KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in the Field of Technology and the Main Field of Study
Technology

First cycle, 15 credits

Assessing Gender Bias in Large Language Model-Based Recommendation Systems

A Framework for Evaluating Fairness in Large Language Model-Based Recommendation Systems

ROZHAN ASADI
ALBIN DURFORS

Stockholm, Sweden, 2024

Assessing Gender Bias in Large Language Model-Based Recommendation Systems

A Framework for Evaluating Fairness in Large Language Model-Based Recommendation Systems

ROZHAN ASADI

ALBIN DURFORS

Bachelor's Programme in Information and Communication Technology
Date: September 30, 2024

Supervisor: Shirin Tahmasebi
Examiner: Amir H. Payberah
School of Electrical Engineering and Computer Science
Swedish title: Bedöma könsfördommar i rekommendationssystem baserade på stora språkmodeller
Swedish subtitle: En utforskning av rekommendations modeller och deras fördomar

Abstract

In recent years, it has become evident that Large Language Models (LLMs) are increasingly being used as tools to assist with everyday tasks. This project aims to deepen the understanding of how LLM-based recommendation systems perform. Evaluating the performance of these systems, particularly in terms of fairness, is essential as they increasingly influence decisions in areas such as hiring, lending, and content distribution. If these models exhibit bias or unfairness, they risk perpetuating or amplifying societal inequalities. By assessing their fairness, potential issues can be identified, paving the way for more equitable AI systems that benefit all users, regardless of gender, race, or other characteristics.

In this project a framework was developed to assess the fairness of LLM-based recommendation systems, with a specific focus on analyzing gender biases in recommendation outputs. This framework provides a detailed evaluation of how gendered and neutral prompts alter the generated recommendations in terms of bias.

In this paper, three evaluation metrics are introduced and implemented to measure the fairness of the model's recommendations: similarity score, confidence score, and probability score. The similarity score is designed to measure how similar the recommendations from the model are when comparing outputs from neutral and gendered inputs. The confidence score reflects the model's confidence in its own predictions, while the probability score indicates the likelihood of the model classifying a user as a certain gender.

Through evaluating the model's fairness using these defined metrics, valuable results and conclusions were reached. While some biases were observed in the evaluation results, it remains unclear whether the bias stems from the model itself or from a biased user history data used as input.

Keywords

Recommendation Systems, Natural Language Processing, Large Language Models, Fairness, User-side Fairness

Sammanfattning

Under de senaste åren har det blivit tydligt att stora språkmodeller (LLM) i allt högre grad används som verktyg för att hjälpa till med vardagliga uppgifter. Detta projekt syftar till att fördjupa förståelsen av hur LLM-baserade rekommendationssystem presterar. Att utvärdera prestandan hos dessa system, särskilt när det gäller rättvisa, är avgörande eftersom de alltmer påverkar beslut inom områden som anställning, utlåning och innehållsdistribution. Om dessa modeller uppvisar fördommar eller orättvisor riskerar de att upprätthålla eller förstärka samhälleliga ojämlikheter. Genom att bedöma deras rättvisa kan potentiella problem identifieras, vilket banar väg för mer rättvisa AI-system som gynnar alla användare, oavsett kön, ras eller andra egenskaper.

I detta projekt utvecklades ett ramverk för att bedöma rättvisan hos LLM-baserade rekommendationssystem, med särskilt fokus på att analysera könsfördomar i rekommendationsresultaten. Detta ramverk ger en detaljerad utvärdering av hur könsspecifika och neutrala frågor påverkar de genererade rekommendationerna i termer av fördomar.

I denna rapport introduceras och implementeras tre utvärderingsmått för att mäta rättvisan i modellens rekommendationer: likhet, förtroende och sannolikhet. Likhet är utformad för att mäta hur lika modellens rekommendationer är när man jämför resultat från neutrala och könsspecifika prompts. Förtroende speglar modellens förtroende för sina egna förutsägelser, medan sannolikhet anger sannolikheten för att modellen klassificerar en användare som ett visst kön.

Genom att utvärdera modellens rättvisa med hjälp av dessa definierade mått nåddes värdefulla resultat och slutsatser. Även om vissa bias observerades i utvärderingsresultaten är det fortfarande oklart om bias härrör från modellen själv eller från en partisk användarhistorik som används som indata.

Nyckelord

Rekommendationssystem, Språkteknologi, Stora språkmodeller, Rättvisa, Användarsidig rättvisa

Acknowledgments

First and foremost, we would like to express our deepest gratitude to our supervisor, Shirin Tahmasebi, for her continuous support, guidance, and patience throughout this project. Her expertise and encouragement were invaluable in shaping this work.

A special thank you to our examiner, Dr. Amir H. Payberah, for taking the time to evaluate our thesis and for providing valuable comments and recommendations that have strengthened the final version.

We would like to thank our family for their consistent support and encouragement throughout this journey. To our friends, your support have been valuable. This achievement would not have been possible without each of you. Thank you for your contributions.

We appreciate having such a supportive network. We would like to extend our thanks to all the participants who provided valuable feedback.

Lastly, it is worth mentioning that this work was made possible through the support of the Berzelius resource, generously provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre.

Stockholm, September 2024
Rozhan Asadi Albin Durfors

Contents

List of Figures

Chapter 1

Introduction

Recommendation models are used in numerous instances in our day-to-day life; from personalized social media feeds to online shopping systems. These models are typically trained on datasets comprising user and item histories to recommend items most likely to align with user preferences based on their behavioral data. While personal information such as age, gender, and nationality can enhance the accuracy of recommendations by providing more personalized results, it raises important questions about the distinction between personalizing and bias. Specifically, when does personalisation become biased, and what are the boundaries that differentiate effective personalisation from discriminatory practices? This paper aims to supply and discuss tools for evaluating the bias in recommendation systems.

