6-2014

Structuring the Unstructured Note: Automatic Organizing and Formatting for lecture notes.

Shiqing He
Union College - Schenectady, NY

Structuring the Unstructured Note:

Automatic Organizing and Formatting for lecture notes.

Shiqing He

Advisor: Kristina Striegnitz

June 13, 2014

Abstract

"Note taking is an important practice during lectures. With the rapid development of electronic devices such as smart phones, tablets and laptops, digital note-taking has become an option for many students. In the classroom setting, note takers might fail to record clearly organized notes due to limitations of time and devices. Therefore, users often need to manually organize long paragraphs of notes by dividing or structuring them into appropriate sections based on content. The goal of this research is to develop a system that automatically structures unorganized notes. Using topic modeling and automatic summarization, we detect and analyze the "structure" of the notes. We then build a webpage based on the structure in order to format notes into a more readable version.

1 Introduction

"Note-taking is among the most powerful contributors to performance in courses having a strong content base." Williams and Eggert (2002) With the rapid development of digital devices such as smart phones, tablets and laptops, taking notes digitally has become significantly easier recently.

Digital notes are normally taken in one of two ways. First, students can use a keyboard or other devices to type notes into text format. In this case, if the lecturers' speaking or writing speeds are faster than the students' typing speed, students might not be able to capture as much information as they want. It is also possible for students to type what lecturers have said word by word, resulting in having long verbose paragraphs of unorganized notes. Secondly, students can take notes in multiple media formats such as photos, audios and videos. Photos are similar to traditional notes on paper. However photo notes are more difficult to edit in comparison to texts. Although audio and video notes contain relatively complete information from lecturers, reviewing and editing these notes can be time-consuming.

Comparing with traditional notes on paper, digital notes are easier to organize and to share, but typing digital note in a lecture setting has several problems. A study by Pam Mueller and Daniel Oppenheimer suggests that typing digital note during lecture generally introduces distraction and leads note-takers to record verbose notes. Therefore taking digital note during lecture would potentially distracts the note taker. Paul (2014)

Besides typing or writing, there is another less explored note-taking strategy. Automatic note taking system, the main topic of this research, would free students from the burden of note-taking and allows them to pay more attention towards the content of the lecture and potentially enhance their studies. When students need to review the lecture, they can obtain complete and detailed lecture notes from the automatic note taking system.

Note taking is a form of information recording; therefore a note taking system requires information from lectures. In order to free users from typing during lectures, this system can acquire data input by recording target lectures into audio or video. Using speech recognition systems to process these multi-media records, we are able to transform multi-media notes into text and improve the reusability of audio notes.

Speech recognition would potentially free users from typing, thereby reducing the distraction of digital devices. Assuming the speech recognition system has a high accuracy, it will form relatively complete lectures records. Nevertheless, such texts are not ideal lecture notes because they are often unorganized long verbose paragraphs. Such texts are difficult to understand unless users manually section them into appropriate format and eliminate irreverent parts.

For example, the following paragraph could be a part of a lecture record that a note taker obtained by using a speech-recognition system:

> "Hi everyone, welcome to the first class of xxx. Today we are going to talk about MinMax search algorithm. MinMax searching algorithms are frequently used search algorithms that have multiple implementations. Here are several different MinMax search algorithms:idea1: set up two counters and keep comparing along the list,which will require n-3 comparisons. idea2: sort the list, and pick beginning and end, which requires O(nlogn) running time. idea3: think recursively. Using divide and conquer, which will be faster :(O(logn))."

The notes demonstrated above are not structured. Note takers need to read through the paragraph and understand the content in order to format this note. In the example given, there are several modifications that can improve this note immediately:

1. Remove irrelevant information such as "Hi everyone, welcome....."
2. Use paragraph break to separate concept and listing.
3. Change the the main topic, "MinMax Search Algorithm", into bold.
4. Change the keyword "idea" into italic.

After these modifications, the example paragraph becomes more organized and more readable.

> "**MinMax search algorithm** MinMax searching algorithms are frequently used search algorithms that have multiple implementations. Here are several different MinMax search algorithms:
>
> - *idea1*: set up two counters and keep comparing along the list,which will require n-3 comparisons.
> - *idea2*: sort the list, and pick beginning and end, which requires O(nlogn) running time.
> - *idea3*: think recursively. Using divide and conquer, which will be faster :(O(logn)).
>
> "

2 Goal and Design

The automatic note taking system can be divided into two parts: first, obtaining a lecture transcription, or raw text, by using a speech recognition system on audio or video recording of lectures; second, organizing the raw texts into readable, concise and organized notes.

