Multilingual Prompts in LLM-Based Recommenders: Performance Across Languages

Makbule Gulcin Ozsoy

London, UK

Abstract

Large language models (LLMs) are increasingly used in natural language processing tasks. Recommender systems traditionally use methods such as collaborative filtering and matrix factorization, as well as advanced techniques like deep learning and reinforcement learning. Although language models have been applied in recommendation, the recent trend have focused on leveraging the generative capabilities of LLMs for more personalized suggestions. While current research focuses on English due to its resource richness, this work explores the impact of non-English prompts on recommendation performance. Using OpenP5, a platform for developing and evaluating LLM-based recommendations, we expanded its English prompt templates to include Spanish and Turkish. Evaluation on three real-world datasets, namely ML1M, LastFM, and Amazon-Beauty, showed that usage of non-English prompts generally reduce performance, especially in less-resourced languages like Turkish. We also retrained an LLM-based recommender model with multilingual prompts to analyze performance variations. Retraining with multilingual prompts resulted in more balanced performance across languages, but slightly reduced English performance. This work highlights the need for diverse language support in LLM-based recommenders and suggests future research on creating evaluation datasets, using newer models and additional languages.

Keywords

Recommender systems, Large language models, Prompting, Multi-language evaluation

1. Introduction

Large language models (LLMs) have become essential tools in natural language processing, excelling in tasks like named entity recognition, text classification, summarisation, and translation. They are designed to understand and generate natural language, making it more intuitive for end-users to interact with machines through applications like search engines and chatbots. Recently, there is increasing interest in using LLMs within recommender systems. Recommender systems estimate user preferences to suggest relevant items, using various techniques, such as collaborative filtering, content-based filtering, and matrix factorization [1, 2, 3, 4], deep learning [5, 6, 7, 8], reinforcement learning [9, 10], and language models-based [11, 12] methods. The recent trend is to leverage LLMs' generative capabilities for more personalized suggestions. Researchers are investigating how LLMs can enhance recommendations by understanding complex user behaviors and language patterns.

Given this trend, we anticipate four phases in the evolution of LLM-based recommenders, as shown in Figure 1: (i) Initial Phase: Users request recommendations through actions like clicking, and black-box recommender systems use traditional methods to generate personalized suggestions. (ii) LLM Integration: In this phase, LLMs are introduced into the recommender system. Although users continue to interact via clicks or scrolls, the system now uses prompts and LLM models to generate recommendations. Prompts are in the language where the LLM performs best, such as English. (iii) Prompt Template Interaction: Users interact with the system using natural language through prompt templates. Instead of relying only on clicks or scrolls, users fill out templates to request recommendations. While in the initial stages, these prompts could be in English, later they could be in the user's native language. (iv) Natural Language Interaction: Users interact with the recommender system directly in their native language, without the need for prompt templates. Considering recent developments in LLM-based recommenders, such as those in [13, 14, 15], it appears the field is already in phase two or even phase three. However, these phases often rely on user interactions in languages where LLMs perform best, primarily English. Current state-of-the-art LLMs cover only a small percentage of the world's spoken languages, favoring those with abundant resources like English [16, 17, 18].

This work investigates how non-English prompts affect recommendation performance using the OpenP5 platform [19, 14], which supports developing, training, and evaluating LLM-based models for generative recommendation. We expanded OpenP5's English templates to include Spanish and Turkish, assessing their impact on performance. Spanish, with more resources than Turkish but fewer than English, was expected to perform slightly worse, while Turkish, being less resourced, was anticipated to perform lower. In addition to using non-English prompts on a pretrained model, we examined the effects of further training the model with multilingual prompts, namely in English, Spanish, and Turkish. The main contributions of this work are[1]:

- Exploring the impact of non-English prompts on LLM-based recommenders by comparing performance with English, Spanish, and Turkish prompts.
- Exploring the effects of further training an LLM-based recommender with multilingual prompts.
- Evaluating LLM-based recommenders on three real-world datasets, namely ML1M, LastFM and Amazon-Beauty, finding that non-English prompts generally reduce performance particularly for languages less similar to English. However, retraining the model with multiple languages led to a more balanced performance, with a slight decrease for English prompts.

