



Interactive Social Media Federated Recommendation System via Chrome Extension

Prepared By

Name: Odai Atef Atiya Abdelbaky

ID No.: 252000312

Submitted in fulfillment of the requirements for the degree Master of
Science in Artificial Intelligence

At

Collage of Informatics

Midocean University

UAE

Supervisor Name: Dr.Ashraf Alsaed

Collage of Informatics, Midocean University, Comoros

Acknowledgements

I am grateful to Dr. Ashraf Alsaed for his support and assistance in my thesis; I am thankful for the many ways in which he helped me achieve my goals. In every way that I can imagine, my thesis has benefited from Dr. Alsaed's support and assistance. I would also like to thank the faculty and staff at Midocean University, College of Informatics, for allowed me to conduct research successfully. The help I received throughout my master's program from them has been greatly appreciated. In addition to thanking Dr. Alsaed and Midocean University, I would like to take advantage of this opportunity to thank all who supported me, including my family, friends, colleagues and those who offered their time and energy by providing me with feedback. Last, I would like to thank my family for their love, support, and patience, and for being my biggest source of inspiration and motivation throughout my academic career.

Abstract

Social media sites create abundant, ever-growing flows of information, making it increasingly hard for users to assess information of interest to them efficiently when it aligns with their interests, goals, and information needs. Current recommendation systems within these platforms mostly rely on passive behavioral signals likes, clicks, scrolling behavior, and dwell time to learn about user preferences. While effective for maximizing user engagement to the platform, these systems lack transparency, build a limit on user autonomy, and offer little to no information on the actual reasoning behind recommendations. Conventional recommendation algorithms also remain closely tied to the social media platform, which stops users from maintaining a consistent preference profile across social media platforms. All these difficulties culminate to lead to information overload, diminished trust, privacy violations, and feelings of distancing between recommended information and users' interests.

This project showcases an alternative, user-directed recommendation framework that is delivered as a Chrome extension, enabling users to explicitly shape their content experience. Rather than depending on platform-native algorithms, the proposed framework allows users to manually annotate posts that they see from their social media feeds as either "**Recommended**" or "**Unwanted**." The user's annotations will form their own data set reflective of their own explicit preferences as opposed to inferred behavioral patterns. A simple ML model that has been locally repurposed via Fine Tuning with techniques such as low-rank adaptation (LoRA) has the ability to use these annotations to learn user-specific relevance patterns and anticipate likely user interactions with upcoming posts based on what users have shared previously. Additionally, since all fine-tuning has taken place on the user device, it helps protect the user's privacy by eliminating dependency on centralized collections of user data and dictating that sensitive information about user behavior should not be collected or shared. The cross-platform website uses a Chrome extension, which provides the ability to customize your experience on both platforms without needing to rely on Private API or Backend Service access. Instead, it relies only on publicly visible text the browser extracts. This model provides compatibility and transparency and is not ethically objectionable, unlike some paradigms based on surveillance-inspired tracking. The recommendation model is inherently dynamic, developing over time as users contribute more annotations, and will become more accurate and personalized while allowing users to retain agency over the recommendations.

The recommendation framework is able to surface relevant posts, deprioritize unwanted content, and provide a transparent alternative to conventional opaque recommendation models. In sum, this project puts forward a novel, privacy-centric, user-centered approach to content recommendation based on explicit feedback, transparency, and user agency. It combines interactive annotation, lightweight learning, and cross-platform functionality to arise at a viable and ethical approach to improve content consumption online when algorithmic filtering and information overload dominate the field.

Table of Contents

Acknowledgements.....	iii
Abstract.....	iv
List of Abbreviations.....	vi
List of Figures	vii
Chapter 1 Introduction.....	1
Chapter 2 Background and Related Work	3
Chapter 3 Proposed Approach (Methodology)	8
Chapter 4 :Experimental Results and Discussion	9
Chapter 5 Future Work.....	20
Chapter 6 Conclusion.....	21
References	22

List of Abbreviations

- API: Application Programming Interface
- CF: Collaborative Filtering
- CARS: Context-Aware Recommender Systems
- Chrome Ext.: Chrome Extension
- DB: Database
- DL: Deep Learning
- JSON: JavaScript Object Notation
- LLM: Large Language Model
- LoRA: Low-Rank Adaptation
- ML: Machine Learning
- MSN: Mobile Social Network
- MySQL: My Structured Query Language
- NLP: Natural Language Processing
- Node.js: JavaScript Runtime Environment
- OCR: Optical Character Recognition
- OSN: Online Social Network
- PEFT: Parameter-Efficient Fine-Tuning
- RS: Recommender System
- UI: User Interface
- URL: Uniform Resource Locator
- XLM-R: Cross-lingual Language Model, RoBERTa
- mT5: Multilingual Text-to-Text Transfer Transformer

List of Figures

- Figure 4.1 Overall architecture of the proposed recommendation system
- Figure 4.2 Workflow of the Chrome extension for post collection and annotation
- Figure 4.3 Backend API processing pipeline for receiving and storing labeled posts
- Figure 4.4 Data flow from social media platforms to backend API and MySQL database
- Figure 4.5 Structure of the labeled dataset collected from X, Facebook, and LinkedIn
- Figure 4.6 User interface of the Chrome extension showing “Recommended” and “Unwanted” labels

Chapter 1 Introduction

Since social media has become the main source of communication, entertainment, education, and news for most, users can no longer keep track of the sheer volume of posts that are produced every day. So to keep up with that, social media sites use recommendation algorithm-based recommendation engines to organize the content in user feeds and prioritize that feed based on the user's previous interactions. The recommendation algorithm-based recommendation engines are a significant portion of what users view and ultimately what they interact with.

1.1 Problem Definition

Although mainstream recommendation systems have increased user engagement, four significant shortcomings of these approaches exist. The first limitation is that the primary method for inferring user preferences is to use implicit behavioral signals (e.g., clicks, watch time, scrolling); however, these types of signals tend to be noisy and often easily misinterpreted. Secondly, because most mainstream recommendation algorithms are opaque (i.e., users do not receive any explanations about the content recommended), users are less likely to trust the algorithm and correct undesired patterns. Thirdly, most mainstream recommendation algorithms tend to optimize engagement for the platform (i.e., encouraging more time spent using the platform) instead of for goals of users, resulting in the emergence of irrelevant, repetitive, and/or sensationalized content in user feeds. Fourthly, most personalization models require extensive centralized tracking, raising significant privacy concerns, and restrict the learned preference profiles to the single platform where they were created, limiting personalizing experiences across platforms.

This research project attempts to solve this issue via user-directed recommendations via a Chrome extension. Users will explicitly label posts as “Recommended” or “Unwanted,” creating clear preference datasets. Using these explicit datasets, a lightweight model that runs locally on the user’s device will learn from user-generated feedback in order to make personalized predictions regarding future ranking and filtering decisions. By transferring control of personalized recommendations from platform-inferred data to user-annotated data and keeping the training local, this proposed system will provide a more transparent, private, and less overwhelming experience for users while providing more consistent personalized experiences across platforms.

1.2 Thesis Objectives

Develop a Chrome extension capable of overlaying annotation buttons (“Recommended” and “Unwanted”) on social media posts within supported platforms.

Collect labeled feedback from users to create a personalized training dataset reflecting individual content preferences.

Train and optimize a recommendation model that learns from these annotations to predict which future posts should be highlighted or hidden.

Implement local or lightweight model training to preserve user privacy and minimize data transfer.

Evaluate the system’s effectiveness in improving content relevance and user satisfaction through both quantitative (accuracy, precision, recall) and qualitative (user experience) metrics.

