CS-410
Text Information Systems
Final Project

Text Miners
Ashish Kumar Pradhan (apradh6@illinois.edu)
Kirti Magadum (magadum2@illinois.edu)
Bhuvaneswari Periasamy (bp14@illinois.edu)

Improving
EducationalWeb System

1. Introduction

The EducationalWeb System is the term given to the Web of Slides (WOS) created at the University of Illinois at Urbana Champaign by Sahiti Labhishetty, Bhavya, Kevin Pei, Assma Boughoula and Prof. Chengxiang Zhai. It aimed to link all the lecture slides of a course which would be easily navigable and searchable. Not only that, with Machine Learning, it is also able to provide a list of related slides which has some similarity to the current slide displayed. It is created as a Python Flask Web App with an interface which is similar to opening up a PowerPoint presentation. Navigation can be primarily done by clicking through the previous and next buttons in the interface as well as from the list of similar slides provided by the system. A separate system allows the search ability through which the navigation can be done as well.

This project consisted of adding a few features which would enhance the System so that it can be used in a more widespread manner. Keeping true to the vision which was to create a system which would interact with each of the slides in a way that the current internet interacts, the modifications are aimed that doing just that.

2. Background

The existing code consists of two major components:

a. The related slides code which computes the similarity between slides using Deep Learning https://github.com/Bhaavya/mooc-web-of-slides
b. The structural part of the application which consists of the rest of the Python Flask Web App. https://github.com/CS410Fall2020/EducationalWeb

Initially, we thought of incorporating both parts of the code into one and then including the modifications but due to the computationally heavy part of the Deep Learning component, we decided to exclude this part completely and focus on adding the new components to the Web App.

3. Code Explanation

Before incorporating the modifications, installing the existing code from the repository was necessary. We have detailed the following steps taken for installing the code in our Windows Machines.

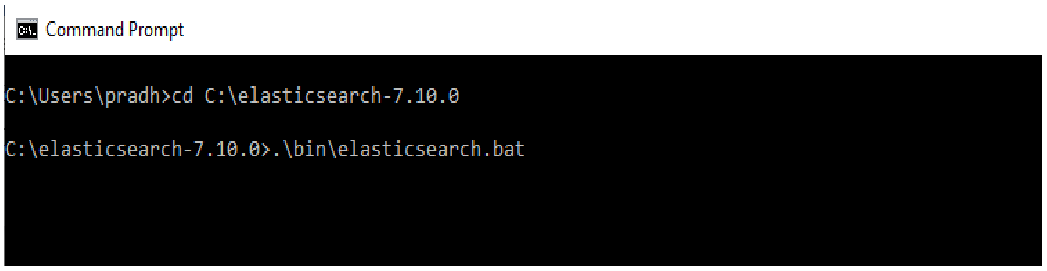
a. Download the
Download the code from the github page (https://github.com/CS410Fall2020/EducationalWeb) to a folder EducationalWeb in your local machine

b. Elastic Search
Elastic Search was installed by downloading the zip file from the Elastic Search Website.
https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.10.1-windows-

x86_64.zip. Unzip the folder in a folder **ElasticSearch** and then run the following code in the command prompt after changing the directory to that location.