The paper is organized as follows: Section 2 provides background on recommender models, with a focus on LLM-based systems, their challenges, and LLMs on non-English content. Section 3 explores the effects of non-English prompts on

The 1st Workshop on Risks, Opportunities, and Evaluation of Generative Models in Recommender Systems (ROEGEN@RecSys 2024), October 2024, Bari, Italy.

makbulegulcin@gmail.com (M. G. Ozsoy)
0000-0001-6013-1668 (M. G. Ozsoy)

[1] For a video summary of the contributions of this work, see: https://vimeo.com/1014071055

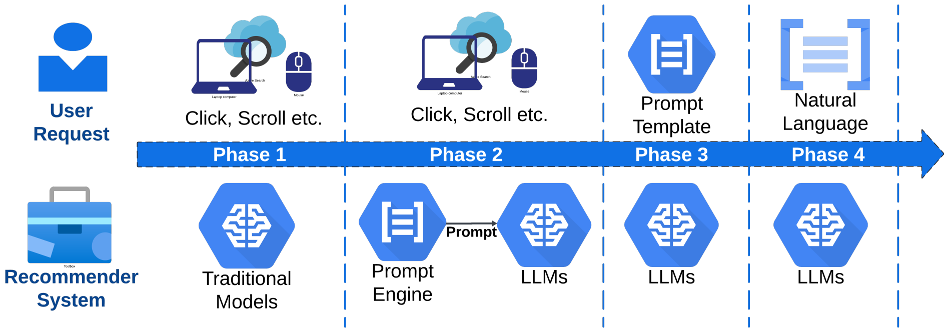

Figure 1: Four phases in the evolution of LLM-based recommenders: (i) Initial Phase: Traditional methods generate suggestions based on user actions like clicks. (ii) LLM Integration: LLMs enhance recommendations using prompts. (iii) Prompt Template Interaction: Users request recommendations via natural language templates. (iv) Natural Language Interaction: Direct native language communication with the system.

LLM-based recommendation systems, including (i) the use of non-English prompts on an already trained LLM-based recommender and (ii) the impact of training a new model with both English and non-English prompts. Section 4 analyzes and discusses the experimental results, while Section 5 presents conclusion and outlines future research directions.

2. Related Work

This section provides an overview of LLM-based recommender systems, the challenges they encounter, and performance of LLMs with non-English content.

2.1. Integration of Recommender Systems and LLMs

Recommender systems aim to estimate users' preferences and recommend items based on historical user-item interaction data. They can use various approaches, such as traditional collaborative filtering, content-based filtering, matrix factorization [1, 2, 3, 4], deep learning [5, 6, 7, 8], reinforcement learning [9, 10], and language model-based methods [11, 20, 21, 12, 22]. Recently, researchers have started to integrate capabilities of LLMs in recommender systems [12, 23, 24].

LLM-Based Recommenders For integrating recommender systems and LLMs, researchers utilize two key strategies: non-tuning-based methods and tuning-based methods. Non-tuning-based methods [25, 26, 27, 28, 29, 30, 31, 32, 33] utilize in-context learning and prompt optimization to interact with LLMs (e.g., Chat-GPT) without training. The main focus in these works is the design of prompts, which require extensive expert knowledge and human labor. Additionally, they might not perform as well as traditional recommendation methods [34, 35]. Tuning-based methods focus on further training LLMs with domain-specific knowledge. These approaches leverage historical interactions and contextual data, incorporating prompts and instructions during the tuning process, though they may suffer from high computational costs. For example, P5 [13], RecSysLLM [36], TALLRec [37], PALR [38], PBNR [39], InstructRec [40], GenRec [41], RecGPT [15], Re2LLM [42], GLRec [43] use various prompts to fine-tune LLMs for recommendation purposes. Additionally, PPR [44] and UniCRS [45] have explored prompt learning techniques. These approaches leverage historical interactions and contextual data and incorporate prompts and instructions during the tuning process. However, they may suffer from high computational costs. Other research efforts aim to support researchers working with LLM-based recommenders. For example, OpenP5 [19, 14] has developed a platform for exploring LLM-based recommendations, providing an environment for the development, training, and evaluation of generative recommender systems for research purposes.

Challenges of LLM-based Recommenders The approaches explained above focus on improving the recommendations performance. However, LLM-based recommenders have some additional challenges. Position bias occurs since LLMs favor items at the top of the list [46]. Popularity bias is another issue, where frequently mentioned items in training data tend to rank higher, potentially reducing diversity and worsening cold-start problems [46]. Fairness bias is also a concern, as LLMs may exhibit biases related to sensitive attributes such as age, gender, or race [46, 23]. Additionally, LLMs face challenges in safety and robustness, as small changes can affect their reliability, risking manipulation or misuse[12]. Hallucinations are another problem, where LLMs might recommend non-existent items, leading to user dissatisfaction [47, 23]. Effective prompt design is crucial, as converting user and item attributes into natural language prompts can be limited by context length. Controlling LLM outputs to meet specific constraints (e.g., price, color) is difficult, and ensuring consistent formatting is challenging [46, 48]. Finally, privacy concerns arise because LLMs use large datasets that may contain sensitive user information, risking data exposure [12]. Even though various studies have explored many challenges in LLM-based recommenders [12, 23, 24, 46], the impact of prompt language has not been thoroughly investigated. Most LLM-based recommender systems predominantly use English prompts, whether for tuning or non-tuning methods. Some notable exceptions include RecSysLLM [36], which uses English and Chinese prompts; both high-resource languages [16]; and FaiRLLM [29], which examines user country information but uses a consistent prompt language throughout. In this work, we investigate how prompt language affects recommendation performance by exploring languages that are relatively low-resource and linguistically distinct from English.

