Final Presentation Fall 2023 Textbook Summarization - ChatGPT

Rachit, Veena, Divya, Pearl

Introduction

- This project was created to help in the digitizing of and Electrical and Computer Engineering Textbook
- Our main goal was to build on the existing project from last semester by creating dynamic summaries of chapters using AI
- Our project stores chapters in a database, generates keywords for each chapter, and then generates summaries of each chapter incorporating the keywords.
- The aim of this implementation is to allow the user to easily navigate through related chapters and be able to see summarized versions of them

Existing Project from Previous Semester

- Static summaries generated for each sub-section of textbook

- Our aim was to improve on the current summaries so they are more useful for the students

Motivation/Goals for Project

Summaries for sub-sections of textbook was based off of existing text from textbook

- Need for more clarity/explanation in summaries

Incorporating and leveraging AI for educational purposes

- Create more dynamic, accurate summaries

Linking similar chapters together for the front end team

- Generate keywords to be used in each summary

Overview

Pipeline

- Store ChatGPT article into database
- Generate keywords for each text selection and store into database
- Make call to ChatGPT API and feed text selection and keywords to generate a summary using keywords

Organization

- Split work into 4 categories and connected all of our individual pieces

Workflow

- Weekly sub-team meetings to talk about progress and next steps

Technologies Used: NLTK Library, OpenAI API, SQL, Python

Keyword Extraction Results - Veena

Approach

- Used NLTK library
  - Tokenization
    - Tokenize input into sentences and words
- Pre-trained model: KeyPhrase Transformer - T5 model
  - Each task is converted to text-to-text format

Organization

- Several preprocessing functions
  - process(), filter(), extract(), get_keywords(), generate_keywords()
- One main function that calls on preprocessing functions and outputs 2 keywords for every 100 words

Results

```
Text: OK, so ChatGPT always picks its next word based on probabilities. But where do those probabilities come from? Let’s start with a simpler problem. Let’s consider generating English text one letter (rather than word) at a time. How can we work out what the probability for each letter should be? A very minimal thing we could do is just take a sample of English text, and calculate how often different letters occur in it. So, for example, this counts letters in the Wikipedia article on “cats”

Keywords extracted: wikipedia,probability theory,letters
```

ChatGPT Article in SQL Results

Create a table in the existing database called “keywords

Keywords stores:

- ID
- Chapter_Name
- Chapter_Text

The text was fed as an input to the keywords extraction model and stored in a table called “Keyword_Store”

Keyword_Store table consists of :

- ID
- Keywords

chapter_name	chapter_text	id
Where Do the Probabilities Come From?	OK, so ChatGPT always picks its next word bas…	2
What Is a Model?	Say you want to know (as Galileo did back in th…	3
Models for Human-Like Tasks	The example we gave above involves making a…	4
Neural Nets	OK, so how do our typical models for tasks like i…	5
The Practice and Lore of Neural Net Training	Particularly over the past decade, there've been…	6
"Surely a Network That's Big Enough Can Do A…	The capabilities of something like ChatGPT see…	7
The Concept of Embeddings	Neural nets—at least as they're currently set up…	8
Inside ChatGPT	OK, so we're finally ready to discuss what's insi…	9

chapter_id	keywords
6	physics ,simple mathematics ,n…

```
5 •  create table keyword_store (chapter_id int, keywords varchar(120), FOREIGN KEY (chapter_id) REFERENCES keywords(id));
```

Keywords Table and Keyword Store Table

ChatGPT API call Results - Divya

- Approach
  - Splits the text to summarize into small chunks
  - Sends in each chunk to the API to create a summary
  - Combine all the individual summaries to form one whole summary
- Organization
  - Two methods to achieve this: split_text() and generate_summary()
- Results
  - A summary of 1-2 sentences for every 2048 characters is generated for the text input from the database

```
Text: Say you want to know (as Galileo did back in the late 1500s) how long it's going to take a cannon ball dropped from each floor of the Tower of Pisa to hit the ground. Well, you could just measure it in ea
ch case and make a table of the results. Or you could do what is the essence of theoretical science: make a model that gives some kind of procedure for computing the answer rather than just measuring and rememb
ering each case.It is worth understanding that there's never a "model-less model". Any model you use has some particular underlying structure—then a certain set of "knobs you can turn" (i.e. parameters you can
set) to fit your data. And in the case of ChatGPT, lots of such "knobs" are used—actually, 175 billion of them. But the remarkable thing is that the underlying structure of ChatGPT—with "just" that many paramet
ers—is sufficient to make a model that computes next-word probabilities "well enough" to give us reasonable essay-length pieces of text.

Keywords extracted: cannon ball,tower of pisa,measurement,data analysis,theoretical science,underlying structure,data model,parameters,chatgpt,lots of such "knobs",next-word probabilities

Summary using the keywords: In the late 1500s, Galileo wanted to determine the time it takes for a cannonball dropped from each floor of the Tower of Pisa to hit the ground. Instead of measuring each case, he c
reated a theoretical model that provided a procedure for computing the answer. Similarly, in the case of ChatGPT, a language model with 175 billion parameters, the underlying structure and numerous adjustable p
arameters allow it to generate text with reasonable next-word probabilities.

--------------------------------------------------
```