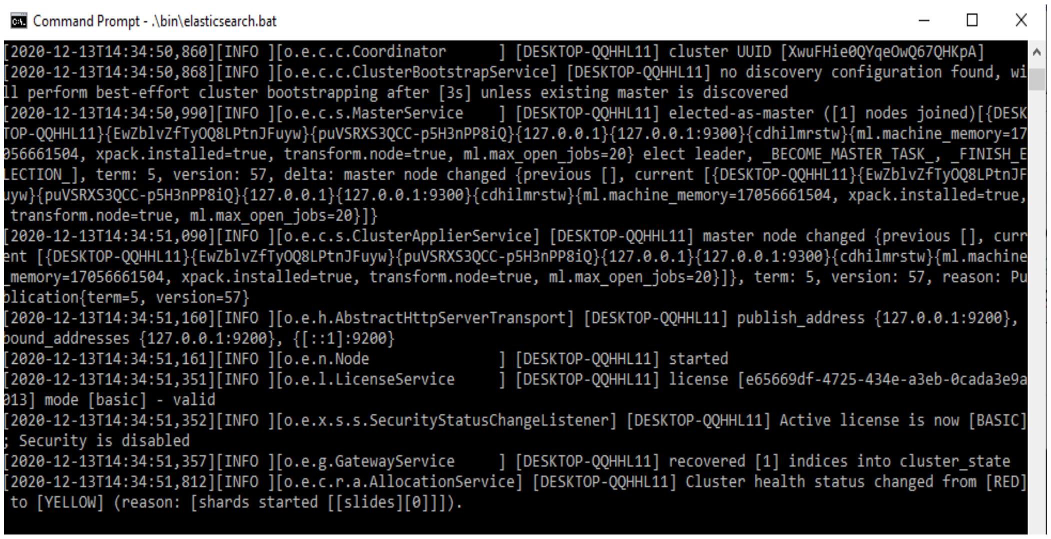
.\bin\elasticsearch.bat

This will start the elastic search in your local machine.



```
C:\Users\pradh>cd C:\elasticsearch-7.10.0

C:\elasticsearch-7.10.0>.\bin\elasticsearch.bat
```

Fig 1



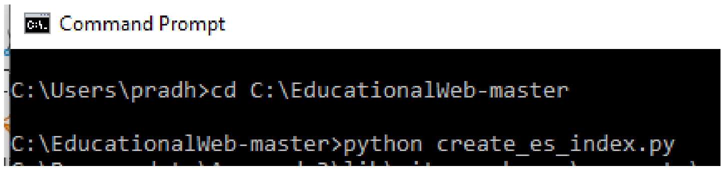
```
[2020-12-13T14:34:50,860][INFO ][o.e.c.c.Coordinator      ] [DESKTOP-QQHHL11] cluster UUID [XwuFHie0QYqeOwQ67QHKpA]
[2020-12-13T14:34:50,868][INFO ][o.e.c.c.ClusterBootstrapService] [DESKTOP-QQHHL11] no discovery configuration found, wi
ll perform best-effort cluster bootstrapping after [3s] unless existing master is discovered
[2020-12-13T14:34:50,990][INFO ][o.e.c.s.MasterService    ] [DESKTOP-QQHHL11] elected-as-master ([1] nodes joined)[{DESK
TOP-QQHHL11}{EwZblvZfTyOQ8LPtnJFuyw}{puVSRXS3QCC-p5H3nPP8iQ}{127.0.0.1}{127.0.0.1:9300}{cdhilmrstw}{ml.machine_memory=17
056661504, xpack.installed=true, transform.node=true, ml.max_open_jobs=20} elect leader, _BECOME_MASTER_TASK_, _FINISH_E
LECTION_], term: 5, version: 57, delta: master node changed {previous [], current [{DESKTOP-QQHHL11}{EwZblvZfTyOQ8LPtnJF
uyw}{puVSRXS3QCC-p5H3nPP8iQ}{127.0.0.1}{127.0.0.1:9300}{cdhilmrstw}{ml.machine_memory=17056661504, xpack.installed=true,
 transform.node=true, ml.max_open_jobs=20}]}
[2020-12-13T14:34:51,090][INFO ][o.e.c.s.ClusterApplierService] [DESKTOP-QQHHL11] master node changed {previous [], curr
ent [{DESKTOP-QQHHL11}{EwZblvZfTyOQ8LPtnJFuyw}{puVSRXS3QCC-p5H3nPP8iQ}{127.0.0.1}{127.0.0.1:9300}{cdhilmrstw}{ml.machine
_memory=17056661504, xpack.installed=true, transform.node=true, ml.max_open_jobs=20}]}, term: 5, version: 57, reason: Pu
blication{term=5, version=57}
[2020-12-13T14:34:51,160][INFO ][o.e.h.AbstractHttpServerTransport] [DESKTOP-QQHHL11] publish_address {127.0.0.1:9200},
bound_addresses {127.0.0.1:9200}, {[::1]:9200}
[2020-12-13T14:34:51,161][INFO ][o.e.n.Node               ] [DESKTOP-QQHHL11] started
[2020-12-13T14:34:51,351][INFO ][o.e.l.LicenseService     ] [DESKTOP-QQHHL11] license [e65669df-4725-434e-a3eb-0cada3e9a
013] mode [basic] - valid
[2020-12-13T14:34:51,352][INFO ][o.e.x.s.s.SecurityStatusChangeListener] [DESKTOP-QQHHL11] Active license is now [BASIC]
; Security is disabled
[2020-12-13T14:34:51,357][INFO ][o.e.g.GatewayService     ] [DESKTOP-QQHHL11] recovered [1] indices into cluster_state
[2020-12-13T14:34:51,812][INFO ][o.e.c.r.a.AllocationService] [DESKTOP-QQHHL11] Cluster health status changed from [RED]
 to [YELLOW] (reason: [shards started [[slides][0]]]).
```