Table 1
The statistics of the datasets

	#Users	#Items	#Interactions	Sparsity
ML1M	6,040	3,616	999,611	0.9516
LastFM	1,090	3,646	52,551	0.9868
Amazon Beauty	22,363	12,101	198,502	0.9993

Table 2
Command asking translations of all prompts, and an example translation

Translation command	Translate following sentences to Spanish: """1. Considering {dataset} user_{user_id} has interacted with {dataset} items {history} . What is the next recommendation for the user ?; {dataset} {target} ..."""
Example translation	Considerando que el user_{user_id} de {dataset} ha interactuado con {history} elementos de {dataset}. ¿Cuál es la próxima recomendación para el usuario?; {dataset} {target}

2.2. LLMs and Non-English Content

Large language models (LLMs) have become the dominant tools for performing various natural language processing tasks. Even though they are mainly build on English texts, researchers extend their capabilities to other non-English languages by building multi-language models [16]. These models are trained on multiple languages simultaneously and can infer connections between them. As the result, they can utilize word associations and grammatical rules acquired from languages with more resources, such as English, and apply them to languages with less available data. However, these models are predominantly trained on English text, leading them to transfer values and assumptions encoded in English into other language contexts [16, 17, 18]. Not all languages have the same resourcedness levels, such that the volume, quality, and diversity of the available data to train language models are different. The disparity in resourcedness levels means that, even when trained on multiple languages, LLMs perform significantly better in higher-resource languages and those similar to them compared to lower-resource languages [16]. There are research works that are focusing on challenges in low-resource languages. Some researchers collect data in specific languages and retrain or fine-tune LLMs [49, 50, 51, 52]. Zhao et al. [17] further analyzed how expanding vocabulary, additional pretraining, and instruction tuning affect the ability of LLMs to generate text and follow instructions in a non-English language. They showed that comparable performance can be achieved with less than 1% of the pre-training data.

3. LLM-based Recommenders Beyond English

We explore the impact of non-English prompts on LLM-based recommender systems in two ways: (i) applying non-English prompts to an existing model and (ii) using both English and non-English prompts to further train the model. In this section, we will first describe the evaluation setup and then discuss the tasks and their results.

3.1. Experimental Setup

We used OpenP5 [19, 14] platform, which is recently developed for developing, training, and evaluating LLM-based models for generative recommendation. It contains implementations of an encoder-decoder LLM, namely T5 [53], and a decoder-only LLM, namely Llama-2 [54] and provides 10 widely recognized public datasets for experiments. For indexing the items, it utilizes three item indexing methods: (i) Random indexing where the item ids are assigned randomly. It is fairly simple method but can introduce artificial relationships among items due to the way tokenizers used by LLMs work. (ii) Sequential indexing which assigns item ids to consecutive numbers based on users' consecutive interactions observed in training data. This approach mitigates tokenizer issues, as items with the same prefix are likely related, reflecting actual user interactions. (iii) Collaborative indexing which assigns item ids by the frequency of co-occurrence of items. For recommendation, OpenP5 focuses on sequential and straightforward recommendation tasks, where the former utilizes user ids along with user history, while the latter relies only on user ids. For each recommendation task, it offers eleven personalized prompts templates, all of which are created by the authors and are all in English. During execution, whether for training or making recommendations, these prompts are filled with personalized information for each user, such as user ids and interaction history as a list of item ids. The prompts are divided into seen and unseen categories. Out of the eleven prompts, one is designated as the unseen prompt and is used to assess the model's zero-shot generalization capability. In addition to offering a platform for LLM-based recommendations, OpenP5 also provides evaluation results. In their experiments, they fine-tuned the T5 [53] model on full parameters, and Llama-2 [54] using the LORA [55] technique.