1.3 Thesis Contribution

This thesis contributes to the domain of social media recommendation systems as it describes a user-centered and privacy-preserving recommendation framework through the implementation of a Chrome extension. Typical recommendation approaches are largely based on implicit behaviour feedback. However, the proposed system...will enable users to provide explicit either a "recommended" or an "unwanted" label to every social media post, and this will create a much clearer and more transparent preference model. Furthermore, the use of lightweight...Low-Rank Adaptation (LoRA) techniques will enable efficient model adaptation to a user device without having to collect large amounts of centralised data. The cross-platform browser-based architecture described in this thesis will also allow the system to operate on publicly-viewable material and will therefore make the approach more flexible and independent from proprietary platform APIs. This thesis provides a practical, transparent and ethical alternative to traditional opaque recommendation systems through the combination of explicit feedback, local learning and cross-platform usability.

1.4 Thesis Organization

The dissertation is organized into five sections. The first section is very introductory: it presents the foundation of the research; describes the principal aims and objectives of the research; provides a list of the primary contributions the study will make; and summarizes how the material in this study is arranged or structured. The second section addresses the theoretical background associated with the study of recommender systems, social media personalization, browser-based interventions, and lightweight adaptation techniques such as Low-Rank Adaptation (LoRA), in the context of existing literature. The third section details how data was collected using a Google Chrome extension by outlining the overall system architecture and design of the extension; describing the data collection process; presenting the backend RESTful API used for data storage; and providing details of how the data is stored in a database. The fourth section provides a description of how the experiments were established, what datasets were used, how they were evaluated, and what results were obtained. The final section provides a summary of the results obtained in each chapter to provide insight into future directions for research.

Chapter 2 Background and Related Work

A survey of Campana et al.(2023) [1] reported on the different types of recommender systems (RS) available for Online Social Networks (OSN) and Mobile Social Networks (MSN) that make use of social relationships and user mobility to improve the design of modern RSs. First, the paper introduced four main approaches to RSs: Collaborative Filtering (CF), Content-Based Filtering, Graph-Based Approaches, and Context-Aware Recommender Systems (CARS). As these approaches depend solely on existing conventional interaction data, each suffers the common shortcomings of this reliance: scalability issues, cold start problems, and sparser feedback due to fewer interactions. The authors argue that social networking environments, such as OSNs, have the potential to significantly boost the quality of recommendations as they offer rich social signals, including friendship links/follow Markup tags, trust channels, tagging behavior, geographical contexts, and the availability of user-generated content. Additionally, they categorised the published literature on recommendation systems into the following system types: Friend Recommendations Based on Link Prediction, Social Trust Enhanced Item Recommendations, Tag Based Recommendations, and Geolocation Based Recommendations. MSN environments create a set of unique challenges for RSs relating to the decentralised nature of communication between devices, the intermittent nature of the connections between devices, limited device resource availability, and privacy issues; thus, RSs are required to work primarily with local device data resources and to rely on opportunistic communication to connect devices, rather than following a centralised model. The authors therefore reviewed multiple distributed and decentralised approaches to RSs including OppCF, MobHinter, differs, locPref, and p-PLIERS, illustrating how RSs may continue to be successful even with the absence of a centralised approach. Piccardi et al.'s (2024) [2] current study describes a newly developed methodology for conducting field studies of social media feed ranking using web browsers as a platform to conduct experiments and/or interventions. The impetus for this study is based upon the growing body of research regarding how feed ranking systems affect user well-being and the way in which users access information through their feeds, as well as the increasing number of restrictions placed on the access that platforms have to feed ranking and data selection systems. The authors have developed a methodology called a browser-based client-side 'reranking pipeline' that allows users to intercept the content of feeds as it is delivered to them, select which content they would like to see and apply the desired interventions to the selected content, thereby re-rendering the modified feed for their respective participants. The authors have also identified that there are three different types of intervention methods—namely, (1) downranking 'bad' content, (2) upranking 'good' and/or properly balanced content and (3) rewriting or editing content. Furthermore, the authors have discussed the technical challenges associated with implementation (e.g., inventory constraints, latency, consistency in handling user interface changes) as well as the ethical issues (consent, privacy and minimization of harm). The authors describe various ways to measure outcomes using a combination of survey data, passive behavioral metrics, and real-time ecological momentary assessments, as well as practical information about enrolling participants in the study, assigning participants to conditions, designing the interface, optimizing performance, and storing participant data securely. The study's conclusion is that extension-based reranking can facilitate ecologically-valid experimental research conducted in social media platforms, even if the social media company does not support such activity; therefore, the experimental designs used will still produce rigorous results for measuring the effect of interventions.

Hu et al.'s (2021) [3] study introduces LoRA (Low-Rank Adaptation) as a low-cost way of adapting a large number of language models (for example: GPT-3) for downstream tasks, without having to fine-tune/old model parameters for those tasks. The authors state that traditional fine-tuning for extremely large models (e.g., GPT-3) is costly in storage and computation, and that traditional fine-tuning methods (full tuning, adapter, prefix tuning) can add inference overhead (e.g., latency) and/or reduce available sequence length. To address these issues, LoRA uses the insight that many model fine-tuning updates are low-dimensional; therefore, if only low-dimensional updates are required to perform the task, then LoRA can be used without the need to fully update a model for all tasks, unlike adapter and prefix tuning. LoRA's method involves freezing the pre-trained weights and representing ΔW , the update, as a product of two small matrices (BA) whose rank is extremely low, generally 1 or 2. The only matrices trained will therefore be BA, which dramatically reduces the number of parameters that need to be trained as well as the amount of memory consumed. The authors of this article find that during their experimentation with RoBERTa, DeBERTa, GPT-2, and GPT-3, LoRA is able to reach performance levels comparable to or higher than that of full fine-tuning, using as much as 10,000x fewer trainable parameters. Furthermore, LoRA allows for the merging of the LoRA updates into the base weights without introducing any notable inference latency. The analysis also proposes that LoRA learns low-rank and meaningful adaptations that are primarily concentrated in the attention projection matrices (e.g., W_q and W_v) and that they are stable across multiple trials. The conclusions drawn indicate that adaptation can be effectively captured in a low-rank space. Overall, the paper establishes LoRA as a valid way of scalably and practically adapting large pre-trained models to new tasks in cases where compute and memory limitations are a primary concern. According to Kong et al.'s (2025) [4] article, utilizing a single instance of a LoRA Module across an entire community of LLM-Based Sequential Recommenders creates a high risk for negative transfer due to model and user differences throughout the entire lifetime of the recommender system. The authors go on to describe how the common LLM-SR process creates the optimal training pathways for the creation of end-to-end training data via PEF-trained LoRA Modules for prompt-texts by converting user histories into prompts and utilizing prompt-context and expected translation prompts as potential output hypotheses within a retraining LLM process; however, they also show that different clusters of behavior have a tendency to present issues in the manner that their respective gradients misalign and lead to conflicts that arise during the creation of an optimal single LoRA Module'. They approach the challenge of large-scale sequence recommendation by developing a single multi-task training system with LoRA. Model updates use an MoE structure with a single gating network, driven by what a user has selected from prior recommendations and captured through a recurrent model like SARPS. Users identify "sequences," or sets of recommendations based on the user's selected sequence, and each sequence identifies its expert "weight." A recommendation generated by the MoE can then be combined with the user's prior selections to update formulations of LoRA for the specific user and sequence. iLoRA provides users with greater accuracy of their recommendations than does LoRA using fewer than 1% additional trainable parameters. A scalable method of providing instance-wise expert activation will also be beneficial to the future of personalization and elimination of negative transfer within sequential recommendation.

In their article, Jannach, Naveed, and Jugovac (2016) [9] analyze how recommender systems can evolve from focusing on system-centered optimization to having user-centered control. The researchers argue past recommenders placed emphasis on either engagement or prediction accuracy and provided limited opportunity for users to direct their recommendations. They categorize different types of user control including editing preferences, providing critique, giving interactive feedback and using transparent features which allow users to view and/or change the

recommendation logic applied. This study indicates that greater user control positively correlates with improved perceptions of fairness, trust, and user satisfaction, particularly when the recommendation is of significant impact or personal sensitivity to the user. However, they also reveal several practical challenges surrounding the use of user control, e.g., high levels of user control may lead to increased cognitive load; poor user interfaces create confusion; , and trade-offs exist between having strict user control & maximum raw predictive performance regarding recommendations. A primary point is that a successful recommender system should have some balance between automaticity (the system performing tasks automatically) and providing real agency (control) to the user through low-friction interactions. This idea directly supports systems that collect explicit labels, such as recommended vs. unwanted, as this type of interaction allows users to help shape their personalization intentionally vs. simply by using many noisy, implicit behaviors.