Fig 2

c. Create index (need to be done only once)

Run the following script after navigating to the EducationalWeb folder in the command prompt

python create_es_index.py

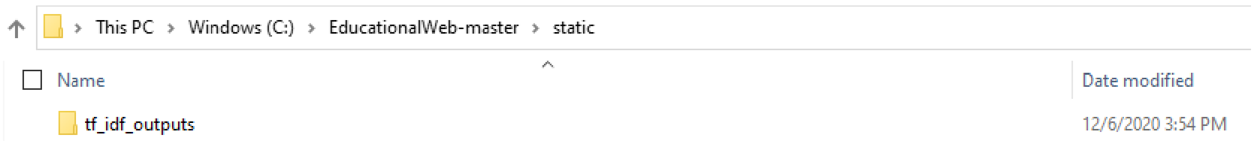


```
Command Prompt

C:\Users\pradh>cd C:\EducationalWeb-master

C:\EducationalWeb-master>python create_es_index.py
```

Fig 3

d. Download tfidf_outputs.zip folder

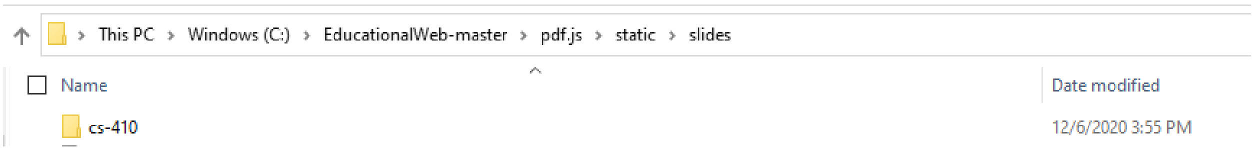
Download it from the following link, unzip it and save it in EducationalWeb/static
https://drive.google.com/file/d/19ia7CqaHnW3KKxASbnfs2clqRlgdTFiw/view?usp=sharing



This PC › Windows (C:) › EducationalWeb-master › static

Name	Date modified
tf_idf_outputs	12/6/2020 3:54 PM

e. **Download the lecture slides**
Use the following link to download the lecture slides and then unzip it to EducationalWeb/pdf.js/static/slides/
https://drive.google.com/file/d/1Xiw9oSavOOeJsy_SIilxPf4aqsuyuuh6/view?usp=sharing