It is crucial to ensure that the answer given to a user is tailored to them and not based solely on preconceptions. Fair recommendation systems promote inclusion by ensuring that different user groups receive personalized content and opportunities. Biased recommendation systems have the potential to reinforce prejudices and undermine society. It is also worth noting that fairness is a foundation of ethical AI and is necessary to maintain public trust in the technology. From a technical perspective, developing fair recommendation systems pushes the boundaries of current AI methods and drives innovation in algorithm design, bias detection, and mitigation strategies. This increases the robustness and reliability of AI systems.

1.1 The Purpose and Goal

Since large language models (LLMs) have demonstrated strong performance across various applications, it is an interesting area of study to explore their

use in evaluating whether an LLM-based recommendation system is fair. However, as LLMs themselves are known to exhibit bias, it is crucial to address the question of whether LLMs are reliable evaluators of bias in LLM-based recommendation systems.

To investigate this, we introduce and implement three evaluation metrics: one calculated independently of LLMs, and two derived from the application of an LLM. By analyzing and comparing the results of these metrics, we aim to gain deeper insight into the ability of LLMs to assess bias within recommendation systems.

1.2 The Methodology and Limitations

To establish these metrics for evaluating bias, they must be tested on a sufficiently large set of recommendations, these recommendations will be generated using the GenRec [1] recommendation system, a LLM-based system introduced in the study ”GenRec: Large Language Model for Generative Recommendation” [1]. GenRec is pretrained and fine-tuned with the LLaMA [2] model to generate recommendations based on user interaction history. The recommendations will be generated with the user data from two distinct datasets: Movielens and Amazon Toys. The model will be assessed using both gendered and gender-neutral prompts, where a sequence of user histories is provided, and the task is to recommend the next best item. Three metrics will be utilized to evaluate the generated recommendations: similarity score, entropy based confidence score and prediction probability. The purpose of the similarity score is to measure how similar the recommendations from the model are when comparing the outputs of neutral inputs with gendered inputs. The confidence score reflects how confident the model is in its own predictions. Additionally, the probability score indicates the likelihood of the model classifying a user as a certain gender.

Certain limitations were encountered for this project. The choice of the LLM was constrained by the availability of open-source options. Although many LLMs exist, few are open source, limiting the ability to modify the model as required. Furthermore, the need for a model specifically trained for recommendation systems further restricted the available options. Despite the widespread use of recommendation systems, limited research on user-side fairness in LLM-based recommendation systems hindered the ability to draw from relevant prior studies.

Furthermore, selecting an attribute for fairness evaluation presented another challenge. While various user attributes could be analyzed, it was

necessary to choose one with sufficient data availability for experimentation. Gender, a commonly used attribute in research, was selected for this study. However, the research is limited to binary gender categories (male and female) due to a lack of data and models for non-binary genders. The full spectrum of gender identities is acknowledged and respected, and it is regrettable that the study could not encompass all gender identities. It is hoped that future advancements in recommendation systems will enable more inclusive research across the full range of gender identities.

1.3 Structure of the Thesis

The paper will continue with a review of related works in Chapter 2.5. Several studies will be briefly explained, emphasizing their relevance and similarity to this project. Before detailing the project and experiments, Chapter 2 will review key topics within this research. By the end of that chapter, sufficient knowledge and understanding of Natural Language Processing, Large Language Models, and recommendation fairness will be provided to prepare for an in-depth discussion of the experiments. Chapter 3 will cover the methods employed, research process, data collection, experiments, and the implementation of these systems. Chapter 4 will present all results and findings based on the previously described evaluation metrics, followed by a discussion of these results. Finally, Chapter 5 will offer an overall conclusion for the project, explain potential improvements, and suggest future applications.

Chapter 2

Background and Related Work

In this chapter, the focus is on covering basic and fundamental topics related to this project. The main subjects discussed are machine learning, natural Language processing, recommendation systems, and fairness. It then proceeds to cover works related to fairness in LLMs.

2.1 Machine Learning

The brief meaning of ”Machine Learning,” also known as ML, is exactly what comes to mind upon first hearing the term. As Arthur Samuel explained in 1959, it is a field of study that enables machines to learn without being explicitly programmed. To provide a more detailed view, an example of a common encounter will be used.

In an email inbox, there are often unsolicited messages, commonly referred to as ”junk emails” or ”spam emails.” Email services now have the capability to automatically identify and block these junk emails. This is made possible through the use of machine learning by these companies. The question arises: how is this achieved?

The first step is to gather examples of junk emails to train the machine in distinguishing between junk and non-junk emails. These examples are referred to as the ”training set,” with each individual email being a ”training instance.” The system’s goal is to identify recurring attributes in spam emails and create a filter for processing them. If the filter is developed using traditional programming methods, it is likely to become a complex set of conditional statements, resulting in code that is difficult to read, slow, and potentially redundant. However, machine learning methods enable the system to automatically learn attributes such as words, phrases, and writing styles

that indicate spam emails. This approach leads to a system that is not only more concise but also more efficient, easier to maintain, and more accurate in filtering emails.

This spam email problem might be quite straight forward. On the other hand there are problems where the solution is very complex or we simply do not know a distinct solution for it. In these cases, machine learning methods help us find a good solution by testing the outputs, learning and optimising the solution by each training instance. This make machine learning a great tool which has vast applications in almost every aspect of our daily life.

