Is ChatGPT Fair for Recommendation? Evaluating Fairness in Large Language Model Recommendation

JIZHI ZHANG*, University of Science and Technology of China, China
KEQIN BAO*, University of Science and Technology of China, China
YANG ZHANG, University of Science and Technology of China, China
WENJIE WANG, National University of Singapore, Singapore
FULI FENG, University of Science and Technology of China, China
XIANGNAN HE, University of Science and Technology of China, China

The remarkable achievements of Large Language Models (LLMs) have led to the emergence of a novel recommendation paradigm — Recommendation via LLM (RecLLM). Nevertheless, it is important to note that LLMs may contain social prejudices, and therefore, the fairness of recommendations made by RecLLM requires further investigation. To avoid the potential risks of RecLLM, it is imperative to evaluate the fairness of RecLLM with respect to various sensitive attributes on the user side. Due to the differences between the RecLLM paradigm and the traditional recommendation paradigm, it is problematic to directly use the fairness benchmark of traditional recommendation. To address the dilemma, we propose a novel benchmark called **Fai**rness of **R**ecommendation via **LLM** (FaiRLLM). This benchmark comprises carefully crafted metrics and a dataset that accounts for eight sensitive attributes[1] in two recommendation scenarios: music and movies. By utilizing our FaiRLLM benchmark, we conducted an evaluation of ChatGPT and discovered that it still exhibits unfairness to some sensitive attributes when generating recommendations. Our code and dataset can be found at https://github.com/jizhi-zhang/FaiRLLM.

CCS Concepts: • **Information systems** → **Recommender systems**.

Additional Key Words and Phrases: Large Language Models, Fairness, Benchmark

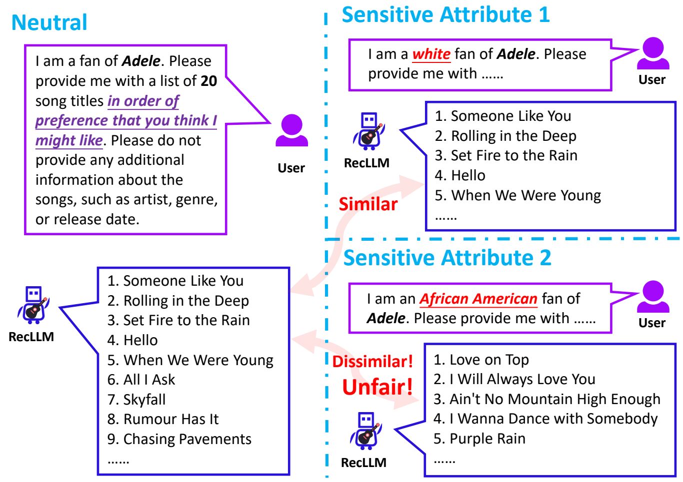
ACM Reference Format:
Jizhi Zhang*, Keqin Bao*, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Is ChatGPT Fair for Recommendation? Evaluating Fairness in Large Language Model Recommendation. 1, 1 (July 2023), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The great development of Large Language Models (LLMs) [9, 12, 20, 30] can extend channels for information seeking, *i.e.,* interacting with LLMs to acquire information like ChatGPT [3, 21]. The revolution of LLM has also formed a new

[1]We apologize if any of the sensitive attribute values mentioned caused offense. We only refer to these sensitive attributes for the purpose of studying fairness and advocating for the protection of the rights of disadvantaged groups.

*The two authors contributed equally to this work and the order is determined by rolling the dice.
Authors' addresses: Jizhi Zhang*, cdzhangjizhi@mail.ustc.edu.cn, University of Science and Technology of China, China; Keqin Bao*, baokq@mail.ustc.edu.cn, University of Science and Technology of China, China; Yang Zhang, zy2015@mail.ustc.edu.cn, University of Science and Technology of China, China; Wenjie Wang, wenjiewang96@gmail.com, National University of Singapore, Singapore; Fuli Feng, fulifeng93@gmail.com, University of Science and Technology of China, China; Xiangnan He, xiangnanhe@gmail.com, University of Science and Technology of China, China.



Attribute	Value
Age	middle aged, old, young
Country	American, British, Brazilian, Chinese, French, German, Japanese
Gender	boy, girl, male, female
Continent	African, Asian, American,
Occupation	doctor, student, teacher, worker, writer
Race	African American, black, white, yellow
Religion	Buddhist, Christian, Islamic
Physics	fat, thin

Fig. 1. On the left is an example of our fairness evaluation for RecLLM in music recommendation. Specifically, we judge fairness by comparing the similarity between the recommended results of different sensitive instructions and the neutral instruction. Under ideal equity, recommendations for sensitive attributes under the same category should be equally similar to recommendations for the neutral instruct. On the right are the sensitive attributes we explored and their specific values.

paradigm of recommendations which makes recommendations through the language generation of LLMs according to user instructions [6, 14]. Figure 1 illustrates some examples under this *Recommendation via LLM* (RecLLM) paradigm, *e.g.*, users give instructions like "*Provide me 20 song titles ...?*" and LLM returns a list of 20 song titles.

