has much more parameters than the representation or matching models, the combined models are hard to converge. We did not show the results of LL-Attn based representation or matching models.

The matching models utilize CNN to extract the matching features, if the matching models are combined with CL-AP or LC-AP, the two CNN models lead to hard convergence. So for the matching model, we only show the results based on LL-AP. For all the matching

Table 5 Experiment results of the basic text model and four hierarchical text models

Model	QWK ↑	SCC ↑	PCC ↑	RMSE ↓
L-AP	0.410	0.459	0.459	2.36
LL-AP	0.440	0.436	0.470	2.38
LL-Attn	0.443	0.437	0.475	2.35
CL-AP	0.474	0.518	0.523	2.25
LC-AP	0.426	0.448	0.463	2.47
RoBERTa	0.511	0.494	0.542	2.27

models, the 2-D convolutional network is used to extract matching features from the similarity matrix, to obtain features from different scales, we set the filter sizes to 3×3 , and 4×4 , and we use 100 filters. **LL-AP-MT** is the model to integrate given topics and **LL-AP-MQ** is the model to utilize quoted posts. Table 6 shows the experimental results of all mentioned models.

By utilizing topic information, all models including LL-AP-RT, LC-AP-RT, and LL-AP-MT, consistently perform better than corresponding hierarchical text models on all evaluation metrics. Especially, LL-AP-MT outperforms all hierarchical text models on three correlation metrics by a large margin, more than 4 percent, and also gains the lowest RMSE score. All these results show that the topic is important to learn more accurate post scores. Besides, LL-AP-MT also outperforms LL-AP-RT, CL-AP-RT, and LC-AP-RT, more than 6 percent on the QWK score, 3 percent on the SCC score, and 4 percent on the PCC score, and obtains the lowest RMSE score. These results prove that the semantic matching model is more suitable for coping with the topic information compared with the representation model. Because CNN based matching model captures sentence-level semantic interactions (the similarity matrix calculated by cross attention) which are helpful to measure the relevance between posts and given topics. Compared with the pre-trained model, RoBERTa, LL-AP-MT also gains better performance on all evaluation metrics. It also proves the effectiveness of incorporating extra topics via the matching model.

To integrate quoted posts, LL-AP-RQ, CL-AP-RQ, LC-AP-RQ, and LL-AP-MQ, consistently outperform corresponding hierarchical text models on all evaluation metrics. More specifically, CL-AP-RQ correlates better with more than 2 percent on the QWK score, 2 percent on the SCC score, and 3 percent on the PCC score. The rest three models correlate better with more than 5 percent on the QWK score, 6 percent on the SCC score, and 6 percent on the PCC score. They also gain lower RMSE scores than that of corresponding hierarchical text models. All these results illustrate that quoted posts are also important to improve the accuracy of predicting students' scores. Furthermore, LL-AP-RQ outperforms LL-AP-MQ on all correlation evaluation metrics by a large margin, more than 2 percent. These results prove that compared with the semantic matching model, the representation model is more effective to integrate quoted posts. Since the representation model used cross attention to select relevant sentences in quoted posts which reveal the student's understanding of quoted posts. Compared with the pre-trained model, RoBERTa, LL-AP-RQ also gains much better

Table 6 Experiment results of matching and representation models

Model	QWK \uparrow	SCC \uparrow	PCC \uparrow	RMSE \downarrow
RoBERTa	0.511	0.494	0.542	2.27
LL-AP	0.440	0.436	0.470	2.38
LL-AP-RT	0.444	0.482	0.504	2.34
LL-AP-MT	0.524	0.539	0.556	2.21
LL-AP-RQ	0.538	0.559	0.572	2.16
LL-AP-MQ	0.517	0.538	0.551	2.26
CL-AP	0.474	0.518	0.523	2.25
CL-AP-RT	0.461	0.507	0.509	2.28
CL-AP-RQ	0.501	0.541	0.559	2.15
LC-AP	0.426	0.448	0.463	2.47
LC-AP-RT	0.446	0.464	0.498	2.33
LC-AP-RQ	0.482	0.511	0.528	2.32

performance on all evaluation metrics. It also proves the effectiveness of incorporating extra quoted posts via the representation model.