This PC › Windows (C:) › EducationalWeb-master › pdf.js › static › slides

Name	Date modified
cs-410	12/6/2020 3:55 PM

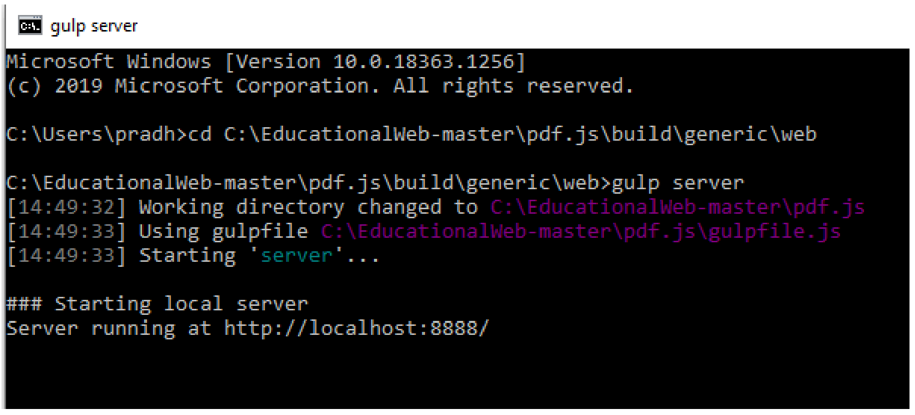
f. **Install gulp server**

Follow the steps in this page (https://gulpjs.com/docs/en/getting-started/quick-start/) to install gulp server

g. **Run Gulp Server**

Open a separate terminal window and navigate to EducationalWeb/pdf.js/build/generic/web and execute the following command to run gulp server

gulp server



```
gulp server
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\pradh>cd C:\EducationalWeb-master\pdf.js\build\generic\web

C:\EducationalWeb-master\pdf.js\build\generic\web>gulp server
[14:49:32] Working directory changed to C:\EducationalWeb-master\pdf.js
[14:49:33] Using gulpfile C:\EducationalWeb-master\pdf.js\gulpfile.js
[14:49:33] Starting 'server'...

### Starting local server
Server running at http://localhost:8888/
```

Fig 4

h. **Run the application**

In another terminal window, navigate to the EducationalWeb folder and run the following script to start the code. Make sure that all the python libraries present in the code are installed before the script is run.

```
python app.py
```

i. **Launch the application**

The application should open in http://localhost:8096/

4. Modifications to the code

Our goal is to add additional courses to the existing code so that users can navigate between courses and utilize the features available to maximize the learning ability from those courses. In order to achieve that, we have made two additions to the code, namely:

a. Created a web crawler which would take as input the url of the webpage which contains all the lecture content and create a list of urls which contains the pdf links
b. Download the pdfs, and subsequently create a repository like the existing one for CS410, containing folders for each lecture pdfs which will contain single page pdfs split automatically by the code.
c. These courses are then incorporated into the existing code, bringing in all the course slides into the EducationalWeb which can then be navigated like the existing CS410 course

Web Crawler

An automatic Web Crawler is essential to any web application which needs to add new content from the web. A web crawler essentially follows all the links from a starting page and then retrieves information from the pages as desired. Our web crawler recursively follows all the links which are contained from a starting page. If it finds a dead end in any of the pages, i.e. there are no additional pages to scrape from that page, it will just go one step up and then continue the scraping of the pages from there on.

The creation of the web crawler was done by using the very useful Beautiful Soup library which parses content from HTML pages using a tree structure. It is very useful in navigating, searching and modifying a parse tree of an html page, all very essential parts of web scraping.

Download the lecture content

After the web crawler scrapes through all the URLs containing lecture slides in each webpage, we need to download them and structure them in such a way so that it fits the educational web code. The existing code handles only single page PDFs which are then navigated and searched by the system. To achieve that for the newly added courses, these are the steps that were taken:

1. Download the pdfs from the URLs scraped by the web crawler.
2. Scrape the lecture title from the webpage
3. Create a folder structure like the existing 410 slides. There needs to be one folder for every Lecture pdf

4. Split the pdfs into single page pdfs and then add them to that Lecture's folder.

For our project, we have included 3 additional University of Illinois courses whose lecture slides are publicly available. Those are:

1. ECE 313 (https://courses.engr.illinois.edu/ece313/sp2013/Slides.html)
2. CS 425 (https://courses.engr.illinois.edu/cs425/fa2019/lectures.html)
3. CS 554 (https://solomonik.cs.illinois.edu/teaching/cs554/index.html)