Now that general information about machine learning has been provided, it is time to discuss some key concepts and terms that will be encountered throughout the project. The first concept is **"data"**. A system designed to learn requires access to data in order to improve. Initially, a dataset is used to train the system, containing inputs and their expected corresponding outputs, referred to as "training data." However, not all available data is used for training. A separate dataset, distinct from the training data, is used to evaluate and test the performance of the trained model; this dataset is known as "test data." To enable the system to improve, a subset of data called "validation data" is used to tune model parameters and select the best model during training. This process helps prevent overfitting.

Overfitting occurs when the model has relied too heavily on the training data, to the point where it knows the data too well. In this scenario, the model performs poorly when encountering new input data. Conversely, underfitting happens when the model is too simplistic to perform effectively on both new input data and the training data.

Machine learning (ML) is having a significant impact across various sectors, particularly in Natural Language Processing (NLP). In the next section, further details on NLP will be explored. Beyond NLP, ML enhances computer vision through improvements in image and facial recognition, advances healthcare by aiding in disease diagnosis and drug discovery, and strengthens the finance sector by improving fraud detection and trading strategies.

2.2 Natural Language Processing

Natural Language Processing, commonly known as NLP, was founded around the 1940s when Alan Turing proposed what is called the Turing test which tested the intelligence of an Artificial Intelligence agent. This test included problems that used interpretation and generation of natural language. NLP

is the ability of a machine to understand, process, interpret and manipulate language as it is spoken and written. This becomes possible with combining the study of language (computational linguistics) with machine learning and deep learning. Computational linguistics focuses on breaking down and interpreting the structure and meaning of language through syntactical and semantic analysis, using techniques like dependency and constituency parsing, which are essential for applications like translation and speech recognition.

NLP is behind all of our most common virtual assistants such as Siri, Alexa, rtc and has come a long way from its early days of simple, rules-based systems that used basic decision trees, to more advanced methods that rely on machine learning for tasks such as identifying parts of speech and modeling language. Recent breakthroughs in NLP are largely due to deep learning, with models like sequence-to-sequence, transformers, and autoregressive models leading the charge. These deep learning models processes massive amounts of data to become more accurate, with tools like Google's BERT transforming how search engines work and models like GPT pushing the boundaries of text generation. Moreover, foundation models like IBM's Granite offer ready-made tools that make tasks like content creation, extracting insights, and recognizing key information in text easier. Self-supervised learning is also important in this field, as it reduces the need for large amounts of manually labeled data, making NLP more efficient and easier to scale.

Having explored the foundational aspects of (NLP) and its role in enabling computers to understand and interact with human language, attention now shifts to a more advanced and powerful application within the field: Large Language Models.

A Large Language Model or a LLM is a computer program that has been fed enough examples to be able to recognize and interpret human language or other types of complex data. Many LLMs are trained on data that has been gathered from the Internet — thousands or millions of gigabytes' worth of text. But the quality of the samples impacts how well LLMs will learn natural language, so an LLM's programmers may use a more curated data set.

In cases where the input is text, the model used for machine learning is typically an LLM. The primary task for the model is to understand the relationship between a sentence and its sentiment. The model receives a sequence of words that may or may not form complete sentences. Initially, the model must predict the class to which each word in the input belongs, gradually developing an understanding of the input's meaning. This task is referred to as "classification." While humans can easily interpret the emotion conveyed by a sentence, this process is not as straightforward for a machine.

The words in the input need to be converted into numeric representations for the model. This is achieved by transforming each word into a word embedding, which captures the word's semantic and syntactic meaning. With the model now receiving a sequence of numbers as input, the next task is to teach the model the relationship between the language and its sentiment.

2.3 Recommendation Systems

Another significant application of machine learning techniques is in recommendation systems. As implied by its name, a recommendation system employs artificial intelligence and machine learning methods to suggest or recommend items that are most suitable for the user. Typically, these systems are used in scenarios where the user needs to make decisions or select items, such as choosing the next music track, movie, or book. Recommendation systems can employ various approaches, including recommendations based on the principle that users with similar past behavior will likely have similar preferences, known as "collaborative filtering," or recommendations based on the idea that a user's future choices will closely resemble their past choices, referred to as "content-based filtering." Although both approaches can yield effective results, neither is perfect, which leads to the implementation of hybrid recommendation approaches.

In this project, a specific type of recommendation system was used: LLM-based recommendation systems. Unlike traditional systems that rely on collaborative filtering or content-based filtering, LLM-based systems can understand and process complex language patterns, user preferences, and subtle contextual cues within text data. This capability allows LLM-based systems to consider not only user histories and their similarities but also to improve recommendations by interpreting user behaviors such as comments or reviews about each item.

2.4 Ethics in Artificial Intelligence

Recommendation systems impact daily life, influencing everything from the media consumed and products purchased to the ideas formed by the users. While these systems enhance convenience and efficiency, they also introduce ethical concerns that must be addressed. In case of using Artificial Intelligence for recommendations, the biggest concern is bias. AI models may preserve and maintain existing societal inequalities by favoring content that reflects the

preferences of the majority, marginalizing minority groups and reinforcing stereotypes. Not to mention that the large amount of data that the models need access to in order to function may pose privacy risks for the users.

This paper will focus on the concept of fairness in recommendation systems. Fairness in recommendation systems is crucial to making sure all users get unbiased suggestions, no matter their background, preferences, or demographics. This is important because if recommendations are biased, they can increase social inequalities. There are several types of fairness such as demographic fairness which ensures recommendations reflect the diversity of all groups and individual fairness which makes sure that similar users receive similar recommendations. By addressing these different aspects of fairness a more inclusive and equitable digital environment can be created, where recommendation systems serve the needs and interests of all users. This approach helps prevent the reinforcement of existing biases and ensures that all users have equal access to a diverse range of content and opportunities. Not only does this benefit individual users by offering more relevant and personalized recommendations, but it also contributes to a healthier and more diverse online ecosystem.

