Final Project Writeup

Project Link: https://github.com/anagha1999/study_resource_recommender

Introduction

This project, "Study Resource Recommender," is designed to help MIMS students access curated study materials based on their preferences for topics, content types, and, most importantly, agreement by TAs. Implemented in Python, the system utilizes structured JSON datasets to store resource metadata, enabling efficient retrieval and manipulation.

Interface for TAs: At the core of the solution is a tagging system that allows TAs to annotate resources independently. The reliability of their annotations is evaluated using Cohen's Kappa, ensuring that only highly agreed-upon resources are prioritized for recommendations. This fosters a robust and consensus-driven approach to resource selection.

Interface for Students: For students, the system offers a faceted navigation interface through an interactive CLI, allowing dynamic filtering of resources based on multiple attributes. Enhanced with autocomplete functionality, this interface ensures ease of use and efficiency. The recommendation engine further refines the experience by integrating student preferences with quality-assessed tags to deliver personalized and reliable study resource suggestions.

Each feature—tagging, navigation, and recommendation—functions as a modular component, collectively creating a comprehensive and reliable system for information discovery and resource recommendation.

Connection to Class Concepts

JSON Dataset

The code uses JSON as a lightweight, structured data format to store and retrieve both resources and tags in separate files. The resources JSON file includes metadata attributes such as ID, title, topic, difficulty, and content type, ensuring a clear and consistent representation of study materials. The tags JSON file records annotations provided by TAs, including the resource ID and corresponding tags, enabling independent storage and retrieval of tagging data. This design aligns with principles discussed in 'Structuring Data' from INFO 202, where metadata enhances consistent representation, efficient search, and interoperability. By maintaining separate JSON files for resources and tags, the system supports modularity and scalability, making it suitable for dynamic tasks like tagging, faceted navigation, and recommendation systems. Leveraging JSON's hierarchical structure, the dataset provides flexibility to organize and process nested attributes with precision,

directly tying to the course concepts of structured data and metadata described in 'Structuring Data' readings.

