Towards Automated Slide Augmentation to Discover Credible and Relevant Links

Dilan Dinushka, Christopher M. Poskitt, Kwan Chin Koh, Heng Ngee Mok, and Hady W. Lauw

Singapore Management University, Singapore
{dinushkasa,cposkitt,kckoh,hnmok,hadywlauw}@smu.edu.sg

Abstract. Learning from concise educational materials, such as lecture notes and presentation slides, often prompts students to seek additional resources. Newcomers to a subject may struggle to find the best keywords or lack confidence in the credibility of the supplementary materials they discover. To address these problems, we introduce SLIDE++, an automated tool that identifies keywords from lecture slides, and uses them to search for relevant links, videos, and Q&As. This interactive website integrates the original slides with recommended resources, and further allows instructors to 'pin' the most important ones. To evaluate the effectiveness of the tool, we trialled the system in four undergraduate computing courses, and invited students to share their experiences via a survey and focus groups at the end of the term. Students shared that they found the generated links to be credible, relevant, and sufficient, and that they became more confident in their understanding of the courses. We reflect on these insights, our experience of using SLIDE++, and explore how Large Language Models might mitigate some augmentation challenges.

1 Introduction

Students go through considerable changes in their learning patterns once they commence their tertiary education. Rather than following a managed learning path, students must transition to an independent learning approach that relies on their own research. Self-learning becomes even more important when the provided course materials are concise or limited, especially when they are the sole reference materials for certain lectures. Thus, it would be beneficial to supplement such materials with credible and relevant links to additional resources.

As an initial attempt to support students in their self-learning journeys, we developed SLIDE++[1], a system for viewing and managing augmented slides (Figure 1). Our platform enables students to access slide sets uploaded by their university instructors, automatically enriched with complementary resources curated from the web. SLIDE++ uses statistical methods to identify the most significant keywords inside slides, then uses them to query the web for relevant articles, videos, and discussions. Instructors can further refine the results by adding/removing resources and 'pinning' the most important recommendations.

[1] https://slideplusplus.preferred.ai/

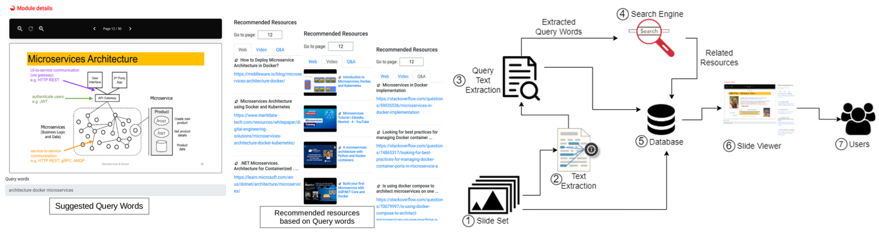

Fig. 1: SLIDE++ website preview (left) and architecture (right): 1. upload slide set; 2. extract & preprocess; 3. query text extraction; 4. search links; 5. store in database; 6. SLIDE++ website; 7. end users (students)

We trialled SLIDE++ in four undergraduate computing courses, then conducted surveys and focus groups at the end of term. Students shared that they found the generated links to be credible, relevant, and sufficient, and increased their confidence in their understanding of the courses. We also explored the use of Large Language Models (LLMs), finding that GPT4 can potentially complement SLIDE++'s statistics-based approach on image-heavy slides.

2 Related Work

Although we are not aware of another tool similar to SLIDE++, there are some works that analyse and augment lecture slides for other purposes. Wolfe [9], for example, creates annotated slides that can be used by 'conversational agents' that students can interact with. Similarly, Shimada et al. [7] analyse lecture slides to generate summarised versions of the materials to read as previews. Like SLIDE++, this tool also automatically analyses a slide set, but instead of providing additional references, it provides a summarised version of its content.