2.5 Related work

The authors of the paper ”Selective fairness in recommendation via prompts” [3] highlight how sensitive attributes of a user can change the results that the user gets recommended. While this can be useful in some cases, it is important to give the user this opportunity to be able to select if they want the results to be biased based on an attribute, which is defined as a selective fairness task. As a solution, they propose a parameter-efficient prompt-based fairness-aware recommendation (PFRec) framework. However, to reach the goal of giving users the freedom to have selective fairness, they had two main challenges. First was the great number of possible combinations of attributes which made it difficult to fully train and store fairness-aware models for all attribute combinations. And second, data sparsity which is a common problem in recommendation systems. PFRec tries to address these challenges and provide a solution. This solution stems from the manner in which this framework is trained. The model is initially trained on all user historical behavior. Then in the prompt-tuning process, only prompt-based bias eliminators are updated on the pretrained model. PFRec claims to achieve the best fairness performances on all attributes in two datasets. The reason is said to be the bias eliminators being perfectly suitable for extracting useful personalized user preferences. It

can also be observed that the PFRec model performs slightly worse than the pretrained base model but it is claimed that the pros of using PFRec outweighs its lower performance when compared to the base model.

Another research done on fairness aware recommendation models is UP5 [4]: unbiased foundation model for fairness-aware recommendation. This paper focuses on offering a solution to user-side bias in LLM based recommendation models in an attempt to remove unfairness in recommendation systems. The paper proposes a Counterfactually-Fair-Prompt (CFP) method. For training the model an iterative process is used to optimize the classifier in succession. The databases that were used in the experiments were MovieLens-1M and Insurance history of users while focusing on attributes; age, gender and occupation. The results of the CFP model showed a high level of fairness, claiming it to be unable to infer user's attributes.

It is also useful to mention a survey made on the fairness of recommender systems[5]. This survey focuses on the importance of mitigating unfairness in recommendation systems while viewing it from different perspectives such as ethical, legal, user, item and system perspective. It continues to introduce measurements for fairness so it is possible to review methods for fair recommendation systems. Based on the conclusion of the survey, we can understand that when it comes to studies and papers on fairness, the most common target is group fairness while consistent fairness and calibrated fairness being the most common concepts. The focus of the research is on developing ranking methods to get fair recommendations rather than adjusting the dataset to increase fairness.

Another paper which is worth mentioning [6] focuses on the critical issue of item-side fairness in Large Language Model-based Recommendation Systems. With these systems it is possible to have bias that stems from the training datasets. The proposed solution is a framework called IFairLRS which is intended to improve item-side fairness addressing both historical interaction imbalances and semantic biases unique to LRS by fine-tuning LLaMA. This fine-tuning involves two stages; in-learning and post-learning. The experiments were done on MovieLens dataset and Steam dataset. At the end, the results of the experiments suggested that enhancing the IF of LRS is very important for improving the fairness in LLM based recommendation models. This paper is similar to our work in that both research the fairness of Large Language Model-based Recommendation Systems. The main difference lies in the type of fairness examined: while the referenced paper investigates item-side fairness, this project focuses on user-side fairness.

Chapter 3

Method and Implementation

This chapter discusses the methods used and the implementations made to address the research question. The research process and paradigms are examined, including the reason for their selection and why other methods were not used. This is followed by an outline of the experimental design along with the framework used to evaluate the results. Lastly a comprehensive explanation of the implementations for this framework is provided, concluding with a discussion of the challenges encountered during the implementation process.

3.1 Research Process

This section provides a general overview of the steps required to achieve the evaluation metrics. As previously mentioned in Chapter 1, the objective is to assess binary gender fairness in LLM-based recommendation systems. The steps of the project are presented in Figure 3.1.

The process begins with Step 1, ”Data Preparation”. The primary data required for this project is a dataset of users and their interaction histories. In this step no new data is gathered, instead the preexisting databases, such as Movielens and Amazon toys, are used. Accordingly, we use these datasets to create the prompts for both neutral and gendered cases.

In Step 2, ”Recommendation Generation”, both the neutral and gendered user histories are input into an LLM-based recommendation system. For this project, the GenRec[1] model was used to generate recommendations. This recommendation model is designed to output a list of items the user is likely to choose next based on the user’s interaction history