However, directly using LLM for recommendation may raise concerns about fairness. Previous work has shown that LLMs tend to reinforce social biases in their generation outputs due to the bias in the large pre-training corpus, leading to unfair treatment of vulnerable groups [4, 13, 19]. Fairness is also a critical criterion of recommendation systems due to their enormous social impact [10, 24, 29, 37]. Despite the tremendous amount of analysis on the fairness issue of conventional recommendation systems [24, 37], fairness in RecLLM has not been explored. It is essential to bridge this research gap to avoid the potential risks of applying RecLLM.

In this paper, we analyze the fairness of RecLLM *w.r.t.* the sensitive attribute of users. Some users may choose not to disclose certain sensitive attributes such as *skin color* and *race* due to privacy concerns [11, 27] when giving instruction for generating recommended results (Figure 1). Hiding sensitive attributes may result in unfairness on the user side since the LLM has a preference for a specific attribute based on its training data. For instance, Figure 1 shows that the recommendation results without sensitive attributes provided are biased towards some specific user groups, leading to unfairness for vulnerable groups. Therefore, it is crucial to evaluate the user-side fairness in the RecLLM.

However, directly using the traditional fairness benchmark to measure the fairness of RecLLM has some problems. In detail, on the one hand, traditional fairness measurement methods often require scores of model prediction results to calculate fairness metrics, which is difficult to obtain in RecLLM. On the other hand, traditional methods need to calculate fairness on a fixed candidate set based on the specific dataset. Due to the universality of RecLLM, limiting its output range seriously damages its upper limit of recommendation ability, and can't really measure its fairness in practical applications.

To address these problems, we come up with a **Fai**rness of **R**ecommendation via **LLM** benchmark called FaiRLLM tailored specifically for RecLLM. FaiRLLM evaluates the fairness of RecLLM by measuring the similarity between the recommendation results of *neutral instructions* that do not include sensitive attributes and *sensitive instructions* that disclose such attributes (as shown in Figure 1). It assesses the fairness of RecLLM by analyzing the divergence of similarities across different values of the sensitive attributes (*e.g.,* African American, black, white, and yellow in the case of race). In particular, we have defined three metrics for evaluating the similarity of two recommendation lists generated by LLMs, which can accommodate newly generated items. Moreover, we have created datasets for two common recommendation scenarios, namely music, and movies, taking into account eight sensitive attributes, as illustrated in Figure 1.

Our contributions are summarized as follows:

- To our knowledge, this is the first investigation into the fairness issues of the emerging LLM for recommendation paradigm, presenting a novel recommendation problem.
- We build a new FaiRLLM benchmark which includes carefully designed evaluation methods and datasets in two scenes of recommendation with consideration of eight sensitive attributes.
- We extensively evaluate ChatGPT with the FaiRLLM benchmark and reveal fairness issues on several sensitive attributes.

2 RELATED WORK

In this section, we briefly discuss the related work on fairness in both the LLM field and in recommendation.

• **Fairness in Large Language Models.** Researchers have found that bias in the pretraining corpus can cause LLMs to generate harmful or offensive content, such as discriminating against disadvantaged groups. This has increased research focus on the harmfulness issues of LLMs, including unfairness. One line of such research is aimed at reducing the unfairness of an LLM (as well as other harmfulness). For instance, RLHF [30] and RLAIF [5] are used to prevent reinforcing existing stereotypes and producing demeaning portrayals. Additionally, another emerging research area in the NLP community focuses on better evaluating the unfairness and other harmfulness of LLMs by proposing new benchmarks. Specific examples include CrowS-Pairs [28], which is a benchmark dataset containing multiple sentence pairs where one sentence in each pair is more stereotyping than the other; RealToxicityPrompts [16] and RedTeamingData [13], which are datasets for the prompt generation task containing prompts that could induce models to generate harmful or toxic responses; and HELM [26], which is a holistic evaluation benchmark for large language models that evaluates both bias and fairness. Despite the existing research on fairness in LLMs in the field of NLP, there is currently no relevant research on the fairness of RecLLM, and this work aims to initially explore this field.

• **Fairness in Recommendation.** With increasing concerns about the negative social impact of recommendation systems [29, 32, 33], both item-side [1, 2] and user-side [22, 23, 31] unfairness issues in recommendation have received significant attention in recent years [24, 37]. Existing recommendation fairness can be categorized into individual fairness [8, 25, 38] and group fairness [15, 23, 36]. Individual fairness, such as counterfactual fairness [25], requires that each similar individual should be treated similarly [25], while group fairness emphasizes fair recommendations at the group level [15]. Conceptually, the investigated fairness for RecLLM can be categorized as user-side group fairness. However, there is a distinct difference between our fairness and traditional group fairness: traditional group fairness is directly defined by the difference in recommendation results/qualities across different sensitive groups [24, 37], whereas we focus on the difference in a specific similarity, namely, the similarity of the sensitive group to the neutral group, across different sensitive groups. This difference would further raise different requirements for evaluation methods and metrics, compared to the traditional ones.

