MOC: Measuring the Originality of Courseware in Online Education Systems

Jiawei Wang
CVTE Research
wangjiawei0531@gmail.com

Jiansheng Fang
CVTE Research
fangjiansheng@cvte.com

Jiao Xu
CVTE Research
xujiao@cvte.com

Shifeng Huang
CVTE Research
huangshifeng@cvte.com

Da Cao
Hunan University
caoda0721@gmail.com

Ming Yang
CVTE Research
yangming@cvte.com

ABSTRACT

In online education systems, the courseware plays a pivotal role in helping educators present and impart knowledge to students. The originality of courseware heavily impacts the choice of educators, because the teaching content evolves and so does courseware. However, how to measure the originality of a courseware is a challenging task, due to the lack of labels and the difficulty of quantification. To this end, we contribute a similarity ranking-based unsupervised approach to measure the originality of a courseware. In particular, we first exploit a pre-trained deep visual-text embedding to obtain the representations of images and texts in a local manner. Next, inspired by the design of capsule neural network, a vector-based pooling network is proposed to learn multimodal representations of images and texts. Finally, we propose a Discriminator to optimize the model by maximizing the mutual information between local features and global features in an unsupervised manner. To evaluate the performance of our proposed model, we further subtly collect a dataset for evaluating the originality of courseware by treating sequential versions of each courseware as ranking lists. Therefore, the learning-to-rank scheme can be utilized to evaluate the similarity-based ranking performance. Extensive experimental results have demonstrated the superiority of our proposed framework as compared to other state-of-the-art competitors.

CCS CONCEPTS

• **Information systems** → **Multimedia and multimodal retrieval**; **Novelty in information retrieval**; • **Social and professional topics** → **K-12 education**.

KEYWORDS

Measurement of Originality; Online Education Systems; Multimodal Learning; Unsupervised Learning

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3351087

ACM Reference Format:
Jiawei Wang, Jiansheng Fang, Jiao Xu, Shifeng Huang, Da Cao, and Ming Yang. 2019. MOC: Measuring the Originality of Courseware in Online Education Systems. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3351087

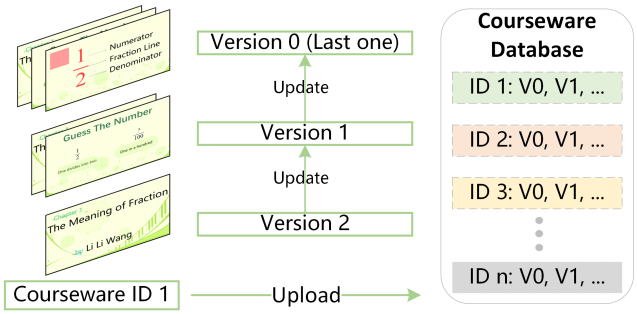


Figure 1: The illustration of a courseware, which is composed of visual and textual content and is continuously updated.

1 INTRODUCTION

Boosted by the proliferation of web technologies, student-oriented education platforms (e.g., KhanAcademy[1], Yuansouti[2], and Zybang[3]) have emerged rapidly in the last decade. Millions of exercises and questions have been accumulated on these platforms and various research topics have been investigated, such as exercise recommendation [17], cognitive analysis [35, 49], and student performance prediction [32]. However, on the other side of the coin, teacher-oriented education platforms (e.g., SmartTech[4], SeeWo[5] and Hite Vision[6]) and their applications also have great impact on online education systems, but they are relatively unexplored. In these platforms, educators frequently upload and update courseware to facilitate the teaching activities. As revealed in Figure 1, the courseware is manually created by integrating textual and visual content and updated by its creators. The sheer number of courseware with frequently updated versions make it difficult for educators to locate their desired courseware. In addition, due to the fact that many teachers create and update courseware by learning from others, there are a lot of similar courseware accumulated in the cloud server. If a user employs the search engine to locate a

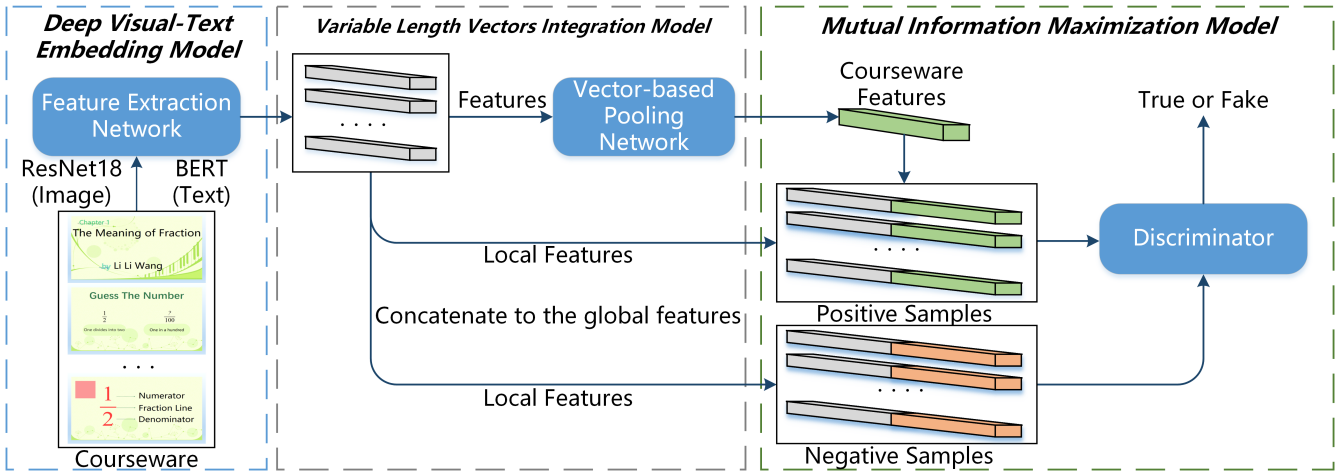


Figure 2: The graphical representation of our propose framework, which is composed of *Deep Visual-text Embedding*, *Variable-length Vectors Integration*, and *Mutual Information Maximization*.

desired courseware without considering the originality or diversity, it may end up with similar ones and adversely hurt the user's experience. To provide a high-quality courseware retrieval service (e.g., courseware search and courseware recommendation), it is inevitable to take the factor of originality into account. The intuition behind is that we believe a courseware with high originality is more likely to be adopted by educators. Thereby, the ability to measure the originality of courseware becomes an urgent task.