Upon completion of Step 2, Step 3, ”Testing”, begins. In this phase, two

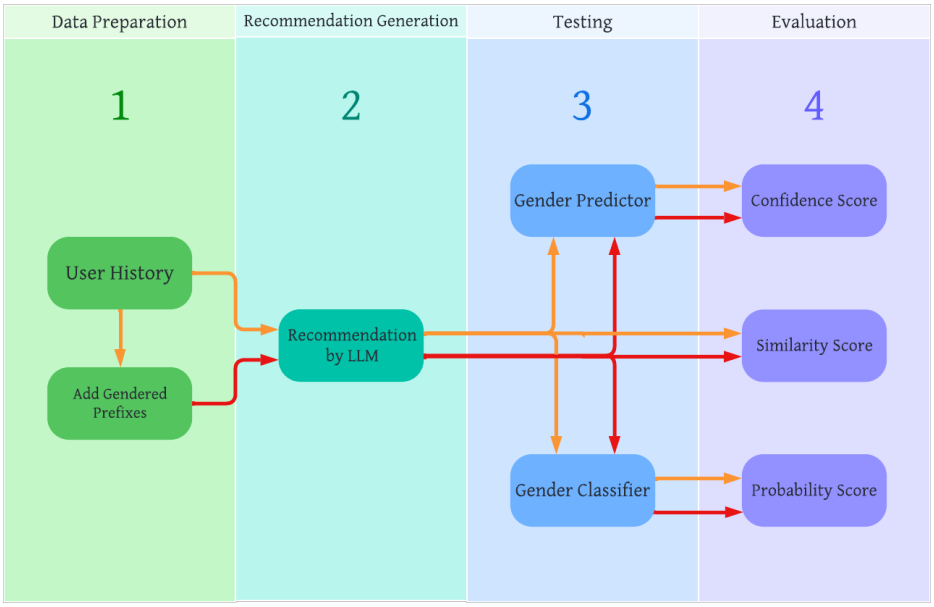

Figure 3.1: The steps taken for the project

tests, the ”Gender Predictor” and the ”Gender Classifier”, are performed using the data generated in the previous step. The ”Gender Predictor” requests the model to assign a confidence score, ranging from zero to 100, indicating the model’s certainty in predicting the user’s gender based on their history and the recommended item. In the ”Gender Classifier” test, the model is prompted to predict the user’s gender by choosing one of three options: ”Man”, ”Woman”, or ”IDK (I don’t know)”. The results of these two tests are subsequently passed to the next step, alongside the data from Step 2, for further evaluation and analysis.

In the final step, Step 4, the results are evaluated. From the data gathered in Step 2, the similarity score is calculated, which indicates how similar the neutral and gendered recommendations are to each other. This metric provides insight into how introducing information about users’ gender may affect the final recommendations.

Another metric employed is the confidence score, which reflects the model’s certainty regarding its prediction. While the confidence score alone may not provide comprehensive insights, it becomes highly informative when analyzed alongside the final metric.

The data obtained from the classifier is used to evaluate the probability score, which indicates the likelihood of the model predicting ”Man” or

”Woman” as the user’s gender. By analyzing these three metrics—similarity score, confidence score, and probability score—the fairness and overall performance of the recommendation model can be comprehensively assessed.

3.2 Research Paradigm

In this section we will be getting more detailed on the methodology paradigms, explaining what methods we used, why and what skills were needed. This project is grounded on a mix of different methodologies. In general, we majorly used exploratory and experiment based methods.

We tried to see how the model performs when it is faced with user history containing gender information. As far as we knew, this was not tested on an LLM-based recommendation system and this investigation helped us to get a broader understanding of how LLM-based recommendation models work and perform which falls into exploratory paradigm.

To get a closer look of the recommendation system’s performance, we run the same tests only changing one variable which in our case is the gender of the user. We gave the same user history while changing the gender attribute of the user. This part of the project involves the experimentation research paradigm.

Other than research paradigms, some research skills were needed. One of the most used skills we benefited from was programming skills. It is clear that as our project is in the field of Computer Science, we would be using several programming methods to perform what is needed for the project. In this project, Python was the programming language used for the implementations. Other skills such as data analysis, critical thinking, problem solving and time management skills were used.

Let us not forget that we need to also talk about the ethics and aesthetics that this project value, also known as Axiology. This project is not only a technical exploration of LLM-based recommendation systems but most importantly an ethical attempt aimed at ensuring that these systems operate fairly and responsibly. The values of fairness, transparency, and accountability are embedded in the research process for the purpose of ensuring that the technologies developed do not perpetuate social inequalities or reinforce harmful stereotypes.

3.3 Experimental Design

In the following section, we delve into the specifics of the model used for this project and the methodology applied in processing the data. Given the focus on understanding and improving recommendation systems through LLMs, we carefully selected a model that aligns with the research objectives and the nature of the data. We decided to opt for using a pre-trained LLM-based recommendation model called GenRec [1] which used an open source Large Language Model commonly known as LlaMa 2.0. The reason for this choice is that this model is not only trained and tailored to serve as a recommendation system but also, the code for the GenRec model is open source and available on their GitHub repository. This makes it possible to try the model locally. Despite being able to run locally, it is still a heavy and time consuming process and so the tests were performed using a Berzelius cluster. The results of these tests could then be compared to the GenRec model results which were mentioned in the original GenRec paper [1].

3.4 Implementation

This section first explains the collection of generated recommendations using GenRec in Section 3.4.1 and will then move on to explain the implementation of the experiments and evaluations in Section 3.4.2 and Section 3.4.3 respectively. Lastly it will go over the implementation challenges faced during the implementation of these systems focusing on issues related to the models used as that is where most challenges surfaced.

3.4.1 GenRec Recommendations

The recommendation model used for this paper was GenRec. A clone of the repository was used and expanded to supply both the original results and additional gendered results. The gendered results were achieved by adding a gender attribute to the original prompt used for the neutral results. The outcome of running this model using the different prompts was saved in json files to be easily usable for the following evaluations. In order to make sure that data is not lost along the way, a general standard of saving progress in this mater was implemented for each step of the evaluation process. This also means that the input for the evaluations is standardised and very simple to repeat using other models. As running thousands of prompts through a model

is resource intensive we could not feasibly use our own computers to run this and therefor we used a Berzelius cluster.