In this research, we focus on the second part of the system and use online courses' video subtitles as raw texts.[1] We consider online courses' video subtitles as complete and accurate records of what lecturers have said, and we assume that students want to take notes from lecturers' talking.

The goal of this research is to design, develop and explore a automatic note-processing system that organizes unstructured, long paragraphs of plain texts based on the content of the texts. This system will analyze and choose certain formats for different sections of the input, and it will generate a web page containing the formatted and analyzed note.

In order to analyze texts and choose appropriate formats, this system needs to understand the contents and structures of inputs. We detect different topics covered by the raw text by using topic modeling. Then we separate the raw input into different topic segments based on the topic information. Once the note is separated into appropriate topic segments, we use automatic summarization techniques to reduce verbosity and extract main ideas within each topic segments. Last but not least, we generate a web page to display the analyzed and organized note.

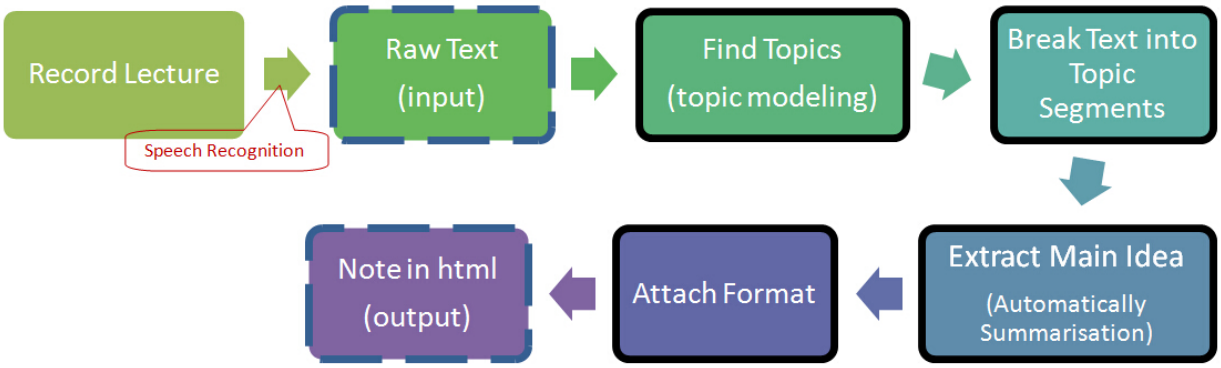


Figure 1: Procedure and Design of the Automatic Note Formatting System

[1] All data are obtained from Coursera.org, "an education platform that partners with top universities and organizations worldwide, to offer courses online for anyone to take, for free." Coursera (2014)

3 Background and Related Work

Many researches have investigated different procedures and approaches of digital note taking. In the 1990s, systems like NoteLook that take video digital files as input have been developed to free note takers from the worry of not recording fast enough.Sarah Reitmeief Patrick Chiu and Wilcox (1999) However, such note are highly likely to be unorganized, and in order to organize such notes efficiently, users might need to extract information from videos. Systems like NoteLook demonstrate the need for a automatic note formatting systems.

There are also investigations on the applicability of property-based document organization. In 2000, Jonathan Huang and Joseph Michiels built a prototype on a "collaborative note sharing system based on document properties rather than implementing traditional hierarchical structures." Huang and Michiels (2000) Using this prototype, users can access and edit documents based on the property of documents. Huang and Michiels bring up the interesting question of "what are the property of a document". This study hints that users might benefit from knowing overviews and properties of documents, and these information might help users to understand these documents better. Out system aims to automatically generate concise notes that assist users' learning, and we hope that the information provided by our system will improve people's note taking experiences.

One of the prototypes of automatic formatting is Microsoft Office Word, in which the software detects users' intention of creating lists, drawing dividing lines and so forth.Microsoft (2014) By providing quick formatting functions and templates, Word allow users to change the layout of the text quickly. Nevertheless, it does not format a paragraph of note into different levels unless the users manually divide it. Also, the formatting process happens simultaneously with typing, and if a whole paragraph of texts is imported to Word, the text stays unstructured. Our system aims to detect users' attention and automatically formatting texts just like Word does, yet we want to detect attention through the content of the texts rather than the action that users perform.