5.3.3 Mixture models combining topics and quoted posts

In this section, to verify the effectiveness of combining given topics and quoted posts simultaneously, we introduce three types of models. Firstly, models use the representation model to integrate both texts based on the three basic hierarchical text models, namely **LL-AP-RTQ**, **CL-AP-RTQ**, **LC-AP-RTQ**. Then, the model utilizes the matching model to integrate both texts based on **LL-AP**, **LL-AP-MTQ**. The last model, **LL-AP-MTRQ**, incorporates given topics by the matching model and quoted posts by the representation model. Since in the former experiment, the representation model performs better to integrate quoted posts, while the matching model gains better performance in using given topics. In this experiment, we compare the mixture models with **LL-AP-RQ**, **CL-AP-RQ**, **LC-AP-RQ**, and **LL-AP-MT** as shown in Table 7.

Focusing on the mixture models, **LL-AP-RTQ**, **LL-AP-MTQ**, and **LC-AP-RTQ** show comparable even better performance, while **CL-AP-RTQ** performs much worse than **CL-AP-RQ**. And in Table 6, **CL-AP-RT** also gains lower performance compared with **CL-AP**. It seems that integrating given topics with representation models will decrease the performance of the hierarchical model **CL-AP**. Since the representation model calculates the interaction between sentence representations from two texts. The semantics captured by the CNN model is quite different from those captured by LSTM. The inconsistency of the two representations hurt the performance. **LL-AP-MTRQ**, the model incorporating topics via the matching model and quoted posts via the representation model, outperforms all other models almost on all evaluation metrics. It shows the effectiveness of incorporating topics and quoted posts with the matching model and representation model respectively. As mentioned before, the representation model is suitable for quoted posts, and the matching model takes advantage of given topics. The mixture model integrating both topics and quoted posts performs better.

5.3.4 Extension of topics

In previous experiments, we extend the topic description. We also run hierarchical text models on the original topic description as shown in Table 8. Models with * utilized the original topic descriptions. All models integrating extended topics show better performance than

Table 7 Experiment results of mixture models

Model	QWK ↑	SCC ↑	PCC ↑	RMSE ↓
LL-AP-RQ	0.538	0.559	0.572	2.16
LL-AP-RTQ	0.551	0.587	0.584	2.17
CL-AP-RQ	0.501	0.541	0.559	2.15
CL-AP-RTQ	0.492	0.531	0.545	2.16
LC-AP-RQ	0.482	0.511	0.528	2.32
LC-AP-RTQ	0.519	0.540	0.553	2.25
LL-AP-MT	0.524	0.539	0.556	2.21
LL-AP-MTQ	0.528	0.549	0.556	2.23
LL-AP-MTRQ	0.561	0.582	0.612	2.13

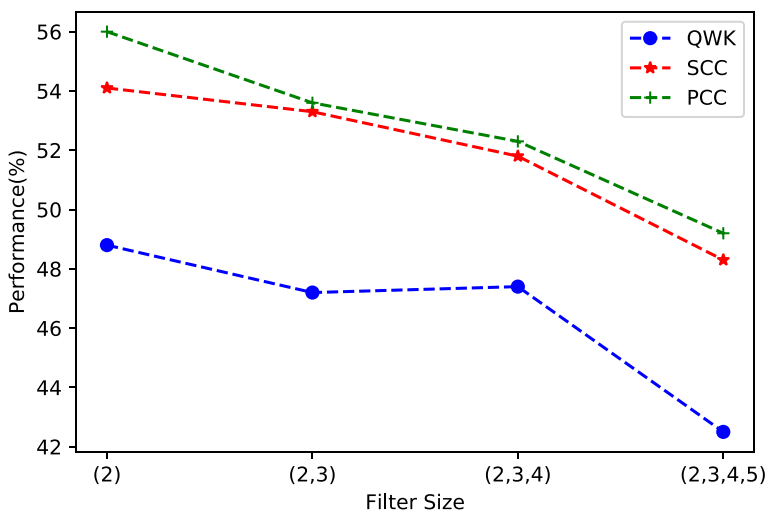
Table 8 Experiment results of models incorporating topics given by instructors