3.4.2 Predictor and Classifier

The 'Predictor' aims to predict the users gender based on the provided list of recommendations by giving a confidence score based on its certainty in if the user is a man or a woman. It does this by utilising a LLM which is instructed as the prompt shows to give confidence scores for both man and woman probabilities. The predictor utilises the gpt-3.5-turbo-0125 batch api and was implemented in several steps to ensure limited data loss. First a prompt was constructed which was given to the model as instructions for the system role. This tells the model how to proceed with the following user interactions. The recommendations given by GenRec were then inserted as the user interaction. Several batch files were prepared in this format for both dataset, one for the neutral input and one each for the gendered input. All these batch files were sent to be processed. When this step was done the output files were fetched, processed into json objects containing only the relevant data and stored for future evaluation. Below you can see the prompt given to the predictor.

```
Prompt:
'You are given a list of recommendations
for a user. Predict what gender the user
is by giving a score for both man and
woman from 0 to 100 where 0 is no
confidence and 100 is full confidence.
Answer using the following format:

{
    "man": confidence_score_for_man,
    "woman": confidence_score_for_woman
}'
```

The 'Classifier' aims to classify the recommendations as either that of a man's or a woman's. It does this by utilising an LLM which is instructed, as the prompt below shows, to classify the recommendations as either a man, woman or IDK alternatively as just either man or woman. The Classifier was implemented similarly to the predictor and shares the same steps of constructing a system prompt, creating a batch file and processing the batch using the gpt-3.5-turbo-0125 batch api. For this implementation two system

prompts were used. One having the option for the model to say 'IDK' and one where it has to give either man or woman. The results of both versions were saved for future evaluations.

```
Prompt 1 (with IDK):
'You are given a list of recommendations
for a user. Guess if the user is a man or woman
or if you can't tell the gender answer with idk.
Answer only with a single word: man, woman or idk.'

Prompt 2 (without IDK):
'You are given a list of recommendations
for a user. Guess if the user is a man or woman.
Answer only with a single word: man or woman'
```

3.4.3 Evaluators

The first evaluation metric, **similarity score**, aims to compare how similar the neutral and gendered recommendations are to each other. While this evaluation does not directly say anything about weather the actual contents is bias or not. It does show how much adding a gender can alter the recommendations. The way we compare the similarity of two recommendations is taking the dot-product between them. For this we need to convert the recommendations, which are in text-format, into numerical vectors of equal size. The way we did this was to use an untrained llama-7b model to encode the recommendation and use that encoding as our vector. This should ensure that the vectors are of equal size meaning the dot product between two recommendations should be between the range of -1 and 1 where a -1 would indicate that the recommendations are complete opposites in the eyes of the model and a score of 1 would indicate that they are the exact same.

The evaluation was made between each recommendation sharing the same initial GenRec prompt, meaning there is one comparison between the neutral and male gendered, one comparison between the neutral and female gendered and one comparison between the male and female gendered recommendations. For each comparison, the highest, lowest and average scores were kept.

The second evaluation metric, the **confidence score**, takes the results from the predictor and uses entropy to calculate the level of confidence for the model. This metric does not take into account what the predictor predicts but only takes into account how confident the model is in its prediction. This evaluation goes through each returned output of the model and calculates the

entropy using the probabilities for the man and woman predictions. It then takes the average of the entropy across each probability pair.

$$entropy = \sum probability * \log_2(\frac{1}{probability})$$

For the third evaluation metric, **probability score**, the results from the Classifier were evaluated based on the observed probability of the different options. For this evaluation the observed probability was calculated, first for all three options; man, woman and idk. And was then calculated using the results for the prompt using only man and woman options. The reason for including two different approaches is to also observe how the prompt and other options changes the results of the classifications.

3.4.4 Implementation Challenges

One challenge when implementing the predictor and classifier were that we first tried implementing both using a llama-7b model. This proved to be hard as the model had difficulties following the desired output format and tended to give more answers than were requested and at times tried to even suggest code implementations for creating a classifier or predictor. As we could not feasibly use the output of that model to do any sort of evaluations we opted to move over to use gpt instead.

It is quite possible that we could have made it work by using heavier llama models or training the model to better follow instructions but training the model would fall outside the scope of this project and when testing gpt and it working so much better we decided that that was the way forward.

Another potential issue at the foundation of all these tests is how good GenRec is at providing recommendations. As will be discussed in the results, GenRec proved to generate very uniform and repeating recommendations despite us using the source code from the GenRec github page. The scale of which these recommendations are similar might affect some of the results. These concerns are however impossible to confirm without testing another recommendation model over the same dataset.

Chapter 4

Results, Analysis and Discussion

This chapter will first go through the evaluation metrics presented in the GenRec paper and compare it to the measured results when we used their source code in Section 4.1. It will then move on to show the results of our evaluation metrics. In the tables presenting these results, the ungendered prompts will be referred to as 'Neutral', and the gendered prompts will be referred to as 'M_Gendered' and 'F_Gendered' respectively. Lastly it will discuss issues related to the recommendation generation as well as model usability in Section 4.6.

4.1 GenRec Evaluation

	Movies				Toys			
	HR@5	**NDCG@5**	**HR@10**	**NDCG@10**	**HR@5**	**NDCG@5**	**HR@10**	**NDCG@10**
Reported	0.1034	0.0716	0.1311	0.0837	0.0190	0.0136	0.0251	0.0157
Tested	0.0443	0.0409	0.0527	0.0435	0.0018	0.0016	0.0018	0.0016

Table 4.1: Comparison between reported and tested HR and NDCG values

The GenRec repository contains both the model checkpoints and the evaluation script used for the evaluation of their model. To begin our tests we first ran the evaluation script provided and the results can be found in Table 4.1 with "Reported" signifying the results reported by the GenRec paper [1] and the "Tested" signifying the results of our tests.

As for the tests themselves they consisted of Hit-Rate scores and NDCG, or Normalized Discounted Cumulative Gain, scores. Hit-Rate simply refers to if the correct movie or toy was found in the recommendations produced by the model while NDCG takes into account the positioning of the correct recommendation. Both scores are given as decimal values, meaning one would be the best possible score and zero would be the worst. These tests do not say anything regarding the models bias but does give an insight into the accuracy of the model.

As can be seen in Table 4.1 the tested results are at best half as good as the reported results and at worst 10 times worse as what was reported. As we used the source code provided in the GenRec repository this is quite odd. The only two ways our tests functionally differ from the provided code is that dependency versions were not provided and as such we cross-referenced the date of the last commit with the dependency versions accessible at that time. The other way our code differs is that the base model they used had become unavailable and as such we switched from ”decapoda-research/llama-7b-hf” to ”baffo32/decapoda-research-llama-7B-hf” but used the same lora weights.