Despite its value and significance, measuring the originality of courseware remains immature due to the following challenges: (1) The measurement of originality is a relatively subjective issue. Different people have different perspectives, even for the same courseware. Therefore, how to find a unified and reliable way to measure the originality of courseware is a non-trivial task. (2) The data on courseware usually exhibit dual-heterogeneities, consisting of textual and visual content. The heterogeneous data is the key evidence for us to understand the rationale for the measurement of originality. How to effectively uncover the information embedded in the hybrid data and seamlessly sew them up remain largely unaddressed research problem. (3) Measuring the originality of courseware is a new task in the research community, which will bring customers with surprising results by encouraging originality. The research topic is new, so is the evaluation method. Thus how to evaluate the performance of an originality-oriented algorithm is of great interest.

To tackle aforementioned challenges, we contribute a novel solution, namely MOC (short for "Measuring the Originality of Courseware"), which transforms the measurement of originality as a similarity ranking problem. Under this paradigm, the ranking task is performed by a ranking model $f(q, k)$, where q denotes the query courseware and k denotes the candidate courseware in the database. If the scores of returned top-k courseware are low, the query courseware has a high originality. The framework of MOC is illustrated in Figure 2. Specifically, we first propose a *Deep Visual-text Embedding* module to obtain the features of both visual and textual content in a common space. Thereafter, we propose a *Variable-length Vectors Integration* method with a vector-based pooling network to acquire the global representations of courseware. During this stage, the local features distilled from the common space are transformed to a global fixed-length feature. Furthermore, inspired by [19], a mutual information estimator Discriminator is proposed to maximize the mutual information between local features and the global feature, which is optimized under an unsupervised setting. Last but not the least, to evaluate the performance of our proposed framework, we subtly construct a test set by taking sequential versions of each courseware as ranking lists. Then our evaluation goal is transformed into a learning-to-rank problem [26]. Therefore, three automatic evaluation metrics are utilized to measure the similarity ranking performance.

The main contributions of this work are summarized as follows:

- We explore the promising yet challenging problem of measuring the originality of courseware, which, to the best of our knowledge, is the first time in the research fields of both multimedia and education.
- We develop a novel framework, MOC, to solve the problem of measuring the originality of courseware by jointly integrating the components of *Deep Visual-text Embedding*, *Variable-length Vectors Integration*, and *Mutual Information Maximization*.
- We collect the first courseware dataset for measuring the originality of courseware, and propose an automatic evaluation metric to objectively evaluate the model performance. Extensive experiments are conducted on a self-constructed dataset to demonstrate the effectiveness and rationality of our solution.

2 RELATED WORK

Three sub-directions in the multimedia field are tightly related to our work, namely, pooling techniques, learning-to-rank methods, and unsupervised learning strategies.

[1]https://www.khanacademy.org
[2]https://www.yuansouti.com
[3]https://www.zybang.com
[4]https://www.smarttech.com
[5]https://www.seewo.com
[6]https://www.hitecloud.cn/res/entranceRescource

2.1 Pooling Techniques

In our framework, one of the most important step is to integrate heterogeneous courseware materials to a fixed-length representation, which is related to the pooling techniques. In the current deep learning research, popular pooling functions including max pooling, average pooling and stochastic pooling [24, 46] have been widely used. These simple pooling operations will completely lose the location information of the feature. For example, the most popular pooling technique max pooling [25] only retains the maximum value and does not consider the other possibly useful features. These forms of pooling techniques are deterministic, efficient, and simple, but have the weakness of hindering the potential for learning optimal network [25]. In another direction, Long Short-Term Memory (LSTM) [20] is also a classic way to solve the problem of variable-length inputs. However, LSTM is not well perform in the long distance propagation, which results in the loss of previous information [16].

Different from aforementioned algorithms, the pooling network in our framework is implemented in a vector-based parallel computing manner. It aligns the semantic space between image and text by using a shared weights capsule neural network.

2.2 Learning-to-Rank Methods

Learning-to-rank is useful for multimedia retrieval [6, 40, 47], recommendation system [7, 41], and many other applications [3, 4, 8, 30]. Traditional ranking methods using word frequency, inverse document frequency and document length as features are too simplified, and are limited in the field of information retrieval [38]. At present, some learning-to-rank methods based on machine learning algorithms are proposed. Compared to the traditional methods, these learning-to-rank algorithms aim to directly generating document ranking results. Pointwise method [9, 11, 30] first converts the document into a feature vector, then uses a machine learning algorithm to score the document according to the classification or regression function learned from the training data. Pairwise approach [10, 48] mainly transforms the ranking problem into a discriminate problem of the document order relationship. The Listwise approach [42, 44, 45] addresses the ranking problem in a more straightforward way, which takes ranking lists as instances in both learning and prediction. The group structure of ranking is maintained and ranking evaluation measures can be more directly incorporated into the loss function in learning.

In our online education scenario, our goal is to perform the originality measurement, the supervised learning approach is not applicable in our case. Therefore, we propose an unsupervised multimodal learning algorithm to learn the representation of a courseware and use learning-to-rank evaluation metrics to indirectly evaluate the proposed model.

2.3 Unsupervised Learning Strategies

There are many popular unsupervised learning strategies for learning image or text representations. In the computer vision field, Deep Adaptive Clustering (DAC) [5], Deep Adaptive Clustering [43], and Noise as Targets [2] approaches are proposed to learn high-level representations from the auxiliary classification task. In another direction, generative models are also commonly used for building representations, such as autoencoders (AE) [37, 39] and flow-based generative models NICE, RealNVP and Flow [14, 15, 23]. And in the filed of natural language processing, learning widely applicable representations of words has been an active area of research. Different language model such as Word2Vec, Glove and BERT have shown to be effective for improving many language processing tasks [13, 29, 31].

However, few of aforementioned methods are available in the multimodal scenario. So we resort to a mutual information-based approach DeepInfo-Max (DIM) [19], which trains an encoder model to maximize the mutual information between a high-level global representation and local parts of the input.