In 2009, Jie Liu and Michelle X. Zhou has developed an interactive smart notepad system that let user to enter a brief note to drive a dynamic information-seeking process based on CENTAUR, a system that generates queries based on existing note context, dictionary and query templates.Liu and Zhou (2009) This system is built based on a text mining system that extracts information, and it demonstrates the possibility of processing text bases on the content and meaning. Similar to this system, we are using natural language processing techniques to detect structures within texts.

4 Find Topics

As shown in Figure 1, the first step of the note formatting process is to find topics that are covered by the lecture. We use topic modeling, a statistical model that "uncover[s] the hidden thematic structure in document collections", to detect topics within the raw text.Blei (2012) A topic model generated based on a document can be used to discover the abstract topics within this document. Each abstract topic is not a single sentence or word, but a cloud of words that collectively define the topic.

MALLET is a "Java-based package for statistical natural language processing, document classification, clustering, topic modeling, information extraction, and other machine learning applications to text."McCallum (2002) We are using Mallet to generate topic models on raw texts. Given the number of topics that are needed, Mallet will generate a list of topics and associated topic words. As shown in Figure 2, the number on the left indicates the topic number, followed by topic words that define this topic.[2] In this case, we have generated five topics and would use this topic information to break raw text into topic segments.

A single word might appear in multiple topics. For example, the word "sound" appears in topic 0 and topic 1. Besides, because of the method and approach that Mallet uses, words such as "ll" and "ve" can possibly be considered as topic words, while "will" and "have" are the real topic words. In the situation where a lot of abbreviations are included, this might introduce noises to the system.

- 0 audio sound maxim ve player play sketch file ll code folder data environment create beat files store hear set
- 1 sound ve ll speed make time bit week good sounds yeah people graphics things stuff thing lot detail type
- 2 width line brush mouse simple brushes music divided map mapping ve idea talk mirrors symmetry distance refresh tells interesting
- 3. android processing device ll run ve running java simple devices javascript ios app desktop write screen ip network address
- 4 color draw position ve mouse screen program drawing colors red green line code rectangle numbers point blue basic background

Figure 2: 5 topics that are detected by Mallet on a sample input.

[2]All example used here after are generated from records of Creative Programming for Digital Media and Mobile Apps taught by Dr Marco Gillies, Dr Matthew Yee-King, Dr Mick Grierson.Dr Marco Gillies (2013)

5 Break Text into Topic Segments

We want to organize the raw text by breaking it into topic segments. Ideally, each segment will concentrate on a single topic, and therefore when the topic shifts from one topic to another, we should generate a topic break. We can detect topic shifts by tagging words with their topic number and calculating topic frequency within a certain length of text.

5.1 Tagging Topic Number

We start by tagging each word within the raw text with the topics that it belongs to. In the example shown in Figure 3 and 2, the word "sound" belongs to topic 0 and 1, and therefore we tag it with 0 and 1. The word "Android" will be tagged with 3. For words that do not belong to any topic, such as the word "basic", we tag it with -1.

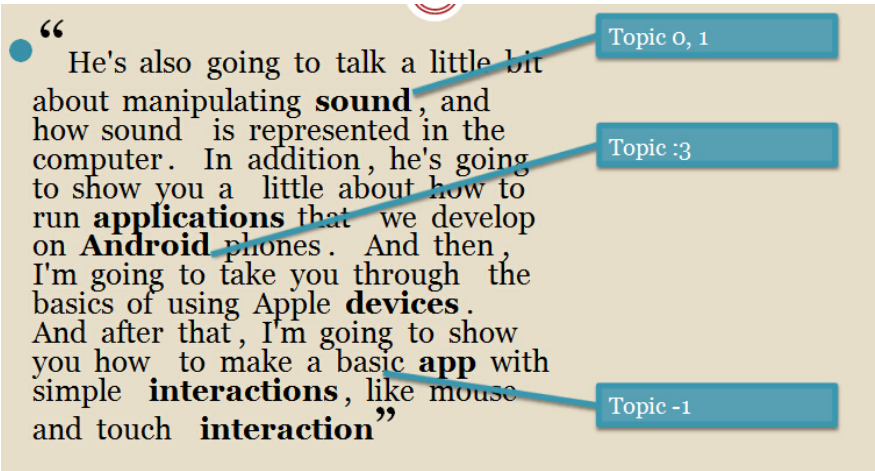


Figure 3: Tag raw text with the topic number

After tagging every word in the raw text, we have an understanding about the distribution of the topics throughout the entire lecture. To view it visually, we can graph the relationship between word index and topic number tags. As shown in Figure 4, it is clear that the first 500 words concentrates on topic 1 and the text indexing from 500 to 1000 covers topic 3 mainly. Intuitively we would put the first 500 words in a different topic segments with the second 500 words. The next section describes how we determine topic shifts automatically.