A related approach to recommending links in slides is recommending content personalised to the student [8]. For example, Oliveira et al. [4] generate such recommendations in learning management systems: they take into account the learning profiles of students in order to suggest activities and materials that better suit their learning styles. Barria-Pineda et al. [2] integrate personalised recommendations into a Java programming practice system, with the system able to explain its recommendations based on the student's prior interactions. These approaches are quite different to SLIDE++ in that existing catalogues of content, activities, or exercises are utilised as the recommendations.

Recently, practitioners are beginning to explore the potential pedagogical benefits of AI tools such as ChatGPT [1, 3, 5], e.g. for generating new content, explaining examples, or analysing error messages. Our preliminary tool is effective at generating keywords using simpler statistical methods, but its approach could also be used in conjunction with advanced AI models, e.g. leveraging ChatGPT to generate better query terms to improve the the relevance of curated resources.

3 Slide++ Tool

The Slide++ website's primary functionality centres around recommending three main categories of resources: web pages, videos, and discussion (Q&A) threads on forums such as Stack Overflow as shown in Figure 1 (left).

The overall architecture and content augmentation process is shown in Figure 1 (right). To initiate the content augmentation process, instructors upload their teaching slides to Slide++ for student viewing. A statistics-based keyword extraction mechanism (detailed shortly) is used to generate potential query words for each slide, which in turn are used to extract and display resources from the web. Instructors can subsequently refine these keywords based on their expertise, and also 'pin' any important links retrieved by the process to highlight those resources to students.

We implemented a logging mechanism to capture student behaviour such as their login patterns, engagement with course content (including slide set viewing and navigation) and their interactions with the provided supplementary materials such as web pages, videos, and Q&A links.

Query Word Extraction. In Slide++, we employ a blend of statistical methods to automate the extraction of query words. The aim is not to generate flawless keywords for each slide but rather to serve as an assisting mechanism for instructors to properly define the query words for a given slide.

We initially transform the text using the TF-IDF technique [6], which gauges the relative importance of a word within the given corpus. (In our context, a single document corresponds to a slide in a slide deck, and a term is a word present in the slide.) The equations 1, 2, 3 provide the related definitions for the representations utilised in our query text extraction mechanism.

$$tf(t,i) = f_{(t,i)} \tag{1}$$

$$idf(t,D_m) = log\left(\frac{|D_m|+1}{df(t,D_m)+1}\right)+1 \tag{2}$$

$$x_t = tfidf(t,i,D_m) = tf(t,i) * idf(t,D_m) \tag{3}$$

The term $f_{(t,i)}$ denotes the frequency of word t in the document i. In our scenario, it can be considered as a count of a word's appearance in a slide in the given slide set. Term $df(t,D_m)$ specifies the number of documents in the corpus D_m in which a word t appears uniquely. The x_t term provides the final representation of word t, with respect to the considered corpus. Beyond word occurrences, to leverage the structural details in PowerPoint slide sets, we use bullet levels inside of a slide and consider it as a weighting factor for word representations. Below are the relevant equations to calculate the weighted values for the considered tokens in a given slide set based on order and level information.

$$penalty_{(t,l)} = tf(t,i) * e^{-log(\beta_l)*l} \tag{4}$$

$$penalty_{(t,o)} = \begin{cases} tf(t,i) * e^{-log(\beta_o)*o} & \text{if } W \geq \omega \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

$$weight_{t,i} = penalty_{(t,l)} * w_l + penalty_{(t,o)} * (1 - w_l) \tag{6}$$

Here, t, l, i, o, and W are symbols denoting the considered token, token level, document (slide), token order value in the slide, and the total number of words in the document respectively. Furthermore, w_l denotes a parameter that adjusts the values coming from level penalty and order penalty, and ω is a threshold parameter we set that limits the effect of penalties on slides with low text content.