Model	QWK \uparrow	SCC \uparrow	PCC \uparrow	RMSE \downarrow
LL-AP-RT*	0.377	0.431	0.436	2.52
LL-AP-RT	0.444	0.482	0.504	2.34
CL-AP-RT*	0.382	0.408	0.448	2.47
CL-AP-RT	0.461	0.507	0.509	2.28
LC-AP-RT*	0.443	0.454	0.505	2.37
LC-AP-RT	0.446	0.464	0.498	2.33
LL-AP-MT*	0.485	0.485	0.518	2.29
LL-AP-MT	0.524	0.539	0.556	2.21

corresponding models integrating original topics. It proves the effectiveness of extending the descriptions of topics. Compared with all hierarchical text models, LL-AP-RT* and CL-AP-RT* perform worse, while LL-AP-MT* achieves much better performance on all four models. It shows that the matching model can make better use of topics information than the representation model. LL-AP-MT* also gains comparable even better performance than the models utilizing extended topics with the representation model including LL-AP-RT, CL-AP-RT, and LC-AP-RT. The results also prove that the matching model is much suitable to integrate topic information compared with the representation model.

5.3.5 Hyper-parameter analysis

In this section, we conduct additional experiments to show the effect of hyper-parameters. We mainly focus on the filter size of the convolutional neural network in CL-AP, CL-AP-RQ, LC-AP, LC-AP-RQ, LL-AP-MT, and LL-AP-RQMT. Since the representation model performs better to integrate quoted posts, and matching model is suitable for given topics. Then we also conduct experiments to see the influence of different learning rates in LL-AP-RQMT, the best model in our work.

**Figure 3** Experimental results of various models with different filter sizes(CL-AP)

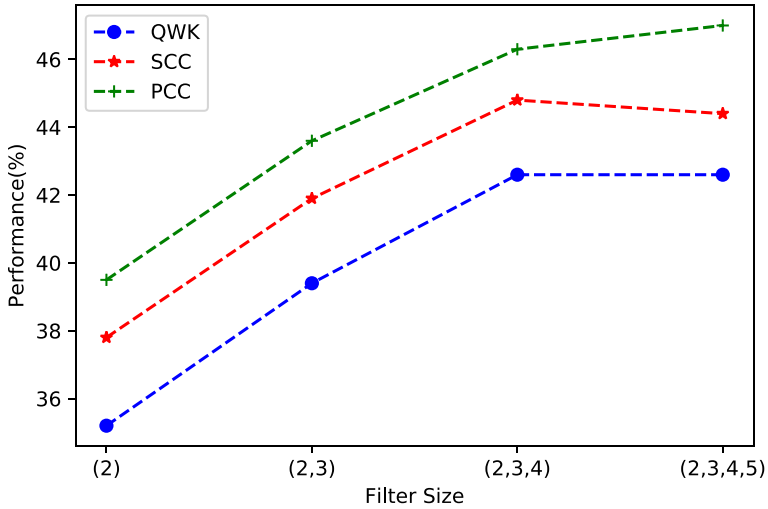


Figure 4 Experimental results of various models with different filter sizes(LC-AP)

Figure 3 shows the experimental results of CL-AP with different filter sizes and evaluation metrics. The results show that with more multi-scale filters, the performance decreases. The filter sizes indicate the window size of adjacent words in each sentence. The reason may be that most key phrases contain two words.