Add the slides to the existing code

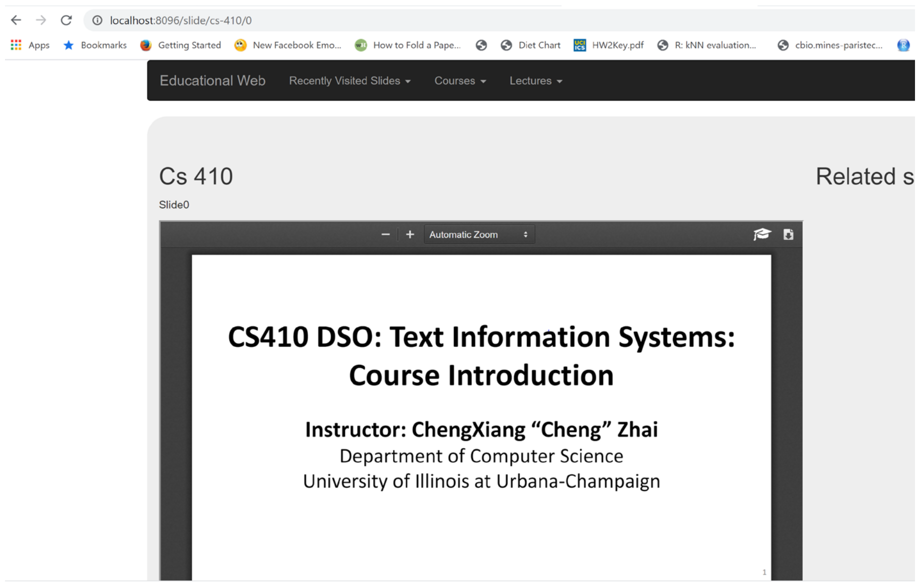
After the slides have been created according to the configuration of the code, modifications had to be made to add them under the drop downs under the "Courses" section. We specifically parsed texts from these webpages to add the details to that course section in the system. The details were present in various tags which needed to be extracted so that we get the relevant titles for the slides.

Running the Modified Code

The code has been integrated in the existing system in such a way that it does not need any additional input from the user.

Changes were made inside the model.py and app.py scripts with the additional functions which were defined being referenced from within the existing functions, thus reducing the need for any user input specifically.

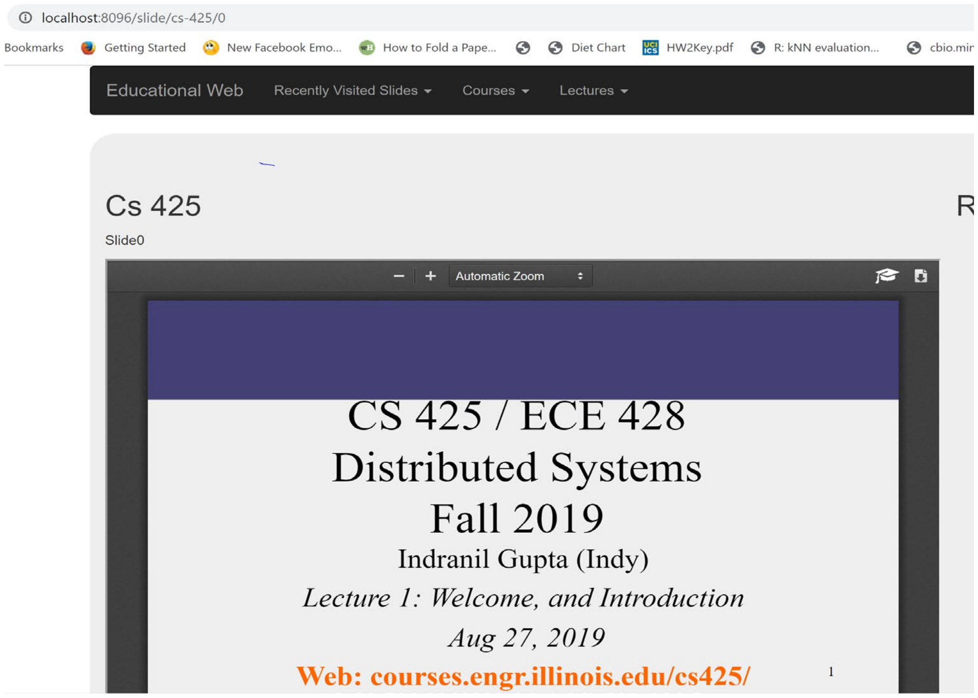
To run the modified code, all that's required is to run the app.py function again which will start the web scraping from the webpages and then create the folder structures for each of the subject and then integrate them into the existing EducationalWeb System.



