Summarization Evaluation

CS 4624
Multimedia, Hypertext, and Information Access
Virginia Tech, Blacksburg VA 24061

Final Report
May 8th, 2023

Professor: Dr. Edward A. Fox
Client: Bipasha Banerjee

Prepared By: Harshil Goel, Varun Choudhary, Anish Dhondi, Parth Desai
# Table of Contents

Figures and Tables ........................................................................................................... 3

1 Abstract .......................................................................................................................... 5  
   1.1 Introduction ............................................................................................................... 5  
   1.2 Intended Users ......................................................................................................... 5

2 Client Requirements ...................................................................................................... 6  
   2.1 Client ...................................................................................................................... 6  
   2.2 Client Requirements ............................................................................................... 6  
   2.3 Associated Capstone Teams .................................................................................... 6

3 Project Research and Tech Stack .................................................................................. 8  
   3.1 Project Research ..................................................................................................... 8  
   3.2 Tech Stack .............................................................................................................. 8

4 Team Organization ........................................................................................................ 10  
   4.1 Team Member Responsibilities .............................................................................. 10  
   4.2 Initially Setup Milestones ...................................................................................... 11  
   4.3 Team Meetings & Team Communication ............................................................... 11

5 Design ........................................................................................................................... 12  
   5.1 Wireframes ............................................................................................................ 12  
   5.2 Methodology ......................................................................................................... 14

6 Deliverables ................................................................................................................... 17  
   Previous Website Walkthrough .................................................................................. 17  
   6.1 Design Reconsiderations ....................................................................................... 18  
   6.2 Walkthrough of New Website ................................................................................ 18

7 Challenges ..................................................................................................................... 22  
   7.1 Backend Data Extraction ...................................................................................... 22  
   7.2 Importing Database ............................................................................................... 22  
   7.3 Front End Handoff ................................................................................................. 22  
   7.4 Insufficient Ground Truth Summaries ................................................................. 22

8 Manual ........................................................................................................................... 23  
   8.1 User Manual ......................................................................................................... 23  
   8.2 Developer’s Manual ............................................................................................... 24  
      8.2.1 Schema Dump ................................................................................................. 24  
      8.2.2 Set Up and Run The Code ............................................................................. 25  
      8.2.3 Server.js considerations ............................................................................... 25  
      8.2.4 Starting The Website ..................................................................................... 26  
      8.2.5 Database Population ...................................................................................... 26

9 Discussion and Future Efforts ....................................................................................... 28  
   9.1 Discussion ............................................................................................................. 28  
   9.1 User Tests ............................................................................................................. 28
# Figures and Tables

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Database tables and their associated fields</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Milestones that were set during initial setup</td>
<td>12</td>
</tr>
<tr>
<td>Figure 2</td>
<td>A wireframe depicting the upload screen</td>
<td>13</td>
</tr>
<tr>
<td>Figure 3</td>
<td>A wireframe depicting the screen that shows the chapter summaries are being generated for</td>
<td>13</td>
</tr>
<tr>
<td>Figure 4</td>
<td>A wireframe depicting what the view for a summary will be</td>
<td>14</td>
</tr>
<tr>
<td>Figure 5</td>
<td>A wireframe depicting how the ranking system should be shown to the user</td>
<td>14</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Sub-goal 1</td>
<td>15</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Sub-goal 2</td>
<td>16</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Workflow diagram breaking down project goal into subtasks</td>
<td>17</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Previous team’s website setup</td>
<td>18</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Initial screen upon first opening the website</td>
<td>19</td>
</tr>
<tr>
<td>Figure 11</td>
<td>ETD Metadata Tab once an ETD is selected</td>
<td>20</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Chapter Text Tab</td>
<td>20</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Summary 1 Tab</td>
<td>21</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Summary 2 Tab</td>
<td>21</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Rate Tab That Allows Users To Evaluate Summaries</td>
<td>22</td>
</tr>
<tr>
<td>Table 2</td>
<td>A table showing what error message each violation in the rating tab leads to</td>
<td>22</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Website home page</td>
<td>24</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Menu bar where the user can navigate to Summary View</td>
<td>24</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Summary View Rate Page</td>
<td>25</td>
</tr>
<tr>
<td>Figure 19</td>
<td>schema must be imported to display data</td>
<td>26</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Files to be adjusted for new port</td>
<td>26</td>
</tr>
<tr>
<td>Figure 21</td>
<td>Line of code to change if setting up server on a different port</td>
<td>27</td>
</tr>
</tbody>
</table>
1 Abstract