3 FRAMEWORK

In our framework, we aim to perform the similarity-based ranking task by learning the global representation of each courseware with the *Mutual Information Maximization* between local features and global features. As shown in Figure 2, we divide our model into three parts: (1) A *Deep Visual-text Embedding* module is proposed to model the image and text in courseware, the representation of image is obtained from a residual network [18], and the representation of text is acquired from a Bi-Transformer-based language model BERT [13]. (2) Due to the fact that different courseware contains variable numbers of images and page texts, we propose a *Variable-length Vectors Integration* method to obtain a global representations of courseware, which is inspired by [34]. (3) Follow the work of [34], we define a *Mutual Information Maximization* strategy to learn global representations by maximizing the mutual information between the input and output of *Variable-length Vectors Integration*.

3.1 Deep Visual-text Embedding

In this work, we denote the image as I_i^j and the text as T_i^j, which refer to the j^{th} image and the j^{th} page text of the i^{th} courseware, respectively. In order to obtain the representations of the inputs, we feed the I_i^j and T_i^j to the *Deep Visual-text Embedding* module, and utilize the final hidden states as the image feature $\mathbf{f}_I^{ij} \in \mathbb{R}^{D_I}$ and text feature $\mathbf{f}_T^{ij} \in \mathbb{R}^{D_T}$, where D_I and D_T represent the dimensions of image feature and text feature. And then, we transform the image feature $\mathbf{f}_I$ and page text feature $\mathbf{f}_T$ to a unify multimodal embedding space:

$$\begin{cases} \mathbf{x}_I = \mathbf{W}_I \cdot (\mathbf{f}_I) + \mathbf{b}_I \\ \mathbf{x}_T = \mathbf{W}_T \cdot (\mathbf{f}_T) + \mathbf{b}_T \end{cases}, \tag{1}$$

where $\mathbf{W}_I \in \mathbb{R}^{D \times D_I}$ and $\mathbf{W}_T \in \mathbb{R}^{D \times D_T}$ are embedding matrices, $\mathbf{b}_I \in \mathbb{R}^D$ and $\mathbf{b}_T \in \mathbb{R}^D$ are bias vectors.

With the help of *Deep Visual-text Embedding* module, we obtain the embedding parameters $\mathbf{W}$, $\mathbf{b}$ and learn the multimodal representations of the input courseware $\mathbf{x}$.

3.2 Variable-length Vectors Integration

Since a courseware has an arbitrary number of images and pages, a fixed-dimensional representation of the input sequence is not available by employing the *Deep Visual-text Embedding* component only. In fact, if we aggregate the final hidden states using naive pooling techniques, such as mean pooling and max pooling, to form a global representation, it would definitely lead to the information

loss for measuring similarity. Therefore, we proposed a *Variable-length Vectors Integration* method to tackle this problem. In order to measure the similarity between courseware, a good courseware representation has to satisfy at least two criteria: (1) the representation has been projected to a fixed-dimensional vector space, and (2) the representation presents accurate information of images and texts in a courseware. Based on these two criteria, we proposed a *Variable-length Vectors Integration* method with a vector-based pooling network.

Inspired by [34], we treat each vector obtained from the embedding space after squashing function as an entity that is presented in the courseware. The activations of the vector represent various attributes of the entity, which may show that there are some different objects within the image, or describe the semantic contents within the text. The goal of our method is to integrate all of the entity information of a courseware into a fixed-dimensional vector. To this end, instead of using naive pooling techniques to gather all of the local features, we proposed a vector-in-vector-out pooling network which is parameterized by the capsule network [34].

3.2.1 Normal Vector-based Pooling Network. Let $\mathbf{s}_j$ denotes the j^{th} high-level vector, which is a weighted sum of the predicted vectors $\mathbf{u}_{ij}$ from the low-level vectors $\mathbf{u}_i$. The predicted vectors $\mathbf{u}_{ij}$ is computed by a weight matrix $\mathbf{W}_{ij}$:

$$\mathbf{u}_{ij} = \mathbf{W}_{ij} \cdot \mathbf{u}_i. \tag{2}$$

$\mathbf{s}_j$ is weighted by a coupling coefficients c_{ij} and the predicted vectors $\mathbf{u}_{ij}$:

$$\mathbf{s}_j = \sum_i c_{ij} \mathbf{u}_{ij}. \tag{3}$$

Thereafter, a squashing function is applied to $\mathbf{s}_j$ to get a high-level activation vector $\mathbf{v}_j$:

$$\mathbf{v}_j = \frac{||\mathbf{s}_j||^2}{1 + ||\mathbf{s}_j||^2} \frac{\mathbf{s}_j}{||\mathbf{s}_j||}. \tag{4}$$

The coupling coefficients c_{ij} shows the correlation between the i^{th} entity vector and all the entity vectors above, which is determined by a correlation logit b_{ij}. The coupling coefficient c_{ij} is expressed as:

$$c_{ij} = \frac{exp(b_{ij})}{\sum_k exp(b_{ik})}, \tag{5}$$

where the b_{ij} is computed by the accumulated dot product of $\mathbf{u}_{ij}$ and $\mathbf{v}_j$, which can be treated as the agreement from the i^{th} low-level entity to the j^{th} high-level entity:

$$b_{ij} = b_{ij} + \mathbf{u}_{ij} \cdot \mathbf{v}_j. \tag{6}$$

We first set all the correlation logits b_{ij} with an initial value of 0, and apply (5)->(3)->(4)->(6) process three times to compute the high-level activation vector, which is named as dynamic routing.

3.2.2 Vector-based Pooling Network with Shared Weights. So far, we have discussed the vector-based pooling network on how to compute the output vector $\mathbf{v}_j$ from the input vector $\mathbf{u}_i$. However, the network can only deal with the fixed-length input. What if we want to handle the outputs of the *Deep Visual-text Embedding* module with variable-length vectors. For this problem, we apply shared weights matrix $\mathbf{W}_{ij}$ across the entire input vectors, which is represented as:

$$\mathbf{W}_{ij} = \mathbf{W}_{kj}, \forall i, k \in 1, 2, ..., N, \tag{7}$$

where N is the number of inputs, and is variable in different courseware. These replicated weights matrix share the same parameters and form the high-level predicted vectors $\mathbf{u}_{ij}$ with fixed length. The idea behind this is that if the j^{th} entity detects a pattern from the i^{th} entity by a weight matrix $\mathbf{W}_{ij}$, it should also can be used to detect the other entity from the same embedding space. There is no need to relearn a weight matrix for every input. In other words, the shared weights matrix not only addresses the problem of variable-length inputs, but also utilizes the entity-invariant structure as an inductive bias.

