LECTOR: An attention-based model to quantify e-book lecture slides and topics relationships

Erwin D. Lopez Z.
Kyushu University
lopez.zapata.erwin.242@s.kyushu-u.ac.jp

Tsubasa Minematsu
Kyushu University
minematsu@limu.ait.kyushu-u.ac.jp

Yuta Taniguchi
Kyushu University
tyuta.taniguchi.y.t@gmail.com

Fumiya Okubo
Kyushu University
fokubo@ait.kyushu-u.ac.jp

Atsushi Shimada
Kyushu University
atsushi@limu.ait.kyushu-u.ac.jp

ABSTRACT
The use of digital lecture slides in e-book platforms allows the analysis of students' reading behavior. Previous works have made important contributions to this task, but they have focused on students' interactions without considering the content they read. The present work complements these works by designing a model able to quantify the e-book LECTure slides and TOpic Relationships (LECTOR). Our results show that LECTOR performs better in extracting important information from lecture slides and suggest that readers' topic preferences extracted by our model are important factors that can explain students' academic performance.

Keywords
e-book, reading behavior, keyphrase extraction, multimodal learning analytics

1. INTRODUCTION
The adoption of e-learning technologies in blended courses can help instructors better understand students' learning behaviors and make more informed revisions of lessons and materials [9]. Examples of these technologies include the e-book reading systems used in university classrooms to distribute lecture materials. By modeling students' interactions on these systems, instructors can analyze their reading behavior and support their learning process [14, 22, 15].

Several works have investigated how to model e-book reading users based on their set of reading characteristics [1, 34, 24, 8, 2]. Nevertheless, their models did not consider the content that students read [31], information that may be important for improving the course content's structures [16], or providing process-oriented feedback to students [27].

Since lecture slide data consists of text and images, their integration into current models poses several challenges to be addressed [7, 31]. Both text and image processing are difficult tasks that recent advances in computer science are attempting to address in different domains. Furthermore, considering multimodal data would require formulating a model able of integrating the different data sources.

In this context, the present work takes the first step by focusing on the text-processing task. We propose the model LECTOR, which uses Natural Language Processing (NLP) techniques to estimate a quantitative relationship between a lecture slide and a topic. By performing this estimation, we can convert a slide-wise set of reading characteristics into a topic-wise set of reading characteristics (Figure 1). Accordingly, we validate LECTOR's performance on this task against previous models.

2. RELATED WORK
2.1 Text processing in e-book lecture slides
Previous studies describe the use of e-book lecture slide text to address various problems, such as slide summarization [28], personalized recommendation [21, 23], and learning footprint transfer [33]. Almost all of these works used the TF-IDF method [26] to process their slides [28, 33, 21]. Other works use hierarchical models to perform this process [32, 5], but they require human labeling of all the text in the slides [3], a task that can be burdensome for teachers.

In addition, a previous study estimated topic reading time from e-book user data by considering only the slides where
the topic was written [31]. We can reformulate this method as a matrix product (Figure 1), where they assigned a relationship of 1 when the topic appears in a given slide, and 0 in other cases (referred to as “Binary score” in this paper).

2.2 Keyphrase extraction from documents
Our problem is reduced to an unsupervised keyphrase extraction task if we consider lecture slides as documents and topics as key phrases. The state-of-the-art studies on this task use pre-trained models (e.g., Doc2Vec [18], ELMo [25], BERT [12]) to represent words as embedding vectors [6, 29, 13]. Then, their methods estimate the similarity between key phrases and documents from the cosine similarity of their corresponding embedding representations [6, 29, 13].

3. PROPOSED MODEL
LECTOR extracts a set of topic candidates from all the slides of a given course and assigns a single score to each slide-topic pair (Figure 2). This score is defined as a linear combination of two different scores, one based on the words’ importance and the other on the similarity between the topic and the slide embeddings.