1.1 Introduction

Electronic Theses and Dissertations (ETDs) are digital versions of academic papers of graduate students. ETDs are highly complicated and lengthy texts: these include multimedia elements and other forms of information. A digital library with summaries for each ETD chapter would enable more people to explore all these texts to learn about various domains. However, most chapters of ETDs don’t have summaries, so the solution was to create AI generated summaries for users across disciplines to read.

Before these summaries are accessible to the public, Bipasha Banerjee will find researchers in various disciplines to evaluate the summaries. Incorporating human feedback into AI-generated summaries results in improved accuracy, relevance, originality, engagement, and satisfaction. Quantitative measures for evaluating AI generated content are great, but qualitative feedback is important too. Subject matter experts can detect errors and inconsistencies in these summaries: this feedback provides guidance.

Our team has developed a website that enables users to view and rank AI-generated summaries of texts against the ground truth (provided) chapter summaries. The users will not know beforehand which is which. The scholars (those with an education level of entering graduate school and beyond) should be able to accurately evaluate the summaries for the ETDs in their field. The purpose of this project is to allow human evaluation of these texts. The ranking feature serves as a form of feedback to perfect the AI generated summaries. The domain experts will use this website to determine the model that serves as most promising for summarization.

1.2 Intended Users

For now, the intended users for this application are subject matter experts at Virginia Tech, so that they can evaluate the summaries. The evaluation will provide an idea of which model performs best for AI generated summaries. Eventually, the final model will be used to serve users outside of Virginia Tech who want to know more about the domain that the ETD is associated with. Providing these summaries allows those outside the domain to grasp the basic concept of the ETD and its associated chapters without having to read the entire paper.
2 Client Requirements

2.1 Client

Bipasha Banerjee is a Ph.D. student and Graduate Research Assistant at Virginia Tech. She completed her master’s at this very school with an impressive 3.84 grade point average, before beginning her education towards her doctorate in 2019. Her focus is on Natural Language Processing and Information Retrieval, specifically from book-length documents. She is working closely with a number of undergraduate student groups in a Multimedia, Hypertext, and Information Access Capstone Class.

2.2 Client Requirements

There are four primary deliverables for this project, that will serve as the guidelines and checkpoints for success as we develop the project:

1. Web User Interface
   ○ An easy to use UI without needing any prior documentation for it.

2. Data Set
   ○ The results from user interaction with the web page should populate a data set that contains the name and content of the ranked summaries, as well as their associated chapters.

3. Ranking Summaries
   ○ A method for ranking summaries must be displayed that will provide the best insight for representing the most fluent, useful, efficient summaries provided.

4. Feedback Stored
   ○ The completed project should provide a system that can store feedback from users either about the rankings or about the WebUI in general, that will be useful for future iterations of the project.

2.3 Associated Capstone Teams

Team Chapter Summarization aims to help readers parse documents faster. Students are creating a gold standard summarization dataset by annotating the chapters boundaries from 100 ETDs. They are then creating a summarization framework that accepts chapter text as input and generates text summaries.

Team Classifying ETDs is working with an existing dataset as well as a gold standard classification dataset. The document metadata contains discipline or department information but is often incomplete or incorrect, making it difficult to categorize documents without manually reading through. They are working with various machine learning and deep learning algorithms
to perform classification for documents with missing metadata, assigning multiple categories to each chapter.