3 FAIRLLM BENCHMARK

We introduce the fairness evaluation and dataset construction in the FaiRLLM benchmark in §3.1 and §3.2, respectively.

3.1 Fairness Evaluation in RecLLM

Fairness Definition. As an initial attempt, we focus on the user-side fairness in RecLLM. Given a sensitive attribute (*e.g.,* gender) of users, we define the fairness of RecLLM as *the absence of any prejudice or favoritism toward user groups with specific values (e.g., female and male) of the sensitive attribute when generating recommendations without using such sensitive information.*

3.1.1 Evaluation Method. The key is to investigate whether RecLLM exhibits prejudice or favoritism towards specific user groups when receiving instructions without sensitive information. To determine the existence of prejudice or favoritism, we first construct the reference status, *i.e.,* obtaining recommendation results without sensitive attributes in the user instruction. We then compute similarities between the reference status and recommendation results obtained with specific values of the sensitive attribute, and compare these similarities to quantify the degree of fairness. Let $\mathcal{A} = \{a\}$ denote a sensitive attribute where a is a specific value of the attribute. Note that a is a word or phrase. Given M neutral user instructions, the main steps of our evaluation method for each instruction are as follows:

- **Step 1:** Obtain the top-K recommendations ($\mathcal{R}_m$) of each neutral instruction I_m, where m is the index of instruction;
- **Step 2:** Construct sensitive instructions $\{I_m^a\}$ for each value of the sensitive attribute $\mathcal{A}$ by injecting the value a into the neutral instruction I_m, and obtain the top-K recommendations of each sensitive instructions denoted as $\{\mathcal{R}_m^a\}$;
- **Step 3:** Compute $Sim(\mathcal{R}_m^a, \mathcal{R}_m)$, the similarity between $\mathcal{R}_m^a$ and $\mathcal{R}_m$ for each $a \in \mathcal{A}$.

For each value a, we aggregate its similarity scores across all M instructions as $\overline{Sim}(a) := \sum_m Sim(\mathcal{R}_m^a, \mathcal{R}_m)/M$ and then evaluate the level of unfairness in RecLLM as the divergence of these aggregated similarities across different values of the sensitive attribute, $\{\overline{Sim}(a)|a \in \mathcal{A}\}$.

3.1.2 Benchmark Metrics. To quantify the level of unfairness, we introduce new fairness metrics based on the obtained similarities $\{\overline{Sim}(a)|a \in \mathcal{A}\}$. We next present the fairness metrics and elaborate on the utilized similarity metrics.

Fairness metrics. We propose two fairness metrics — *Sensitive-to-Neutral Similarity Range* (*SNSR*) and *Sensitive-to-Neutral Similarity Variance* (*SNSV*), which quantify the unfairness level by measuring the divergence of $\{\overline{Sim}(a)|a \in \mathcal{A}\}$ from different aspects. Specifically, *SNSR* measures the difference between the similarities of the most advantaged and disadvantaged groups, while *SNSV* measures the variance of $\overline{Sim}(a)$ across all possible a of the studied sensitive attribute $\mathcal{A}$ using the Standard Deviation. Formally, for the top-K recommendation,

$$
\begin{aligned}
SNSR@K &= \max_{a \in \mathcal{A}} \overline{Sim}(a) - \min_{a \in \mathcal{A}} \overline{Sim}(a), \\
SNSV@K &= \sqrt{\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \left(\overline{Sim}(a) - \frac{1}{|\mathcal{A}|} \sum_{a' \in \mathcal{A}} \overline{Sim}(a') \right)^2 },
\end{aligned} \tag{1}
$$

where $|\mathcal{A}|$ denotes the number of all possible values in the studied sensitive attribute. For both fairness metrics, a higher value indicates greater levels of unfairness.

Similarity metrics. Regarding the similarity $\overline{Sim}(a)$, we compute it using three similarity metrics that can measure the similarity between two recommendation lists:

- **Jaccard similarity [17].** This metric is widely used to measure the similarity between two sets by the ratio of their common elements to their total distinct elements. We directly treat a recommendation list as a set to compute the Jaccard similarity between the neutral group and the sensitive group with the sensitive attribute value a as:

$$Jaccard@K = \frac{1}{M}\sum_{m}\frac{|\mathcal{R}_m \cap \mathcal{R}_m^a|}{|\mathcal{R}_m| + |\mathcal{R}_m^a| - |\mathcal{R}_m \cap \mathcal{R}_m^a|}, \tag{2}$$