CS410 DSO: Text Information Systems: Course Introduction

Instructor: ChengXiang "Cheng" Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign

Fig 5


localhost:8096/slide/cs-425/0

Bookmarks Getting Started New Facebook Emo... How to Fold a Pape... Diet Chart HW2Key.pdf R: kNN evaluation... cbio.mir

Educational Web Recently Visited Slides Courses Lectures

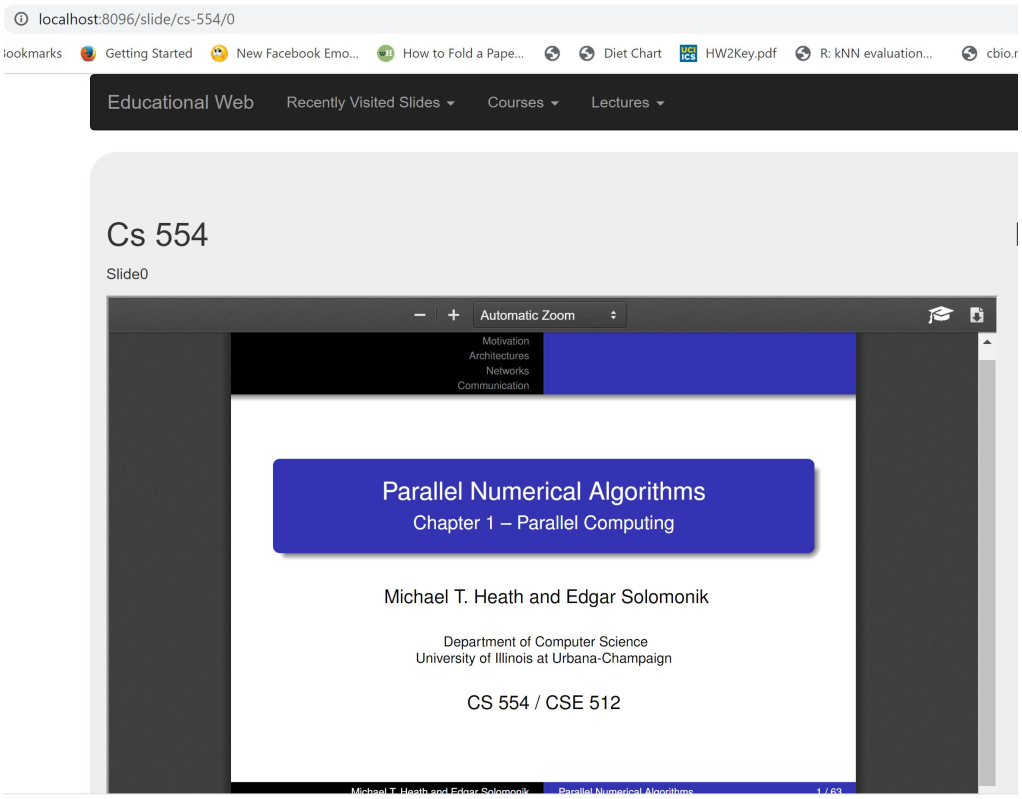
Cs 425

Slide0

Automatic Zoom

CS 425 / ECE 428
Distributed Systems
Fall 2019
Indranil Gupta (Indy)
Lecture 1: Welcome, and Introduction
Aug 27, 2019
Web: courses.engr.illinois.edu/cs425/ 1