Team Recommendation System aims to recommend similar ETD chapters, likely to be of interest to the user. This semester, the team focus is on identifying and configuring clustering methods to assist. A future plan is to analyze user click history on abstracts, metadata, figures, tables, captions, chapters, sections, etc. Such can serve as a form of feedback to the recommendation system.
3 Project Research and Tech Stack

3.1 Project Research

We used a YouTube video, titled ReactJS Frontend Web Development [4], to learn the basics of the React framework, before starting this project. We also used a webpage on geeksforgeeks [2] that was a simple walkthrough of a React application, and how to run it, along with a webpage on code.visualstudio [7] that taught us how to do all of this in our preferred IDE of Visual Studio Code. Put together we used these links to kickstart our work in this new environment. We also used ChatGPT for assistance in the early stages because it helped us figure out missing dependencies we needed to install and to give us a crash course on the new technologies.

Having never used React before, we relied heavily on the React documentation provided on the developer website [6]. Reading through the main features and studying the old code base as a template/starting point helped us launch into development. The PostgreSQL [1] and pgAdmin [5] documentation was also helpful to learn basic SQL commands, procedures, and overall schema and table structure.

3.2 Tech Stack

The finalized tech stack that we landed on was a React based frontend that connects to a PostgreSQL database via a server with various API endpoints that act as a bridge between the two components. The PostgreSQL database was managed using PGAdmin as the primary GUI. Of the 4 tables (chapter, ETD, sum_eval, summary), all except sum_eval were manually populated with the associated data. The chapter table’s primary purpose was to hold chapter text, while the ETD table holds metadata about each ETD (title, author, university, abstract text, etc.). The summary table holds ground truth summaries as well as the AI generated summaries for each chapter in each ETD. The sum_eval table is populated each time a user interacts with the website and provides ranks and ratings for each summary.

Table 1 shows what fields or columns are held by each table in the database:

<table>
<thead>
<tr>
<th>Table</th>
<th>Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>chapter</td>
<td>id, etd_id, chapter_text, chapter_no</td>
</tr>
<tr>
<td>etd</td>
<td>id, title, author, advisor, year, abstract, university,</td>
</tr>
<tr>
<td></td>
<td>degree, uri, department, discipline, language, schooltype, load classifier</td>
</tr>
<tr>
<td>sum_eval</td>
<td>id, etd_id, chapter_no, summary_id, rank, fluency,</td>
</tr>
<tr>
<td>Summary</td>
<td>Id, etd_id, chapter_no, summarization_text, algorithm_used, local_path</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>content_coverage, consistency, overall_rating, user_comments</td>
<td></td>
</tr>
</tbody>
</table>

*Table 1: Database tables and their associated fields*
4 Team Organization

4.1 Team Member Responsibilities

Harshil Goel
Team Lead
Backend Development

Varun Choudhary
Research & Development

Anish Dhondhi
Frontend Development

Parth Desai
Frontend Development
4.2 Initial Milestones

<table>
<thead>
<tr>
<th>Feb 14th</th>
<th>Feb 28th</th>
<th>Mar 14th</th>
<th>Mar 31st</th>
<th>Apr 7th</th>
<th>Apr 13th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Literature Review</td>
<td>Connect backend to frontend</td>
<td>Complete user tests</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Finalize Tech Stacks</td>
<td>End-to-End system ready</td>
<td>Iterative improvements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Background Reading</td>
<td>Ready for User Tests</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UI Wireframes</td>
<td>Data Cleaning</td>
<td>Testing phase</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Barebones UI</td>
<td>Quality assurance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implementation (using</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>previous work)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 1: Milestones that were set initially*

4.3 Team Meetings & Team Communication

The team worked closely with our client to come up with realistic, biweekly milestones for the semester. The team met with Bipasha Banerjee weekly on Thursdays at 12:30pm for thirty minutes. The team has also been meeting on weekly Wednesdays at 1:10pm for an hour or two. The team has been in communication with our client over Slack, and has been in communication internally over I-Message.
5 Design

5.1 Wireframes

Our team worked on initially creating wireframes for the Web UI. These wireframes helped us determine how the UI should look, and to come up with design ideas for the website. The end goal was to create working renditions of these sketches.