Figure 4 illustrates the performance of LC-AP. The filter size refers to the window size of adjacent sentences. The results show that considering a proper number of sentences, we can obtain better text representations.

The influence of different filter sizes with different evaluation metrics of CL-AP-RQ is shown in Figure 5. The performance shows similar trends with CL-AP. Since using the representation model to incorporate quoted posts will not affect the representations of posts.

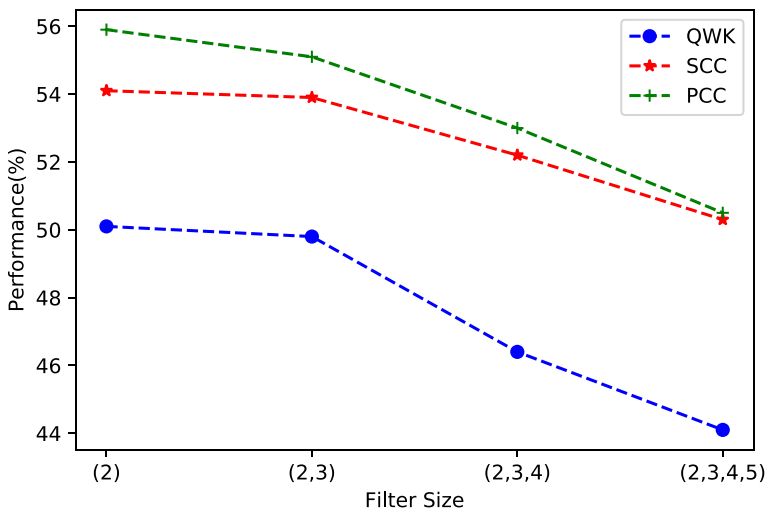


Figure 5 Experimental results of various models with different filter sizes(CL-AP.RQ)

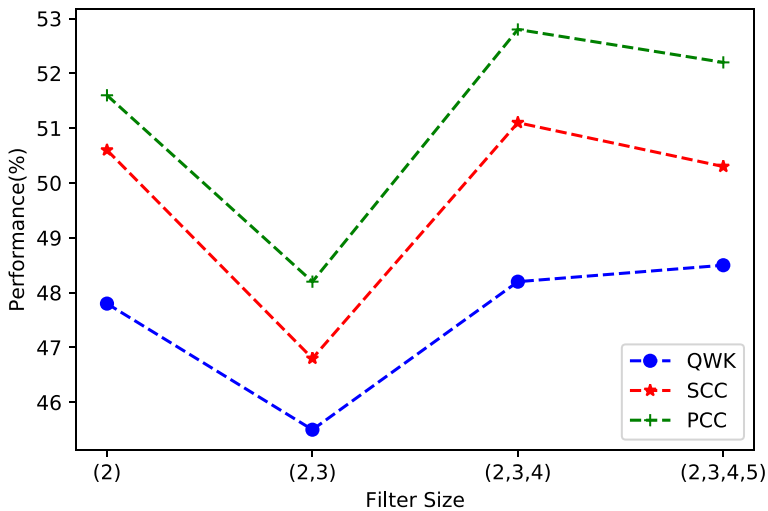


Figure 6 Experimental results of various models with different filter sizes(LC-AP-RQ)

In Figure 6, we also show the influence of different filter sizes on LC-AP-RQ. With extra quoted posts, the filter size has more effect on the performance.

In Figure 7, the experimental results illustrate the influence of different filter sizes on the matching model, LL-AP-MT. Proper multi-scale filter sizes gain better performance.

We also show filter sizes' influence in Figure 8 of the mixture model, LL-AP-MTRQ. Since the model is more complex, multi-scale filter sizes do gain better performance.

With the illustration of the influence of different filter sizes, we also conduct experiments to see how the learning rate affects the mixture model, LL-AP-MTRQ.

Figure 9 shows the performance of LL-AP-MTRQ with different learning rates. Slightly higher learning rates lead to better performance.