Cramer et al. (2008) [\[10\]](#) look into how visibility affects users' levels of trust and acceptance with the use of recommended systems, as seen through experimental studies conducted with test subjects. They study one particular content-based recommended artwork system and test multiple types of visibility/information about how the user-provided data relates to the related items that were offered as recommendations. They found that providing this kind of visibility increases the user's ability to understand the logic used to create the recommendations, and consequently, it can improve levels of trust and perceived usefulness as well as the likelihood of using the system. Furthermore, this study reveals that visibility is not only a technical aspect of recommended systems but rather, it is part of the user-centered design and will influence how confident a user feels toward using the system and how much they will continue to use it. Additionally, using the criteria of how the quality of the visibility provided influences usability, too much complexity in the visibility may hinder usability while short-form, simple-to-understand visibility will assist users with their decision-making. The work is highly applicable to the area of user-directed recommendations where users will be asked for explicit feedback on how their content was ranked (given priority) and how they were filtered from other content. The authors acknowledge that trust-aware, transparent interfaces (or visibility) could increase the effectiveness of recommended systems beyond using solely their raw accuracy metrics.

Sets of studies have looked at how fairness and transparency are critical to recommendation systems from both the algorithmic point of view as well as from the user's perspective (Sonboli et al., 2021 [\[11\]](#)). In other words, they advocate a user-centric definition of fairness in a system, requiring that treatment, visibility and opportunity be aligned with a user's own perceptions and impressions of these characteristics in both the content that is presented and the creators that produce and distribute it. Many technical definitions of fairness can be used effectively but usually have limitations associated with the user's understanding of how they are treated within a given social context and by a given guideline/goal of the recommendation system (authors). The authors conclude that these technical definitions are limited in that they do not provide the user with sufficient information needed to interpret recommendation outcomes and ultimate fairness assessments associated with each outcome, and therefore, the authors encourage that developers incorporate "transparency mechanisms" and/or "explanation interfaces" to aid users in determining an equitable outcome associated with their recommendations would be. Finally, the authors explore the conflicting nature of personalization, diversity, and fairness and indicate that developing or optimizing one of these elements will negatively impact the other two. A significant contribution of the research will be to call for user-centred evaluation frameworks that employ both qualitative evidence gathered from user experience and useful quantitative indicators of fairness in support of developing fair and trustworthy recommender systems. In conclusion, this research emphasises the importance of ensuring the design of a trustworthy recommender

system includes attention to fairness and transparency so they reflect real user experience and not only technical evaluation measures used to optimise algorithms internally.

Wang et al. (2024) [12] provide a thorough review of horizontal federated recommendation systems that allow users to keep their local interaction data while training a collaborative model, including the architecture of core components (client-server coordination, gradient aggregation, personalization), the challenges of implementing federated systems (data heterogeneity, sparse feedback, client drift, communication overhead, and cold-start issue due to federated constraints), and privacy/securities issues related to federated systems (e.g., inference attacks and poisoning attacks) as well as the mitigation methods available (e.g., secure aggregation, differential privacy, and robust optimization). Furthermore, the authors introduce a taxonomy that connects algorithm design choices to the tradeoffs incurred when deploying those algorithms with respect to accuracy, efficiency, privacy and scalability. In addition, the authors note that federated recommendations need to not only match the performance of centralized recommendations but also support the real-world constraints of the devices where federated recommendations will be delivered, and that users can dynamically choose to participate in the federated model training. Therefore, these findings will be useful for recommendation projects that are concerned about keeping users' raw data private by using federated systems to build personalized recommendations and maintaining a high level of recommendation accuracy and reliability without storing any user data centrally.

The 2024 paper by Chronis et al [13] surveys the use of federated learning to create privacy-preserving recommender systems and assesses how utilizing distributed training approaches can lower the exposure of centralised data. The authors cover multiple types of commonly used recommendation approaches including collaborative filtering, neural networks (which include neural recommenders), and hybrid approaches — and identify how each one has been altered for usage in federated environments. The main implementation challenges highlighted by the authors include the presence of non-IID (independent and identically distributed) data for users, the potential for users to be unstable participants, the costs associated with communication, and the potential to converge models given limited resources on devices. The paper also discusses the limitations of multiple privacy-enhancing technologies, including but not limited to secure aggregations, differential privacy, and various approaches which use encryption, with focus on how they impact the utility of the resulting recommendation model and training efficiency. A significant contribution of the paper is the thorough analysis of the trade-offs that exist between recommendation quality, the depth of personalisation, and privacy guarantees. The authors argue that any real-world system implemented using federated learning for recommendations must seek to balance these three objectives relative to the contextual environment in which they operate versus maximising one objective at the expense of the others. As such, this survey will have very high relevancy to user-centric recommendation platforms as it provides guidance for making informed decisions about federated architecture design and implementing privacy controls that support the retaining of personalised value while not collecting raw behavioural data in a centralised backend.

According to Zhang et al. (2023) [14], Outlinings of privacy risks associated with federating recommendation systems through attribute inferring attacks demonstrated that when training federated recommender systems, federated training still puts private attributes of users at risk from being known to third parties. In this paper, the authors conducted a full-scale evaluation of potential attacks against recommendation systems under realistic conditions, and showed how attack methods exploiting shared gradients/models could expose private attributes about a user's demographic/behavioural characteristics. They also evaluated multiple defence techniques, including differential privacy, gradient perturbation and secure aggregation techniques, and

quantified utility/privacy tradeoffs for each technique. The authors provide empirical evidence that the risks associated with an individual's privacy while using a federated recommender system are not exclusively determined by whether the recommender is decentralised; instead, they also depend on model architecture, training methodology and adversarial capabilities. The authors stressed that any real-world implementation of a federated recommender system must have a threat model and a measurable privacy safeguard built into it from the outset of development. As a result, any project based on providing privacy-first recommendations would benefit from the knowledge that learning methods through federating do not inherently make user data secure; thus, robust defence mechanisms must be developed to ensure protection for end users while maintaining quality of recommendations.

Chapter 3 Proposed Approach (Methodology)

The implementation and methodology for creating an interactive social media recommendation system using a web-based application framework for users to label their own preferred social media items. This differs from most current recommendation systems that utilize only implicit user signals, such as click-through rates and how long users view items. The proposed interactive recommendation system will use explicit user feedback through classifications of recommended and unwanted traffic as the basis for providing more transparent, user-centered, and privacy-aware recommendations.

The framework created to implement this interactive recommendation system has three major components: a Chrome extension, a backend Node.JS application to handle all business logic, and a MySQL database for storing user data. The Chrome extension works on the social media platforms of Facebook, LinkedIn, and X. The purpose of this extension is to identify visible posts in the user's social media feed, extract relevant content from these posts, and provide a straightforward interface for the user to annotate whether each post is recommended or unwanted. The interactive recommendation system will capture explicit preference signals that will later be used to create personalized recommendations and filtered content.

Each time a user labels a post, an automated extension pulls relevant text snippets and/or metadata related to the post from the interface. This data includes the content of posts, where they came from (platform), a unique local identifier for each post and the label assigned (as well as when a user has interacted with the post). After a user has labeled the post, the extension sends those user-labeled posts to a back end API built on Node.js. The back end API validates and processes the records, then stores them to a MySQL database. The MySQL database is the main source for finding examples users have labeled, and provides the basis for future analysis, machine learning model development, and evaluation.