Figure 2: A wireframe depicting the upload screen

Figure 2 reveals the upload text page, which is intended to let the user upload a text they need a summary for. This functionality made sense at the time; however as we progressed, it seemed unnecessary for the current scope, and we had to take it out.

Figure 3: A wireframe that shows the chapter text, along with summaries and ranking options
Figure 3 then reveals the page that results from uploading a chapter. The first tab is the chapter text itself, i.e., what the user uploaded. In the current setup the chapter text does not come from the upload but rather the database.

Figure 4: A wireframe depicting what the view for a summary will be

Figure 4 shows what each summary tab should yield: a block of text (e.g., AI generated summary) and a table showing metrics related to it (length, word count, etc.).

Figure 5: A wireframe depicting how the ranking system should be shown to the user
Figure 5 shows the ranking tab, which is intended to allow the user to drag and drop summaries in order of which one was best. We would provide a variety of categories in which the summaries would be rated in order to find the strengths and weaknesses of each.

5.2 Methodology

We further our design process by taking our solution system and breaking it down into 2 sub-goals of allowing users to read and rank summaries, and storing user feedback for each summary to later evaluate. These goals are dependent and together would give us our optimal solution to rank these generated summaries to find the best one.

As can be seen in figure 6, We start with breaking down allowing users to read and rank summaries. This is the first step in our design process, as once we obtain the summaries, we display them in a user-friendly environment. To reach this goal we start from the bottom of our diagram giving the user the option to upload their text to be summarized. After uploading, summaries are generated and displayed each in their own tab view, along with a rating tab where the user can achieve this goal to read and rank summaries.
After displaying the summaries in each tab, we put our focus on the rate page. This page allows the user to rate the overall qualities of each of the summaries along with specifying rankings in which each succeeds or struggles with. After the user has selected their options in the ranking tab, they are able to save their results, which would then be stored back into our database. This is beneficial, because it achieves the purpose of the users to rank these summaries. That allow us to acquire this data regarding whether summaries are successful or not, and to use it to improve for the future.
Figure 8 shows the full breakdown of the sub-goals and system solutions process in this design process. The workflow breakdown allows us as developers to focus on the importance of each step in the design process and produce the best solution possible as a result.
6 Deliverables

**Previous Website Walkthrough**

In Fall 2022, the Team Summarization Methods in CS 5604 [3] developed a website that has a login screen, search capability for ETDs, and pages for each ETD. The page for each ETD provides a summary alongside information about the publication and authors. However, there was no support for user evaluation of chapter summaries.

Another current CS4624 team is responsible for generating chapter summaries using AI, along with displaying these summaries. Our job is to provide an evaluation method for these summaries that could help improve the AI summary generation. Our input would be 3-4 AI generated summaries for the chapters of a provided ETD and one “gold standard” summary which would not be identified to the user. Our platform would then allow the user to rank each of these summaries to determine where these AI generated summaries stand relative to the gold standard summary.

Due to not having access to data that the previous site was based on, and in order to preserve the integrity of the previous website, our team created a new project modeled off the old code base, and added new tabs for Summaries. Our frontend connects to a PostgreSQL database that contains all the data concerning ETD metadata, chapter summaries (both ground truth and generated), chapter texts, and the evaluations received for the summaries from each user.
6.1 Design Reconsiderations

Although the wireframes were a useful starting point for the final design, many things changed throughout the development process. The biggest changes came in two forms: the upload text functionality and the ranking framework. Although the wireframes featured an upload text button, the idea was scrapped as it was beyond the scope of our objectives. Allowing the user to upload their own text is a great functionality once the best AI model has been chosen. When trying to decide exactly which model to use, we found it more useful to provide our own dataset so that there are ground truth summaries associated with it. This provides a better evaluation criteria for the AI models to be compared against. The ranking framework also changed to accommodate an easier backend integration. Rather than dragging and dropping the summaries, the user simply ranks them via a drop down menu. The user also has the ability to rate specific factors for each summary: fluency, content coverage, consistency, an overall rating, and a user comments section.