Through the *Variable-length Vectors Integration* method, the final hidden states $\mathbf{s}_j$ of the vector-based pooling network holds the semantic information of the whole input vectors $\mathbf{x}$, so we can take all of the final hidden states $\mathbf{s}_j$ as the unified semantic representations for the courseware.

3.3 Mutual Information Maximization

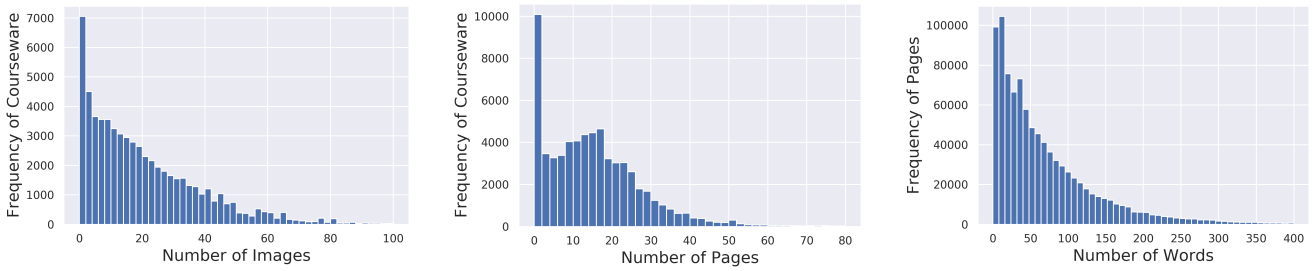
Our approach to learn the vector-based pooling network relies on *Mutual Information Maximization* strategy, which means we aim to find a global representation that contains the local information of the entire inputs, represented by a courseware feature $\mathbf{z} \in \mathbb{R}^F$. We define the local features as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\} \in \mathbb{R}^{N \times D}$, where $\mathbf{x}_i$ denotes the image features $\mathbf{x}_I$ or text features $\mathbf{x}_T$ from the embedding space. As a proxy for maximizing the local mutual information between $\mathbf{x}_i$ and $\mathbf{z}$, we adopt a Discriminator, $\mathcal{D} : \mathbb{R}^{N \times D} \times \mathbb{R}^F \rightarrow \mathbb{R}$, which assigns a probability score to the local-global pair. If the pair comes from the same courseware (positive sample), the probability score should be higher. For the negative sample, in which the global feature $\mathbf{z}$ is from another courseware, $\mathcal{D}$ will assign a lower probability score. For the loss function, we use a binary cross-entropy as follows:

$$\max_{\mathcal{D}} \mathop{\mathbb{E}}_{x,z \sim p(x,z)} [log(D(x,z))] + \mathop{\mathbb{E}}_{x,z \sim p(x)p(z)} [log(1 - D(x,z))], \tag{8}$$

where $p(\mathbf{x}, \mathbf{z})$ denotes the joint distribution of local features and global features, while $p(\mathbf{x})$ and $p(\mathbf{z})$ are marginal distributions. This approach corresponds to the Jensen-Shannon divergence of the joint and the product of marginal, which is superior to the standard KL-divergence-based definition of mutual information since the standard mutual information estimator is unstable for training [19].

3.4 Similarity-based Ranking Method

Similarity-based ranking method aims to calculate the similarity score of each courseware pair, which can be used to rank candidate courseware to find similar ones. Specifically, we first represent each courseware as a multimodal global feature by employing the *Deep Visual-text Embedding* module and the vector-based pooling network. Next, the similarity score of the input pair is computed by the cosine similarity of their global features. Finally, we can rank the candidate courseware and forms a ranking list according to the similarity scores.



(a) Courseware distribution w.r.t. the number of images. (b) Courseware distribution w.r.t. the number of pages. (c) Page distribution w.r.t. the number of words.

Figure 3: Distributions of the dataset set.

In summary, this method precisely measures the similarity scores by jointly considering the heterogeneous information in the courseware. And those candidates with the largest similarity score will be returned as the most similar one of the given query. We believe that if the top-k ranking scores are high, the courseware is lack of originality (e.g. Many of them are equal to 1, indicating that there are many plagiarized courseware among them). In other words, a courseware is evaluated with high originality if its similarity scores with other courseware are low, and vice versa. Formally, the originality score is defined as follows:

$$O_{pred} = 1 - \frac{\sum_i^k (k - i + 1) \times L_i}{\sum_i^k i}, \tag{9}$$

where L_i is the i^{th} similarity scores in a given ranking list, and k is the length of the top-k ranking list.

4 EXPERIMENTS

In this section, we conducted extensive experiments on a self-collected dataset to answer the following four research questions:

- **RQ1** How does our proposed MOC framework perform as compared to other state-of-the art competitors?
- **RQ2** Are different modalities equally important? How does MOC perform by employing the visual and textual content respectively?
- **RQ3** How do the local and global features contribute to the performance of MOC? How do different pooling strategies perform in aggregating local features?
- **RQ4** How consistent between the algorithm results and human subjective assessment results?

4.1 Experimental Settings

4.1.1 Dataset. We experimented on a real-world dataset released by an education-oriented platform SeeWo[1]. SeeWo is a specialized community for educators, students, and parents to participate in the progress of K-12 education where numerous exercises and courseware are accumulated. We collected the courseware which contain visual and textual content simultaneously. Based on these criteria, we finally obtained 64, 209 courseware. And on average, each courseware contains 21.1 images and 15.1 pages, and each page contains 80.3 words. The detailed data distributions are revealed in Figure 3, which obey decay distributions.