Connecting it all together using SQL - Rachit

MySQL Connector (Keywords side)

- Fetching textbook content from MySQL database
- Running generate_keywords() function on that text
- Updating the new keywords into the database

MySQL Connector (Summary side)

- Fetching the generated keywords from the database
- Feeding those keywords into the summarization algorithm

```
python3 app.py

Text: OK, so ChatGPT always picks its next word based on probabilities. But where do those probabilities come from? Let's start with a simpler problem. Let's consider generating English text one letter (rather
than word) at a time. How can we work out what the probability for each letter should be? A very minimal thing we could do is just take a sample of English text, and calculate how often different letters occur
in it. So, for example, this counts letters in the Wikipedia article on "cats"

Keywords extracted: wikipedia,probability theory,letters

Summary using the keywords: The probabilities for generating English text one letter at a time can be determined by analyzing the frequency of occurrence of different letters in a sample of English text, such a
s the Wikipedia article on "cats". This approach utilizes probability theory to calculate the likelihood of each letter appearing in the generated text.

--------------------

Text: Say you want to know (as Galileo did back in the late 1500s) how long it's going to take a cannon ball dropped from each floor of the Tower of Pisa to hit the ground. Well, you could just measure it in ea
ch case and make a table of the results. Or you could do what is the essence of theoretical science: make a model that gives some kind of procedure for computing the answer rather than just measuring and rememb
ering each case.It is worth understanding that there's never a "model-less model". Any model you use has some particular underlying structure—then a certain set of "knobs you can turn" (i.e. parameters you can
set) to fit your data. And in the case of ChatGPT, lots of such "knobs" are used—actually, 175 billion of them. But the remarkable thing is that the underlying structure of ChatGPT—with "just" that many paramet
ers—is sufficient to make a model that computes next-word probabilities "well enough" to give us reasonable essay-length pieces of text.

Keywords extracted: cannon ball,tower of pisa,measurement,data analysis,theoretical science,underlying structure,data model,parameters,chatgpt,lots of such "knobs",next-word probabilities

Summary using the keywords: In the late 1500s, Galileo wanted to determine the time it takes for a cannonball dropped from each floor of the Tower of Pisa to hit the ground. Instead of measuring each case, he c
reated a theoretical model that provided a procedure for computing the answer. Similarly, in the case of ChatGPT, a language model with 175 billion parameters, the underlying structure and numerous adjustable p
arameters allow it to generate text with reasonable next-word probabilities.

--------------------

Text: The example we gave above involves making a model for numerical data that essentially comes from simple physics—where we've known for several centuries that "simple mathematics applies". But for ChatGPT w
e have to make a model of human-language text of the kind produced by a human brain. And for something like that we don't (at least yet) have anything like "simple mathematics". So what might a model of it be l
ike?When we made a model for our numerical data above, we were able to take a numerical value x that we were given, and just compute a + b x for particular a and b. So if we treat the gray-level value of each p
ixel here as some variable xi is there some function of all those variables that—when evaluated—tells us what digit the image is of? It turns out that it's possible to construct such a function. Not surprisingl
y, it's not particularly simple, though. And a typical example might involve perhaps half a million mathematical operations.

Keywords extracted: physics,simple mathematics,numerical data,chatgpt,human-language text,arithmetic,model,gray-level image,function,mathematical operations

Summary using the keywords: To create a model for human-language text, we don't have a simple mathematical approach like in physics. Instead, we need to develop a function that can process and understand the te
xt, which involves complex algorithms and computations. For example, to determine the digit represented by a gray-level image, a function can be constructed, but it would typically require around half a million
 mathematical operations.

--------------------
```

Keyword and summary generation + MySQL database

Result Grid Filter Rows: Search Edit: Export/Import:

keywords_extracted	chapter_name	chapter_content	id
wikipedia,probability theory,letters	Where Do the Probabilities Come From?	OK, so ChatGPT always picks its next word based on probabilities. But where…	2
cannon ball,tower of pisa,measurement,…	What Is a Model?	Say you want to know (as Galileo did back in the late 1500s) how long it's goi…	3
physics,simple mathematics,numerical d…	Models for Human-Like Tasks	The example we gave above involves making a model for numerical data that…	4
NULL	NULL	NULL	NULL

Future Work

Include weights for keywords

- To indicate relevance/importance
- Use as sorting mechanism to find the top-most keywords of all keywords generated

Implement our work to the ECE textbook

- To create more effective summaries for sub-sections of textbook