3.1 Topics extraction
We consider a topic to be an observable entity (keyphrase). Models such as EmbedRank [6] and AttentionRank [13] use the Part-Of-Speech to generate noun phrases that become their possible key phrases. In our case, we work with slides written in Japanese and use the Bi-LSTM-based NLP library Nagisa to identify the nouns. Then, we define single nouns and n-gram sequences (n=2) of nouns as our topics.

3.2 Word embeddings and attention matrix
We use a BERT model (fine-tuned on all the course slides’ text in the MLM task [12]) to estimate a self-attention matrix $A^t$ and a set of word embeddings $E^t$ for each slide. We then correct these token-wise values to word-wise values [10].

3.3 LECTOR’s importance score
For a given slide $s_i$, we quantify the attention $a_{ij}$ that words $w$ belonging to a given topic $t_j$ receive from all the other words $w$ within the slide $s_i$ by summing the different weights of the matrix $A^t$ as shown in Equation 1.

$$a_{ij} = \sum_{w \in t_j} \sum_{w' \in s_i \setminus \{w\}} A^t_{w'w}$$

Since this score is strongly influenced by the frequency of the topic’s words $f_j$, the importance score ($ss_{ij}$) is calculated by considering the Smooth Inverse Frequency [4] (Equation 2).

$$ss_{ij} = a_{ij} \left( \frac{k}{k + f_j} \right)$$

3.4 LECTOR’s similarity score
For a given slide $s_i$, we estimate its embedding representation $P^i_s$ as a weighted average of its corresponding word embeddings $E^i$ (Equation 3).

$$P^i_s = \sum_{w \in s_i} Weight(w) E^i_w$$

We define the word weight as the probability of belonging to the discourse of the given slide. We consider that this discourse is given by a general discourse introduced in the first slide of the lecture material and a specific discourse introduced by the title of the respective slide (Figure 3).

Accordingly, given the set of title and body embeddings $E^t_{st}$ and $E^b_{sb}$, the Weights are calculated as shown in Equation 4. In Appendix A, we detail the formulation and estimation of these Weights from the set of word embeddings.

$$Weight = Pr(w_t \in s_t|s_1) \cdot Pr(w_b \in s_b|s_1)$$

Finally, the similarity score is given by the cosine similarity between the topic $t_j$ and slide $s_i$ embeddings [6, 29, 13].

$$b_{ij} = \frac{P^i_s \cdot E^j_t}{||P^i_s|| \cdot ||E^j_t||}$$

$$cs_{ij} = \left( \frac{1}{f_j} \sum_{topicj} b_{ij} \right) f_j^\alpha, \alpha \in [0, 0.25]$$

3.5 LECTOR’s final score
The final score for a given topic $t_j$ and slide $s_i$ is a linear combination of the previously normalized importance and similarity scores (Equation 7). The parameter $d$ defines the importance of each score value.

$$score_{ij} = d \cdot ss_{ij} + (1 - d) \cdot cs_{ij}$$

LECTOR’s final output is the matrix $M$, whose elements $M_{ij}$ are the final scores between slides $s_i$ and topics $t_j$.

4. RESULTS AND DISCUSSION
4.1 Dataset
Our dataset consists of the textual content of 620 slides from 22 e-book materials delivered in the course “Programming Theory” in the year 2019 (before the pandemic restrictions). This course was offered by the School of Engineering at Kyushu University for 7 weeks.
4.2 First Experiment formulation

The ground-truth values to evaluate LECTOR’s estimates are given by the relationships between different topics and slides. However, to find them empirically, we would need a large number of samples because these relationships are perceived differently by different people. Furthermore, given the large number of topics and slides in a course, we would need millions of ground truth labels for each sample.

For this reason, our experiment is designed to indirectly evaluate the estimates of the models. Similar to works on keyphrase extraction, we assume that the most important topics should have the highest relationships with the course content (the different slides). For a given topic $t_j$, we define its keyphrase candidate score $mt_j$ as the sum of the scores obtained across all slides (Equation 8).

$$mt_j = \sum_{i=1}^{\#slides} M_{ij} \tag{8}$$

We use the $mt_j$ values to extract the most important topics of the course. Our ground-truth labels are given by the course keywords extracted from the course syllabus (“Scheme”, “Data Structure”, “List Processing”, “Recursion”, “Expression”, “Condition”, “Design Recipe”, “Function”, “High-level function”). We define $\emptyset n$ as the set that contains the top $n$ topics according to the scores $mt_j$. By comparing this set to the ground truth, we can measure the model performance.