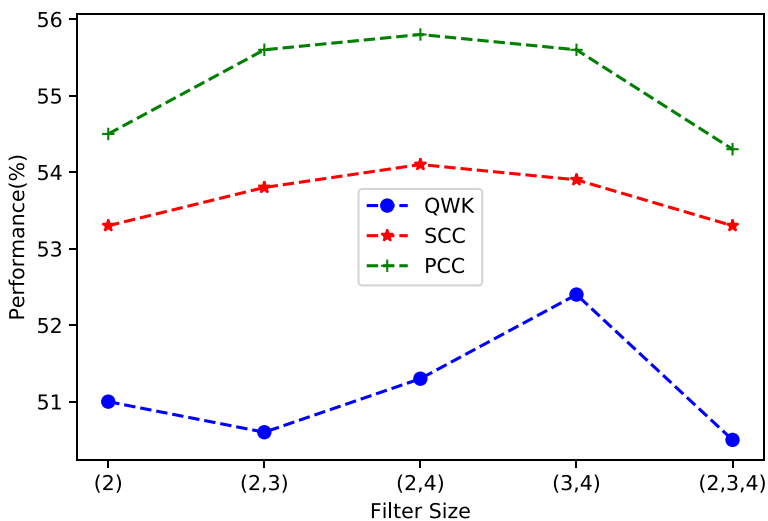


Figure 7 Experimental results of various models with different filter sizes(LL-AP-MT)

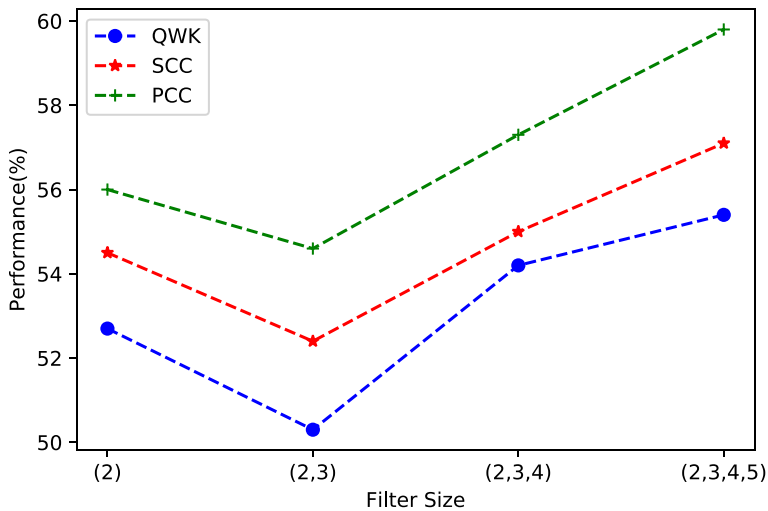


Figure 8 Experimental results of various models with different filter sizes(LL-AP-MTRQ)

5.3.6 Model parameters analysis

In this section, we compare the model complexity measured by the number of the parameters of hierarchical text models, as well as representation models and matching models. Meanwhile, we also show the running time of different models. All the statistics are shown in Table 9.

We calculate the number of parameters of various models first. For the representation models, we utilize multiple attention to implement the self-attention, so these models do not require extra parameters. For the models using convolutional neural networks, the total number of the parameters highly relies on the multi-scale filter sizes. For LL-AP-MTRQ,