For testing, we subtly construct a test set to evaluate the originality of courseware by treating sequential versions of each courseware as ranking lists. The motivation and operation are shown as follows: (1) The courseware uploaded by an educator will be tagged with a courseware ID. Every time the educator edits the courseware and update it, a new version will be uploaded to the server. So each courseware ID will correspond to multiple different versions. (2) We extract all uploaded version based on the courseware ID, and sort them in descending order by their uploaded history. So that for a particular courseware ID, we have a ranking list of courseware with multiple versions. (3) We consider the similarity between the first courseware and other courseware in the ranking list should be decreased in turns of version. Therefore, we treat the ranking of versions as the ideal ranking list, which can be used to evaluate our model. Ultimately, we evaluated our model on a test set contains 9, 626 courseware, which forms 471 ranking lists. With this test set, measuring the originality of courseware is linked to the learning-to-rank problem [26].

4.1.2 Automatic Evaluation Metrics. The evaluation of originality is generally a difficult task and there are no established metrics in existing works. So we transform the task into a similarity-based ranking problem. To better measure the ranking quality, we propose to evaluate it in both automatic and manual way. For automatic metrics, we propose to employ Mean Reciprocal Rank (MRR) [12] and Normalized Discounted Cumulative Gain (NDCG) [21], which are inspired by the learning-to-rank problem. Specifically, we use the first version to calculate the MRR, and take the top-10 versions as the correct retrievals to calculate the NDCG. However, these evaluation metrics only consider the information of the top-k ranking results. According to the difference between our task and the ranking task, we further propose an Inversion Number Sorting Accuracy (INSA) metric to measure the difference between the predicted ranking list and the ideal ranking list with the complete sequence, which is defined by the inversion number (IN). Let $\mathbb{I}$ be the indication function, then the IN is denoted as:

$$IN = \sum_{i>j}^{p} \mathbb{I}[\pi(i) < \pi(j)], \tag{10}$$

where π is the permutation which also refers to the version ranking list in our setting, and the i, j are the indexes of the list position.

[1]https://www.seewo.com

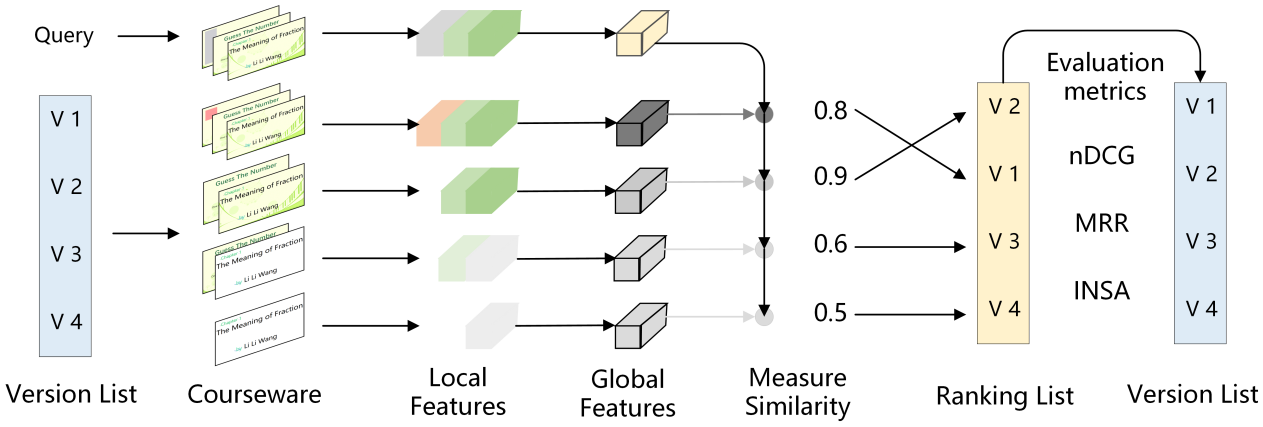


Figure 4: The illustration of how to evaluate the framework.

Table 1: Experimental results and comparison using global features on NDCG, MRR and INSA (Section 4.2).

	NDCG		MRR		INSA	
	score	p-value	score	p-value	score	p-value
Mean	0.644	3.4e-15	0.350	1.9e-16	0.397	2.4e-17
Max	0.806	1.9e-11	0.481	3.6e-14	0.575	3.1e-14
Bi-GRU	0.853	1.7e-05	0.533	1.3e-08	0.657	6.7e-06
Bi-LSTM	0.829	5.3e-06	0.510	3.2e-08	0.611	2.6e-07
MCNN	0.757	2.8e-10	0.370	4.3e-14	0.492	1.1e-10
MANN	0.846	1.6e-07	0.518	1.9e-10	0.628	2.8e-09
MOC	**0.889**	-	**0.622**	-	**0.698**	-

The IN measures the chaos of a permutation, which should be 0 when the ranking list is in ascending order. If a ranking list with the length of n is in descending order, its IN is equals to $\frac{n\times(n-1)}{2}$, which is the maximum of that length. The bigger the IN, the greater mismatch between the predicted and the labeled ranking lists is. To this end, we propose an INSA metric as follows:

$$INSA = 1 - \frac{2 \times IN}{n \times (n-1)} \tag{11}$$

And then, we average all INSA of ranking lists and obtain an evaluation value to measure the sorting performance. Last but not least, the t-test is conducted on these three metrics based on the 5 times experimental results. In addition, the independent samples t-test is used in Tables 1 and 2, and paired samples t-test is applied to INSA metric in Table 3.

4.1.3 Human Evaluation. We conducted human evaluation using the courseware training set. In particular, two types of tasks are assigned:

Task1: to explore the effectiveness of our similarity-based ranking method using the MOC algorithm, annotators were requested to give a originality score ranging from 0 to 10 and a confidence value ranging from 0 to 3 to measure the originality of a courseware given its top-10 similar ones.

Task2: this task aims to compare the predictive originality by different methods for one courseware. To compare the performance of our MOC algorithm with the non-parametric algorithm, we further exploit a non-parametric algorithm using max pooling features to apply similarity-based ranking method and perform human evaluation as the same as Task1. More details will be discussed later.

For human evaluation experiments, we first randomly pick 102 query courseware in training set, and then perform similarity-based ranking method to acquire top-10 similar courseware for each courseware. Note the above two task will result in different top-10 ranking results since different algorithms are applied. And then, due to the subjectiveness of the originality measurement, each task is assigned to five annotators for reducing the annotation bias.