We considered three baselines. The first is given by the TF-IDF model [26], which is predominant in the slide text processing literature. The second is given by the AttentionRank model [13], which represents the state-of-the-art in unsupervised keyphrase extraction. The third model is given by the previously described Binary score model proposed by [31].

4.3 First Experiment results

Our results are summarized in Table 1. We can see that AttentionRank outperforms all the other models with an F-score of 28.68% when considering the 5 most important topics. This result shows the high performance of this state-of-the-art model even in a different domain (slides unstructured text). This F-score was achieved by identifying 2 keyphrases in its five most important topics. As we can see in Table 2, we can also note that despite all the other models achieving the same F-score, the TF-IDF and Binary models are more influenced by the frequency of the topics, estimating topics such as “i” and “define” as one of their most important ones.

At $n = 10$, we can see that the attention-based models outperform the TF-IDF and Binary models. Specifically, AttentionRank, LECTOR Similarity score, and LECTOR achieve an F-score of 31.68%. At $n = 15$, LECTOR outperforms all the other models with an F-score of 33.44%. We can see the same result when comparing the best F-score obtained by each model and the mean of the results obtained in the first $n@100$ sets. These results show that AttentionRank has difficulty finding new keyphrases, whereas LECTOR does not.

In Table 2, we can see that AttentionRank tends to give high scores also to minor topics such as “define”, “else”, or “empty” which may explain its lower performance.

The mentioned problem of AttentionRank has two reasons. The first is that its “Accumulated Self-Attention” is influenced by the word frequencies. In their paper, the authors pointed out that this characteristic can be beneficial in large documents. However, in the context of lecture slides, several words from the domain knowledge of the course can appear repeatedly. For example, the mentioned “define” and “else” are well used in the program examples of the course “Programming Theory”. On the other hand, the design we considered in the LECTOR’s importance score limits the influence of the frequency of the words.

However, in the AttentionRank model, topics must also achieve a high “Cross-Attention” value in order to get a high final score. The reason that words like “define” and “else” are important topics of the model is due to the two discourse hypotheses of AttentionRank. For a given slide, the first assumes that the topic candidate defines the slide discourse, and the second assumes that the slide defines the topic discourse. In the context of noisy and unstructured slide text, this consideration can lead to some problems.

For example, given the topic “define” and a slide that contains a programming code example about list processing,
Table 2: Most important topics of each model. ENG: a word originally written in English.

<table>
<thead>
<tr>
<th>n</th>
<th>TF-IDF</th>
<th>AttentionRank</th>
<th>Binary score</th>
<th>LECTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>function</td>
<td>function</td>
<td>function</td>
<td>function</td>
</tr>
<tr>
<td>2</td>
<td>list</td>
<td>example problem</td>
<td>list</td>
<td>data</td>
</tr>
<tr>
<td>3</td>
<td>list (ENG)</td>
<td>definition</td>
<td>define (ENG)</td>
<td>list</td>
</tr>
<tr>
<td>4</td>
<td>i (ENG)</td>
<td>recursion</td>
<td>definition</td>
<td>definition</td>
</tr>
<tr>
<td>5</td>
<td>define (ENG)</td>
<td>example</td>
<td>cond (ENG)</td>
<td>program</td>
</tr>
<tr>
<td>6</td>
<td>definition</td>
<td>value</td>
<td>data</td>
<td>computation</td>
</tr>
<tr>
<td>7</td>
<td>page</td>
<td>define (ENG)</td>
<td>list (ENG)</td>
<td>function definition</td>
</tr>
<tr>
<td>8</td>
<td>data</td>
<td>expression</td>
<td>empty</td>
<td>expression</td>
</tr>
<tr>
<td>9</td>
<td>count</td>
<td>argument</td>
<td>count</td>
<td>example problem</td>
</tr>
<tr>
<td>10</td>
<td>program</td>
<td>computation</td>
<td>i (ENG)</td>
<td>recursion</td>
</tr>
<tr>
<td>11</td>
<td>value</td>
<td>list</td>
<td>value</td>
<td>data definition</td>
</tr>
<tr>
<td>12</td>
<td>expression</td>
<td>else (ENG)</td>
<td>expression</td>
<td>list processing</td>
</tr>
<tr>
<td>13</td>
<td>cond (ENG)</td>
<td>empty (ENG)</td>
<td>recursion</td>
<td>program design</td>
</tr>
<tr>
<td>14</td>
<td>example</td>
<td>element</td>
<td>else (ENG)</td>
<td>recursion function</td>
</tr>
<tr>
<td>15</td>
<td>recursion</td>
<td>count</td>
<td>element</td>
<td>exercises</td>
</tr>
</tbody>
</table>

