Discussion with Professor (Email):

blue text is from Ruohan; highlighted in yellow means follow-up is needed.

1. Strengthen the Motivation of Research:

Motivation: Consider adding more background information to better motivate your research question. Recommendation System plays a significant role in everyday life--perhaps include examples that highlight their impact in domains like e-commerce, streaming platforms, or personalized learning.

2. Setup Base-Line Evaluation Model & Process Report:

Evaluation baselines: It would be helpful to include baseline methods that are straightforward to implement and provide an intuitive comparison for your method. For instance, you could consider baselines like random recommendations or the best uniform recommendations where everyone receives the same suggestion.

Our ultimate solution will use a hybrid approach (some uniform and tailored models combined). In the sense of that, should we also present the results using either one as well for comparison? We are also planning to see if we would highlight some minor tunning we adjust for better performance (e.g. loss function in some model, how we deal with sampling bias, etc.). Should we also include these parts as well while they are not the focus?

For 2, yes it would be good to compare the final hybrid model with each one of them. Adding details of the tunning process would also be beneficial, which reflects the systematic approach you are taking.

3. Heuristic/Qualitative Evaluation:

Heuristic evaluation: can you elaborate more on what you mean by the heuristic approach? If your aim is to evaluate your method's real-world performance, consider conducting surveys or lab experiments. You might also explore department resources for support in running these evaluations.

Yes, the heuristic evaluation we are talking here is similar to real-world performance or qualitative assessment. We did discuss about conducting surveys as one of the evaluation. It is possible to do it for our final outcome, but it might be difficult to do comparison. It may be difficult for us to conduct serval surveys in this way. On top of this, my question is more about the phrasal evaluation. We are going to continue the study in some phases, so we are thinking if we can evaluate their performance in a heuristic way? It is ok if we can only do the quantitative approaches (using traditional evaluation metric, e.g. accuracy, recall, ROC, etc.).

For 3, it's ok to only use the quantitative approaches, but adding real-world evaluations would make your results much more compelling and interesting. You may also consider using GPT as an "AI"-evaluator.

Study Methodologies:

Our project aims to study content recommendation systems and address common problems in the industrial domain. Consequently, the problem-solution approach serves as the main framework for our research. Below are the methodologies we employ:

1. Comparative Evaluation

This method forms the backbone of the study and involves systematically comparing the baseline and target models to assess their performance under different conditions.

Steps:

1. Setup Evaluation Metrics: Define clear and measurable metrics to assess the models. Common metrics include:
   - Accuracy, Precision, Recall, F1-Score: Measure relevance and correctness of recommendations.
   - User Satisfaction: Use a proxy metric or qualitative measure if applicable.
   - Intra-List Diversity (Optional): Assess the variety within the recommendation list.
   - Novelty (Optional): Evaluate how new or unexpected the recommendations are.
   - Coverage (Optional): Measure how much of the dataset the system can recommend effectively.
2. Select Target Users/Items: Choose the evaluation group based on the focus of the study. This could include:
   - The entire testing dataset for general evaluation.
   - Specific groups like cold-start users, long-tail items, or niche categories for focused analysis.
3. Select Base-Line Model: Define the existing or simple model for comparison, such as Collaborative Filtering or Content-Based Filtering.
   - *Suggested Base-Line Models (from Ruohan): Random Recommendation, Best Uniform Recommendation (same recommendation to all users).*
4. Select Target Model: Define the enhanced model under evaluation, such as DSSM or a hybrid recommender system.
5. Statistical and Visual Analysis:
   - Visualization: Use bar charts, ROC curves, and trade-off plots to present findings clearly.
   - Statistical Testing (Optional): Validate differences in performance using paired t-tests or similar methods.

2. User Acceptance Test (UAT)

This method emphasizes direct involvement of users to evaluate the system's performance from a qualitative perspective. (Similar to the user journey map)

Steps:

1. Define User Group: Set up profiles for user groups through questionaries and assumptions. This group should reflect the diversity of the target audience.
2. Design Scenarios: Present users with predefined scenarios and recommendations generated by the system.
3. Collect Feedback:

- Use surveys or interviews to gather qualitative insights about relevance, usability, and satisfaction.
- Ask users to rank or rate recommendations and provide reasoning for their preferences.
- *Recommendation (from Ruohan): Consider using GPT as an AI-evaluator.*

4. Analyze Results:

- Identify patterns in feedback, such as common likes/dislikes or usability issues.
- Use the insights to complement quantitative findings from Comparative Evaluation.

3. Case Studies and Scenario Simulation (Optional)

This method involves testing the system's behavior in specific, well-defined cases to illustrate its strengths and limitations.

Steps:

1. Define Use Cases: Identify scenarios that highlight specific challenges or features, such as:

- Recommending niche items with limited data (long-tail problem).
- Handling users with unique preferences.
- Adjusting recommendations to balance diversity and relevance.