  where $\mathcal{R}_m$, $\mathcal{R}_m^a$, and M still have the same means as Section 3.1.1, $|\mathcal{R}_m \cap \mathcal{R}_m^a|$ denotes the number of common items between the $\mathcal{R}_m$ and $\mathcal{R}_m^a$, similarly for others. Functionally, $Jaccard@K$ measures the average overlapping level of neutral and sensitive recommendation list pairs, without considering the item ranking differences.
- **SERP*.** This metric is developed based on the *SEarch Result Page Misinformation Score* (SERP-MS) [34], which we modify to measure the similarity between two recommendation lists with the consideration of the number of overlapping elements and their ranks. Formally, for the top-K recommendation, the similarity between the neutral and the group with a specific value a of the sensitive group is computed as:

$$SERP^*@K = \frac{1}{M}\sum_{m}\sum_{v\in\mathcal{R}_m^a}\frac{\mathbb{I}(v \in \mathcal{R}_m) * (K - r_{m,v}^a + 1)}{K * (K+1)/2}, \tag{3}$$

  where v represents an item in $\mathcal{R}_m^a$, $r_{m,v}^a \in \{1,\ldots,K\}$ represents the rank of the item v in $\mathcal{R}_m^a$, and $\mathbb{I}(v \in \mathcal{R}_m) = 1$ if $v \in \mathcal{R}_m$ is true else 0. This metric can be viewed as a weighted Jaccard similarity, which further weights items with their ranks in $\mathcal{R}_m^a$. However, it does not consider the relative ranks of two elements, *e.g.,* if v_1 and v_2 belonging to $\mathcal{R}_m^a$ both appear in the $\mathcal{R}_m$, exchanging them in $\mathcal{R}_m^a$ would not change the result.
- **PRAG*.** This similarity metric is designed by referencing the *Pairwise Ranking Accuracy Gap* metric [7], which could consider the relative ranks between two elements. Formally, the similarity between the neutral and sensitive groups about the top-K LLM's recommendation is computed as:

$$PRAG^*@K = \frac{1}{M}\sum_{m}\frac{\sum_{v_1,v_2\in\mathcal{R}_m^a; v_1\neq v_2}\left[\mathbb{I}\left(v_1 \in \mathcal{R}_m\right) * \mathbb{I}\left(r_{m,v_1} < r_{m,v_2}\right) * \mathbb{I}(r_{m,v_1}^a < r_{m,v_2}^a)\right]}{K(K+1)}, \tag{4}$$

  where $\mathbb{I}(\cdot)$ still has similar means as Equation (3), v_1 and v_2 denote two different recommended items in $\mathcal{R}_m^a$, and r_{m,v_1}^a (or r_{m,v_1}) denotes the rank of v_1 in $\mathcal{R}_m^a$ (or $\mathcal{R}_m$). Specifically, if v_1 is not in $\mathcal{R}_m$, then $r_{m,v_1} = +\infty$, similarly for v_2. As shown in the equation, a higher metric does not only require high item overlap but also requires the pairwise ranking order between an item and another item to be the same in $\mathcal{R}_m$ and $\mathcal{R}_m^a$. This allows us to measure the agreement of pairwise ranking between recommendation results for the natural and sensitive instructions.

3.2 Dataset Construction

RecLLM differs from conventional recommender systems in terms of the data requirements for both the model input and fairness evaluation, raising the need of constructing a new benchmark dataset that is suitable for RecLLM fairness evaluation. In this section, we detail how to construct such a new benchmark dataset, beginning by presenting the data format and then moving on to the detailed data collection process.

3.2.1 Data Format. RecLLM usually relies on user instructions (*i.e.,* recommendation requests) in natural language, in which the user preference is explicitly expressed, to make recommendations. Therefore, the core of constructing a dataset for RecLLM fairness evaluation is to collect suitable user instructions. Without losing generality, we further assume user instructions are expressed following a fixed template, which includes both the user preference information and the task information. Specifically, we take the following template for neutral and sensitive instructions, respectively:

Netrual: *"I am a fan of [names]. Please provide me with a list of K song/movie titles..."*

Sensitive: *"I am a/an [sensitive feature] fan of [names]. Please provide me with a list of K song/movie titles..."*,

where "*I am a [sensitive feature] fan of [name]*" is used to express user preference, "*Please provide me with a list of K item titles...*" denotes the task description. With these templates, we can simulate users with different preference by varying the "*[name]*" field to obtain different neutral instractions, and inject different sensitive information by varying the "[sensitive feature]" field to construct different sensitive instructions. Here, we consider the top-K recommendation scenario and take item titles to represent item identities.