the mentioned model will focus on the context words of “define” in the code (including the “define” itself) resulting in a high Cross-attention score in this case. Then, when we consider the topics “list processing” or “example code”, even if the model manages to estimate high scores for these topics, they will be relatively as important as “define”.

Similarly, the presence of noise in the slides can highly influence the relative scores, sometimes estimating low scores for a closely related topic and slide pair. In contrast, LECTOR’s similarity score considers a singular discourse defined by the main title and slide title that give relatively high scores to topics highly related to this discourse. In the previous example, LECTOR would give higher scores to “list processing” and “example code” rather than “define”, and also would give a higher score to “define” rather than a random noise word.

4.4 Second Experiment formulation

Previous studies of students’ eye-tracking data have concluded that each student has a different preference for learning content [20]. Accordingly, this experiment aims to compare the topic preferences of students with different grades.

We extract their reading time on the different slides (inside and outside of class) and obtain their slide preferences by normalizing the reading time values across the week. Then, we use LECTOR to quantify their Relative Reading Times for the different topics (Topic RRT), as shown in Figure 1. Finally, we group the students according to their grades (A=24, B=6, C=4, D=6, F=10) and compare both their reading time and RRT distributions. We measure the separability of the distributions by using the Fisher Discriminant Ratio (FDR) and statistically validated them with a T-test.

4.5 Second Experiment results

We can see an example of our results in Figure 4. Figure 4a shows the distribution of the reading time of the students with final grades A and B in the second week after the lecture (out-class). Both distributions overlap, so the FDR is 0.0502 and the significance level (p) of the T-test is 0.3302. In Figure 4b we see the same distributions when we consider the relative time spent reading about “Design method”. Here, students with a final grade of A tend to read more on this topic, resulting in a higher FDR of 5.5802 and a lower p of 0.037 in the T-test.

Our different results are summarized in Table 3. We considered the first 3 weeks of the course because of insufficient data in later weeks due to dropouts. As shown in this table, we have included 5 cases, comparing students with consecutive grades (A-B, B-C, C-D, D-F) and at-risk students (students who failed the course) with non-risk students. The result shown in Figure 4 can be found in the first column and fourth row of the table.

In the results of Reading Time, we can see that students from different groups tend to read the same amount of time. In the case of at-risk and non-risk student groups, we find...
Table 3: Fisher Discriminant Ratio between different groups of students in the first 3 weeks of the course.