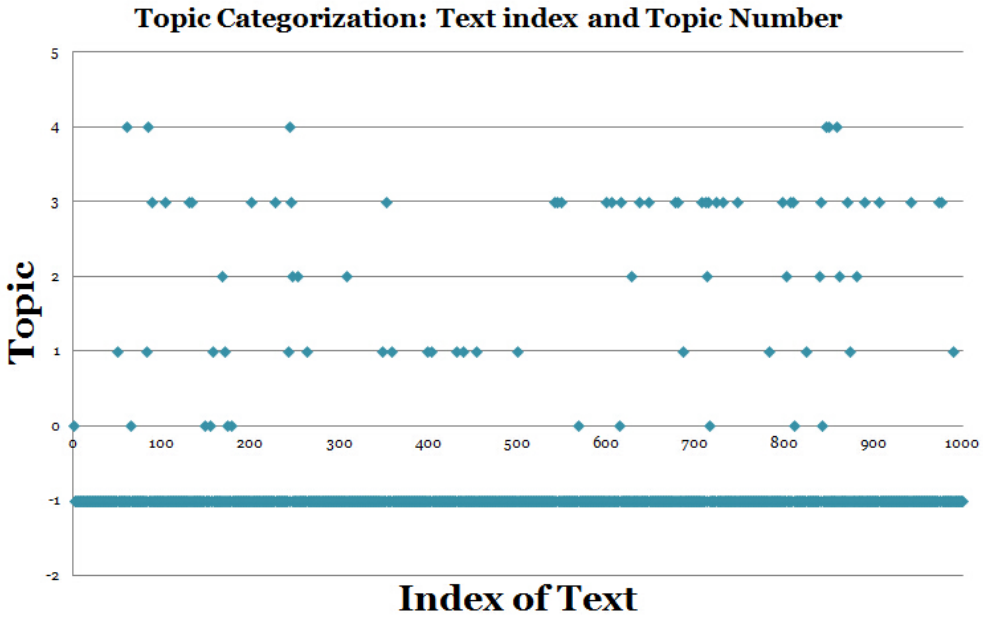


Figure 4: Distribution of topic

5.2 Calculate frequency and determine topic shift

In this step, we create a "window" that contains a certain number of texts from the input. Within the window, we rank each topic according to how often the words from that topic occur in the window. The topic with the highest frequency will be marked as the main topic within that window.

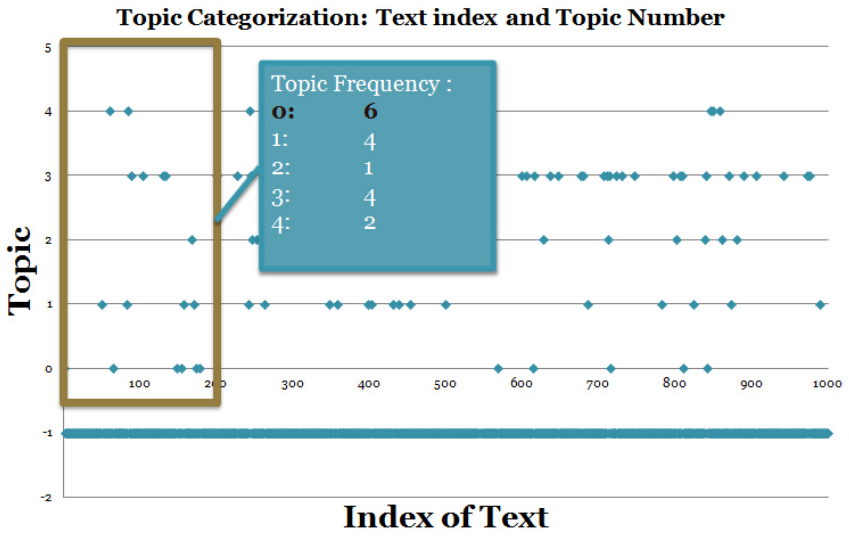
We continue the calculation and marking main topic by moving the window one word closer to the end of the input and repeat this frequency calculation. In the situation where only one main topic for the entire text exists, the top ranked topic should remain the same for all windows. In the situation where topic shifts, the top ranked topic would change and we can use the location of the window as a break point. Figure 5a and 5b demonstrate an example of this calculation.