In addition to considering the importance of word ordering within a slide, we utilise the importance of slide titles as an additional factor along with a Gaussian window-based method to incorporate information from neighbouring slides.

$$token_{(t,i)} = weight_{(t,i)} * \alpha + title_{(t,i)} * (1 - \alpha) \tag{7}$$

$$x_{(t,i)} = token_{(t,i)} * P(i \mid \mathcal{N}(0, \sigma^2)) \tag{8}$$

Here, $x_{(t,i)}$ represents the value of term t in document i. The Gaussian distribution $\mathcal{N}(0, \sigma^2)$ represents a predefined variance σ^2 and zero mean. We calculate $x_{(t,i)}$ for a given slide text content around its neighbouring slides (a hyper-parameter) to calculate the final representation values for all vocabulary tokens.

For a given slide we select the highest ranking tokens based on the above weights and display those in the Slide++ tool as the selected query words. Note that these query words are *suggestions*, in that they can be overridden by the instructor, but they help to reduce the load on instructors.

4 Study and Discussion

4.1 Study Design

To evaluate the effectiveness of Slide++, we deployed the tool in a field study involving four undergraduate computing courses (running in parallel) at our institution. Instructors of the four courses agreed to participate by utilising the website to upload their slides and verifying the generated keywords and supplementary resources. In the first lesson of the term, instructors demonstrated the Slide++ website, and invited students to use it as a learning resource throughout the course. Students were not forced to use the tool (slides were also distributed through the usual learning management system). At the end of the term, students were invited to participate in an online survey and a focus group session. This participation was entirely optional for the students, but came with a small monetary token of appreciation.

We trialled Slide++ on two programming courses (*Data Structures & Algorithms*, *Web Application Development*) as well as two software engineering courses (*Software Project Management*, *IT Solution Lifecycle Management*). Based on information recorded in the system, over 400 unique student accounts

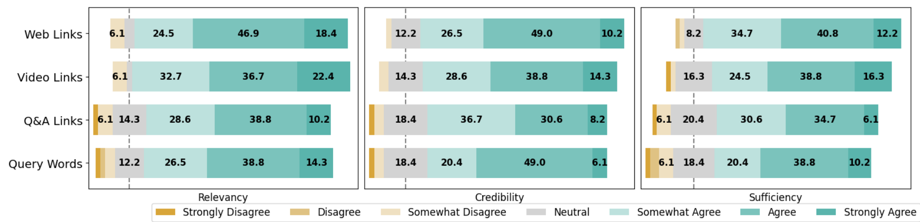

Fig. 2: Extent students agree that resources are relevant, credible, and sufficient

were registered on the website throughout the term, and over 40 slide sets were uploaded by instructors across the four courses.

For the survey, we were able to gather 49 student responses, with the following distribution across the four courses respectively: 33.33%, 11.11%, 22.22% and 33.33%. Among the respondents, 57% of them reported their gender as female, 39% as male, and the remaining 4% preferred not to say. (Note that our student cohort is approximately 40% female and 60% male.) Furthermore, 73% of the students shared that they were taking our Information Systems degree, with the others all taking Computer Science. Of these 49 students, 13 students accepted our invitations to join one of three focus group sessions.

Based on the data we collected from Slide++ over the considered term, the 'activity count' on the website hit around 6,500 actions (clicking links, opening slide sets, traversing slides) per week in total. Among the students who clicked at least one recommended link, the overall average number of clicks rises to 7.64, with a maximum of 41. This data indicates a reasonable amount of tool usage and provides motivation to explore the use-cases further.

4.2 Relevancy, Credibility, and Sufficiency of Slide++

In our survey, we focused on questions to explore the participants' thoughts on the generated query words and recommended web resources (web links, videos, and Q&A). We used a 7-point Likert scale to measure each participant's evaluation of the relevance, credibility, and sufficiency of the recommended resources.

Figure 2 summarises the responses received to these questions. Overall, students agreed that Slide++ was able to provide relevant, credible and sufficient additional resources. Web links received the highest percentage of positive responses, whereas Q&A links and query words received slightly lower responses. We believe this is because good Q&A links (i.e. specific technical discussions on Stack Overflow) may not exist for the content of every slide, especially when compared to the links curated from around the broader web.