<table>
<thead>
<tr>
<th>Week 1 (IN-CLASS)</th>
<th>Topic</th>
<th>(Expressions)</th>
<th>Data (Exercises)</th>
<th>Design (Execution)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading Time</td>
<td>0.0342</td>
<td>0.2615</td>
<td>2.782</td>
<td>0.0229</td>
</tr>
<tr>
<td>Topic RRT</td>
<td>1.4517</td>
<td><strong>612.44</strong></td>
<td>46.861</td>
<td><strong>3.233</strong></td>
</tr>
<tr>
<td><em>p&lt;0.05</em>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 1 (OUT-CLASS)</td>
<td>Topic</td>
<td>(Auxiliary Functions)</td>
<td>Program Design</td>
<td>Problems</td>
</tr>
<tr>
<td>Reading Time</td>
<td>0.0409</td>
<td>0.0023</td>
<td>0.0436</td>
<td>0.0085</td>
</tr>
<tr>
<td>Topic RRT</td>
<td>3.0031</td>
<td><strong>29.3069</strong></td>
<td><strong>653.11</strong></td>
<td>72.649</td>
</tr>
<tr>
<td><strong>p&lt;0.01</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 2 (IN-CLASS)</td>
<td>Topic</td>
<td>(Problems)</td>
<td>Boolean Value</td>
<td>Program</td>
</tr>
<tr>
<td>Reading Time</td>
<td>1.0128</td>
<td>0.6902</td>
<td>0.1735</td>
<td>0.0021</td>
</tr>
<tr>
<td>Topic RRT</td>
<td>6.3908</td>
<td>8.4876</td>
<td><strong>588.83</strong></td>
<td>1.7794</td>
</tr>
<tr>
<td><strong>p&lt;0.01</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 2 (OUT-CLASS)</td>
<td>Topic</td>
<td>(Design)</td>
<td>(Cond)</td>
<td>(Body)</td>
</tr>
<tr>
<td>Reading Time</td>
<td>0.0502</td>
<td>0.0855</td>
<td>0.2629</td>
<td>0.0913</td>
</tr>
<tr>
<td>Topic RRT</td>
<td><strong>5.5802</strong></td>
<td>29.9718</td>
<td><strong>241.1</strong></td>
<td>2.8445</td>
</tr>
<tr>
<td><strong>p&lt;0.01</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 3 (IN-CLASS)</td>
<td>Topic</td>
<td>(Exercise Problems)</td>
<td>(Synthetic Data)</td>
<td>(Sorting) (Examples)</td>
</tr>
<tr>
<td>Reading Time</td>
<td>0.0503</td>
<td>0.4597</td>
<td>0.1141</td>
<td>0.0142</td>
</tr>
<tr>
<td>Topic RRT</td>
<td>11.8214</td>
<td>8.263</td>
<td>7.998</td>
<td>15.061</td>
</tr>
<tr>
<td><strong>p&lt;0.01</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 3 (OUT-CLASS)</td>
<td>Topic</td>
<td>(Templates)</td>
<td>(Element Count)</td>
<td>(Exercise Problems)</td>
</tr>
<tr>
<td>Reading Time</td>
<td>0.0234</td>
<td>0.0008</td>
<td>0.2131</td>
<td>1.4279</td>
</tr>
<tr>
<td>Topic RRT</td>
<td><strong>15.166</strong></td>
<td><strong>168.33</strong></td>
<td><strong>286.84</strong></td>
<td>42.266</td>
</tr>
<tr>
<td><strong>p&lt;0.01</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*~*p<0.05 **~p<0.01

We can consider student’s reading preferences for further analysis. For example, as mentioned earlier, at-risk students engage less outside of class in the second and third weeks. In Table 3, we also see that they tend to focus more on exercises problems. This is a signal that at-risk students adopt a surface learning approach [17], focusing on the content directly related to the assessments. Thus, previous works [1, 34] that have analyzed the students’ reading behavior can use the topic preferences to make better reports.

5. LIMITATIONS

The first limitation is the indirect evaluation of the models’ estimates. As previously discussed, collecting labels for a direct evaluation is impractical, but if we limit the number of topics to the most important ones we can collect a limited set of labels to conduct a more direct evaluation.

The second limitation is the size of our dataset. To evaluate the generalizability of our model, we need to consider slides from different courses. In a science course, the slides are less structured and include equations or code. In this case, the robustness of LECTOR plays an important role.

In addition, our slides are in Japanese and the generality of our results may be affected by the use of other methods for topic extraction in different languages.

