AC-2 Data Visualizer for Topic Models

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Course: 4805-03

Semester: Spring 2023

Professor Perry

Date 02/09/2023

Team

Project Owner: Arthur Choi

Team Members:

Kalp Patel, Aditya Shiroya, Michael Cooper, James Roll

Table of Contents

1.0 Introduction

In many fields, managing and examining large document collections is an essential task. Many document collections, however, are disorganized, and manual organization is impractical. As a result, automated techniques for discovering and visualizing a collection's structure are crucial to facilitating content exploration. A topic model is a statistical model that represents the latent topical structure of text. The main objective of this project is to automatically discover the topics that appear in Wikipedia articles, by using only the words appearing in the articles themselves. The system aims to analyze, explore, and visualize a topic model that has been learned from a dataset, such as Wikipedia, and to incorporate a recommendation engine into the interactive visualization. Moreover, our recommendation engine will also show how similar the suggested information is to the article the user is currently viewing, using the similarity matrix/heat map. By using our system, users will be able to easily explore and understand the latent topics that exist in any given text dataset. We hope our system will provide a valuable tool to researchers, analysts, and anyone interested in understanding complex text datasets.

1.1 Overview

The aim of this project is to create a web-based system for organizing, summarizing, and displaying a large collection of documents, such as Wikipedia articles. The underlying thematic structure of the articles will be found using topic modeling techniques, which will then represent each text as a collection of topics. The system will give users a simple interface to navigate between high-level summaries (the "themes") and the specific articles themselves as they browse the corpus. The visualization will also include a recommendation engine that will propose articles with related themes and display how similar the suggested material is to the item being viewed right now. The system will ultimately help people explore and understand vast amounts of unstructured documents.

1.2 Project Goals

The main goal of this project is to create a web-based system that utilizes topic modeling to automatically identify and display the thematic organization of a sizable corpus of text, like Wikipedia articles. The system will offer a user-friendly interface that enables users to move between high-level found summaries (the "themes") and individual texts to explore and browse the corpus. The system will also have a recommendation engine that suggests articles with related themes and shows how similar different pieces of information are to one another. This research will make a contribution to the field of information retrieval and data mining by offering an effective tool for arranging and summarizing large amounts of unstructured data.

1.3 Definitions and Acronyms

Definitions:

Topic Model: A statistical model that represents the latent topical structure of text.

Wikipedia Dataset: A collection of text articles from the website Wikipedia.

Recommendation Engine: A software program that suggests relevant content to users based on their behavior and preferences.

Similarity Matrix: A matrix that measures the similarity between two or more items.

Corpus: a large collection of written or spoken texts, treated as a whole

Graph: A visual representation of data or information.

Heat Map: A visual representation of data that uses color to show different levels of intensity.

Web-based System: A server-based software system that is accessed via a web browser.

Acronyms:

API: Application Programming Interface.

AI: Artificial Intelligence

HTML: Hypertext Markup Language.

CSS: Cascading Style Sheets.

JS: JavaScript.

UI: User Interface.

UX: User Experience.

DB: Database.

HTTP: Hypertext Transfer Protocol.

HTTPS: Hypertext Transfer Protocol Secure.

SQL: Structured Query Language.

CSV: Comma-Separated Values.

1.4 Assumptions

- The system will be designed and developed for use with English language text.
- Client/user has an active Internet Connection or has access to one to view the website.
- The topic model will be generated using Latent Dirichlet Allocation (LDA) algorithm.
- The user interacting with the system will have a basic understanding of topic modeling.
- The system will be used in a web-based environment.
- The system will be accessed through a standard web browser.
- The system will be developed using open-source technologies.
- The system will be tested and evaluated on Wikipedia dataset.

2.0 Design Constraints

2.1 Environment

The website will be developed and tested on local computers during the development period. However, the product will be containerized and installed on a server during production using Kubernetes. There are no particular requirements for database management because we use an embedded MongoDB database. MongoDB is the best database for our needs because it is a NoSQL database, and we will just be storing simple data in the form of documents.

2.2 User Characteristics

This website is intended for students or researchers who need extra resources for their research projects. Using probabilistic topic modeling, the website will suggest related content in response to user search queries. We will serve an extensive range of users, including those who are new to the issue and those who already have a solid understanding, as the website intends to educate people about topic modeling. We will include suggestion boxes that describe how the results are created in order to make the website more user-friendly.

2.3 System

The foundation of our website is probabilistic topic modeling. We will update the system as necessary to keep it current and relevant if there are any substantial advancements in the field. We will also monitor changes to the React and Spring frameworks, which we will be utilizing to develop the website. We will update the system to maintain compatibility and guarantee seamless performance if there are any significant updates or changes.

