[AI-related FYP] AI for Personalized Content Recommendations

Prof Ruohan Zhan

Names of Participants:

CHENG, Yan Hei (Cathy)

WAN, Nga Chi (GiGi)

CHAN, Tony Yuen Yeung (Tony)

Project Summary

i) What is the problem?

The primary challenge in building a useful content recommendation system (CR) is effectively capturing and adapting to user preferences in dynamic environments. This challenge is compounded by issues such as data sparsity, which arises from limited user-item interactions, and the cold-start problem for new users and items. Additionally, scalability is crucial for processing large datasets and delivering timely recommendations. This project focuses on studying various techniques and models in CR systems to discover solutions for these common issues, ultimately aiming to enhance the effectiveness and user satisfaction of recommendation systems.

ii) How do you hope to solve it?

1. **Study Different CR Techniques/Models**:
- Conduct a comprehensive literature review of existing CR techniques, including collaborative filtering, content-based filtering, and hybrid approaches.
- Investigate temporal dynamics in recommendations to understand how user preferences change over time.
2. **Research Common Issues in CR Systems**:
- Identify prevalent challenges in CR systems, such as the cold-start problem, scalability, and user satisfaction.
- Explore industrial approaches to address these issues, focusing on effective solutions.
3. **Build an Interactive CR System**:
- Backend: Develop a system that processes user requests (queries) and applies various CR models based on user interactions and preferences.

- Frontend: Create a user-friendly interface that stores cookies and allows users to interact seamlessly with the system.

4. **Incorporate Extra Features**:

- Integrate additional functionalities such as sentiment analysis and large language model (LLM) extraction to enhance the recommendation process.

5. **Evaluate the Performance of the CR System**:

- Gather user feedback through user acceptance testing (UAT) and conduct simulations to assess the system's performance.
- Measure key performance metrics, including precision and recall, to evaluate the effectiveness of recommendations.

6. **Review and Adjust Techniques/Models**:

- Analyze the performance results and adjust the models, parameters, and structures based on user feedback and simulation outcomes.

iii) What result do you expect to achieve? (A time schedule should also be provided)

1. **Interactive CR System**:

- A fully functional CR system that effectively processes user queries and provides relevant recommendations.
- Backend: Capable of applying different models based on user input.
- Frontend: User interface that enhances user interaction and experience.

2. **Research Paper**:

- Existing Techniques/Models: A comprehensive overview of popular industrial approaches to CR techniques and models studied in the project.
- Common Issues/Problems: Identification of general issues encountered in CR systems and the proposed solutions, along with limitations of the approaches.
- Performance Evaluation: Detailed analysis of UAT results, comparing performance metrics (precision and recall) across different stages to demonstrate improvements.
- Limitations and Future Research Directions: Discussion of the project's limitations and suggestions for future research avenues.

3. **Project Video**:

- Provide a brief overview of the content recommendation system (CR) project, highlighting its key objectives and significance in enhancing user experiences.
- Showcase the interactive CR system in action, demonstrating how it processes user queries to generate personalized recommendations.

iv) What is important about your expected result?

1. **Relevance of Recommendations**: The CR system should be able to return relevant search results based on user keywords and feedback, enhancing user satisfaction and engagement.
2. **Performance and Diversity of Techniques/Models**:

- Basic CR Models (60%): Implement foundational models such as Matrix Completion and Collaborative Filtering.
- Machine Learning Techniques (30%): Incorporate advanced techniques like embeddings to improve recommendation quality.
- Additional Techniques (10%): Utilize extra techniques such as LLMs to enhance the system's capabilities.

3. **Delivery of the Whole Project**: The successful execution of the project will be evaluated at each stage, focusing on:

- Data Pre-Processing:
    - Identify key features and insights from the dataset.
    - Clean, standardize, and embed the data for effective processing.
- CR System Functionality:
    - Implement a structured retrieval process: Retrieval => Pre-Ranking => Ranking => Re-Ranking.
- User Experience: Ensure the functionality and UI/UX of the interactive CR system are intuitive and user-friendly.