6.2 Walkthrough of New Website

Figure 10 reveals the main page the user sees when they click on the Summaries tab. Although the ETD Metadata, Chapter Text, and Rate tabs are shown, no data is displayed until the user selects an ETD and Chapter to view.
Figure 11 occurs when the user has chosen an ETD and chapter. Notice how the Summary 1 and Summary 2 tabs are now also present. The ETD Metadata tab is now populated with information about the selected ETD. If the database does not contain a certain piece of metadata for an ETD, that information is simply left blank on the page, as can be seen for the discipline heading.

Figure 12: Chapter Text Tab
Figure 12 shows what the chapter text tab looks like. Upon selecting a chapter, this tab is populated with the actual chapter text. This tab is essential so that the subject matter experts can read the chapter and accurately assess whether or not the summary covers the content it needs to.

Figure 13: Summary 1 Tab

Chapter 5

In this dissertation, I explore the role of financial slack in municipal finance. The subject of study is of significant importance and policy relevance because municipal governments are the layer of governments that are closest to citizens. In the past decade, American cities have experienced an economic peak in 2007, an economic recession between late 2007 and 2009, and economic recovery from 2010. During this turbulent economic time, how cities managed their finances directly shape the outcomes of public goods and service provision that matters to people’s daily life. In the management literature, organization theorists posit that financial slack increases an organization’s capacity in absorbing internal and external shocks. It offers managers more discretion to respond to uncertainty. In the turbulent economic time when government officials face more uncertainty in managing their cities’ revenues and expenditures, what roles does financial slack play in municipal financial management? Prior studies on how financial slack affects government management primarily focus on financial slack’s expenditure stabilization role in state governments. Findings from these studies have reached a relatively consistent conclusion that financial slack has a counter-cyclical stabilization effect on state expenditures. Studies on financial slack’s role in local governments are much scarce. The majority of these studies also focus on financial slack’s expenditure stabilization role, but findings from these studies are mixed. Some find that financial slack stabilizes local expenditures (Marlow 2005; Wang 2015), others do not find such effect (Hendrick 2006; Wang and Hou 2012). Almost all these 129 studies at the local level use samples of cities (or counties) in one state or a particular metropolitan area. Results from these studies have relatively low external validity. This dissertation addresses the limitations in the current research on financial slack in municipal finance. The revenue chapter investigates whether tax revenue volatility contributes to municipal financial slack accumulation. Based on a nationwide sample of cities, I use a dynamic panel analysis approach to conduct the empirical analysis. I find that tax revenue volatility as a risk factor increases a city’s financial slack level. In addition to this, other risk factors in a city’s fiscal system such as credit rating, economic condition, and political factors contribute to municipal financial slack accumulation.
**Figure 14: Summary 2 Tab**

Figures 13 and 14 show each of the summary tabs. In this case, Summary 1 is the ground truth summary and Summary 2 is an AI generated summary. This is not known to the user, however, and is only known to the developer.