3.0 Functional Requirements

- Using a given dataset, the system should be able to produce topic models.
- Users should be able to explore and visualize the topic model using the system's interactive web interface.
- The system should include a recommendation engine that proposes articles with related themes and rates how similar the material is to the one being viewed.
- Users should be able to sort and filter search results using a variety of criteria using the system.
- For the aim of easing the study and understanding of the topic model, the system should include visualizations such word clouds, topic clouds, and heat maps.
- Users should be able to export search results and visualizations from the system for use in other applications and analysis.
- Large datasets should be manageable by the system, and its search and visualization tools should be quick and effective.
- Users with various levels of technical expertise should be able to utilize the system without any trouble.
- To allow further updates and improvements to the topic modeling method and web interface, the system should be scalable.

4.0 Non-Functional Requirements

4.1 Security

To safeguard user data and stop illegal access, the system should include security safeguards. The system must make sure that user data is kept private and secure. This involves developing safe login and authentication processes, encrypting private information, and guarding against unwanted system access.

4.2 Capacity

The system must have the capacity to manage several users and documents. Without encountering performance problems, it needs to be able to process and store a sizable amount of data. A considerable increase in traffic must be handled by the system without any disruptions or slowdowns.

4.3 Usability

The program must be simple to use and navigate. Users should be able to instantly look for and obtain relevant documents, as well as view and investigate the topics and relationships among them. In order to assist users who might not be familiar with topic modeling, the system should additionally offer useful hints and explanations.

4.4 Other

Performance, maintainability, and reliability may be seen as additional non-functional requirements. The website should be built with a minimum amount of latency and reaction time to operate swiftly and effectively. A clear, modular code structure that is simple to update and maintain over time should also be considered while designing the website. Lastly, the website should be built with dependability in mind. Backups and disaster recovery plans should be in place to guarantee that the website can be swiftly restored in the case of a system failure or other difficulty.

5.0 External Interface Requirements

5.1 User Interface Requirements

The primary objective of the user interface is to give users a simple, user-friendly website where they can search for articles and browse related articles using a topic model. Users can use keywords relating to their research topic or the article's name to search for it. To help viewers better understand how the connected articles relate to one another, the related articles will be presented in a visually appealing fashion, like a graph or heat map. The user interface also seeks to educate users about topic models by offering details on the likelihood values and the justifications for why particular articles appear to be linked. The interface will be made to be user-friendly, with easy-to-follow instructions and straightforward navigation.

5.2 Hardware Interface Requirements

- The system will be web-based and usable from any device with an internet connection.
- Modern desktop and mobile browsers will be able to access the website with no problems.

- For accessing the system, no particular hardware requirements are necessary.

5.3 Software Interface Requirements

The system will make use of the Java Spring Framework, Maven, Lombok API, and Tomcat web services along with a REST API. These tools will be used to build a connection between the back-end topic model system and the web application. In the future, other software interfaces could be implemented as necessary.

5.4 Communication Interface Requirements

The user and the system will communicate via a web-based graphical user interface (GUI). The GUI will be created to be responsive, user-friendly, and easy to browse. To retrieve and process the data, the system will also interface with other APIs, such as Wikipedia. The HTTP/HTTPS protocol will be used by the system to facilitate communication between the client and the server. The system will give the user feedback in the form of alerts, warnings, and status messages. These messages, which will be helpful and simple to comprehend, will be shown on the user interface. In the event of problems or exceptions, the system will also provide error messages. In order to be analyzed afterwards, these signals will be recorded in a different error log file.

APPENDICES

Appendix A: Use Cases

1. Main Success Scenario

   User selects "Create Account" from the menu.

   The user is shown a registration form by the system.

   The user completes and submits the registration form.

   The system sets up the user's account and logs them in.

2. Main Success Scenario

   The user selects the "Forgot Password" option.

   The system shows the user a form to input their email address.

   The user clicks "Submit" after entering their email address.

   User receives an email from the system with instructions for changing their password.

   User complies with the instructions and changes their password.

   Using their new password, the user has signed in.

3. Search for article:

Primary Actor: User

Precondition: User is now on the website's main page.

Postcondition: User is brought to the article page.

4. The user enters the article's name in the search window and hits "Search."

   A list of articles matching the search query is displayed once the system searches for the article.

   The user chooses the desired item from the list.

   The system finds and shows the article page.

   View Related Articles

   Primary Actor: User

   Precondition: User is currently on an article page.

   Postcondition: Related articles are displayed to the user.

5. When a user visits an article page, they see a section titled "Related Articles."

   A related article link is clicked by the user.

   The related article page is retrieved and displayed by the system.

   Analyze the topic model:

   Primary Actor: User

   Precondition: User is currently on an article page.

   Postcondition: The topic model is visualized for the user.