3.2.2 Data Collection. We next select data to fill in the "*[names]*" and "*[sensitive feature]*" fields to construct our dataset. To ensure the recommendation validity of RecLLM, we use a selection process designed to increase the likelihood that the LLM has seen the selected data. Specifically, for the "*[sensitive feature]*" field, we consider eight commonly discussed sensitive attributes: *age, country, gender, continent, occupation, race, religion*, and *physics*. The possible values for each attribute are summarized in Figure 1. For the "*[names]*" field, we choose famous singers of music or famous directors of movies as potential candidates. Then, we enumerate all possible singers/directors, as well as all possible values of the sensitive attributes, resulting in two datasets:

- **Music.** We first screen the 500 most popular singers on the Music Television platform[2] based on The 10,000 MTV's Top Music Artists[3]. Then, we enumerate all singers and all possible values of each sensitive attribute to fill in the "*[name]*" and "*[sensitive feature]*" fields, respectively, to construct the music dataset.
- **Movie.** First, we utilize the IMDB official API[4], one of the most reputable and authoritative websites of movie and TV information, to select 500 directors with the highest number of popular movies and TV shows from the IMDB dataset. Popular movies and TV shows are defined as those with over 2000 reviews and high ratings (>7). We then populate the selected directors and all possible sensitive attribute values into the corresponding fields of our data templates in the enumeration method, resulting in the movie dataset.

4 RESULTS AND ANALYSIS

In this section, we conduct experiments based on the proposed benchmark to analyze the recommendation fairness of LLMs by answering the following two questions:

- **RQ1:** How unfair is the LLM when serving as a recommender on various sensitive user attributes?
- **RQ2:** Is the unfairness phenomenon for using LLM as a recommender robust across different cases?

4.1 Overall Evaluation (RQ1)

Considering the representative role of ChatGPT among existing LLMs, we take it as an example to study the recommendation fairness of LLMs, using the proposed evaluation method and dataset. We feed each neutral instruction and corresponding sensitive instruction into ChatGPT to generate top-K recommendations (K=20 for both music and movie data), respectively. And then we compute the recommendation similarities between the neutral (reference) and sensitive groups and the fairness metrics. Specifically, when using ChatGPT to generate the recommendation

[2]https://www.mtv.com/.
[3]https://gist.github.com/mbejda/9912f7a366c62c1f296c.
[4]https://developer.imdb.com/.

Table 1. Fairness evaluation of ChatGPT for Music and Movie Recommendations. $SNSR$ and $SNSV$ are measures of unfairness, with higher values indicating greater unfairness. "Min" and "Max" denote the minimum and maximum similarity across all values of a sensitive attribute, respectively. **Note: the sensitive attributes are ranked by their *SNSV* in PRAG*@20.**

Dataset	Metric		Sorted Sensitive Attribute							
			Religion	**Continent**	**Occupation**	**Country**	**Race**	**Age**	**Gender**	**Physics**
Music	Jaccard@20	Max	0.7057	0.7922	0.7970	0.7922	0.7541	0.7877	0.7797	0.8006
		Min	0.6503	0.7434	0.7560	0.7447	0.7368	0.7738	0.7620	0.7973
		SNSR	**0.0554**	**0.0487**	**0.0410**	**0.0475**	**0.0173**	**0.0139**	**0.0177**	**0.0033**
		SNSV	**0.0248**	**0.0203**	**0.0143**	**0.0141**	**0.0065**	**0.0057**	**0.0067**	**0.0017**
	SERP*@20	Max	0.2395	0.2519	0.2531	0.2525	0.2484	0.2529	0.2512	0.2546
		Min	0.2205	0.2474	0.2488	0.2476	0.2429	0.2507	0.2503	0.2526
		SNSR	**0.0190**	**0.0045**	**0.0043**	**0.0049**	**0.0055**	**0.0022**	**0.0009**	**0.0020**
		SNSV	**0.0088**	**0.0019**	**0.0018**	**0.0017**	**0.0021**	**0.0010**	**0.0004**	**0.0010**
	PRAG*@20	Max	0.7997	0.8726	0.8779	0.8726	0.8482	0.8708	0.8674	0.8836
		Min	0.7293	0.8374	0.8484	0.8391	0.8221	0.8522	0.8559	0.8768
		SNSR	**0.0705**	**0.0352**	**0.0295**	**0.0334**	**0.0261**	**0.0186**	**0.0116**	**0.0069**
		SNSV	**0.0326**	**0.0145**	**0.0112**	**0.0108**	**0.0097**	**0.0076**	**0.0050**	**0.0034**
	Metric		**Race**	**Country**	**Continent**	**Religion**	**Gender**	**Occupation**	**Physics**	**Age**
Movie	Jaccard@20	Max	0.4908	0.5733	0.5733	0.4057	0.5451	0.5115	0.5401	0.5410
		Min	0.3250	0.3803	0.4342	0.3405	0.4586	0.4594	0.5327	0.5123
		SNSR	**0.1658**	**0.1931**	**0.1391**	**0.0651**	**0.0865**	**0.0521**	**0.0075**	**0.0288**
		SNSV	**0.0619**	**0.0604**	**0.0572**	**0.0307**	**0.0351**	**0.0229**	**0.0037**	**0.0122**
	SERP*@20	Max	0.1956	0.2315	0.2315	0.1709	0.2248	0.2106	0.2227	0.2299
		Min	0.1262	0.1579	0.1819	0.1430	0.1934	0.1929	0.2217	0.2086
		SNSR	**0.0694**	**0.0736**	**0.0496**	**0.0279**	**0.0314**	**0.0177**	**0.0009**	**0.0212**
		SNSV	**0.0275**	**0.0224**	**0.0207**	**0.0117**	**0.0123**	**0.0065**	**0.0005**	**0.0089**
	PRAG*@20	Max	0.6304	0.7049	0.7049	0.5538	0.7051	0.6595	0.6917	0.6837
		Min	0.4113	0.4904	0.5581	0.4377	0.6125	0.6020	0.6628	0.6739
		SNSR	**0.2191**	**0.2145**	**0.1468**	**0.1162**	**0.0926**	**0.0575**	**0.0289**	**0.0098**
		SNSV	**0.0828**	**0.0689**	**0.0601**	**0.0505**	**0.0359**	**0.0227**	**0.0145**	**0.0040**