**Figure 15: Rate Tab That Allows Users To Evaluate Summaries.**

Figure 15 shows the rate tab. Here, the user can rank the summaries as well as rate each metric on a scale of 1 through 5. The ranking and rating values are required; however the user comments section is optional. The rating tab supports three error messages:

<table>
<thead>
<tr>
<th>Violation</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>The user does not fill out all required values</td>
<td>“Please rate all categories for all summaries.”</td>
</tr>
<tr>
<td>The user does not rank all summaries</td>
<td>“Please select a rank for all summaries.”</td>
</tr>
<tr>
<td>The user provides the same rank for 2 summaries</td>
<td>“Each summary must have a unique rank.”</td>
</tr>
</tbody>
</table>

*Table 2: A table showing what error message each violation in the rating tab leads to*
7 Challenges

7.1 Backend Data Extraction

The team developed a Python script that extracted ground truth chapter summaries for ETDs. However, the script was also picking up other pieces of information: without standardized formats and headings and decoding errors, it proved difficult to automate the abstraction. The team decided it was more important to move forward and focus on more pressing features and deliverables. The team split the ETDs evenly among each other and members extracted the abstracts and chapter summaries manually.

7.2 Importing Database

Our backend lead created the database with all the ETDs on his machine, but we had problems exporting and then importing that database to the local machines of the rest of the team. This slowed down our process of testing as when the backend and frontend were connected, only one person had access to the database to test it. We ended up working past it, meeting a couple of times per week in person to complete the project, and by creating the frontend structure first and adding backend support afterwards.

7.3 Frontend Handoff

Many team members ran into issues with running the original codebase, prolonging the period to assess the existing work. The missing dependencies were resolved as the development environment was being set up. We also had some problems after running the original codebase, as each page couldn’t be displayed until user login credentials were provided, and that was not possible in our run cases to start off because we didn’t have access to the previous webpage’s backend. To work around this, we developed a new tab on the existing webpage and began our implementation from there.

7.4 Insufficient Ground Truth Summaries

Some ETDs did not have summaries for all chapters and many ETDs did not have even one chapter where a summary or abstract was provided. Although the Chapter Summarization team was still able to provide an AI generated summary for those ETDs, the evaluation framework may fail since there is no ground truth summary to compare the generated ones to.
8 Manual

8.1 User Manual

Figure 16: Website home page

Figure 17: Menu bar where the user can navigate to Summary View
Upon first logging into the website, you will see the home page shown in Figure 16. Click on the 3 lines so you see the menu displayed in Figure 17. Navigate towards the bottom and click on Summary View to get to the Summarization Evaluation page for the site. Once you navigate to that page (see Figure 18), you can begin viewing the summaries and evaluating them.

![Figure 18: Summary View Rate Page](image)

Once you have navigated to the summary view page (Figure 18), make sure to select an ETD and a Chapter from the drop down menus; otherwise you will not be able to view any data. Use the tabs at the top to navigate through each section. Once you have read the ETD Metadata, Chapter Text, and all the Summaries, navigate to the Rate tab to provide your evaluation on the summaries. You are purposefully not provided insight into which summary is the ground truth and which is AI generated in order to get a fair evaluation. You are required to fill out each field, except for user comments which are encouraged but not required. Upon clicking save, your evaluation will be saved to the database for the developer team to take into consideration.

### 8.2 Developer’s Manual

#### 8.2.1 Schema Dump

The project files include a PostgreSQL schema dump under the name 'Summarization_Schema.sql’. In order for the code to display any data, this schema must be
imported and set up locally. The file itself may require some edits to actually cooperate with the local machine. The easiest way to do this is to first install PostgreSQL and PGAdmin on your local machine. Once that is set up, create a new database called SummarizationEvaluation, and then click on the import option. Importing the SQL file should populate the database with the required tables and data.

In server.js, make sure to update the following line of code in Figure 19:

```javascript
const sequelize = new Sequelize(
    "postgres://postgres:CapstoneProj@localhost:5432/SummarizationEvaluation"
);
```

*Figure 19: schema must be imported to display data*

so that the path directs to your local PostgreSQL setup. Specifically, change the //postgres and CapstoneProj values, as that is the username and password combination for the database.