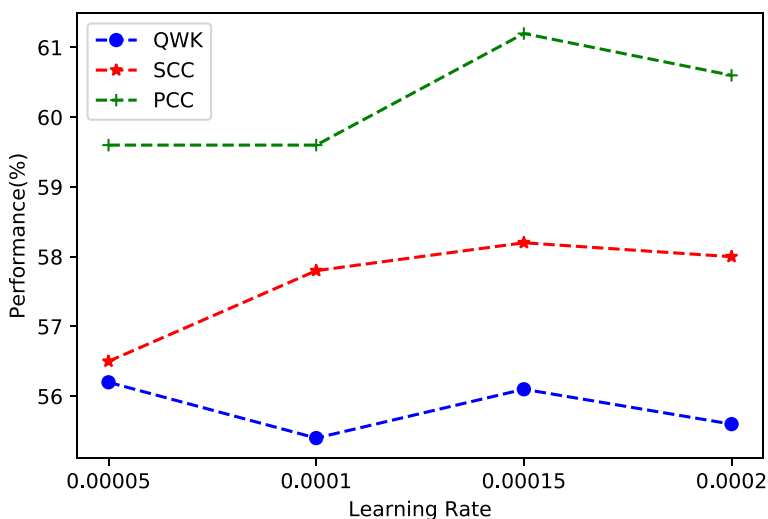


Figure 9 Experimental results of LL-AP-MTRQ with different learning rates

Table 9 Experiment results of models incorporating topics given by instructors

Model	LL-AP	CL-AP	LC-AP	LL-AP-RQ
#Parameters	1.8M	0.96M	1.17M	1.8M
Runtime (BPS)	14.0	18.0	15.0	7.7
Model	CL-AP-RQ	LC-AP-RQ	LL-AP-MT	LL-AP-MTRQ
#Parameters	0.96M	1.17M	>1.8M	>1.8M
Runtime (BPS)	8.5	8.1	5.2	5.1

the representation model has no extra parameters, and the matching model contains several 2-D filters. However, the total number of the parameters of the filters is significantly less than that of the LSTM. So we use $\approx 1.8M$ as the number of the parameters of the LL-AP-MTRQ model. So as LL-AP-MT. The total number of parameters of the filters is extremely small than that of LSTM.

To compare the run-time of each model, we utilize Batch Per Second (BPS) as the metric, which means that how many batches can be trained during one second. Although representation models do not require extra parameters, they still contain complicated calculations. Compared with the three hierarchical models including LL-AP, CL-AP, and LC-AP, LL-AP-RQ, CL-AP-RQ, as well as LC-AP-RQ have the same number of parameters as the corresponding models. However, these models require more runtime.

6 Conclusion and future works

In this paper, we propose a new task called APS to measure the writing quality and relevance of the posts with extra topics and quoted posts. To solve the proposed task, we propose a mixture model including a hierarchical text model to measure the writing quality, a semantic matching model to utilize topics, and a representation model to integrate quoted posts. Experimental results show that integrating topics and quoted posts can improve the performance on the correlation metrics by more than 6 percent. These models also perform better than integrating topics or quoted posts respectively. Furthermore, integrating topics via the matching model and quoted posts via the representation model achieves the best performance almost on all evaluation metrics, which proves that the semantic matching model is suitable to integrate topics and the representation model can make better use of quoted posts.

Although our experiments obtained higher performance, there are still several limitations. On the one hand, the accuracy is not higher enough to assess each student as the instructor. On the other hand, the models can not be interpreted. So there are two future directions, one is to design more accurate models, and the other one is to explain to the models so that instructors and students could believe the calculated scores. Currently, these models can be only used to assist the assessment of instructors and alleviate their burden.

Acknowledgments The work was supported by PolyU Teaching Development with project code 1.61.xx.9A5V and the Hong Kong Jockey Club Charities Trust (PolyU Project ID: P0038517). It was also supported by PolyU Internal Start-up Fund (P0035274).