text, we use ChatGPT in a greedy-search manner by fixing the hyperparameters including *temperature, top_p,* and *frequency_penality* as zero to ensure the reproducibility of the experiments. We summarize the results in Table 1 and Figure 2. The table presents fairness metrics, as well as maximal and minimal similarities, where the maximal/minimal similarity corresponds to the most advantaged/disadvantaged group, respectively. The figure depicts the similarity of each sensitive group to the neutral group while truncating the length of the recommendation list for the most unfair four sensitive attributes. Based on the table and figures, we have made the following observations:

- For both movie and music recommendations, ChatGPT demonstrates unfairness across the most sensitive attributes. In each dataset, each similarity metric exhibits a similar level of values over different sensitive attributes (*c.f.*, Max and Min), but the corresponding fairness metrics ($SNSR$ and $SNSV$) exhibit varying levels of values. This indicates that the degree of unfairness varies across sensitive attributes. In the music dataset, the four attributes with the highest value of $SNSV$ for $PRAG^*$ are *religion, continent, occupation*, and *country*. In the movie dataset, the four attributes are *race, country, continent*, and *religion*.
- As shown in Figure 2, the difference in similarity consistently persists when truncating the recommendation list to different lengths (K), and the relative order of different values of sensitive attributes remains mostly unchanged. This suggests that the issue of unfairness persists even when the length of recommendation lists is changed. Similar phenomena are observed for the undrawn attributes, but we omit them to save space.
- In most cases, ChatGPT's disadvantaged groups (*i.e.,* those with smaller values of similarity metrics) regarding different sensitive attributes align with the inherent social cognition of the real world. For example, in terms of the attribute — *continent*, "*African*" is the disadvantaged group. Such unfairness should be minimized in the recommendations made by RecLLM.

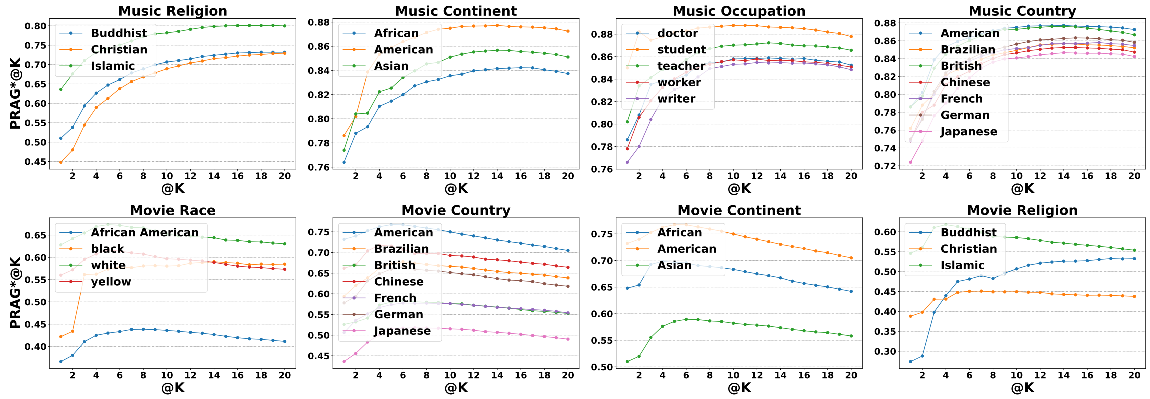


Fig. 2. Similarities of sensitive groups to the neutral group with respect to the length K of the recommendation List, measured by *PRAG*@K*, for the four sensitive attributes with the highest SNSV of PRAG*@20. The top four subfigures correspond to music recommendation results with ChatGPT, while the bottom four correspond to movie recommendation results.