Based upon this workflow, the system has three key components. In the data acquisition stage, user-generated content (e.g., posts) is acquired from X, Facebook and LinkedIn using the Chrome extension. In the user annotating stage, users annotate the acquired post based on the post's relevancy. In the back end processing and storage stage, user-labeled posts are sent through the API and stored in the database. Using this process, user-specific data matching actual user preferences can be created, which can be done more accurately (due to direct interaction) than by traditional engagement data (i.e. impressions/click-through rates).

One of the significant advantages of this methodology is the identification of explicit user interaction as the means of produce data, rather than data being inferred through behaviour. This gives more transparency since the users are made familiar with how the data they provide is generated and used, in addition to enhancing the quality of the data since the data will be derived from intentional and interpretable feedback rather than from uncertain behavioural traces.

Having cross-platform functionality of the Chrome extension means that the identical process of annotating content would be applied to multiple social media platforms; therefore, an enhanced, versatile, and comprehensive profile of user preference could be developed by directly accessing the same authenticated database that stores the data produced through the Chrome extension.

The database of the data that is stored will be used to create or modify a recommended model that predicts whether or not a particular user will regard any newly encountered post as being relevant to or not relevant to them. Thus, the current implementation adequately provides the necessary systems to allow the development of an interactive and federated recommendation systems infrastructure. Collectively, the Chrome extension, database, and back end API comprise the

basic structure through which large amounts of explicit preference user generated data can be collected for the development of user specific recommendation systems research. This approach to collecting data and storing it also include a component that uses artificial intelligence, as defined by the technique of Low-Rank Adaptation (LoRA). By collecting labeled posts through a Chrome extension, we have access to a supervised dataset that can be used to train or adapt a recommendation model that incorporates the individual preferences of users. LoRA provides an efficient means of fine-tuning when compared to retraining a large language or text model, as it does not require a large number of parameters to be updated, but rather only requires that some portion of the model's parameters to be updated while the original model weight is fixed. This approach greatly reduces the computational resource requirements (cost/memory), and the time taken to develop the recommendation model, and as such is more feasible when developing personalized recommendations. Furthermore, in terms of scalability and being privacy-aware, the use of LoRA is appropriate in the context of this research, because of its ability to perform user-specific adaptations with limited resources. By continually collecting additional labeled examples, the recommendation model could be incrementally improved in its ability to make the distinction between content that is recommended, and content that is not wanted (i.e. spam), thereby increasing the overall accuracy and relevance of the recommendations being provided.

Chapter 4 :Experimental Results and Discussion

This chapter introduces the evaluation of the proposed federated recommendation system using an experiment. Evaluation of the system will be carried out both in terms of evaluating how well the ML model performs as well as verifying that the integration of all three components of the recommendation system (Model - Backend - Browser) works correctly. To do this, the model trained by the recommendation system was evaluated by classifying social media posts into two target classes: recommended and unwanted. Using downloaded datasets from Kaggle, and ChatGPT for generating realistic user behavior and label scenarios for testing and experimentation, social media posts were classified into post categories such as entertainment, sports, news, etc., and grouped into their respective thematic classes for personalized content recommendation. Thus controlled experiments could be performed to understand the recommendation system's reaction to different preferences of content and different patterns of labels. One of the design criteria that must be satisfied for the proposed system is that the output should consist of one or more final recommendation "scores" which are not merely hard class labels but also derived from the softmax probability of the corresponding class label being equal to 1 (TRUE). This makes the output much easier to interpret and more appropriate for real-world use, such as through a browser extension.

4.1 Experimental Setup

The system testing was accomplished through an experimental environment consisting of three main components; a backend API used for both training and inference, an interactive web browser extension that interacts with various forms of social media content, and a machine learning pipeline utilizing LoRA-based personalization in conjunction with transformer model(s). The aim of creating this complete end-to-end recommendation workflow as opposed to a single isolated classification model.

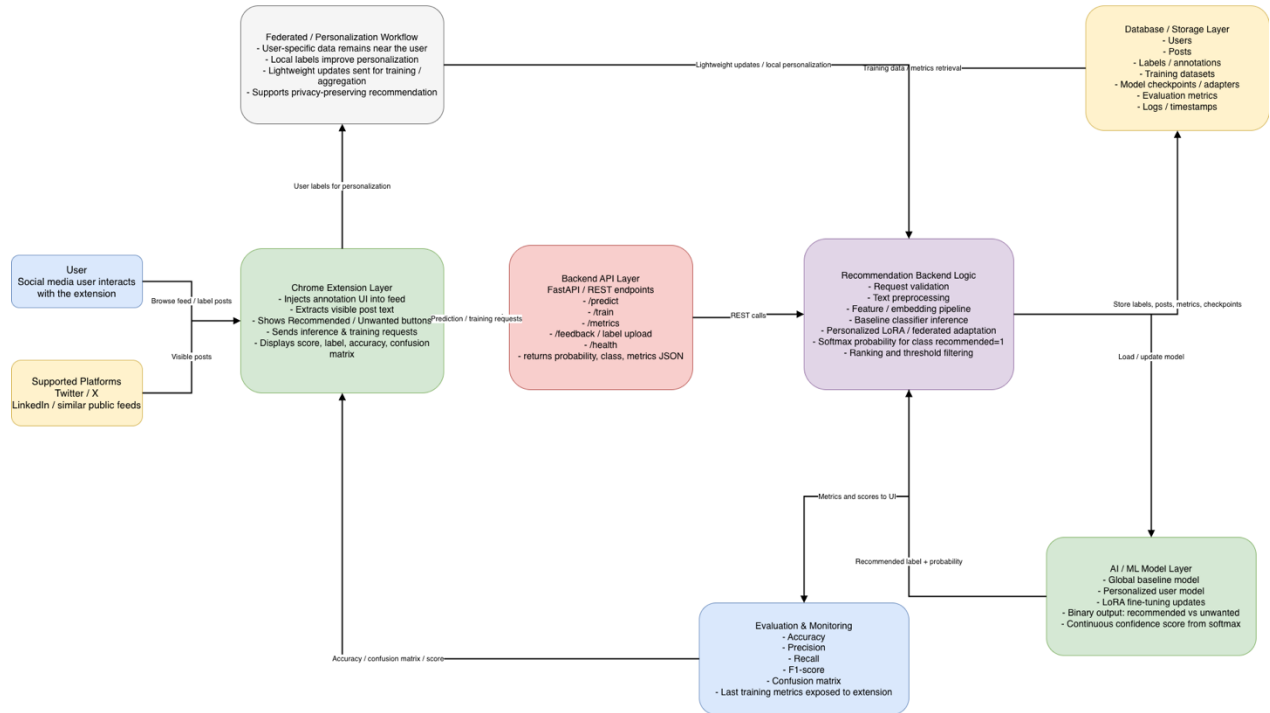


Figure 4.1 Overall architecture of the proposed recommendation system

4.1.1 Hardware Setup

The experiment was conducted on:

CPU: Apple M4 Pro

RAM: 24 GB

Operating System: MacOS Tahoe

Python Version: 3.9.6

Node.js Version: 24.13.1

Browser for Extension Testing: Google Chrome V146.0.7680.178

This hardware will ensure reproducibility and document the actual environment used for training, running the backend, and testing extensions.

4.1.2 Software Stack

The backend framework is FastAPI, Uvicorn API runtime server, Python programming language, Machine Learning Libraries (PyTorch, Transformers, PEFT, Datasets, Scikit-learn, NumPy, Pandas and Accelerate), JWT Authentication using python-jose, Password Security with passlib, SQLAlchemy Database ORM/Access Layer, MySQL Database Engine using PyMySQL, REST API API Style with JSON responses, Swagger Interactive Documentation via /docs. The extension type is Browser Extension; Manifest Version V3; Programming Language: JavaScript; Storage Layer: chrome.storage.local and Local Browser Storage; Frontend Interaction via Pop-up Based Interface With Content Scripts; Target Platforms: Facebook, X, LinkedIn.