2. Qualitative Evaluation: Examine the recommendations generated for these cases:

- Highlight successful examples where the system performed well.
- Discuss limitations or unexpected outcomes to identify areas for improvement.

Study Focus (Problems):

Below is the list of problems we identified and the step-by-step strategies to address them. Some problems are prioritized higher due to their significant impact on recommendation performance. For lower-priority issues, detailed strategies may not yet be developed.

1. Data Sparsity (High Priority)

Definition: Data sparsity refers to the lack of sufficient user-item interaction data, which hampers the system's ability to generate accurate recommendations.

Impact: Collaborative Filtering models, which heavily rely on user interactions, often struggle with sparse data, leading to less reliable recommendations and limited personalization.

Methodologies: *Comparative Evaluation*, *User Acceptance Test (UAT)*

Solution/Strategy Flow:

Problem	Demonstrate that the dataset exhibits sparsity, a common issue in content with limited interactions (e.g., reviews without click-rate or like conversion data).
	Provide statistics on items/users with minimal interactions (e.g., number of users with fewer than 5 reviews).
	(Optional) Show the limitations of Collaborative Filtering through Leave-One-Out Cross-Validation or metrics like *Hit Rate* and *Top-K Evaluation*.
Solution	Illustrate how models like DSSM mitigate the sparsity issue by leveraging additional data or embeddings.
	Highlight improvements in performance metrics (e.g., accuracy, F1-score) after parameter tuning or using alternative loss functions.
	Demonstrate how the overall recommendation system improves despite sparse data.

2. Cold-Start Problem (High Priority)

Definition: The cold-start problem occurs when new users or items lack interaction history, making it challenging to provide personalized recommendations.

Impact: New users receive generic suggestions, and new items struggle to gain visibility, reducing user engagement and satisfaction.

Methodologies: *Comparative Evaluation*, *User Acceptance Test (UAT), Case Studies and Scenario Simulation (Optional)*

Solution/Strategy Flow:

Problem	(Optional) Relate to data sparsity statistics (e.g., users/items with zero interactions).

	Describe scenarios illustrating the user journey for new users or the introduction of new items.
Solution	Detail how solutions like assigning initial labels/interests or leveraging demographic data can improve recommendations for new users.
	Showcase the user journey improvements with tailored recommendations.
	(Optional) Provide updated statistics demonstrating increased engagement for new users/items (same set).

3. Long Tail Items (High Priority)

Definition: Long-tail items are less popular items with fewer interactions, often overlooked by recommendation systems.

Impact: Ignoring long-tail items reduces content diversity and user satisfaction, while perpetuating the Pareto Principle (popular items dominate recommendations).

Methodologies: *Comparative Evaluation*, *Case Studies and Scenario Simulation (Optional)*

Solution/Strategy Flow: describes what is used to boost unpopular items and how the performance is.

Problem	Explain the Pareto Principle and how it affects content recommendation.
	Highlight challenges in training models with long-tail items, such as biased negative sampling in DSSM or Self-Supervised Learning.
	Provide statistics showing the low exposure of long-tail items.
Solution	Outline strategies to promote long-tail items, such as tuning model parameters, using balanced sampling methods, or adding filtering layers.
	Present updated exposure statistics to demonstrate improved visibility for long-tail items.

4. Scalability (Medium Priority)

Definition: Scalability refers to the system's ability to handle a growing number of users and items efficiently without significant performance degradation.

Impact: As the platform expands, computational demands increase, potentially leading to slower response times and reduced performance.

Methodologies: *Comparative Evaluation*

Strategy Flow:

- Evaluate system response times and performance metrics as data volume increases.

- Incorporate scalability-focused techniques like caching, approximate nearest neighbor search, or distributed training. Compare system performance before and after scaling (Can use synthetic datasets to simulate the situation).

5. Diversity vs. Relevance Trade-off (Low Priority)

Definition: Balancing diversity in recommendations with relevance to user preferences.

Impact: Overemphasizing relevance leads to narrow recommendations, while focusing on diversity may reduce personalization.

Solution/Strategy Flow:

- Explore strategies to balance diversity and relevance, such as multi-objective optimization.
- Measure trade-offs using metrics like intra-list diversity and relevance scores.

6. Privacy Concerns (Low Priority)

Definition: Ensuring user data is handled responsibly to protect privacy while delivering personalized recommendations.

Impact: Mismanagement of personal data can lead to privacy breaches, loss of trust, and regulatory non-compliance.

7. Evaluation Challenges (Low Priority)

Definition: Difficulties in assessing recommendation algorithms accurately, particularly with limited ground truth or real-time feedback data.

Impact: Inadequate evaluation metrics may lead to deploying suboptimal models, reducing user satisfaction.