Project Timeline:

Task	Duration	Expect End Date	Status
Study the Background of CR	3 months	17/11	Completed
~~Study and Select the Dataset~~	~~2 months~~	~~17/11~~	Adjusted
Research on System Setup	1 month	17/11	Completed
~~Data Pre-Processing~~	~~2 weeks~~	~~31/11~~	Removed
1st Stage BE (Retrieval)	1 month	31/12	In Progress
2nd Stage BE (Pre-Ranking + Ranking)	1 month	31/1	/
1st Stage FE (Draft for UAT)	1 month	31/1	/
1st UAT + 3rd Stage BE (Revamp + Re-Ranking)	1 month	28/2	/
2nd Stage FE (80%)	1 month	28/2	/
Final Report (1st half)	2 weeks	16/3	/
2nd UAT + 4th Stage BE (Revamp + Other Features)	1 month	31/3	/
Final Report (2nd half)	2 weeks	6/4	/
3rd UAT (Final Evaluation)	2 weeks	13/4	/
3rd Stage FE (Finalize)	2 weeks	13/4	/
Video Production	1 month	30/4 (TBC)	/

Task Update:

1. **Study the Background of CR (Completed)**

- Completed an initial review of retrieval models, including User/Item Collaborative Filtering, Deep Structured Semantic Model (DSSM), and Deep Retrieval according to a tutorial of Red Recommendation System.
- Will continue exploring models addressing specific challenges and documenting pain points as part of further research.

2. **Study and Select the Dataset (Adjusted)**

- Currently using the Yelp Dataset for learning and testing.
- Open to exploring additional datasets that might align better with project goals.

3. **Research on System Setup (Completed)**

- Modelling: *Python* (using *TensorFlow* & *Scikit-learn*; may expand).
- Backend: *Python* (*Flask* for API and real-time processing).
- Frontend: *React* (default *Flask* pages used temporarily).
- Database: *SQLite* (considering *Redis* or *Milvus* for vector-based queries).
- Hosting: Linux machine provided by the department (exploring *AWS/Azure* as alternatives).

4. **Data Pre-Processing (Removed)**

   Pre-processing will depend on the selected models. This task has been deferred to allow flexibility.