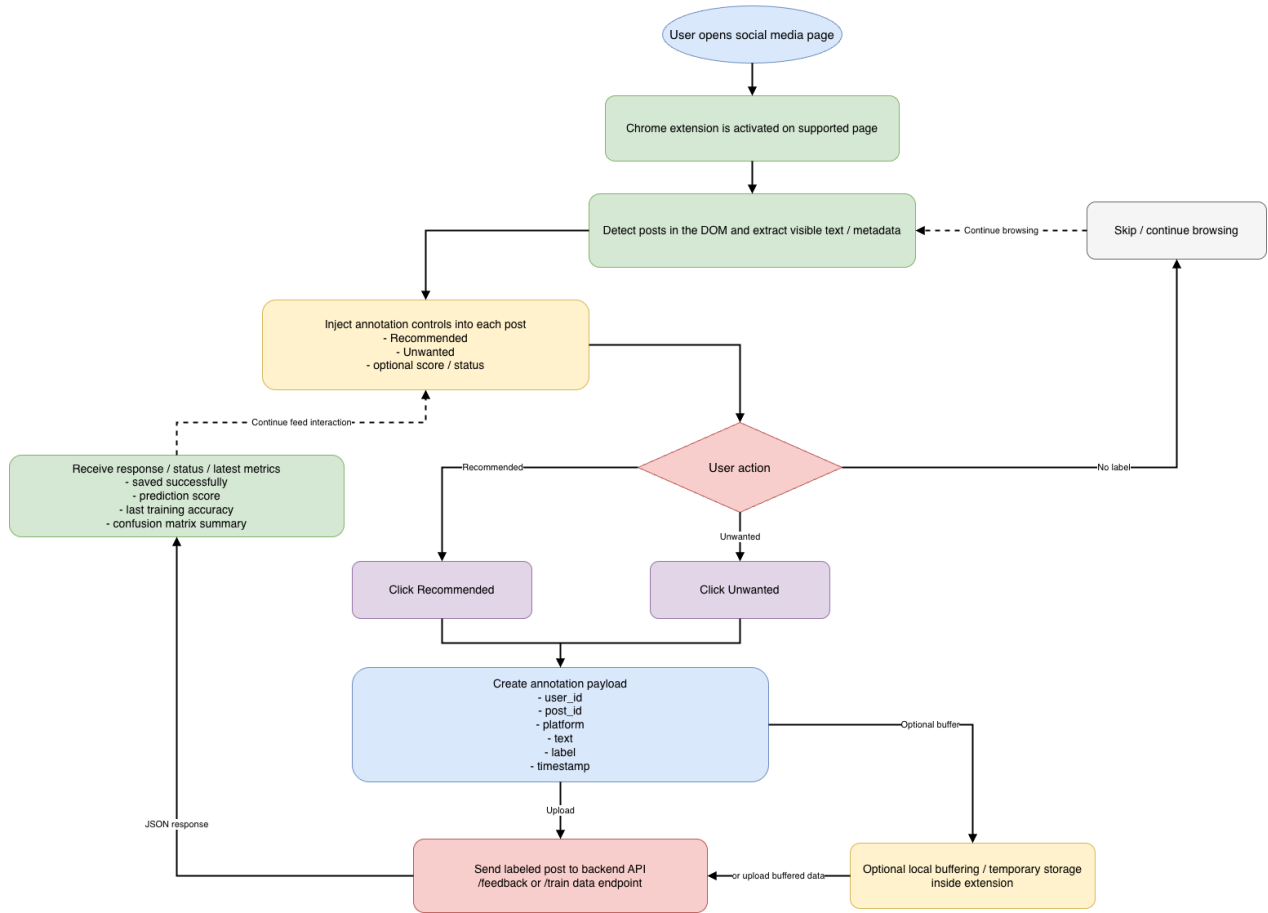


Figure 4.2 Workflow of the Chrome extension for post collection and annotation

4.1.3 Configuration of the Model

To personalize user responses using the backend training pipeline, a transformer-based sequence classification model with Low-Rank Adaptation (LoRA) was used.

Base Model Name: xlm-roberta-base

The training endpoint was configured with the following defaults:

- No. of Epochs: 3
- Batch Size: 8
- Learning Rate: 0.0002
- Maximum Sequence Length: 256
- The minimum number of samples needed for training: 20
- The maximum number of posts used to train with at a time: 1000
- Ratio of validation/test: 0.3
- Ratio of training/test: 70%/30%

In terms of user-specific adapter directories where users can train and use their own LoRA adapter separately.

4.1.4 Language of the Application and AI-Assisted Data Preparation

The application was designed to support the Arabic language. ChatGPT was also used during the experimental preparation phase to help simulate user preferences and user behavior scenarios.

This simulation helped produce more realistic labeled interactions and evaluate how the system would behave for users who prefer different category types.

The test post dataset used to create used to run tests was a collected dataset from Kaggle where posts were downloaded, categorized into classes (Recommended vs Unwanted) and then labeled. Categories of content that the dataset was used to classify include:

- Entertainment
- Sports
- News
- Fashion & Style
- Food & Cooking
- Travel
- Home & Decor
- Relationships
- Marketing / Promotions
- Entrepreneurship
- Career Advice
- Industry Insights

This created a data set that would allow for systematic assessment.

4.2 Backend Implementation and Evaluation

The proposed system's backend is essential because it processes the authentication workflows for the application, manages the collection of posts and their subsequent training, and generates metrics for both user inference and for each model when utilized. The backend was developed with FastAPI, and has a number of main application endpoints that include user registration, user log-in, collecting posts, training the models used, and inferring from the models.

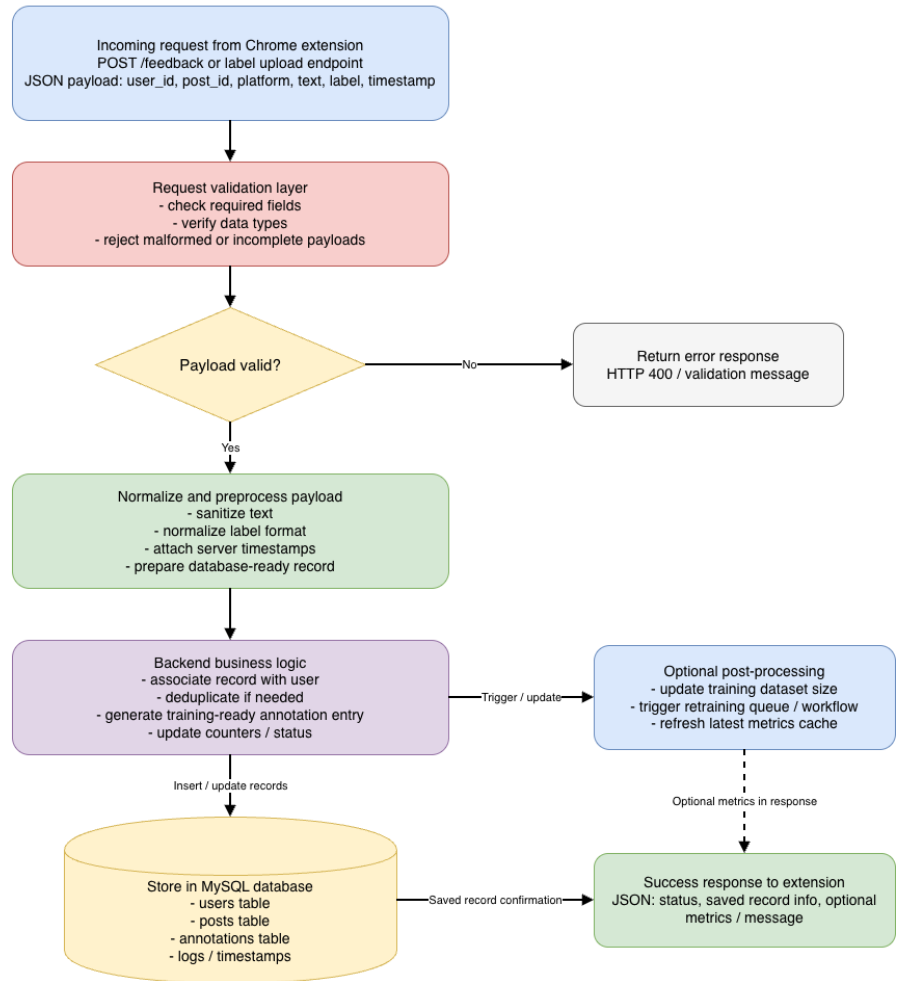


Figure 4.3 Backend API processing pipeline for receiving and storing labeled posts