### 8.2.2 Set Up and Run The Code

There are many dependencies required to run the code. When running `npm install`, make sure to use the `--legacy-peer-deps` flag to ignore the errors concerning different versions of React. These warnings are not fatal and the code will still run smoothly if that flag is applied. The server side code provided in server.js is what connects the React modules to the PostgreSQL database. In order to access the database the server should be run on a separate terminal via 'node server.js'. If the startup fails, it is most likely due to dependencies not being installed, so make sure to install them via `npm start <library or dependency name> --legacy-peer-deps'`. Again, `--legacy-peer-deps` is required here so that installation is compatible with different versions of React. Upon installing all dependencies, the server side code should run.

### 8.2.3 Server.js considerations

Make sure there is no other process using port 3001 on your local machine, as that is the port that is used by the server. If there is, either close the application running on 3001, or update the server side code to use a different port. If the server is changed to use a different port, the files given in Figure 20 will need to be changed as well, to accommodate for the new port.

<table>
<thead>
<tr>
<th>File to be changed</th>
<th>Filepath</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter.js</td>
<td>src/pages/Chapter.js</td>
</tr>
<tr>
<td>ChapterText.js</td>
<td>src/pages/ChapterText.js</td>
</tr>
</tbody>
</table>
The ‘await fetch()’ (Figure 21) call specifically is what requires changing in each file.

```
const fetchChapterText = async () => {
  try {
    const response = await fetch(`http://localhost:3001/api/etd/${etdID}/chapter/${chapter_no}`);
    const chapterData = await response.json();
    setChapterContent(chapterData);
  } catch (error) {
    console.error("Error fetching chapter text:", error);
  }
};
```

*Figure 21: Line of code to change if setting up server on a different port*

### 8.2.4 Starting The Website

Once the setup is completed, run `npm start` on a separate terminal (i.e., not the one node server.js is already running on). Running that command should immediately open up the website in the default browser. If not, simply navigate to ‘http://localhost:3000/summaryview’ in the browser of your choice and you should be connected.

### 8.2.5 Database Population

As different users interact with the website and submit their evaluations, those evaluations will be stored in the sum_eval table in the database. Examine that table to understand the summary ratings and make changes to the AI models and AI generated summaries as required.

Any new entries that need to be added to the etd table, chapter table, or the summary table would need to be done manually for now. Our process involved creating an Excel sheet
with all the data and iteratively adding to it as we received more data. Upon each iteration we imported the data into the respective tables in the database schema. Once the database is updated, the server must be restarted to account for the changes and the data to be reflected in the website. You are also free to use any PostgreSQL GUI you want, as everything is local to your machine and not specific to pgAdmin.
9 Discussion and Future Efforts

9.1 Discussion

By the end of the time we had for the project, we were able to successfully build a website that allows for evaluation for the generated summaries, so we completed the main objective we had set for ourselves. However, many things changed along the way and one goal of ours was not met due to the challenges faced. Although the wireframes inspired the final design for the site, much of the functionality from the wireframes is not included in the current design. This was due to a shift in focus from providing users with maximum customizability to focusing more on the development of the AI models themselves and evaluating the summaries against the ground truths we had as a baseline. Our biggest setback was automating the extraction of chapter summaries. The lack of a consistent structure for each ETD and the various decoding errors diminished our hopes of automating the entire pipeline as much as possible. We were trying to future proof the development setup by making it easy to import data, but we spent too much time trying to fix the discrepancies and ended up being set back by 2 weeks. Although we still met most of our objectives, we missed one main one: user tests.

9.1 User Tests

We planned to conduct user tests and to complete iterative improvements based on the user feedback. We were not able to meet this deliverable on time for this report. The plans for the last 2 weeks of the project were to have the client run the project on her personal computer and have her share it with her colleagues at the lab. After completing adjustments, we planned to get feedback from classmates. We have researched important factors for a “Gold Standard” summary for the user to evaluate each summary before ranking but we were looking to see if the students in the lab or class have other suggestions for these factors.
10 References


    https://www.geeksforgeeks.org/build-a-basic-react-app-that-display-hello-world