5. **1st Stage BE (Retrieval) (In Progress)**

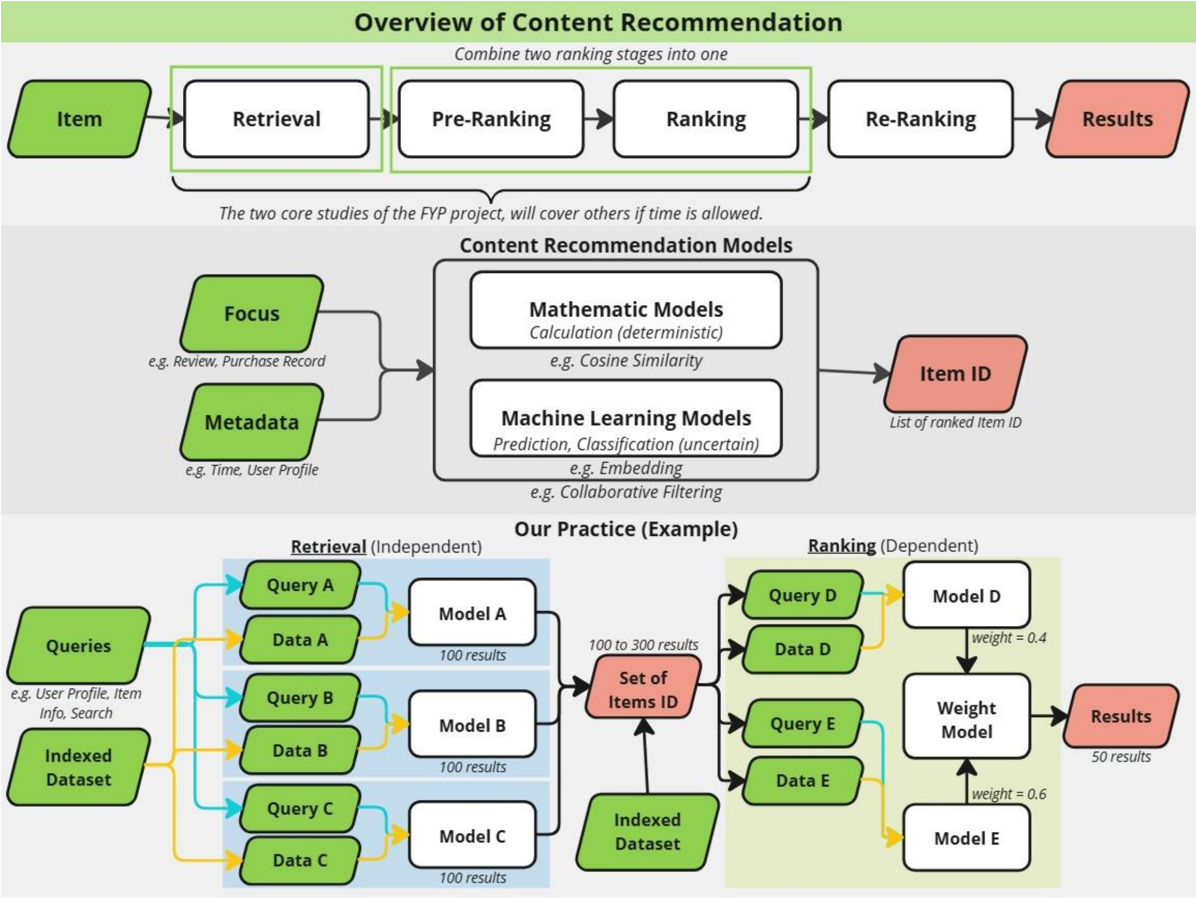
- Implemented a draft ***Item Collaborative Filtering*** (***Item CF***) system using Flask.

- Currently working on the ***Deep Structured Semantic Model*** (***DSSM***).
- Planned Retrieval Models:
  (1) **Item CF**, (2) **User CF**, (3) **DSSM**, (4) **Deep Retrieval/Tree-based Deep Model** (**TDM**), (5) **GeoHash** and other simple retrieval models.

Additional Updates:

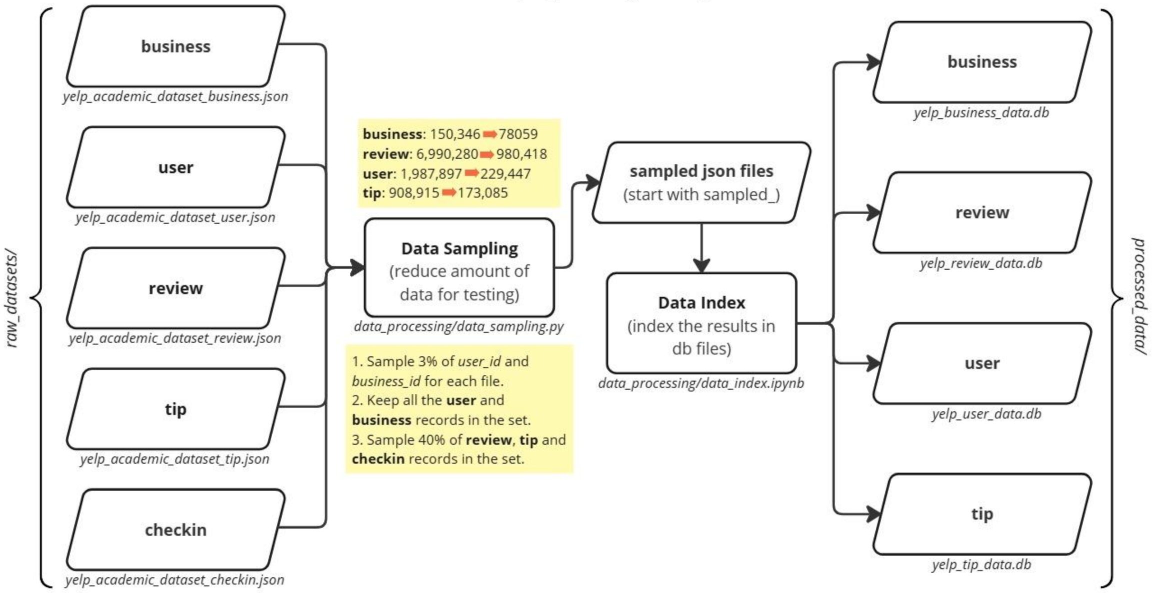
1. Task Management: Trello for project tracking.
2. Ideation & Visualization: Miro for brainstorming, planning, and visualization.
3. GitHub Repository: Content Recommendation Project