4.2.1 The Backend Endpoints

The main endpoints used for experimentation purposes are as follows:

- /auth/register - Method for creating a new extension user.
- /auth/login - Method for authenticating and receiving the JWT access token for the user. Also returns the last training history if available.
- /posts - Method for posting new posts into the backend database with labels.
- /posts/stats - Method for returning aggregate statistics from the collected posts including aggregates and totals by individual platform type.
- /Train - Method for training a user-specific LoRA adapter utilizing the latest labeled posts and returning training metrics.
- /inference/batch - Method for running inference through a pre-trained (.LoRA) user adapter on a batch of pre-labeled posts for the user.

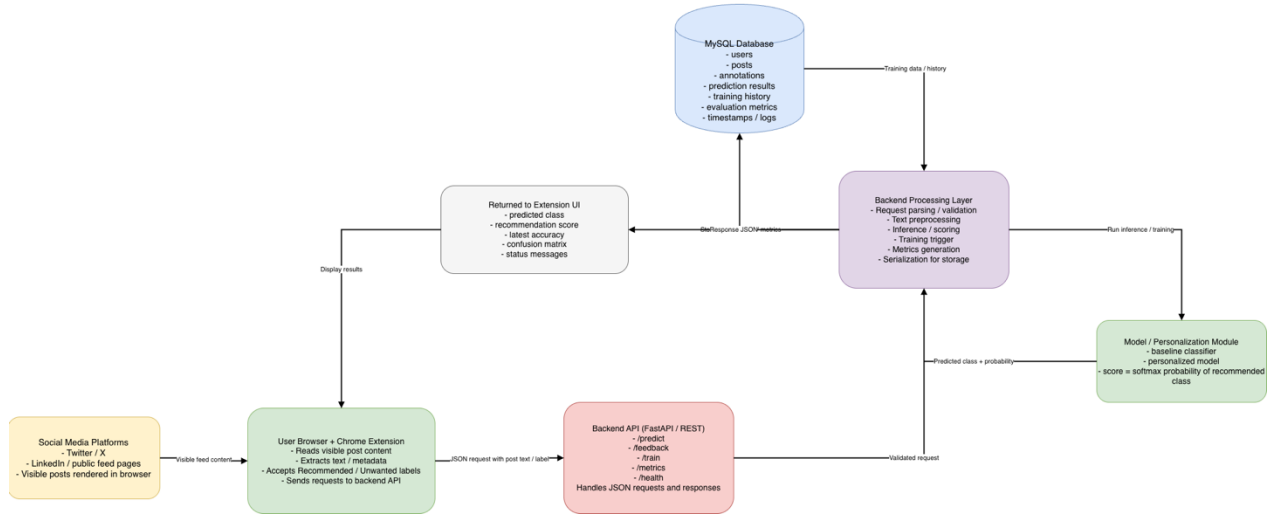


Figure 4.4 Data flow from social media platforms to backend API and MySQL database

4.2.2 Metrics Returned

The backend received a major improvement in that metrics returned from the /train and /auth/login endpoints are explicitly returned to the API consumer as key metric values in the context of training resources (i.e., above all user data at the top level of the API response):

Accuracy (training accuracy)

Precision / percision (precision score)

Recall

F1 (F1 score)

Confusion Matrix (confusion matrix)

4.3 Evaluating Performance of the Recommendation System

Evaluation of the recommendation system was done using conventional binary classification evaluation metrics: accuracy, precision, recall, and F1. In addition, a confusion matrix was used to give further insight into how well the model is able to identify elements of the positive or recommended class by distinguishing unwanted items from items that have been correctly predicted to be recommended items.

The backend calculates the confusion matrix as follows:

- True Negative (TN): Unwanted items that were predicted correctly are unwanted.
- False Positive (FP): Unwanted items that were incorrectly predicted as being recommended.
- False Negative (FN): Recommended items that have been incorrectly predicted to be unwanted.
- True Positive (TP): Recommended items that have been correctly predicted to be recommended.

The confusion matrix can then be exported to structured JSON format for ease of data analysis on the backend and for use in displaying the recommendation data on the frontend.

4.3.1 Recommendation Scores

The recommendation score was an important design decision by using the confidence level from the model for the recommendation class, where recommendation class = 1. Rather than using only the predicted class label, the use of the softmax probability for the positive (recommended) class as the recommendation score provides a continuous representation of the confidence level of the recommendation.

This provides several advantages:

- Improves the model's ability to explain its decisions.
- Allows for ranking and filtering based on thresholds, thereby allowing for better use of the recommendations.
- Provides a better output for the frontend for displaying the recommendation.
- Helps to differentiate between weak recommendations and strong recommendations.

4.3.2 Evaluation Metrics

Model Chosen: xlm-roberta-base

How Was the Model Trained: LoRA-based transformer fine-tuning

Where Was Data Obtained: Posts Downloaded from Kaggle + Simulated Preference Scenarios

Training Data Split: 70% Training & 30% Testing

of Samples Analyzed: 181

Accuracy: 87%

Precision: 80%

Recall: 100%

F1 Score: 89%

Confusion Matrix:

True\Prediction	Unwanted	Recommended
Unwanted	18	7
Recommended	0	29

4.3.3 Discussion of Model Results

An evaluation metric used to evaluate the performance of a recommender system is accuracy based on overall classification. An additional evaluation metric used to evaluate the performance of a recommender system is precision based on the reliability of positive recommendations.

Recall is used as a metric to evaluate how well the recommender system is identifying relevants in terms of recommended content. The F1-score has been used to report the overall performance of a recommender system. In terms of identifying whether a recommender system encourages over or under-recommendation of recommended content, the confusion matrix is useful. For example:

a high number of false positives indicates to users that they receive too many recommendations for unwanted items and

a high number of false negatives indicates to users that valuable recommendations were missed.

In recommender systems, analysis of this type is important due to how user experience is affected by not only raw accuracy but also by the nature of errors made when classifying items.

4.4 Datasets

This section describes the datasets utilized for the creation and assessment of a content only recommendation classifier (recommended and not recommended) throughout this project. Both public datasets available via Kaggle will be employed to establish, validate and evaluate the Language Understanding component, and a custom dataset classified by ChatGpt will be used as the primary data source for personalization. The datasets are listed below, with a description of the source they originate from and what fields are present in each dataset and how they work within the overall architecture of the recommendation system.

Public Datasets (Kaggle Sources):

The datasets provide a wealth of example text that was necessary for initial experimentation, testing of preprocessing and pre-training and/or warm-starting the text classifier on the datasets before it is further personalised with LoRA.