5.3 Optimization and Evaluation

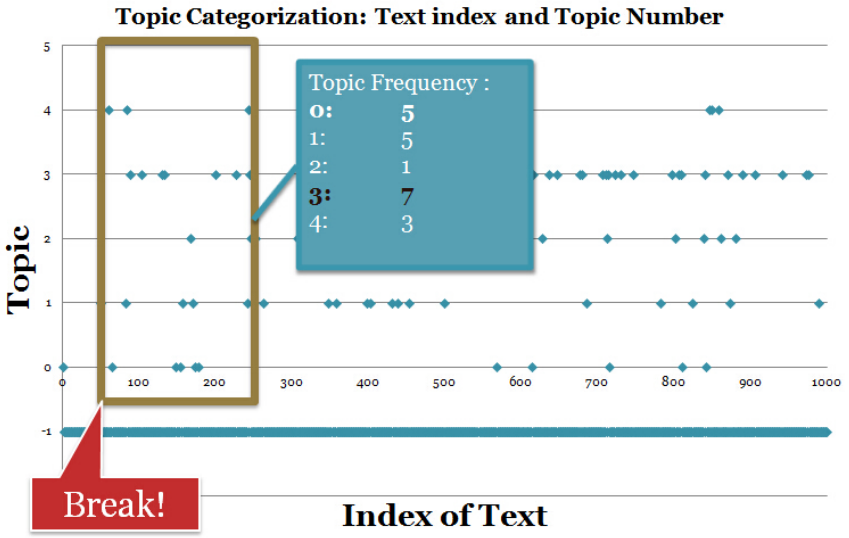
There are three major optimizations that we have implemented in our system.

1. Add major topic ignoring mechanism to detect subtopic shift.

   In the situation where a single topic dominants the whole text, this algorithm would not detect any topic shift and therefore fail to break the text into topic segments. This situation happens often because ideally a lecture should be about a single topic. But in order to introduce and explain that



(a) Calculating topic frequency



(b) Breaks are generated when topic shifts

Figure 5: Calculating frequency and detect topic shifts

topic, lecturers' might need to give different definitions, examples and backgrounds, and the note taking system will need to detect the break between these subtopics. We added an "ignore topic" mechanism that targets to solve this problem by removing the single main topic that dominant the whole lecture and calculating the frequency again.

2. Add latency mechanism and force length mechanism to remove topic segments that are too short.

As breaks generated by this algorithm are extremely sensitive to individual word, in the situation where multiple topics appears near to each other, this system is going to suggest many breaks, or jump back and forth between topics. Topic changing within short amount of text might introduce noise to the break points. For instance, the main topic is topic 0 in window 10, and in window 11 the main topic becomes topic 1, ideally we would have a topic break at window 11. However if window 12 has the main topic of 0 again, the topic break is considered as noise because the topic has not really shifted.

In order to reduce such topic noise, we introduced latency into our system. Latency is the number of window that the topic shift has to last in order to make a break point. Any topic shift that last shorter than the latency would be ignored. For instance, when we set latency to be 5, in the situation where window 10's main topic is 0, and window 11 has main topic of 1. We are not creating the break point until we confirm that window 12, 13, 14, 15 also has topic 1 as main topic.

Even with the latency mechanism, the system generates break points that are too close sometimes, so we introduced a clean up mechanism to force each text segment to be longer than a certain length by removing breakpoints that are too close together.

3. Move breaks to the end of sentences to maintain complete sentence structure.

   The window calculations generate a set of break points that represent the topic shift in the entire input text. Figure 6 is a visual representation of the result that we generated with our system. Because we generate breaks based on word instead of sentence, we need to move these breaks to the closest sentences end in order to maintain complete sentences.

The subtitles that we obtained from Coursera are separated into different topics originally. We record these breaks and merge these subtitles into one raw text as our input. We consider these breaks as real breaks and briefly evaluated our system by calculating the word distance between analyzed break and the real break. The fact that the evaluation result is highly sensitive to the data input should be taken into account. Our system might behave better on courses taught with certain lecture styles than others. The evaluation result also varies significantly when the system setting including number of topic to detect, window size, latency and minimal topic segment length changes. Running on a single course containing nine lectures, when the topic number is 5, window size is 1 percent of the whole text length, latency is 0.8 percent of the text length, and minimal topic segment length is 50 words, the generated breaks are 121 words away from the real breaks on average.

