

2.3 EducationalWeb System

The EducationalWeb system (<http://timan102.cs.illinois.edu/explanation//slide/cs-410/0>) is a tool to help students learn from course slides. It has two main functionalities currently: 1) Retrieve and recommend relevant slides for each slide. You can read more about this in the following papers [Web of Slides](#), [WOS Demo](#); 2) Find an explanation of a term/phrase on the slide by highlighting it and then clicking on the "cap/scholar" button on the top-right of a slide. It will try to retrieve a relevant section from the Professor's textbook that contains an explanation of the selected phrase. You can read more about the underlying algorithm [here](#).

The code for the system is available [here](#). Below are some ideas to improve and expand this system. You may choose to integrate your code with the existing system, or borrow some ideas from it, or build your own systems/algorithms.

- *Improving the usability and reach of the existing system*

Some of you might have used the system and identified potential areas of improvement. The aim of this subtopic is to refine the current version of EducationWeb. Some specific ideas include (many are borrowed from [this Piazza post](#)):

1. **Scale up the current system.** Add more slides and courses from multiple sources e.g. Coursera, UIUC courses, etc. and run the existing algorithms on them. Again, it might be useful to think about automatic crawling similar to the subtopic in 2.2 above. It would be very interesting to see the interaction between slides/textbooks at a large scale!!
2. **Improve the performance of the system.** Currently, loading each slide takes time.
3. **Allow downloading slides in bulk.** Currently, we can only download one slide at a time.
4. **Add more context to the explanations** (e.g. link to the specific page in the textbook)
5. **Allow adding additional courses/lectures directly from the web interface.** This would also involve dynamically identifying the recommended/relevant slides for a new slide. Currently, a static file is used which contains pre-computed recommendations for each slide.
6. **Integrate the tool with Piazza/Coursera**, i.e. maybe link Piazza/Coursera to the tool or vice-versa. Alternatively, add discussion forum and video capabilities to the tool so that it serves as a one-stop-shop for all users' educational needs.
7. **Link to latest related research articles:** In this way, the lecture content can be automatically updated
8. You could also work on **improving** the current recommendation, search and explanation mining **algorithms** (described in the papers at the beginning of this section 2.3)

- *Automatically creating teaching material for in-demand skills*

This subtopic is an extended version of the existing EducationWeb system. There is an increasing demand for skilled workers in the industry. Quality education is not easily accessible to everyone due to barriers such as high cost, geographical and language barriers, etc. Also, instructors cannot be available 24*7 to provide personalized support to all learners. In this subtopic, the overarching aim is to tackle some of these issues. In particular, the following tasks might be good starting points.

- **Identifying in-demand skills:** You can crawl and analyze relevant sections of job boards, news articles, scientific articles, social media, etc. to automatically identify the *emerging keywords/topics*. For this, you may refer to some papers on contextual text mining (mentioned in Option 1 of this document).
- **Creating lectures and tutorials for those skills:** For this, you may consider lecture slides (e.g. from Coursera courses) as the basic units of knowledge. Then, the task could be to find the most relevant slides or clusters of slides (could be across multiple courses/lectures) for a given skill (topic). You may borrow some ideas from the EducationWeb system for this.
You may also use the slides in existing lectures on some topics as the “relevant slides” for those topics. In this way, you can automatically generate training data for supervised learning.

You could also combine knowledge from multiple sources (e.g. textbook sections, slides, videos, blogs, codebases) for creating more comprehensive tutorials.

A more challenging task would be to automatically *generate* the lectures/tutorials using techniques from natural language generation and abstractive text summarization. Another interesting idea is to automatically *generate agents*, e.g. using Virtual Agent Interaction Framework (VAIF). This goes beyond the material covered in class but could lead to some highly innovative and state-of-the-art projects!

If you choose this option, please answer the following questions in your proposal:

1. What are the names and NetIDs of all your team members? Who is the captain? The captain will have more administrative duties than team members.
2. What system have you chosen? Which subtopic(s) under the system?
3. Briefly describe the datasets, algorithms or techniques you plan to use
4. If you are adding a function, how will you demonstrate that it works as expected? If you are improving a function, how will you show your implementation actually works better?
5. How will your code communicate with or utilize the system? It is also fine to build your own systems, just please state your plan clearly
6. Which programming language do you plan to use?

7. Please justify that the workload of your topic is at least $20 \cdot N$ hours, N being the total number of students in your team. You may list the main tasks to be completed, and the estimated time cost for each task.

At the final stage of your project, you need to deliver the following:

- Your documented source code.
- A demo that shows your implementation actually works. If you are improving a function, compare your results to the previously available function. If your implementation works better, show it off. If not, discuss why.