4.1.4 Training and Evaluation Details. In the *Deep Visual-text Embedding* module, the image features of a courseware are extracted from ResNet18 pretrained on ImageNet [33], which are denoted as $\mathbf{f}_I^{ij} \in \mathbb{R}^{512}$; and the text features are extracted from BERT [13] resulting in text feature $\mathbf{f}_T^{ij} \in \mathbb{R}^{768}$. Then the features are transformed to a unified multimodal embedding space with the dimension of 128. Note here we apply a position encoding technique [36] to the embedding space of image and text for retaining their position information in the courseware. In order to acquire a global representation of each courseware, we perform three stacked vector pooling networks above the input entity vectors, all vector pooling networks are with the same setting, in which the number and dimension of high-level entity are 128 and 16, respectively. Then we take the last output entity vectors as the global feature $\mathbf{z} \in \mathbb{R}^{2048}$. Note here we only use shared weights in the first vector pooling network. To maximize the mutual information between the local feature $\mathbf{x}$ and global feature $\mathbf{z}$, we use a discriminator to estimate the mutual information with two fully connected layers. We train the model with a batch size of 64, in which the half of samples are positive ones and the other half are negative ones. The negative samples are acquired by shuffling $\mathbf{z}$ in positive samples. The training loss is the average over all samples. We use Adam [22] with the learning rate of 0.001 and train the model with 300 epochs.

For evaluation, we first compute the global features of a courseware, and take the outputs of *Deep Visual-text Embedding* module as local features. For each courseware ID in the test set, we treat the feature of the latest uploaded courseware as the query, and the others as the keys. Next, we obtain a ranking list sorted by the cosine similarity scores between the query and keys, which is detailed in Figure 4.

Table 2: The impact of image and text features. Experimental results are revealed w.r.t. NDCG, MRR and INSA (Section 4.3).

	Image						Text					
	NDCG		MRR		INSA		NDCG		MRR		INSA	
	score	p-value	score	p-value	score	p-value	score	p-value	score	p-value	score	p-value
Mean	0.544	3.7e-14	0.247	2.7e-14	0.194	1.6e-15	0.673	2.2e-13	0.344	1.2e-14	0.400	9.8e-13
Max	0.710	7.1e-09	0.341	1.0e-10	0.383	5.4e-11	0.772	2.9e-10	0.410	3.2e-13	0.499	5.1e-10
Bi-GRU	0.750	3.1e-02	0.374	2.6e-05	0.446	3.1e-03	0.798	6.6e-07	0.448	2.4e-10	0.546	4.1e-06
Bi-LSTM	0.732	3.4e-06	0.357	2.1e-07	0.419	2.3e-07	0.784	3.5e-09	0.425	5.5e-12	0.519	1.7e-08
MOC	**0.756**	-	**0.393**	-	**0.453**	-	**0.840**	-	**0.540**	-	**0.582**	-

Table 3: The impact of local and global features. Experimental results are revealed w.r.t. NDCG, MRR and INSA (Section 4.4).

	Image				Text				Fused			
	NDCG	MRR	INSA	p-value	NDCG	MRR	INSA	p-value	NDCG	MRR	INSA	p-value
Local (mean)	0.561	0.254	0.206	8.9e-09	0.716	0.350	0.452	5.3e-06	0.694	0.363	0.462	2.2e-07
Local (max)	0.722	0.341	0.396	4.2e-06	0.789	0.444	0.529	1.8e-04	0.836	0.520	0.622	2.2e-05
Global	**0.756**	**0.393**	**0.453**	-	**0.840**	**0.540**	**0.582**	-	**0.889**	**0.622**	**0.698**	-
Local (mean) & Global	0.739	0.376	0.418	7.8e-05	0.804	0.443	0.550	1.3e-03	0.853	0.543	0.651	1.1e-04
Local (max) & Global	0.723	0.340	0.398	9.7e-06	0.798	0.457	0.539	4.9e-04	0.840	0.525	0.629	4.0e-05

4.2 Overall Performance Comparison (RQ1)

We first compare several competitive baseline methods using the global feature considering different pooling techniques, which are detailed as follows: (1) The method of Mean obtains the global features by applying the mean pooling operation to all image and text local features, respectively and then concatenating the pooled features of image and text to form a global features. (2) The method of Max obtains the global feature in the same way as the Mean method, except the use of max pooling operation. (3) We use a bidirectional GRU (Bi-GRU) [1] and a bidirectional LSTM (Bi-LSTM) [20] to gather all of the local features of images and texts, and take the last hidden states as the global features. (4) MCNN [28] is a multimodal CNN for integrating texts and images into a fixed-length representation. (5) MANN [27] is an attention-based LSTM for learning a unified semantic representation of images and texts in a multimodal way.

Table 1 shows the comparison results among different methods, we have the following observations: (1) Our MOC method achieves the best performance as compared to other competitors. All the p-values between our model and each baseline are much smaller than 0.05, indicating the improvements are statistically significant. The major difference between our model and other baselines is the vector-based pooling network, which shows that it is more effective for similarity-based ranking task by integrating the image and text features in a multimodal way. (2) By comparing the parametric models with the non-parametric models (i.e., Mean, Max), we can see that the former almost outperforms the latter on three metrics, which shows that our *Mutual Information Maximization* component can help to learn a semantic global representation. (3) The previous state-of-the-art method MANN does not perform well as compared to our proposed model. The reason behind is that the visual and textual content in the courseware are weakly related, using the image-guided attention features is unsuitable for the courseware data.

4.3 Impact of Different Modalities (RQ2)

The motivation of this work is to learn a unified semantic representations from the heterogenous data. But how do different modalities affect the ranking performance? To validate the effectiveness of different modalities, we train our model by employing the features of image and text, respectively.

Experimental results are revealed in Table 2. We can see that (1) By comparing the results of our solution with other methods, our proposed solution outperforms all of the other methods on three evaluation metrics, verified by the small p-values. It clearly demonstrates that our vector-based pooling network works better in handling variable-length inputs of images and texts. (2) The method taking text features into account consistently outperform the one of image features, showing the fact that text features are more discriminated than image features. It makes sense because the text is more able to indicate the theme of a courseware than the image does. (3) We further compare the performance of different modalities between Table 1 and Table 2, which reveals that fusing image and text information can greatly improve the performance of all methods.