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Attali, Y., Burstein, J.: Automated essay scoring with e-rater® v. 2. *The Journal of Technology Learning and Assessment* 4(3) (2006)
2. Beigman Klebanov, B., Flor, M., Gyawali, B.: Topicality-based indices for essay scoring. In: *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 63–72. Association for Computational Linguistics, San Diego (2016). <https://doi.org/10.18653/v1/W16-0507>, <https://www.aclweb.org/anthology/W16-0507>
3. Chandrasekaran, D., Mago, V.: Evolution of semantic similarity—a survey. *ACM Comput Surv (CSUR)* 54(2), 1–37 (2021)
4. Chen, H., He, B.: Automated essay scoring by maximizing human-machine agreement. In: *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1741–1752. Association for Computational Linguistics, Seattle (2013). <https://www.aclweb.org/anthology/D13-1180>
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp 4171–4186 (2019)
6. 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)*, pp. 153–162. Association for Computational Linguistics, Vancouver (2017). <https://doi.org/10.18653/v1/K17-1017>, <https://www.aclweb.org/anthology/K17-1017>
7. Elliot, S.: Intellimetric: from here to validity. *Automated essay scoring: a cross-disciplinary perspective*, 71–86 (2003)
8. Heilman, M., Madnani, N.: ETS: domain adaptation and stacking for short answer scoring. In: *Proceedings of the 7th international workshop on semantic evaluation, SemEval@NAACL-HLT 2013*, Atlanta, Georgia, USA, June 14–15, 2013, pp 275–279. <https://www.aclweb.org/anthology/S13-2046/> (2013)
9. Higgins, D., Burstein, J., Marcu, D., Gentile, C.: Evaluating multiple aspects of coherence in student essays. In: *Proceedings of the human language technology conference of the North American chapter of the association for computational linguistics: HLT-NAACL 2004*, pp 185–192 (2004)
10. Jiménez, S., Becerra, C.J., Gelbukh, A.F.: SOFTCARDINALITY: hierarchical text overlap for student response analysis. In: *Proceedings of the 7th international workshop on semantic evaluation, SemEval@NAACL-HLT 2013*, Atlanta, Georgia, USA, June 14–15, 2013, pp 280–284. <https://www.aclweb.org/anthology/S13-2047/> (2013)
11. Kumar, S., Chakrabarti, S., Roy, S.: Earth mover’s distance pooling over siamese lstms for automatic short answer grading. In: *Proceedings of the twenty-sixth international joint conference on artificial intelligence, IJCAI 2017*, Melbourne, Australia, August 19–25, 2017, pp 2046–2052 (2017). <https://doi.org/10.24963/ijcai.2017/284>
12. Landauer, T.K.: Automated scoring and annotation of essays with the intelligent essay assessor. *Automated essay scoring. A cross-disciplinary perspective* (2003)
13. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: a robustly optimized bert pretraining approach, arXiv:1907.11692 (2019)
14. Louis, A., Higgins, D.: 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*, pp 92–95 (2010)
15. Mohler, M., Bunesco, R., Mihalcea, R.: Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In: *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, vol . Association for Computational Linguistics1*, pp 752–762 (2011)
16. Ott, N., Ziai, R., Hahn, N., Meurers, D.: Comet: integrating different levels of linguistic modeling for meaning assessment. In: *Proceedings of the 7th international workshop on semantic evaluation, SemEval@NAACL-HLT 2013*, Atlanta, Georgia, USA, June 14–15, 2013, pp 608–616. <https://www.aclweb.org/anthology/S13-2102/> (2013)
17. Paudel, P.: Online education: benefits, challenges and strategies during and after covid-19 in higher education. *Int J Stud Educ* 3(2), 70–85 (2021)
18. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. <https://www.aclweb.org/anthology/D14-1162/> (2014)
19. Riordan, B., Horbach, A., Cahill, A., Zesch, T., Lee, C.M.: Investigating neural architectures for short answer scoring. In: *Proceedings of the 12th workshop on innovative use of NLP for building educational applications, BEA@EMNLP 2017*, Copenhagen, Denmark, September 8, 2017, pp 159–168. <https://www.aclweb.org/anthology/W17-5017/> (2017)