Fig 6


localhost:8096/slide/cs-554/0

ookmarks Getting Started New Facebook Emo... How to Fold a Pape... Diet Chart HW2Key.pdf R: kNN evaluation... cbio.r

Educational Web Recently Visited Slides Courses Lectures

Cs 554

Slide0

Automatic Zoom

Motivation
Architectures
Networks
Communication

Parallel Numerical Algorithms
Chapter 1 – Parallel Computing

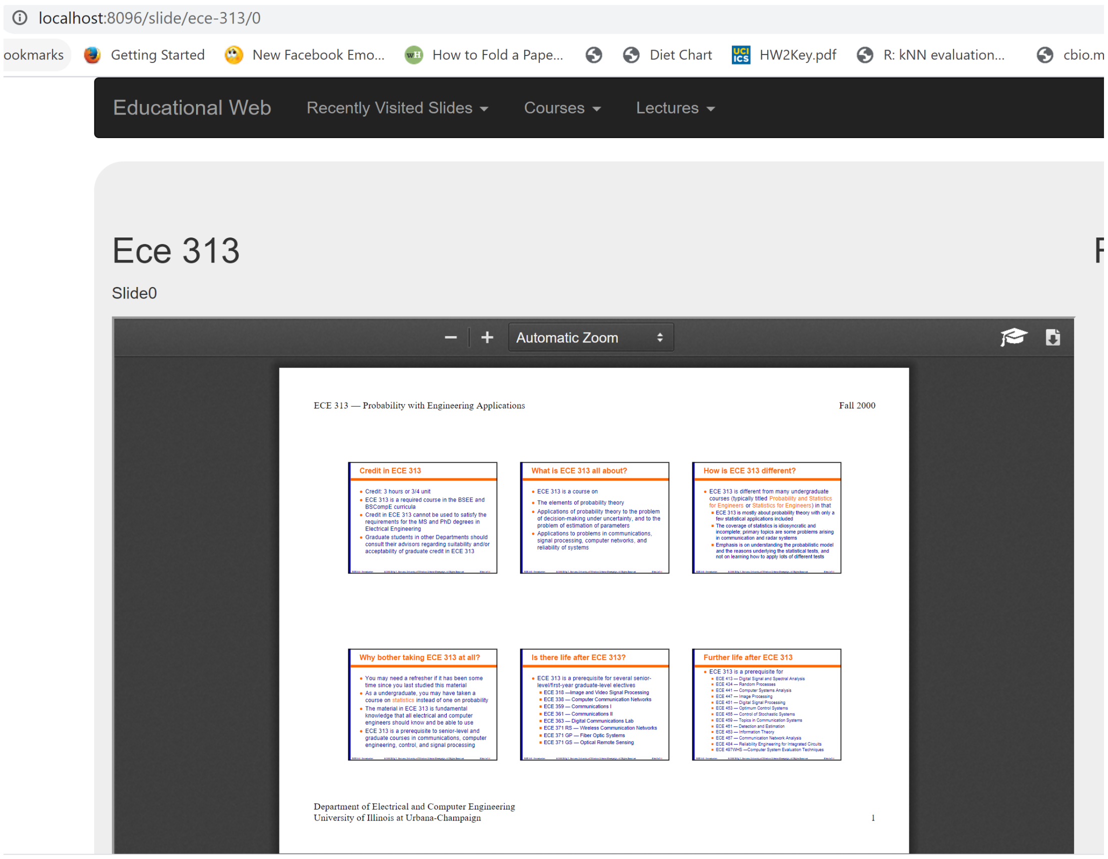
Michael T. Heath and Edgar Solomonik

Department of Computer Science
University of Illinois at Urbana-Champaign

CS 554 / CSE 512

Michael T. Heath and Edgar Solomonik Parallel Numerical Algorithms 1 / 63

Fig 7



Ece 313

Slide0

ECE 313 — Probability with Engineering Applications

Fall 2000

Credit in ECE 313

- Credit: 3 hours or 3/4 unit
- ECE 313 is a required course in the BSEE and BSCompE curricula
- Credit in ECE 313 cannot be used to satisfy the requirements for the MS and PhD degrees in Electrical Engineering
- Graduate students in other Departments should consult their advisors regarding suitability and/or acceptability of graduate credit in ECE 313

What is ECE 313 all about?