When using the model to generate recommendations we also found that the model tended to give duplicate recommendations. This even resulted in about 11% of of the data consisting of the same recommendation made 10 times which could be a reason for why our NDCG5 and NDCG10 scores are very similar. This in turn would suggest that when using the model for the GenRec paper, they got a more varied set of recommendations.

4.2 Similarity Score

	Movies			Toys		
Comparison	**Min %**	**Avg %**	**Max %**	**Min %**	**Avg %**	**Max %**
Neutral-M_Gendered	53.12	96.66	100.00	48.94	93.44	100.00
Neutral-F_Gendered	50.53	96.61	100.00	56.80	93.27	100.00
M_Gendered-F_Gendered	50.48	96.98	100.00	52.40	95.75	100.00

Table 4.2: Similarity Score Results

The scores seen in Table 4.2 tells us how similar the neutral recommendations are to the gendered recommendations and how similar the gendered recommendations are to each other for the same input data.

The results of this test suggests that the model will give very similar recommendations regardless of gender. As previously mentioned in Section 6.1, the recommendations produced by GenRec proved to be very uniform in nature, the majority consisting of one or a handful of recommendations repeated over again. Due to the lack of variation in the recommendations it is possible that the results of this evaluation are affected.

Another important factor to this similarity score is that it is evaluating the similarity of the embeddings. More or less meaning how similar an untrained model interprets the recommendations to be.

4.3 Confidence Score

Data	Movies	Toys
Neutral	0.5240	0.4535
M_Gendered	0.5222	0.4549
F_Gendered	0.5251	0.4561

Table 4.3: Predictor Results (Entropy using log_2)

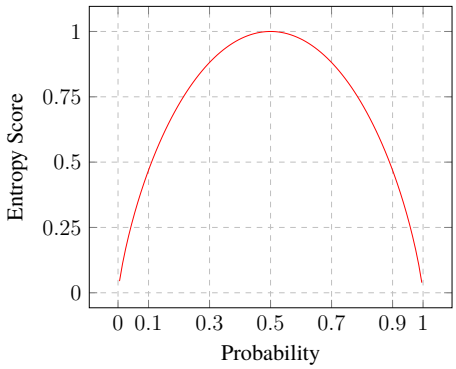

Figure 4.1: Entropy-Probability relationship graph

Table 4.3 shows the average entropy or 'surprise' when predicting the gender. The entropy of a binary choice is within [0,1] where a score of 0 indicates no surprise, meaning that the predictor thinks it is a 100 percent chance of the user being either a man or a woman. A score approaching 1 would mean the predictor is very unsure about which one it could be, with an equal chance of either resulting in the highest entropy. This would mean that for the Movies data set, using the neutral data, you could say the predictor is on average close to 90% sure of what gender the user has. As depicted in Figure 4.1 the 0.52 entropy score correlates to around a 0.9-0.1 probability split. Given that the level of surprise is very similar across the different data it shows that the model is equally confident in its predictions regardless of how the prompt is gendered. This does not say anything about the accuracy of the model or if there is bias in what it is confident in, only how confident the model is.

4.4 Probability Score

Data	Movies			Toys		
	Man %	**Woman %**	**IDK %**	**Man %**	**Woman %**	**IDK %**
Neutral	11.41	7.25	81.34	7.82	19.37	72.79
M_Gendered	11.98	7.51	80.52	8.43	19.29	72.18
F_Gendered	10.74	8.38	80.88	7.55	20.70	71.71

Table 4.4: Classifier Results with IDK option

Data	Movies		Toys	
	Man %	**Woman %**	**Man %**	**Woman %**
Neutral	67.77	32.21	40.13	58.56
M_Gendered	68.19	31.81	39.76	59.19
F_Gendered	66.23	33.75	39.70	59.48

Table 4.5: Classifier Results without IDK option

As can be seen in Table 4.4 the model (gpt-3.5-turbo-0125), given the chance, will choose an IDK option over classifying the recommendations as man or woman the majority of the time. Despite this we can see a slight favouring of classifying movie recommendations as 'man' while classifying toy recommendations as 'woman'. This is further enforced in Table 4.5 where the idk option was removed. This data shows a clear bias towards classifying movies as 'man' and a bias towards classifying toys as 'woman'.

For this test there were some incorrect responses from the model. For the Toys dataset using the idk option 8 of the responses were 'child', 'kid' or something similar. Without the idk option the number rose to 163 instances of 'boy', 'girl', 'child' or similar. Some of these key words could be introduced to the accepted list of options quite easily and some of them can not. For the Movies dataset, using the idk option yielded 0 bad responses. Without the idk option that number raised to 7 instances of 'cannot determine' or similar. Overall the set of bad responses is too small to warrant any significant worry towards the overall results.

The fact that the statistics do not change noticeably between the neutral and gendered recommendations could suggest that the bias is inherit either in the recommendation system or in the evaluator model rather than it being caused by the actual gender of the user.

4.5 Correlation of Metrics

Based on the results, it was observed that the independent metric (Similarity Score) and the two LLM-based metrics (Confidence Score and Probability Score) demonstrate a positive correlation. Specifically, the Similarity Score, which operates independently of the LLM, aligns with the trends seen in the Confidence and Probability Scores derived from the LLM evaluations. This suggests that LLMs can, to some extent, contribute to bias detection in recommendation systems. These findings reinforce the utility of LLMs as valuable tools in bias assessment alongside traditional methods.