Visualizations and updates are shared on Miro. Feedback & suggestions are welcome!



Data Sampling and Indexing for Yelp Dataset

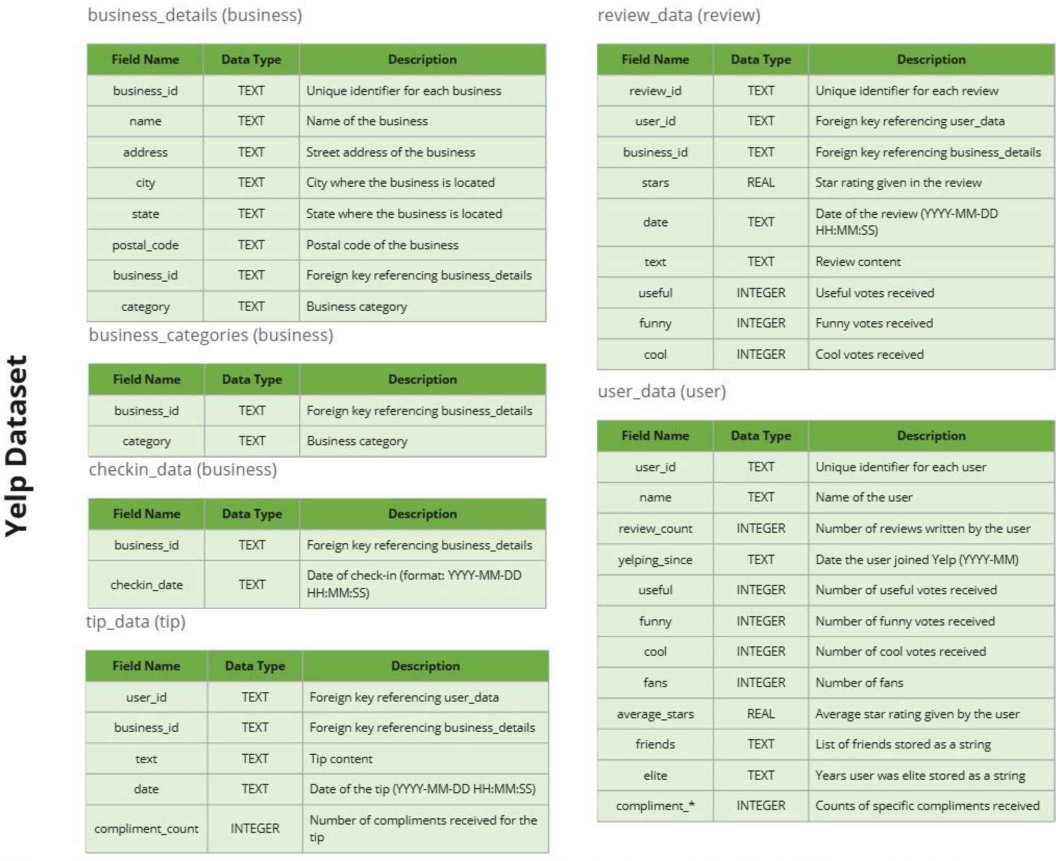
To reduce data size which reduce the time and complexity for training and testing, will use the full dataset in final version.