- ECE 313 is a course on
- The elements of probability theory
- Applications of probability theory to the problem of decision-making under uncertainty, and to the problem of estimation of parameters
- Applications to problems in communications, signal processing, computer networks, and reliability of systems

How is ECE 313 different?

- ECE 313 is different from many undergraduate courses (typically titled Probability and Statistics for Engineers or Statistics for Engineers) in that
  - ECE 313 is mostly about probability theory with only a few statistical applications included
  - The coverage of statistics is idiosyncratic and incomplete; primary topics are some problems arising in communication and radar systems
  - Emphasis is on understanding the probabilistic model and the reasons underlying the statistical tests, and not on learning how to apply lots of different tests

Why bother taking ECE 313 at all?

- You may need a refresher if it has been some time since you last studied this material
- As a undergraduate, you may have taken a course on statistics instead of one on probability
- The material in ECE 313 is fundamental knowledge that all electrical and computer engineers should know and be able to use
- ECE 313 is a prerequisite to senior-level and graduate courses in communications, computer engineering, control, and signal processing

Is there life after ECE 313?

- ECE 313 is a prerequisite for several senior-level/first-year graduate-level electives
  - ECE 318 —Image and Video Signal Processing
  - ECE 338 — Computer Communication Networks
  - ECE 359 — Communications I
  - ECE 361 — Communications II
  - ECE 363 — Digital Communications Lab
  - ECE 371 RS — Wireless Communication Networks
  - ECE 371 GP — Fiber Optic Systems
  - ECE 371 GS — Optical Remote Sensing

Further life after ECE 313

- ECE 313 is a prerequisite for
  - ECE 413 — Digital Signal and Spectral Analysis
  - ECE 434 — Random Processes
  - ECE 441 — Computer Systems Analysis
  - ECE 447 — Image Processing
  - ECE 451 — Digital Signal Processing
  - ECE 453 — Optimum Control Systems
  - ECE 455 — Control of Stochastic Systems
  - ECE 459 — Topics in Communication Systems
  - ECE 461 — Detection and Estimation
  - ECE 463 — Information Theory
  - ECE 467 — Communication Network Analysis
  - ECE 484 — Reliability Engineering for Integrated Circuits
  - ECE 497WHS —Computer System Evaluation Techniques

Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

1

Fig 8

5. Limitations and Scope for Further Work

1. Due to an absence of a training dataset, a classification model could not be created to correctly classify a webpage as one which contains lecture slides. Due to this we had to explicitly input the starting URL for web scraping. Creating that model required much more time than what could be allotted to this project.
2. The related slides section which uses Deep Learning to find out the similarity between the slides could not be incorporated. Additional work needs to be done to integrate it into the existing code
3. Documentation of the code and how to navigate between the scripts would have been helpful to understand the code in the initial stages and not spend a lot of time in that.

6. Contribution of Team Members

Team Member	Contribution
Ashish Pradhan (Captain)	Web Crawler, Documentation
Kirti Magadum	PDF splitting code, integration to the existing code
Bhuvaneswari Periasamy	Modeling code, Text parsing code

7. Conclusion

The EducationalWeb is a powerful tool which can assist the learners from learning in an optimized manner without spending a lot of time in searching and navigating through slides across different topics. There is tremendous potential in this application and the scope of incorporating a lot of features is immense.

The modifications which we added to the existing code paves the way for additional work which would improve the system. As stated in the scope above, creating a classification model and adding it to our web scraper would make it largely automated and be able to scrape the pdf urls from anywhere in the web