20. Saha, S., Dhamecha, T.I., Marvaniya, S., Sindhgatta, R., Sengupta, B.: Sentence level or token level features for automatic short answer grading?: Use both. In: Artificial intelligence in education - 19th international conference, AIED 2018, London, UK, June 27-30, 2018, Proceedings, Part I, pp 503–517. https://doi.org/10.1007/978-3-319-93843-1_37 (2018)
21. Shermis, M.D., Burstein, J.C.: Automated essay scoring: a cross-disciplinary perspective. Routledge (2003)
22. Smolentzov, A.: Automated essay scoring. Scoring essays in Swedish (2013)
23. Sultan, M.A., Salazar, C., Sumner, T.: Fast and easy short answer grading with high accuracy. In: NAACL HLT 2016, the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, San Diego California, USA, June 12-17, 2016, pp 1070–1075. <https://www.aclweb.org/anthology/N16-1123/> (2016)
24. 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, pp. 1882–1891. Association for Computational Linguistics, Austin (2016). <https://doi.org/10.18653/v1/D16-1193>, <https://www.aclweb.org/anthology/D16-1193>
25. Tay, Y., Phan, M.C., Tuan, L.A., Hui, S.C.: Skipflow: incorporating neural coherence features for end-to-end automatic text scoring. In: Proceedings of the thirty-second AAAI conference on artificial intelligence, (AAAI-18), the 30th innovative applications of artificial intelligence, pp 5948–5955. <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16431> (2018)
26. Van Dongen, S., Enright, A.J.: Metric distances derived from cosine similarity and pearson and spearman correlations. arXiv:1208.3145 (2012)
27. Vanbelle, S.: A new interpretation of the weighted kappa coefficients. *Psychometrika* **81**(2), 399–410 (2016)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems 30: annual conference on neural information processing systems 2017, 4-9 December 2017, Long Beach, CA, USA, pp 5998–6008. <http://papers.nips.cc/paper/7181-attention-is-all-you-need> (2017)
29. Wang, Y., Wei, Z., Zhou, Y., Huang, X.: Automatic essay scoring incorporating rating schema via reinforcement learning. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp. 791–797. Association for Computational Linguistics, Brussels, Belgium (2018). <https://doi.org/10.18653/v1/D18-1090>, <https://www.aclweb.org/anthology/D18-1090>
30. Wishart, C., Guy, R.: Analyzing responses, moves, and roles in online discussions. *Interdiscip J E-Learn Learn Objects* **5**(1), 129–144 (2009)
31. 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: Findings of the association for computational linguistics: EMNLP 2020, pp. 1560–1569. Association for Computational Linguistics, Online (2020). <https://doi.org/10.18653/v1/2020.findings-emnlp.141>, <https://www.aclweb.org/anthology/2020.findings-emnlp.141>
32. Yang, Y., Cao, J., Shen, J., Yang, R., Wen, Z.: Blended learning. Education in a smart learning environment - 13th international conference, icbl 2020, Bangkok, Thailand, August 24-27, 2020, proceedings, lecture notes in computer science, vol 12218. Springer, pp 15–24. In: Cheung, S.K.S., Li, R., Phusavat, K., Paoprasert, N., Kwok, L.F. (eds.) (2020). https://doi.org/10.1007/978-3-030-51968-1_2
33. Yannakoudakis, H., Briscoe, T., Medlock, B.: 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, pp. 180–189. Association for Computational Linguistics, Portland (2011). <https://www.aclweb.org/anthology/P11-1019>
34. Yuan, J., Kim, C.: Guidelines for facilitating the development of learning communities in online courses. *J Comput Assist Learn* **30**(3), 220–232 (2014)
35. Zhou, X., Li, L., Dong, D., Liu, Y., Chen, Y., Zhao, W.X., Yu, D., Wu, H.: Multi-turn response selection for chatbots with deep attention matching network. In: Proceedings of the 56th annual meeting of the association for computational linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, volume 1: Long Papers, pp. 1118–1127 (2018). <https://doi.org/10.18653/v1/P18-1103>, <https://www.aclweb.org/anthology/P18-1103/>