Evaluation Strategy: Two-Tier Evaluation

The Two-Tier Evaluation approach involves assessing your recommendation model using two distinct evaluation settings that reflect both its scoring (or prediction) capability and its performance in the production-style retrieval scenario. This dual evaluation strategy helps bridge the gap between offline model performance and real-world system behavior.

Tier 1: Prediction/Scoring Evaluation

Objective: To measure how accurately the model predicts whether a given user–item pair is positive (or the predicted rating). This evaluation is carried out on a controlled test set where you have ground-truth labels (e.g., positive/negative reviews).

Method:

Input	A set of user–item pairs (usually sampled or curated from historical interactions).
Process	1. Use the model to predict a rating or a probability score indicating the likelihood that the user will rate the item positively. 2. Compare the predicted values against the actual labels (positive or negative).
Metrics	***Accuracy***, ***Precision***, ***Recall***, ***AUC*** (Area Under the ROC Curve), *F1-**score***, etc.

Benefits:

1. Provides a clear picture of how well the model differentiates between positive and negative pairs.
2. Helps in diagnosing issues related to model training, feature representation, and classification/regression capability.
3. Easier to iterate on since the candidate set is smaller and the evaluation is straightforward.

Tier 2: Retrieval Evaluation (Production Simulation)

Objective: To simulate the production environment where the model is used to retrieve the top-k candidates (items) for a given user from a large catalog. This evaluation measures how effectively the model's embedding or similarity function surfaces items that the user actually likes.

Method:

Input	A user identifier and the entire (or a large subset of) item catalog.
Process	1. Retrieve the top-k nearest neighbor items based on the model's embedding or similarity score.

	2. Determine whether the retrieved items include those that the user has positively interacted with (as per historical data).
Metrics	1. ***Recall@k***: Fraction of the truly positive items that appear within the top-k list. 2. **Precision@k**: Fraction of the items in the top-k list that are truly positive. 3. **NDCG** (Normalized Discounted Cumulative Gain): To capture the ranking quality of the retrieved list.

Benefits:

1. Directly mimics the production scenario, allowing you to see how well the model performs when the retrieval task is constrained by real-world factors like runtime efficiency and large candidate sets.
2. Highlights potential issues such as the “needle in a haystack” problem, where even a good model might have low recall if only a very small subset of relevant items exists in a large pool.
3. Helps determine if the model’s training objectives align well with the retrieval task. If the model is only optimized for prediction but not for ranking, performance may suffer in this tier.

Why Both Tiers Are Important

Complementary Insights:

- Prediction/Scoring Evaluation reveals if your model can tell apart positive from negative interactions under controlled conditions.
- Retrieval Evaluation shows whether these predictions translate into effective ranking in a real-world scenario, where you need to quickly find a few relevant items out of many.

Training vs. Production Alignment:

- A model might score well in prediction because it was optimized to distinguish on a pairwise basis, yet it might not yield high recall when asked to rank an entire catalog.
- Running both evaluations helps identify if further tuning (or even a change in loss function, such as switching to a ranking loss) is needed to better align the model with production goals.

Feedback for Model Improvement:

- If performance in Tier 1 is strong but Tier 2 is lagging, you can investigate factors such as:
    - The quality and structure of your embedding space.

- The efficiency and accuracy of your nearest neighbor search (e.g., whether an approximate nearest neighbor method is causing retrieval errors).
- The need for post-retrieval re-ranking steps to boost performance.

Operational Considerations:

- Tier 2 evaluation forces you to consider runtime constraints and large-scale retrieval challenges, ensuring that improvements made during development translate into actual benefits when deployed.

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	Completed
1st Stage Testing & Evaluation Design	2 weeks	9/2	/
2nd Stage BE (Retrieval + Ranking)	*2 months*	*28/2*	/
1st Stage Models Optimization	2 months	28/2	/
1st Stage FE	1 month	16/3	/
2nd Stage Testing & Evaluation Design	2 weeks	16/3	/
3rd Stage BE (Ranking)	1 month	31/3	/
Final Report (1st half)	2 weeks	31/3	/
2nd Stage Model Optimization	1 month	13/4	/
2nd Stage FE	1 month	13/4	/
Features Add-on (Optional)	1 month	20/4	/
Final Report (2nd half)	2 weeks	20/4	/
Video Production	1 month	30/4 (TBC)	/

Check Trello for task detail and distribution.

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* (Structure DB), *Faiss* (Vector DB), *npy/pkl* (Simple Key/Index).
- 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 removed to allow flexibility.

5. **1st Stage BE (Retrieval) (Adjusted)**

- Designed and trained the minimum performance models for Item Collaborative Filtering (Item CF) and Deep Structured Semantic Model (DSSM).
- Integrated the two models into the CR system for qualitative assessment using Flask.