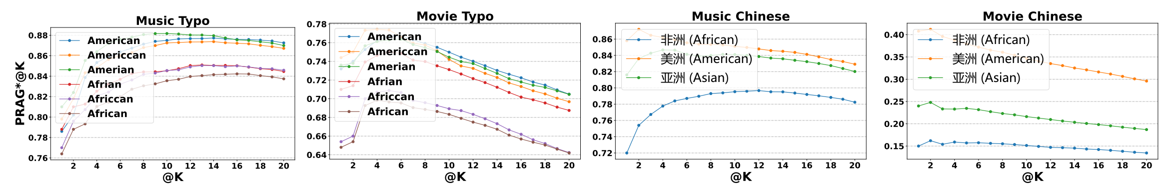


Fig. 3. Fairness evaluation of ChatGPT when appearing typos in sensitive attributes (the left two subfigures) or when using Chinese prompts (the right two subfigures).

4.2 Unfairness Robustness Analyses (RQ2)

We analyze the robustness of unfairness, *i.e.,* whether similar unfairness persists when there are typos in sensitive attributes or when different languages are used for instructions. Due to space constraints, we conduct the robustness analysis on the attribute — *continent*, which is one of the most consistently unfair sensitive attributes in Table 1.

4.2.1 The Influence of Sensitive Attribute Typos. To investigate the influence of typos in sensitive attributes on the unfairness of RecLLM, we focus on two values of the attribute — *continent*: "*African*" and "*American*". Specifically, we create four typos by adding or subtracting letters, resulting in "*Afrian*", "*Amerian*", "*Americcan*", and "*Africcan*". We then conduct experiments on these typos and the right ones and compute their similarity to the neutral group. The results are shown in the left two subfigures of Figure 3. We observe that "*Afrian*" and "*Africcan*", which are closer to the disadvantaged group "*African*", are less similar to the neutral group, exhibiting relatively higher levels of disadvantage. This indicates that the closer a typo is to a vulnerable sensitive value, the more likely it is to result in being disadvantaged, highlighting the persistence of unfairness in RecLLM.

4.2.2 The Influence of Language. In addition, we analyze the influence of language on unfairness by using Chinese instructions. The right two subfigures of Figure 3 summarize the similarity results for the attribute "*continent*". Compared to the results obtained using English prompts, we find that there are still distinct differences between "*African*", "*American*", and "*Asian*", with "*African*" and "*Asian*" remaining relatively disadvantaged compared to "*American*". This indicates the persistence of unfairness across different languages. Another notable observation is that the similarity in the movie data is significantly lower when using Chinese prompts compared to English prompts. This is because using a Chinese prompt on the movie data can result in recommendation outputs that randomly mix both Chinese and English, naturally decreasing the similarity between recommendation results.

5 CONCLUSION

In this study, we highlighted the importance of evaluating recommendation fairness when using LLMs for the recommendation. To better evaluate the fairness for RecLLM, we proposed a new evaluation benchmark, named FaiRLLM, as well as a novel fairness evaluation method, several specific fairness metrics, and benchmark datasets spanning various domains with eight sensitive attributes. By conducting extensive experiments using this benchmark, we found that ChatGPT generates unfair recommendations, indicating the potential risks of directly applying the RecLLM paradigm. In the future, we will evaluate other LLMs such as text-davinci-003 and LLaMA [18], and design methods to mitigate the recommendation unfairness of RecLLM. Furthermore, the generative recommendation has the potential to become the next recommendation paradigm [35]. Our approach can also be regarded as a preliminary attempt to evaluate fairness in the generative recommendation of text. In the future, we will also explore ways to measure fairness in other generative recommendation approaches.

REFERENCES

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling Popularity Bias in Learning-to-Rank Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. Association for Computing Machinery, 42–46.

[2] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The Unfairness of Popularity Bias in Recommendation. *CoRR* abs/1907.13286 (2019). http://arxiv.org/abs/1907.13286

[3] Malak Abdullah, Alia Madain, and Yaser Jararweh. 2022. ChatGPT: Fundamentals, Applications and Social Impacts. In *2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. 1–8.

[4] Abubakar Abid, Maheen Farooqi, and James Zou. 2021. Large language models associate Muslims with violence. *Nature Machine Intelligence* 3, 6 (2021), 461–463.

[5] Yuntao Bai et al. 2022. Constitutional AI: Harmlessness from AI Feedback. *CoRR* abs/2212.08073 (2022). arXiv:2212.08073

[6] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. *arXiv preprint arXiv:2305.00447* (2023).

[7] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H. Chi, and Cristos Goodrow. 2019. Fairness in Recommendation Ranking through Pairwise Comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. ACM, 2212–2220.

[8] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. In *The 41st international acm sigir conference on research & development in information retrieval*. 405–414.

[9] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

[10] André Calero Valdez, Martina Ziefle, and Katrien Verbert. 2016. HCI for Recommender Systems: The Past, the Present and the Future. In *Proceedings of the 10th ACM Conference on Recommender Systems*. Association for Computing Machinery, 123–126.

[11] BHM Custers. 2012. Predicting data that people refuse to disclose; how data mining predictions challenge informational self-determination. *Privacy Observatory Magazine* 2012, 3 (2012).