Platform	Dataset	Description	Ref
Twitter	Sentiment140	1.6M tweets labeled by sentiment (positive, negative, neutral). Can be used to pretrain your embedding and text classification components.	[5]
Twitter	Twitter US Airline Sentiment	14K tweets labeled for sentiment. Smaller and good for experiments.	[6]
LinkedIn-style text	Job Posts Dataset	Job posts with titles and descriptions — simulates LinkedIn-style professional text.	[7]
1.3M LinkedIn Jobs & Skills	Linkedin Jobs	This dataset contains 1.3 million job listings scraped from LinkedIn in the year 2024.	[8]

id	user_id	platform	text	recommended	created_at
66	140	3	X	0	2026-03-14 23:10:28
67	141	3	X	0	2026-03-14 23:10:28
68	142	3	X	0	2026-03-14 23:10:28
69	143	3	X	0	2026-03-14 23:10:28
70	144	3	X	0	2026-03-14 23:10:28
71	145	3	X	0	2026-03-14 23:10:28
72	146	3	LinkedIn	0	2026-03-14 23:10:28
73	147	3	LinkedIn	0	2026-03-14 23:10:28
74	148	3	LinkedIn	0	2026-03-14 23:10:28
75	149	3	LinkedIn	0	2026-03-14 23:10:28
76	150	3	LinkedIn	0	2026-03-14 23:10:28
77	151	3	LinkedIn	0	2026-03-14 23:10:28
78	152	3	LinkedIn	0	2026-03-14 23:10:28
79	153	3	LinkedIn	0	2026-03-14 23:10:28
80	154	3	LinkedIn	0	2026-03-14 23:10:28
81	155	3	LinkedIn	0	2026-03-14 23:10:28
82	156	3	Facebook	0	2026-03-14 23:10:28
83	157	3	Facebook	0	2026-03-14 23:10:28
84	158	3	Facebook	0	2026-03-14 23:10:28
85	159	3	Facebook	0	2026-03-14 23:10:28
86	160	3	Facebook	0	2026-03-14 23:10:28
87	161	3	Facebook	0	2026-03-14 23:10:28
88	162	3	Facebook	0	2026-03-14 23:10:28
89	163	3	Facebook	0	2026-03-14 23:10:28
90	164	3	Facebook	0	2026-03-14 23:10:28
91	165	3	Facebook	0	2026-03-14 23:10:28
92	166	3	Twitter	1	2026-03-14 23:18:49
93	167	3	Facebook	1	2026-03-14 23:18:49
94	168	3	LinkedIn	1	2026-03-14 23:18:49
95	169	3	Facebook	1	2026-03-14 23:18:49
96	170	3	Twitter	1	2026-03-14 23:18:49
97	171	3	LinkedIn	1	2026-03-14 23:18:49
98	172	3	Twitter	1	2026-03-14 23:18:49
99	173	3	Facebook	1	2026-03-14 23:18:49
100	174	3	Twitter	1	2026-03-14 23:18:49
101	175	3	LinkedIn	1	2026-03-14 23:18:49
102	176	3	LinkedIn	1	2026-03-14 23:18:49
103	177	3	Facebook	1	2026-03-14 23:18:49

Figure 4.5 Structure of the labeled dataset collected from X, Facebook, and LinkedIn

4.5 Conclusion

The results reported within this research indicate that the proposed system has successfully bridged the existing gap between the machine learning capabilities required to create a useful recommendation application and the actual practical usefulness of such applications. Rather than considering the recommendation model as being isolated from other components, this project demonstrated the end-to-end process by integrating the personalized training (of user-specific preference data), reporting of backend metrics, and frontend visualization into a cohesive environment for user-specific recommendation. From a model standpoint, the implementation of personalized user-specific recommendations using LoRA provides the means to effectively adapt transformer-based models, and therefore the recommendation results produced from those models, to the user's specific preferences. This will be especially beneficial for users who may interpret what is relevant (or undesirable) in very different ways but require the same base information within the recommendation process.

Furthermore, utilizing the softmax probability associated with a class of 'recommended = 1' to calculate the recommendation score provides a continuous measure of the confidence level (as opposed to a binary measure) associated with a recommendation. From a system architecture standpoint, the backend is critical to providing transparency and assurance to the platform. The backend has made the model and the resulting recommendations measurable and reproducible by explicitly providing the evaluation metrics that it produced, such as the accuracy, precision, recall, F1-score, and confusion matrix. This is particularly important in intelligent systems, particularly when the quality of the recommendations produced could potentially impact the level of trust that an end user has in the system. The user interface of the browser extension demonstrates that integration of recommendation transparency into the user-facing layer is possible. By showing users and evaluators the model's metrics, training history, and use of confusion matrix values as references for how well a recommendation has been made, the system offers users and evaluators greater insight into the system's operation than most AI-assisted applications, thus reducing the sense of ambiguity that exists with

many AI-assisted systems through the “black-boxing” effect of those systems. By implementing a full-stack architecture involving machine learning algorithms, backend engineering, and frontend usability, the system has made a significant contribution to the body of knowledge about operationalizing personalized recommendations. The combination of these three components of the system represents a much greater contribution to practice than the assessment of any individual components by themselves.

4.6 Summary of Examined Experimental Results and Discussion

The results of the experiments performed on the proposed federated recommendation system demonstrate promising predictions for all of the levels examined. The machine learning portion of the system provided the capability of providing useful and personalized classification through the use of LoRA adaptation to the xlm-roberta-base model. The backend infrastructure of the proposed federated recommendation system provided reliable training, storage, and delivery of metrics used to evaluate the accuracy of recommendations made by the recommendation workflow. Finally, the browser extension will display metrics generated by the above systems to the user and allow interaction with their personalized recommendation workflows across supported social media platforms. Having a controlled experimental framework was established using a combination of Kaggle based datasets; ChatGPT based user behaviours with the model to create labelled test scenarios across multiple different content categories: such as sport, entertainment or news... With this, the evaluation of the variability of how different recommendation decisions were made depending on the type of content was made possible.

The overall use of the softmax probability for ‘class recommended = 1’ for recommendation score. Helps to improve interpretability of model output as well as making the recommendation process more informative to the end user; and the elements used to achieve this (e.g. accuracy, precision, recall, F1-score, and confusion matrix) were included both in the backend responses as well as extension visualization - which also helps improve the transparency of the system.

In conclusion - the proposed solution suggests that an integrated personalised recommendation system can be built by integrating each component of the system (model evaluation, backend function and frontend display) in such a way that they reinforce each other; and that although there are still a number of large scale evaluations to be done, the evidence presented in this paper around the performance of the system as technically coherent, experimentally valid and build on to form future development and research is compelling.