4.4 Impact of Local and Global Features (RQ3)

To validate the effectiveness of local and global features, we perform some micro-level analyses. Specifically, we employ *Global* to represent the global features, *Local(mean)* to represent the local features which applies the mean pooling operation, and *Local(max)* to represent the local features which applies the max pooling operation. Furthermore, to combine the representation ability of local features and global features, we concatenate the pooled local features with the global features to form a new courseware representation, denoted by *Local(mean) & Global* and *Local(max) & Global*.

Table 3 shows the experimental results. As can be seen, by comparing the experimental results of *Local(mean)* and *Local(max)*, one interesting observation is that the mean pooling operation greatly

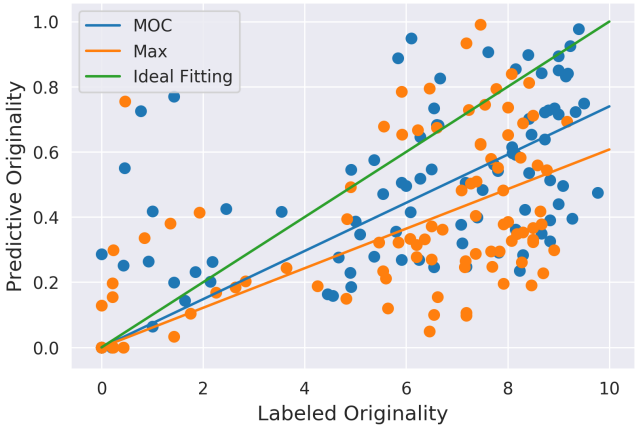


Figure 5: The diagram unfolds a clear comparison between parametric and non-parametric algorithms using prediction-annotation scatter plot. The lines of different colors are the fitting curve of the corresponding data points.

harms the representation ability of the local features. A probable reason is that the average pooling operation may obscure the representation boundary of the courseware vector, making it difficult to measure the similarity between courseware. On the contrary, the *Local(max)* is a strong baseline in the ranking task, which demonstrates that it can preserve most of the information of image and text in a courseware. The *Global* method achieves the best performance as compared to other methods, verified by the small p-values, indicating the global feature is a good representation of the courseware. Furthermore, the performance of concatenated features are better than of *Local(mean)* and *Local(max)*, but slightly worse than the performance of *Global* method, which shows the fact that the global features have extracted the most of discriminative information in the courseware.

4.5 Originality Evaluation (RQ4)

We aim to validate the consistency between algorithm results and human subjective assessment results, which considers the predictive originality and human-label originality with confidence. First, given the experimental results of 5 annotators, we calculate the weighted originality to reduce bias of the annotations as follows:

$$O_{label} = \frac{\sum_i^5 C_i \times O_i}{\sum_i^5 C_i}, \tag{12}$$

where O_i is the originality and C_i is the corresponding confidence assessed by the i^{th} annotator. And then, to measure the consistency between predictive originality and labeled originality, we apply the Pearson correlation coefficient and linear regression coefficient to measure the linear correlation between the two variables.

Figure 5 shows the scatter plot of the annotations and predictions. We find that our MOC model performs well in predicting originality, verified by the small margin between MOC fitting curve and ideal fitting curve. And it is less accurate for the Max algorithm since its top-10 query results lead to less consistent annotations across annotators.

Table 4: The comparison of consistency between parametric and non-parametric algorithms (Section 4.5).

	Pearson Correlation	Fitting Slope
Max (non-parametric)	0.5956	0.0607
MOC (parametric)	**0.7036**	**0.0740**
Ideal	1.0000	0.1000

Table 4 shows qualitative results of the assessment consistency. The Pearson correlation measures the linear correlation between the prediction and annotation. And the fitting slope measures the margin between the prediction fitting curve and the ideal fitting curve. By comparing the experimental results of MOC and Max, we can observe that the predictive originality of parametric model is far more consistent with human evaluation results. We believe the reason behind is that MOC algorithm can extract the most discriminative global features from the courseware by using vector pooling network and mutual information maximization model, while the non-parametric algorithm will lose information due to the naive pooling technique.

In conclusion, although the measurement of originality is a very subjective problem, as shown in the figure and table above, our MOC algorithm is able to achieve consistent results with human evaluation. Moreover, our algorithm can also perform better in predicting originality than the non-parametric algorithm using naive pooling technique, which clearly demonstrates the effectiveness.

5 CONCLUSIONS AND FUTURE WORK

In this work, we proposed a similarity ranking-based unsupervised approach to measure the originality of courseware in online education systems. For modeling the heterogeneous data of courseware into global representations, a *Deep Visual-text Embedding* module and a *Variable-length Vectors Integration* method were proposed. Specifically, given the images and texts of courseware, we first exploited a pre-trained ResNet18 and a language model BERT with embedding layers to generate the image and page text representations in a local manner. Then, we designed a vector-based pooling network to learn global semantic representations of courseware in a multimodal way, in which a capsule neural network with shared weights is used to fused image and text representations. Finally, a Discriminator was proposed to optimize the model by maximizing the mutual information between the local features and the global features. By experimenting on a self-collected dataset, we have demonstrated the effectiveness and rationality of our proposed solution on both overall performance comparison and micro-scope analyses.

In the future, we plan to extend our work in the following two directions. First, besides the semantic information like image and text, we would like to consider other structured information of courseware, such as page number, font size, and so on. Second, we will consider to design a co-training scheme to jointly optimize the *Deep Visual-text Embedding* module and the *Variable-length Vectors Integration* module in a unified framework. It will further improve the similarity-based ranking performance, which helps better measure the originality of courseware.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*. ACM.

[2] Piotr Bojanowski and Armand Joulin. 2017. Unsupervised learning by predicting noise. In *Proceedings of the International Conference on Machine Learning*, Vol. 70. ACM, 517–526.

[3] Da Cao, Xiangnan He, Lianhai Miao, Yahui An, Chao Yang, and Richang Hong. 2018. Attentive group recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 645–654.

[4] Da Cao, Xiangnan He, Liqiang Nie, Xiaochi Wei, Xia Hu, Shunxiang Wu, and Tat-Seng Chua. 2017. Cross-platform app recommendation by jointly modeling ratings and texts. *ACM Transactions on Information Systems* 35, 4 (2017), 37.