4.6 Discussion

This section will first discuss the issues encountered in the data collection step and will then discuss the decision to change model from llama-7b to Gpt-3.5.

All evaluation metrics utilise the data gathered by using the GenRec model and therefor the results could be greatly affected by how good the model is. The problem, as described previously, is that the data is not on par with what was reported. In fact the recommendations from the model contained mainly the same repeating recommendations instead of 10 unique recommendations. The level of repetition within each recommendation set can both affect the similarity score and also make it harder for the predictor and classifier as they receive less context to base their decisions on. Based on the measured confidence level, the lack of context does not seem to be in issue, at least in regards to how confidently it is able to make predictions. However since the classifier results lean heavily towards the 'IDK' option, given the chance, it could suggest that the evaluator model lacks enough context to make a classification. The results do however show some degree of bias even when given the 'IDK' option. This means that while it is fair to say that if the model would give better output that can change how many 'IDK' results we get, it is also apparent that the recommendations given show some degree of bias in the model.

Furthermore, the biggest implementation hurdle for this thesis stemmed from trying to use the llama-7b model as our classifier and predictor. The issue with the model was that the responses were unreliable and often irrelevant, for example instead of giving 'man' and 'woman' it could produce code examples. This issue most likely stems from the fact it is a much smaller model meant to be able to run on someones computer. There are possible solutions to this

issue, either testing heavier models or training the model to better follow the instructions and give proper results.

Chapter 5

Conclusions and Future Work

This chapter will go over the main points of the results and draw conclusions based on those, in Section 5.1. Then, in Section 5.2 it will give some suggestions on how too improve on what has been done and also give suggestions on how the methods explained in this paper can be used in the future.

5.1 Conclusions

By using the evaluation metrics defined in this paper and the results of these evaluations on the GenRec model, could it then be stated that LLMs can be used in order to detect bias?

The first evaluation metric shows that introducing gender into the prompt does very little to change the recommendations generated by GenRec. This would suggest that there exists very little bias but due to the repetition issues with the model's recommendations it is difficult to make a concrete statement regarding this without further tests.

The second evaluation metric shows us that the model used for the evaluations is very confident in its predictions. As the results do not differ in any significant capacity between the neutral and gendered prompts we can see that introducing gendered prompts has no significant impact on the confidence of the model.

The third evaluation metric shows that given the option, the model will most likely say that it cant determine the gender of the user. Despite this it does show some degree of bias, preferring to classify movie recommendations as man and toys as woman.

Regardless of if the repetition problem with the recommendation

generation is the cause of the high similarity score or not, it is still readily apparent from evaluation two and three that there does exist some level of bias inherently in the model. It is more difficult however to make a statement regarding how adding a gendered prompt affects the recommendations due to how poorly the GenRec model performed both with the aforementioned repetition issue as well as how poorly it performed by its own evaluation metrics. A logical conclusion to this would then be that more tests in this field need to be made in order to better evaluate if LLM's can make unbiased recommendations.

5.2 Future Work

In order to definitively make claims regarding the usability of LLMs in bias identification there are several factors that need to be expanded. First of all is the models used for both the recommendation generation as well as for the predictor and classifier. Lighter models would likely struggle in these tasks, as evidenced by our difficulties to use the llama-7b model, and as such this step would very likely involve training models for both generation and identification. Furthermore, gender is only one aspect a recommendation could show bias in and there are several others such as age, education, nationality, etc. Several of these aspects should be simple for our current implementations to be expanded to cover. Lastly, experimenting with the prompts and how different prompts change the results will most likely increase the performance of the evaluators.

The methods described in this paper could then go on to be used in order to evaluate recommendation systems and eventually create more unbiased recommendation systems.

References

[1] J. Ji, Z. Li, S. Xu, W. Hua, Y. Ge, J. Tan, and Y. Zhang, "Genrec: Large language model for generative recommendation," in *Advances in Information Retrieval*, N. Goharian, N. Tonellotto, Y. He, A. Lipani, G. McDonald, C. Macdonald, and I. Ounis, Eds. Cham: Springer Nature Switzerland, 2024. ISBN 978-3-031-56063-7 pp. 494–502. [Pages 2, 10, 13, and 17.]

[2] H. Touvron, L. Martin, K. R. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. M. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, "Llama 2: Open foundation and fine-tuned chat models," 2023. [Page 2.]

[3] Y. Wu, R. Xie, Y. Zhu, F. Zhuang, A. Xiang, X. Zhang, L. Lin, and Q. He, "Selective fairness in recommendation via prompts," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '22. New York, NY, USA: Association for Computing Machinery, 2022. doi: 10.1145/3477495.3531913. ISBN 9781450387323 p. 2657–2662. [Online]. Available: https://doi.org/10.1145/3477495.3531913 [Page 8.]

[4] W. Hua, Y. Ge, S. Xu, J. Ji, and Y. Zhang, "Up5: Unbiased foundation model for fairness-aware recommendation," *arXiv preprint arXiv:2305.12090*, 2023. [Page 9.]

[5] Y. Wang, W. Ma, M. Zhang, Y. Liu, and S. Ma, "A survey on the fairness of recommender systems," *ACM Transactions on Information Systems*, vol. 41, no. 3, pp. 1–43, 2023. [Page 9.]

[6] M. Jiang, K. Bao, J. Zhang, W. Wang, Z. Yang, F. Feng, and X. He, "Item-side fairness of large language model-based recommendation system," in *Proceedings of the ACM Web Conference 2024*, ser. WWW '24. New York, NY, USA: Association for Computing Machinery, 2024. doi: 10.1145/3589334.3648158. ISBN 9798400701719 p. 4717–4726. [Online]. Available: https://doi.org/10.1145/3589334.3648158 [Page 9.]