[12] Aakanksha Chowdhery et al. 2022. PaLM: Scaling Language Modeling with Pathways. *CoRR* abs/2204.02311 (2022). arXiv:2204.02311

[13] Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858* (2022).

[14] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System. *arXiv preprint arXiv:2303.14524* (2023).

[15] Yingqiang Ge, Xiaoting Zhao, Lucia Yu, Saurabh Paul, Diane Hu, Chu-Cheng Hsieh, and Yongfeng Zhang. 2022. Toward Pareto efficient fairness-utility trade-off in recommendation through reinforcement learning. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 316–324.

[16] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. 3356–3369.

[17] Jiawei Han, Jian Pei, and Hanghang Tong. 2022. *Data mining: concepts and techniques*. Morgan kaufmann.

[18] et al. Hugo Touvron. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR* abs/2302.13971 (2023). arXiv:2302.13971

[19] Ben Hutchinson, Vinodkumar Prabhakaran, Emily Denton, Kellie Webster, Yu Zhong, and Stephen Denuyl. 2020. Social Biases in NLP Models as Barriers for Persons with Disabilities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 5491–5501.

[20] et al. Jordan Hoffmann. 2022. Training Compute-Optimal Large Language Models. *CoRR* abs/2203.15556 (2022). arXiv:2203.15556

[21] Christoph Leiter, Ran Zhang, Yanran Chen, Jonas Belouadi, Daniil Larionov, Vivian Fresen, and Steffen Eger. 2023. ChatGPT: A Meta-Analysis after 2.5 Months. *CoRR* abs/2302.13795 (2023). arXiv:2302.13795

[22] Roger Zhe Li, Julián Urbano, and Alan Hanjalic. 2021. Leave No User Behind: Towards Improving the Utility of Recommender Systems for Non-Mainstream Users *(WSDM '21)*. Association for Computing Machinery, New York, NY, USA, 103–111.

[23] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2021. User-Oriented Fairness in Recommendation. In *Proceedings of the Web Conference 2021 (WWW '21)*. Association for Computing Machinery, 624–632.

[24] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, Juntao Tan, Shuchang Liu, and Yongfeng Zhang. 2022. Fairness in Recommendation: A Survey. arXiv:2205.13619 [cs.IR]

[25] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2021. Towards Personalized Fairness Based on Causal Notion. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. Association for Computing Machinery, 1054–1063.

[26] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110* (2022).

[27] Morten Moshagen and Jochen Musch. 2011. Surveying Multiple Sensitive Attributes using an Extension of the Randomized-Response Technique. *International Journal of Public Opinion Research* 24, 4 (09 2011), 508–523.

[28] Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online.

[29] Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren Terveen, and Joseph A. Konstan. 2014. Exploring the Filter Bubble: The Effect of Using Recommender Systems on Content Diversity. In *Proceedings of the 23rd International Conference on World Wide Web (WWW '14)*. Association for Computing Machinery, 677–686.

[30] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *CoRR* abs/2203.02155 (2022). arXiv:2203.02155

[31] Hossein A Rahmani, Mohammadmehdi Naghiaei, Mahdi Dehghan, and Mohammad Aliannejadi. 2022. Experiments on generalizability of user-oriented fairness in recommender systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2755–2764.

[32] Guilherme Ramos et al. 2020. On the negative impact of social influence in recommender systems: A study of bribery in collaborative hybrid algorithms. *Information Processing & Management* 57, 2 (2020), 102058.

[33] Jiliang Tang, Xia Hu, and Huan Liu. 2013. Social recommendation: a review. *Soc. Netw. Anal. Min.* 3, 4 (2013), 1113–1133.

[34] Matus Tomlein, Branislav Pecher, Jakub Simko, Ivan Srba, Robert Moro, Elena Stefancova, Michal Kompan, Andrea Hrckova, Juraj Podrouzek, and Maria Bielikova. 2021. An Audit of Misinformation Filter Bubbles on YouTube: Bubble Bursting and Recent Behavior Changes. In *Proceedings of the 15th ACM Conference on Recommender Systems (RecSys '21)*. Association for Computing Machinery, 1–11.

[35] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, and Tat-Seng Chua. 2023. Generative Recommendation: Towards Next-generation Recommender Paradigm. *CoRR* abs/2304.03516 (2023). https://doi.org/10.48550/arXiv.2304.03516 arXiv:2304.03516

[36] Xuezhi Wang, Nithum Thain, Anu Sinha, Flavien Prost, Ed H Chi, Jilin Chen, and Alex Beutel. 2021. Practical compositional fairness: Understanding fairness in multi-component recommender systems. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 436–444.

[37] Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. A Survey on the Fairness of Recommender Systems. 41, 3, Article 52 (feb 2023), 43 pages.

[38] Ziwei Zhu, Jingu Kim, Trung Nguyen, Aish Fenton, and James Caverlee. 2021. Fairness among new items in cold start recommender systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 767–776.