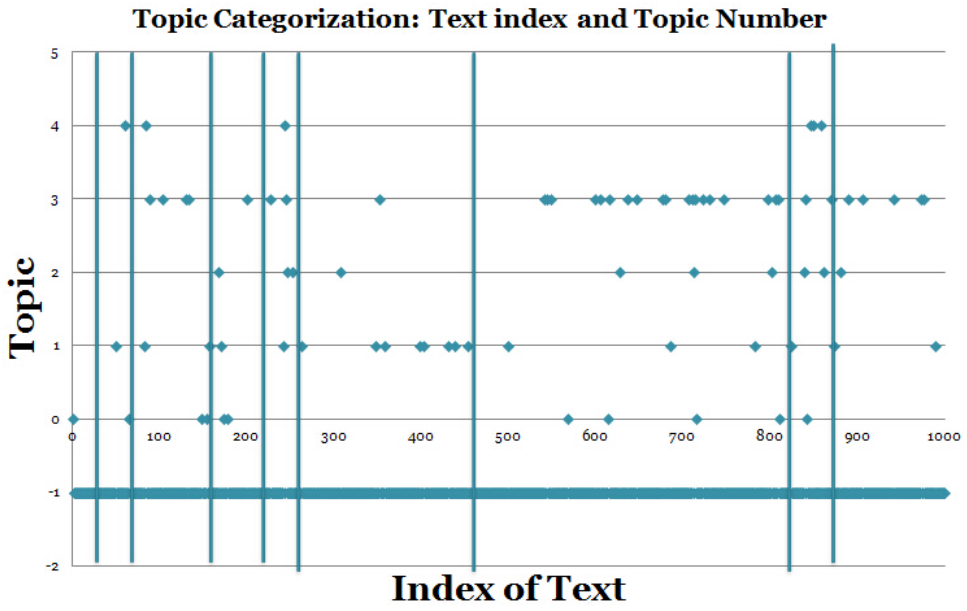


Figure 6: Breaks detected by our algorithm

6 Extract Main Idea and Formatting

After separating the raw texts into topic segments, we are using extractive summarization package to extract main idea of each topic segments. Extractive Summarization "involves determining the most important sentences in a document and putting them together to make a summary."Caleb Butterfiel and Thede (2007) In other words, extractive summarization algorithms assess the importance of each sentence. Given the number of sentences needed for the summarization and a single topic segment, the java based package SummarizeLib will return the top ranked sentences out of this topic segment. YauhenMinsk (2013)

We are using SummarizeLib to generate two summarizations. First, we select one percent of the text from the raw input as the overview of the raw text. Second, we select one to five sentences out of each topic segment as the main idea of the individual topic segment.

Our system stores all the information in a single html file. We choose to html format in order to add the fold/expand function, and thereby users can choose to read main sentences and to expand the entire topic segment when needed. Figure 7 is an example demonstrating the output of our system. The directory of the raw text is placed on the top of the web page. Users can choose to insert other information, such as the title of the course under the raw text directory. Following the title, the overview of the document is enclosed in square brackets. Users can use the overview to gain a basic understanding of the lecture and then investigate

in different topic segments.Start with a star sign, the main sentences of topic segments normally consist one to five sentences. By default, only the main sentences of each topic segment are displayed when the system generates the webpage. Users can use these main sentences to navigate through different topics. If users need more information about a particular topic, they can click on the main sentence and expand the entire topic segment for further reading.