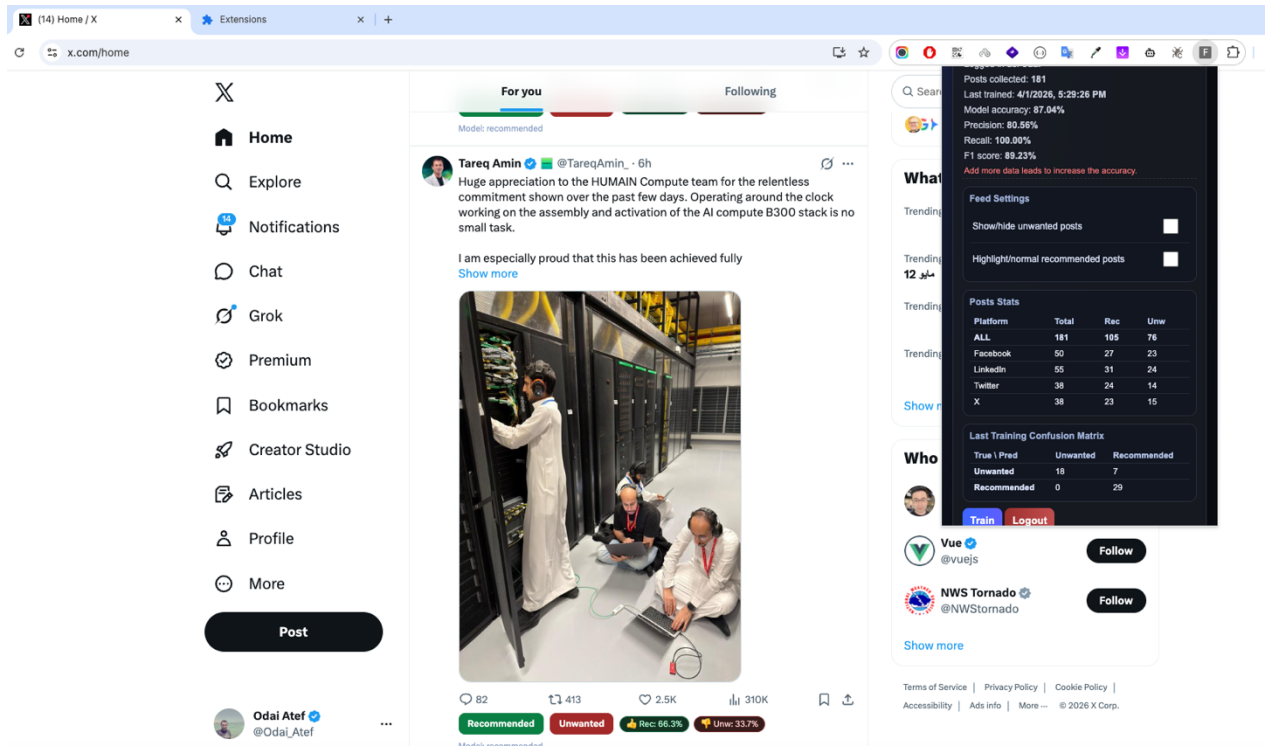


Figure 4.5 User interface of the Chrome extension showing “Recommended” and “Unwanted” labels

Chapter 5 Future Work

While this current implementation has successfully shown the ability to collect and store cross-platform labeled data, it serves merely as the first step toward a fully adaptive recommendation framework; currently, the system's focus is on creating the infrastructure for explicitly collecting data and for storing it in the backend. Future work will still expand this implementation into real-time recommendation generation, retraining, content re-ranking, and automatically predicting user preference through user input annotations. In particular, developing a recommendation model based on these collected annotations so that new posts can automatically be classified as either relevant or irrelevant will continue to evolve. There are many other potential improvements to make as well. The next phase of system development could involve expanding the number of social media platforms supported within the system, improving the ability to process multilingual texts, especially Arabic-language content, and expanding the recommendation pipeline to provide for handling multimedia formats such as video and images. Finally, further research on integrating federated or on-device learning methods to lessen the risk of privacy violations by reducing the need for centralized model training will be an important area of future work. Additionally, increasing transparency and accountability of the recommendation mechanism by providing users with explanations of why specific posts were suggested or filtered should be a part of the future development of this system in order to enhance users' confidence and trust in it and to increase the interpretation of the recommendations provided to them by the system.

Chapter 6 Conclusion

A framework on developing a user-centric and interactive recommendation for social networking platforms through a chrome extension has been designed in order to overcome numerous limitations of traditional recommendation systems: reliance on implicit behavioural cues; limited transparency; and a reliance on central collection of data. By offering users the ability to label posts as either "recommended" or "unwanted," A user would get a more transparent and personalised experience when it comes to content recommendations as opposed to currently available recommendations. The architecture consists of a chrome extension, Node.js based backend API, and MySQL database that collectively enables end-users and the platform to collect, send, store, and manage user preferences across multiple platforms. One of the key contributions of this research is the establishment of a practical framework for explicit feedback-based recommendation systems. This framework will allow end-users to have a more significant say in how their experience is shaped, rather than solely using engagement-based algorithms provided by the platform. In addition, the transparency of the recommendations provided to the end-user on the platform will be enhanced and a more useable and understandable preference dataset will be created for future use in developing recommendation models. The research also demonstrated the value of using multilingual AI models and lightweight adaptation techniques (i.e., LoRA) to help create efficient, scalable, and privacy-aware personalized recommendations, including those based on Arabic-language data. In conclusion , This Research Has demonstrated that Browser Extension-Driven Explicit Feedback Recommendations Are a Viable Alternative to Traditional Opaque Recommendations Systems. With the Combination of Cross-Platform Data Collection, User-Based Annotation, and a Lightweight Personalized Model, The Proposed Social Media Recommender System Has the Potential to Provide a More Ethical, Transparent, and User-Focused Approach to Recommendations for Online Content. Results Suggest That When Users Empowered to Influence Recommendations Directly, The Relevance and Trustworthiness of Online Content Will be Increased.

References

- [1] M. G. Campana and F. Delmastro, “Recommender systems for online and mobile social networks: A survey,” *arXiv*, 2023. [Online]. Available: <https://arxiv.org/abs/2307.01207>.
- [2] T. Piccardi *et al.*, “Reranking social media feeds: A practical guide for field experiments,” *arXiv*, 2024. [Online]. Available: <https://arxiv.org/abs/2406.19571>.
- [3] E. J. Hu *et al.*, “LoRA: Low-rank adaptation of large language models,” *arXiv*, 2021. [Online]. Available: <https://arxiv.org/abs/2106.09685>.
- [4] X. Kong *et al.*, “Customizing language models with instance-wise LoRA for sequential recommendation,” *arXiv*, 2025. [Online]. Available: <https://arxiv.org/abs/2408.10159>.
- [5] A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision,” Stanford Univ., CS224N Project Rep., 2009.
- [6] CrowdFlower, “Twitter US Airline Sentiment,” *Kaggle*, 2015. [Online]. Available: <https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment>.
- [7] K. Regmi, “Job Title and Job Description Dataset,” *Kaggle*. [Online]. Available: <https://www.kaggle.com/datasets/kshitizregmi/jobs-and-job-description>.
- [8] asaniczka, “1.3M LinkedIn Jobs & Skills (2024),” *Kaggle*, 2024. [Online]. Available: <https://www.kaggle.com/datasets/asaniczka/1-3m-linkedin-jobs-and-skills-2024>.
- [9] D. Jannach, N. Naveed, and M. Jugovac, “User Control in Recommender Systems: Overview and Interaction Challenges,” in *Electronic Commerce and Web Technologies*, 2016, pp. 1–21. [Online]. Available: https://doi.org/10.1007/978-3-319-53676-7_2
- [10] H. Cramer, V. Evers, S. Ramlal, M. van Someren, L. Rutledge, N. Stash, L. Aroyo, and B. Wielinga, “The effects of transparency on trust in and acceptance of a content-based art recommender,” *User Modeling and User-Adapted Interaction*, vol. 18, no. 5, pp. 455–496, 2008. [Online]. Available: <https://doi.org/10.1007/s11257-008-9051-3>
- [11] N. Sonboli, J. J. Smith, F. Cabral Berenfus, R. Burke, and C. Fiesler, “Fairness and Transparency in Recommendation [Online]. Available: <https://doi.org/10.1145/3450613.3456835>
- [12] L. Wang, H. Zhou, Y. Bao, X. Yan, G. Shen, and X. Kong, “Horizontal Federated Recommender System: A Survey,” *ACM Computing Surveys*, 2024. [Online]. Available: <https://doi.org/10.1145/3656165>
- [13] C. Chronis, I. Varlamis, Y. Himeur, A. N. Sayed, T. M. Al-Hasan, A. Nhlabatsi, F. Bensaali, and G. Dimitrakopoulos, “A Survey on the use of Federated Learning in Privacy-Preserving Recommender Systems,” *IEEE Open Journal of the Computer Society*, vol. 5, pp. 542–561, 2024. [Online]. Available: <https://doi.org/10.1109/OJCS.2024.3396344>

- [14] S. Zhang, W. Yuan, and H. Yin, “Comprehensive Privacy Analysis on Federated Recommender System Against Attribute Inference Attacks,” arXiv preprint arXiv:2205.11857, 2023. [Online]. Available: <https://arxiv.org/abs/2205.11857>
- [15] Odai Atef, “FedRec Chrome Extension GitHub repository, 2026. [Online]. Available: <https://github.com/Odai-Atef/fedrec-extension>
- [16] Odai Atef, “Federated Recommendation LoRA API GitHub repository, 2026. [Online]. Available: <https://github.com/Odai-Atef/fedrec-lora-personalization>