During the focus group sessions we discovered that even though students appreciated the links generated by the Slide++, they still preferred to have an explicit indication of credibility regarding the suggested contents. The reasoning behind this request seems to be originating from their experiences with the 'hallucination' effects of novel generative models. In focus group sessions, students mentioned issues with tools like ChatGPT and it seems that such technologies have made them suspicious of anything automatically generated:

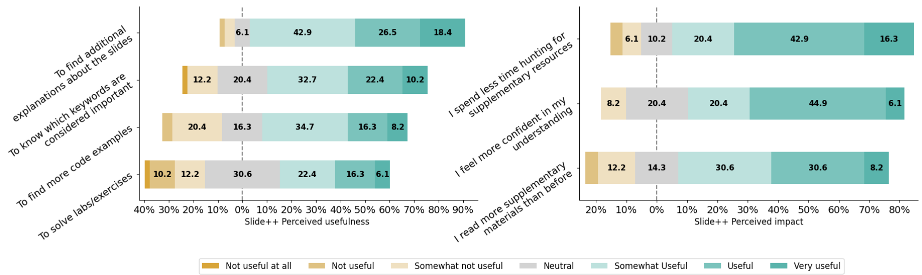

Fig. 3: Perceived usefulness and impact of Slide++

> *"... I actually don't use ChatGPT; instead I use Bing AI. It's just based on ChatGPT, but because it provides links and sources to what it's talking about, [I can verify] if it's hallucinating or not."*

4.3 Perceived Benefits and Impact of Slide++

In our survey, we asked students for their feedback on how useful they found the Slide++ tool for four objectives. Figure 3 (left) summarises how useful they perceived the tool for those objectives. The overall responses lean positive, with perceived usefulness for finding additional explanations being the highest.

During the focus groups, some students shared that the tool was particularly helpful for more 'subjective' topics (e.g. the agile software process 'scrum' as taught in Software Project Management) and appreciated having links to additional articles with different perspectives to those of their instructors.

While the curated resources themselves were found useful, the focus group participants conveyed that they did not find it helpful to see the query words that were generated. We suspect that this is because Slide++ currently displays them as a list of keywords, rather than as a description of the slide.

Finally, we investigated the impact of Slide++ on students' confidence in their understanding, as well as how the tool has changed their approach to finding and using supplementary material. Figure 3 (right) summarises the responses. Overall, the responses suggest that students feel more confident in their understanding of course materials and seemingly spend less time finding resources.

A key lesson we learnt in running this study is the importance of ensuring new pedagogical tools integrate seamlessly with the learning methods and systems students normally use. For example, some students suggested to integrate Slide++ into the university learning management systems, or even directly in slides via a plug-in (as they prefer to annotate slides directly as part of their learning process). We hope to explore such possibilities in the future.

5 Generative Model-based Query Sentence Generation

Slide++ currently employs an efficient statistics-based approach, but we also wanted to assess whether LLMs may help to improve the query words generated—especially for image-heavy slides that lack keywords to extract.

Table 1: BERT F1-scores of statistics-based and LLM-based summaries

Course Name	**Statistics-based**	**LLM-based**
Web Application Development	0.77	0.81
Data Structures and Algorithms	0.83	0.80
IT Solution Lifecycle Management	0.80	0.80
Software Project Management	0.83	0.81

Table 2: Examples of query text from statistics-based and LLM-based approaches

Statistics-based	**LLM-based**
type linked declaring array arrays elements initialize java	Initializing Java arrays with types and sizes.
analysis algorithm exists proof	Quantifiers in logical statements described.
dom getElementsByTagName javascript array index	Accessing elements in HTML by tag name.
granularity service team size microservices	Microservice size varies, consider business needs and two-pizza rule.