raw_datasets/
business
yelp_academic_dataset_business.json
user
yelp_academic_dataset_user.json
review
yelp_academic_dataset_review.json
tip
yelp_academic_dataset_tip.json
checkin
yelp_academic_dataset_checkin.json
Data Sampling
(reduce amount of data for testing)
data_processing/data_sampling.py
sampled json files
(start with sampled_)
Data Index
(index the results in db files)
data_processing/data_index.ipynb
business
yelp_business_data.db
review
yelp_review_data.db
user
yelp_user_data.db
tip
yelp_tip_data.db
processed_data/

business: 150,346➡78059
review: 6,990,280➡980,418
user: 1,987,897➡229,447
tip: 908,915➡173,085

1. Sample 3% of *user_id* and *business_id* for each file.
2. Keep all the **user** and **business** records in the set.
3. Sample 40% of **review**, **tip** and **checkin** records in the set.

We ignore the **photo** feature of the dataset now, but we may use it as some model features in the future.



Yelp Dataset

business_details (business)

Field Name	Data Type	Description
business_id	TEXT	Unique identifier for each business
name	TEXT	Name of the business
address	TEXT	Street address of the business
city	TEXT	City where the business is located
state	TEXT	State where the business is located
postal_code	TEXT	Postal code of the business
business_id	TEXT	Foreign key referencing business_details
category	TEXT	Business category

business_categories (business)

Field Name	Data Type	Description
business_id	TEXT	Foreign key referencing business_details
category	TEXT	Business category

checkin_data (business)

Field Name	Data Type	Description
business_id	TEXT	Foreign key referencing business_details
checkin_date	TEXT	Date of check-in (format: YYYY-MM-DD HH:MM:SS)

tip_data (tip)

Field Name	Data Type	Description
user_id	TEXT	Foreign key referencing user_data
business_id	TEXT	Foreign key referencing business_details
text	TEXT	Tip content
date	TEXT	Date of the tip (YYYY-MM-DD HH:MM:SS)
compliment_count	INTEGER	Number of compliments received for the tip

review_data (review)

Field Name	Data Type	Description
review_id	TEXT	Unique identifier for each review
user_id	TEXT	Foreign key referencing user_data
business_id	TEXT	Foreign key referencing business_details
stars	REAL	Star rating given in the review
date	TEXT	Date of the review (YYYY-MM-DD HH:MM:SS)
text	TEXT	Review content
useful	INTEGER	Useful votes received
funny	INTEGER	Funny votes received
cool	INTEGER	Cool votes received

user_data (user)

Field Name	Data Type	Description
user_id	TEXT	Unique identifier for each user
name	TEXT	Name of the user
review_count	INTEGER	Number of reviews written by the user
yelping_since	TEXT	Date the user joined Yelp (YYYY-MM)
useful	INTEGER	Number of useful votes received
funny	INTEGER	Number of funny votes received
cool	INTEGER	Number of cool votes received
fans	INTEGER	Number of fans
average_stars	REAL	Average star rating given by the user
friends	TEXT	List of friends stored as a string
elite	TEXT	Years user was elite stored as a string
compliment_*	INTEGER	Counts of specific compliments received

We use **SQLite** to store all data as db files at this moment, may use other database tools (e.g. **Redis**, **Milvus**, etc.) in the future.

The Author List

Name(s) of the report writer(s). (For multiple writers, indicate sections and/or percentages done by each.)

Section	Participant
i) What is the problem?	Cathy
ii) How do you hope to solve it?	Cathy (50%), Tony (50%)
iii) What result do you expect to achieve? (A time schedule should also be provided)	GiGi (50%), Tony (50%)
iv) What is important about your expected result?	GiGi