[5] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. 2017. Deep adaptive image clustering. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 5879–5887.

[6] Jingjing Chen, Chong-Wah Ngo, Fuli Feng, and Tat-Seng Chua. 2018. Deep Understanding of Cooking Procedure for Cross-modal Recipe Retrieval. In *Proceedings of the ACM International Conference on Multimedia*. ACM, 1020–1028.

[7] Zhiyong Cheng, Xiaojun Chang, Lei Zhu, Rose C. Kanjirathinkal, and Moha Kankanhalli. 2018. MMALFM: Explainable recommendation by leveraging reviews and images. *ACM Transactions on Information Systems* 37 (2018), 2.

[8] Zhiyong Cheng, Jialie Shen, Lei Zhu, Mohan S Kankanhalli, and Liqiang Nie. 2017. Exploiting Music Play Sequence for Music Recommendation. In *International Joint Conference on Artificial Intelligence*, Vol. 17. Morgan Kaufmann, 3654–3660.

[9] Wei Chu and S Sathiya Keerthi. 2005. New approaches to support vector ordinal regression. In *Proceedings of the International Conference on Machine Learning*. ACM, 145–152.

[10] Corinna Cortes, Mehryar Mohri, and Ashish Rastogi. 2007. Magnitude-preserving ranking algorithms. In *Proceedings of the International Conference on Machine Learning*. ACM, 169–176.

[11] David Cossock and Tong Zhang. 2006. Subset ranking using regression. In *International Conference on Computational Learning Theory*. Springer, 605–619.

[12] Nick Craswell. 2009. Mean reciprocal rank. *Encyclopedia of Database Systems* (2009), 1703–1703.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[14] Laurent Dinh, David Krueger, and Yoshua Bengio. 2014. NICE: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516* (2014).

[15] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2016. Density estimation using Real NVP. *CoRR* abs/1605.08803 (2016).

[16] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to Forget: Continual Prediction with LSTM. *Neural Computation* 12, 10 (2000), 2451–2471.

[17] Hicham Hage and Esma Aïmeur. 2005. Exam Question Recommender System. In *Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*. IOS Press, 249–257.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 770–778.

[19] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. In *Proceedings of the International Conference on Learning Representations*. ACM.

[20] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[21] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* 20 (2002), 422–446.

[22] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[23] Durk P Kingma and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*. MIT Press, 10236–10245.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. MIT Press, 1097–1105.

[25] Chen-Yu Lee, Patrick W Gallagher, and Zhuowen Tu. 2016. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In *Artificial Intelligence and Statistics*. JMLR, 464–472.

[26] Hang Li. 2011. A short introduction to learning to rank. *IEICE Transactions on Information and Systems* 94, 10 (2011), 1854–1862.

[27] Qi Liu, Zai Huang, Zhenya Huang, Chuanren Liu, Enhong Chen, Yu Su, and Guoping Hu. 2018. Finding similar exercises in online education systems. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1821–1830.

[28] Lin Ma, Zhengdong Lu, and Hang Li. 2016. Learning to answer questions from image using convolutional neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI.

[29] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[30] Ramesh Nallapati. 2004. Discriminative models for information retrieval. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 64–71.

[31] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. ACL, 1532–1543.

[32] Jirí Rihák and Radek Pelánek. 2017. Measuring Similarity of Educational Items Using Data on Learners' Performance.. In *Proceedings of the International Conference on Educational Data Mining*. 16–23.

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. 2014. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 3 (2014), 211–252.

[34] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*. MIT Press, 3856–3866.

[35] Mohammad E Shiri, A Esma Aïmeur, and Claude Frasson. 1998. Student modelling by case based Reasoning. In *International Conference on Intelligent Tutoring Systems*. Springer, 394–403.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. MIT Press, 5998–6008.

[37] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research* 11 (2010), 3371–3408.

[38] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 515–524.

[39] Jiayu Wang, Wengang Zhou, Jinhui Tang, Zhongqian Fu, Qi Tian, and Houqiang Li. 2018. Unregularized Auto-Encoder with Generative Adversarial Networks for Image Generation. In *Proceedings of the ACM International Conference on Multimedia*. ACM, 709–717.

[40] Shuhui Wang, Yangyu Chen, Junbao Zhuo, Qingming Huang, and Qi Tian. 2018. Joint Global and Co-Attentive Representation Learning for Image-Sentence Retrieval. In *Proceedings of the ACM International Conference on Multimedia*. ACM, 1398–1406.

[41] Suyun Wei, Ye Ning, Shuo Zhang, Huang Xia, and Zhu Jian. 2012. Item-Based Collaborative Filtering Recommendation Algorithm Combining Item Category with Interestingness Measure. In *International Conference on Computer Science & Service System*. IEEE, 2038–2041.

[42] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the International Conference on Machine Learning*. ACM, 1192–1199.

[43] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *Proceedings of the International Conference on Machine Learning*. ACM, 478–487.

[44] Jun Xu and Hang Li. 2007. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 391–398.

[45] Jen-Yuan Yeh, Jung-Yi Lin, Hao-Ren Ke, and Wei-Pang Yang. 2007. Learning to rank for information retrieval using genetic programming. In *Proceedings of SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*.

[46] Matthew Zeiler and Robert Fergus. 2013. Stochastic pooling for regularization of deep convolutional neural networks. In *Proceedings of the International Conference on Learning Representation*. ACM.

[47] Liang Zhang, Bingpeng Ma, Guorong Li, Qingming Huang, and Qi Tian. 2017. Multi-networks joint learning for large-scale cross-modal retrieval. In *Proceedings of the ACM International Conference on Multimedia*. ACM, 907–915.

[48] Zhaohui Zheng, Keke Chen, Gordon Sun, and Hongyuan Zha. 2007. A regression framework for learning ranking functions using relative relevance judgments. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 287–294.

[49] Tianyu Zhu, Qi Liu, Zhenya Huang, Enhong Chen, Defu Lian, Yu Su, and Guoping Hu. 2018. MT-MCD: A Multi-task Cognitive Diagnosis Framework for Student Assessment. In *International Conference on Database Systems for Advanced Applications*. Springer, 318–335.