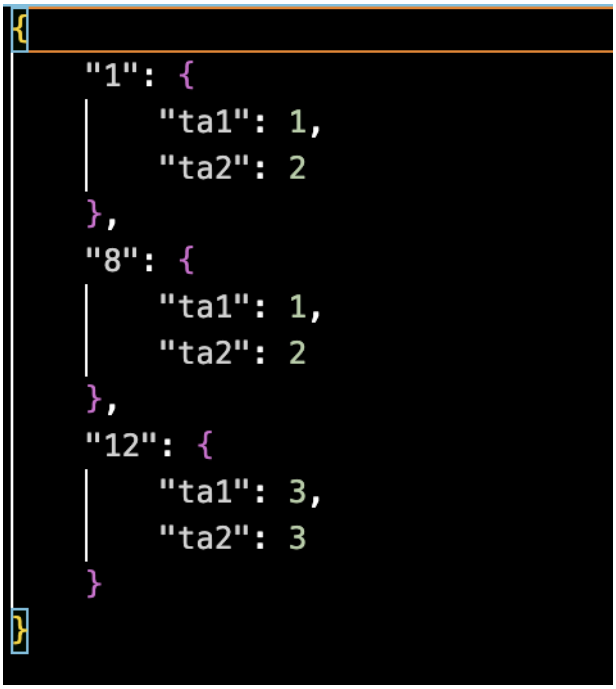

Image 1: tags.json

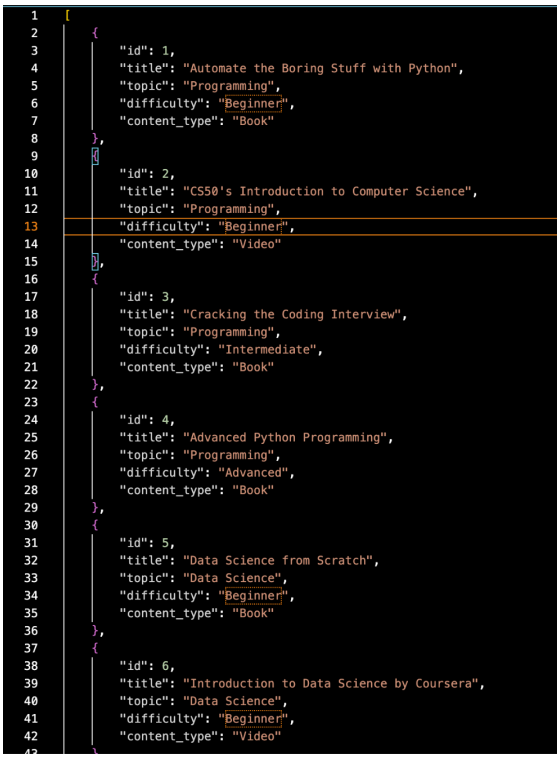

Image 2: resources.json

Cohen's Kappa

Cohen's Kappa is used in the tagging functionality to assess inter-annotator agreement. The code involves two TAs tagging resources independently, and the agreement between their annotations is calculated using Cohen's Kappa formula. This implementation reflects the importance of inter-annotator reliability highlighted in INFO 202 lecture materials on Inter-Rater Reliability. Cohen's Kappa accounts for chance agreement, providing a robust measure of annotation quality. Resources with high agreement are prioritized for recommendations, ensuring reliability. The tagging process is consistent with guidelines for computational grounded coding and annotation pipelines as described in 'Inter-Rater Reliability' readings.

```
anagha@Anaghas-MacBook-Pro study_recommendation_system % python3 resource_tagging.py

Available Resources:
1: Automate the Boring Stuff with Python (Programming, Beginner, Book)
2: CS50's Introduction to Computer Science (Programming, Beginner, Video)
3: Cracking the Coding Interview (Programming, Intermediate, Book)
4: Advanced Python Programming (Programming, Advanced, Book)
5: Data Science from Scratch (Data Science, Beginner, Book)
6: Introduction to Data Science by Coursera (Data Science, Beginner, Video)
7: Practical Statistics for Data Scientists (Data Science, Intermediate, Book)
8: Python for Data Analysis (Data Science, Intermediate, Book)
9: Deep Learning Specialization by Andrew Ng (AI/ML, Beginner, Video)
10: Hands-On Machine Learning with Scikit-Learn and TensorFlow (AI/ML, Intermediate, Book)
11: Reinforcement Learning: An Introduction (AI/ML, Advanced, Book)
12: AI for Everyone by Andrew Ng (AI/ML, Beginner, Video)
13: The Design of Everyday Things (HCI, Beginner, Book)
14: Don't Make Me Think: A Common Sense Approach to Web Usability (HCI, Intermediate, Book)
15: Human-Computer Interaction (HCI, Advanced, Book)
16: Interaction Design Foundation Courses (HCI, Beginner, Video)
17: Cybersecurity for Beginners (Security, Beginner, Book)
18: The Web Application Hacker's Handbook (Security, Intermediate, Book)
19: Practical Malware Analysis (Security, Advanced, Book)
20: Introduction to Cybersecurity by IBM (Security, Beginner, Video)

Select resource IDs to tag. Type 'done' to finish.
Enter resource ID to tag or 'done': 1
Selected Resource: Automate the Boring Stuff with Python
TA1: Enter topic tag (1-Programming, 2-Data Science, 3-AI/ML, 4-HCI, 5-Security): 1
TA2: Enter topic tag (1-Programming, 2-Data Science, 3-AI/ML, 4-HCI, 5-Security): 1
Enter resource ID to tag or 'done': 12
Selected Resource: AI for Everyone by Andrew Ng
TA1: Enter topic tag (1-Programming, 2-Data Science, 3-AI/ML, 4-HCI, 5-Security): 2
TA2: Enter topic tag (1-Programming, 2-Data Science, 3-AI/ML, 4-HCI, 5-Security): 3
Enter resource ID to tag or 'done': done

Cohen's Kappa Score for Inter-Annotator Agreement: 0.3333333333333333
Tags saved in tags.json
```

Image 3: Demonstration of Cohen's Kappa for two Teaching Assistants, after tagging two resources. They agree that the first is a "Programming" resource, and disagree on the next.

Faceted Navigation

Faceted navigation, implemented in the code, allows users to filter resources dynamically by topic, difficulty, and content type. This method connects directly to the concept of faceted metadata discussed in INFO 202 materials. Faceted navigation empowers users to construct flexible queries and refine their search iteratively. This project features like autocomplete, the code enhances user experience, reducing input errors and increasing efficiency. The

system's design adheres to principles of mixing and matching facets to accommodate various navigation paths. This design philosophy aligns with the advantages of faceted navigation for metadata-rich systems described in 'Faceted Navigation' readings.

```
anagha@Anaghas-MacBook-Pro study_recommendation_system % python3 faceted_navigation.py
Faceted Navigation:
Enter topic (autocomplete available): AI/ML
Enter difficulty (autocomplete available): Beginner
Enter content type (autocomplete available): Video

Filtered Resources:
- Deep Learning Specialization by Andrew Ng (AI/ML, Beginner, Video)
- AI for Everyone by Andrew Ng (AI/ML, Beginner, Video)
```

Image 4: Faceted Navigation interface for Students sorted by Inter-Annotator Agreement.