6. CONCLUSIONS

We proposed LECTOR, a new model that adapts state-of-the-art keyphrase extraction models to the domain of lecture slides. From our results, we conclude that LECTOR can quantitatively extract the relationships between topics and e-book lecture slides better than previous models when considering noisy text from scientific lecture slides. LECTOR was able to extract important topics (higher F-score) while avoiding frequent out-of-context topics.

LECTOR’s topic-wise representation of e-book reading characteristics provides new insights into the students reading behavior. Specifically, it allows to access the students’ preferences for some topics and use them to model more detailed behaviors. Our results show that this new model preserves the differences related to reading preferences that exist between students with different final grades.

These responses validate the benefits of integrating attention-based models like LECTOR into reading behavior models. Accordingly, it allows future works to consider students reading preferences in their models. Also, our model can be used for other text processing tasks, such as slide summarization, content recommendation, etc.

7. ACKNOWLEDGMENTS

This work was supported by JST CREST Grant Number JPMJCR22D1 and JSPS KAKENHI Grant Number JP22H00551, Japan.
8. REFERENCES


APPENDIX

A. WORDS’ WEIGHTS ESTIMATION

A.1 Preliminary definition

Given a set of words $A = \{w_1^a, w_2^a, \ldots\}$ and $B = \{w_1^b, w_2^b, \ldots\}$, we will estimate $Pr(w_a \in A|B)$: The probability of each word in A being generated under the discourse (context) of the set of words B.

First, the probability that a given word $w_a$ is generated under a given context word $w_b$ is proportional to the inner product of their word embeddings (Equation 9) [4, 19].

\[
Pr(w_a | w_b) \propto \exp \left( e_a \cdot e_b^T \right)
\]

(9)

With this equation, we can estimate the probability of each word $w_a$ in the set A to be generated under the single context word $w_b$, as shown in Equation 10.

\[
Pr(w_a \in A | w_b) = [k_1 \exp \left( e_a \cdot e_b^T \right), k_2 \exp \left( e_a \cdot e_b^T \right), \ldots]
\]

(10)

We assume a common proportional constant $(k_1 = k_2 = \ldots)$. Then, we can represent Equation 10 as the softmax of the matrix product between the set of embeddings $E_a = [e_a^1, e_a^2, \ldots]$ and the context embedding $e_b$, as shown in Equation 11 (the parameter $\varphi$ preserves the influence of the proportional constant). This equation can also be interpreted as the cross-attention between the Query $e_b$ and the Key $E_a$.

\[
Pr(w_a \in A | w_b) = \text{Softmax} \left( \frac{E_a \cdot E_a^T}{\varphi \sqrt{d_k}} \right)
\]

(11)

Finally, we can generalize this equation to the context $B = \{w_1^b, w_2^b, \ldots\}$ by using the approach “Attention over attention” proposed in the study [11].

\[
S = \frac{E_b \cdot E_a^T}{\varphi \sqrt{d_k}}
\]

(12)

\[
Pr(w_a \in A | B) = A_{row} (SF_{col}(S)) SF_{row}(S)
\]

(13)

where $A_{row}$ means average along the row axis, $SF_{col}$ means softmax along the column axis, and $SF_{row}$ means softmax along the row axis.

A.2 Formulation

Given the set of words embeddings $E_i$ for each slide, we split it into the set of title and body embeddings $E_{ti}^i$ and $E_{bi}^i$. Then, the words’ Weights are estimated using Equations 11 and 13 as follows:

\[
S = \frac{E_{ti} \cdot E_{ti}^T}{\varphi \sqrt{d_k}}
\]

(14)

\[
Pr(w_t \in st_1|st_2) = A_{row} (SF_{col}(S)) SF_{row}(S)
\]

(15)

\[
Pr(w_t \in sb|st) = \text{Softmax} \left( \frac{E_{st}^i \cdot E_{sb}^i}{\varphi \sqrt{d_k}} \right)
\]

(16)

Weight $= Pr(w_t \in st_1|st_2) Pr(w_t \in sb|st_1)$

(17)