We implemented a proof of concept query word generator using the OpenAI GPT4-Vision API, and used it to extract key descriptions of slide contents. Instead of just using the text inside of a slide, we incorporated the full slide image as an input. Furthermore, while the statistics-based approach generates a list of keywords, we wanted to explore whether GPT4 could present those keywords in a more intelligible *sentence* describing the content of the slide (while also being useful to extract resources from the web).

Table 1 shows the course-wise BERTScore [10] comparisons between the statistics-based approach and GPT4-based summaries. The score differences between summary values are not significant, indicating that the statistics-based approach performs well, despite being a more cost-efficient method. However, the query sentences generated by GPT4 appear to be more readable than those in the statistics-based approach as can be seen from the examples in Table 2.

We will be developing Slide++ further to explore combinations of these possibilities. For example, the statistics-based approach could be used for text-heavy slides, with the LLM-based approach for image-heavy ones. LLMs may also be helpful for generating summaries of the whole slide set, similar to the approach of Shimada et al. [7].

6 Conclusion

In order to provide students with an enhanced learning experience, we developed a website, Slide++, that offers functionality for automatically generating query words and links to supplementary materials. We trialled it in four undergraduate computing courses, inviting students to participate in a survey and focus group discussion at the end of the term. We found that students deemed the generated links to be relevant, credible, and sufficient, and that Slide++ helped to improve their confidence in their understanding of the course content. Students conveyed the usefulness of the tool in a variety of scenarios, e.g. generating links to explain diagrams in slide sets, finding interesting articles for class reflection exercises, and discovering different (but credible) perspectives to those of their instructors.

In the future, we plan to further explore LLM-based approaches for more advanced query extraction (especially on image-heavy slides). In addition, we would like to explore how the slide augmentation features of Slide++ might be integrated into common educational tools such as learning management systems, in order to reduce the friction currently imposed by requiring the use of a standalone website.

Acknowledgements. This research/project is supported by the Ministry of Education, Singapore under its Tertiary Education Research Fund (MOE Reference Number: MOE2021-TRF-013). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore. Hady W. Lauw gratefully acknowledges the support by the Lee Kong Chian Fellowship awarded by Singapore Management University.

References

1. Adeshola, I., Adepoju, A.P.: The opportunities and challenges of ChatGPT in education. Interactive Learning Environments **0**(0), 1–14 (2023)
2. Barria-Pineda, J., Akhuseyinoglu, K., Zelem-Celap, S., Brusilovsky, P., Klasnja-Milicevic, A., Ivanovic, M.: Explainable recommendations in a personalized programming practice system. In: AIED (1). Lecture Notes in Computer Science, vol. 12748, pp. 64–76. Springer (2021)
3. Montenegro-Rueda, M., Fernández-Cerero, J., Fernández-Batanero, J.M., López-Meneses, E.: Impact of the implementation of ChatGPT in education: A systematic review. Computers **12**(8) (2023)
4. Oliveira, A., Teixeira, M.M., Neto, C.d.S.S.: Recommendation of educational content to improve student performance: An approach based on learning styles. In: CSEDU (2). pp. 359–365 (2020)
5. Ouh, E.L., Gan, B.K.S., Shim, K.J., Wlodkowski, S.: ChatGPT, can you generate solutions for my coding exercises? An evaluation on its effectiveness in an undergraduate Java programming course. In: ITiCSE (1). pp. 54–60. ACM (2023)
6. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc. (1986)
7. Shimada, A., Okubo, F., Yin, C., Ogata, H.: Automatic summarization of lecture slides for enhanced student preview-technical report and user study-. IEEE Trans. Learn. Technol. **11**(2), 165–178 (2018)
8. Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachsler, H., Bosnic, I., Duval, E.: Context-aware recommender systems for learning: A survey and future challenges. IEEE Trans. Learn. Technol. **5**(4), 318–335 (2012)
9. Wölfel, M.: Towards the automatic generation of pedagogical conversational agents from lecture slides. In: Multimedia Technology and Enhanced Learning. pp. 216–229. Springer (2021)
10. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: Evaluating text generation with BERT. In: ICLR. OpenReview.net (2020)