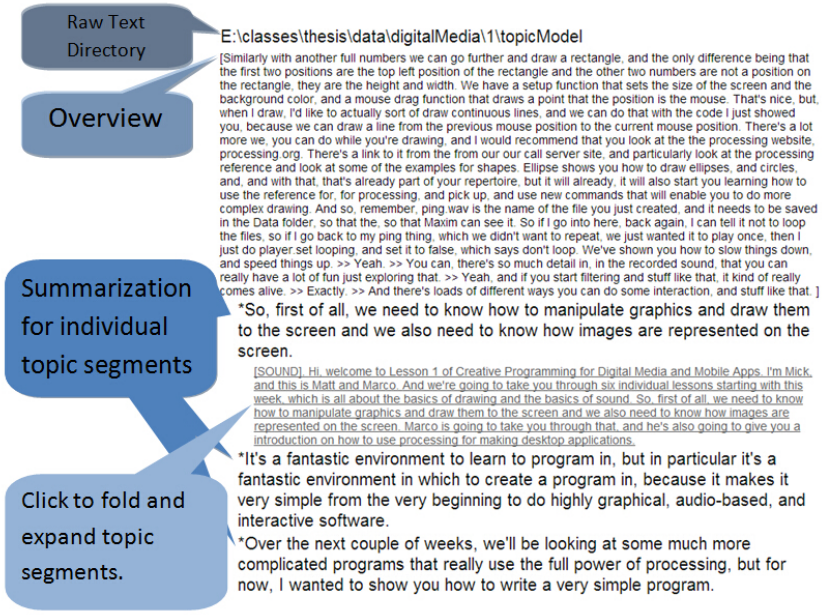


Figure 7: A sample webpage generate by our system.

7 Conclusion and Future Work

We design, build and explore a system that can analyze and structure the unstructured, long paragraphs of lecture records based on the content of the lecture. By detecting topics within lecture records, our system breaks the lecture record into topic segments. As each topic segment centers around a single topic, our system eliminates irrelevant information by using extractive summarization system to generate summarization for individual topic segment. We put summarizations and topic segments into an interactive webpage so that users can choose to expand or fold information. This system can potentially free students from the burden of taking notes during lectures.

We notice that results produced by our system are highly sensitive to lecture contents and styles. Therefore our system performs better on lectures with certain teaching styles. We are interested in finding out

the what makes a particular teaching style more suitable for our system than other styles. In addition, we hope to compare our system's performance on different type of courses.

As system setting varies, we can have completely different results on the same input. Therefore it is crucial to conduct more testings and evaluations in the future in order to develop a more intelligent system that adjusts its setting according to the lecture style or users' preferences.

We also notice that extractive summarization system might not be the most suitable summarization algorithm to use for this task. The sentences extracted by our current extractive summarization strategy might not represent the essence of the topic segment completely. In the future we would love to explore different summarization techniques and packages, especially abstractive summarization systems that select phrases instead of sentences as output.

We believe that our system has provided users tools to explore this new note-taking strategy. We are looking forward to developing this system further and building a better user interface. By making our system available to more users, we are eager to explore the advantage, disadvantage, efficiency and applicability of this note-taking strategy.

References

Blei, D. M. (2012). Probabilistic topic models. https://www.cs.princeton.edu/~blei/topicmodeling.html and https://www.cs.princeton.edu/~blei/papers/Blei2012.pdf.

Caleb Butterfiel, R. M. and S. Thede (2007). Sentence selection for extractive summaries in pare. http://cs.brown.edu/people/rebecca/pare07.pdf.

Coursera (2014). coursera.org. https://www.coursera.org/about/.

Dr Marco Gillies, Dr Matthew Yee-King, D. M. G. (2013). Creative programming for digital media and mobile apps. https://class.coursera.org/digitalmedia-001.

Huang, J. and J. Michiels (2000, april). Exploring property-based document organization in a collaborate note-sharing system. *Technical Report CHI*.

Liu, J. and M. X. Zhou (2009, February). An interactive , smart notepad for context-sensitive information seeking. *Technical Report IUI09*.

McCallum, A. K. (2002). *MALLET: A Machine Learning for Language Toolkit*. UMass Amherst. http://mallet.cs.umass.edu.

Microsoft (2014). Use automatic formatting as you type. http://office.microsoft.com/en-us/excel-help/use-automatic-formatting-as-you-type-HP010082297.aspx.

Paul, A. M. (2014). Pen-and-paper notetaking superior to typing on a laptop. http://anniemurphypaul.com/2014/01/pen-and-paper-note-taking-superior-to-typing-on-a-laptop/.

Sarah Reitmeief Patrick Chiu, A. K. and L. Wilcox (1999). Notelook: Taking notes in meetings with digital video and ink. *Technical Report ACM Multimedia*.

Williams, R. L. and A. C. Eggert (2002). Notetaking in college classes: Student patterns and instructional strategies. *The Journal of General Education Vol. 51 No. 3*. http://www.jstor.org/stable/27797918.

YauhenMinsk (2013). Summarizelib. Summarization (en) Java implementation Lightweight and without dependencies https://github.com/YauhenMinsk/SummaryLib.