In this work, we used random (R) and sequential (S) indexing techniques, omitting collaborative indexing as we couldn't achieve consistent performance, even with English prompts. We employed the OpenP5-T5 model, a fine-tuned version of T5 [53] optimized for recommendation tasks. Experiments were conducted on three well-known recommen-

Table 3
Performance on ML1M dataset with [Random (R), Sequential (S)] indexing, [seen (s), unseen (u)] prompts for [Sequential, Straightforward] recommendation in [EN: English, ES: Spanish, TR: Turkish] (H: Hitrate, N: NDCG)

Indexing (Prompt)	Lng	Sequential					Straightforward				
		H@5	N@5	H@10	N@10	Avg%	H@5	N@5	H@10	N@10	Avg%
R (s)	EN	0.1098	0.0734	0.1573	0.0888	-	0.0215	0.0133	0.0346	0.0175	-
	ES	0.1118	0.0728	0.1586	0.0879	0.21	0.0220	0.0140	0.0351	0.0181	3.12
	TR	0.1071	0.0722	0.1543	0.0875	-1.87	0.0207	0.0128	0.0343	0.0171	-2.66
R (u)	EN	0.1060	0.0693	0.1533	0.0846	-	0.0219	0.0138	0.0341	0.0177	-
	ES	0.1061	0.0700	0.1561	0.0862	1.21	0.0219	0.0136	0.0364	0.0183	2.17
	TR	0.0232	0.0148	0.0364	0.0190	-77.64	0.0220	0.0139	0.0354	0.0182	1.95
S (s)	EN	0.2101	0.1430	0.3053	0.1737	-	0.0310	0.0192	0.0571	0.0275	-
	ES	0.1490	0.0982	0.2220	0.1218	-29.39	0.0320	0.0198	0.0568	0.0277	1.64
	TR	0.0624	0.0399	0.0985	0.0515	-70.12	0.0286	0.0183	0.0561	0.0272	-3.82
S (u)	EN	0.2116	0.1436	0.3055	0.1737	-	0.0316	0.0193	0.0566	0.0272	-
	ES	0.1697	0.1137	0.2493	0.1394	-19.69	0.0311	0.0193	0.0579	0.0279	0.82
	TR	0.0296	0.0191	0.0517	0.0261	-85.19	0.0298	0.0186	0.0556	0.0269	-3.05

Table 4
Performance on LastFM dataset with [Random (R), Sequential (S)] indexing, [seen (s), unseen (u)] prompts for [Sequential, Straightforward] recommendation in [EN: English, ES: Spanish, TR: Turkish] (H: Hitrate, N: NDCG)

Indexing (Prompt)	Lng	Sequential					Straightforward				
		H@5	N@5	H@10	N@10	Avg%	H@5	N@5	H@10	N@10	Avg%
R (s)	EN	0.0156	0.0104	0.0312	0.0153	-	0.0239	0.0151	0.0294	0.0169	-
	ES	0.0165	0.0094	0.0303	0.0139	-3.97	0.0202	0.0121	0.0339	0.0167	-5.31
	TR	0.0174	0.0118	0.0266	0.0147	1.59	0.0202	0.0128	0.0312	0.0164	-6.89
R (u)	EN	0.0128	0.0072	0.0248	0.0110	-	0.0211	0.0130	0.0358	0.0178	-
	ES	0.0147	0.0087	0.0294	0.0135	19.24	0.0229	0.0137	0.0349	0.0176	2.57
	TR	0.0184	0.0121	0.0312	0.0163	46.45	0.0220	0.0134	0.0294	0.0158	-5.44
S (s)	EN	0.0395	0.0262	0.0587	0.0323	-	0.0376	0.0259	0.0661	0.0350	-
	ES	0.0321	0.0215	0.0523	0.0280	-15.22	0.0330	0.0212	0.0578	0.0293	-14.81
	TR	0.0275	0.0167	0.0459	0.0226	-29.62	0.0312	0.0221	0.0523	0.0289	-17.50
S (u)	EN	0.0403	0.0265	0.0606	0.0331	-	0.0403	0.0282	0.0679	0.0370	-
	ES	0.0358	0.0230	0.0532	0.0288	-12.40	0.0376	0.0239	0.0551	0.0297	-15.13
	TR	0.0275	0.0195	0.0468	0.0256	-25.90	0.0330	0.0212	0.0505	0.0269	-23.97

dation datasets, namely ML1M, LastFM, and Amazon Beauty, which are pre-split, pre-processed and publicly available [2]. The statistics of the datasets are summarized in Table 1. Performance was measured using HitRate and NDCG metrics @5 and @10, following the OpenP5 framework [19, 14]. Computations were performed on a P100 GPU with a 30-hour weekly usage limit. All experiments were re-run, and results are discussed in the following sections.

3.2. Impact of Non-English Prompts on a PreTrained LLM-Based Recommender

In order to explore the impact of non-English prompts on a pretrained LLM-based recommender, we used prompts in Spanish and Turkish alongside English. English is a extremely-high resource language by several orders of magnitude, while Spanish is a high-resource and Turkish is medium-resource language [16, 56]. Previous research [57] shows that linguistic similarity influences model performance. Spanish and English belong to the Indo-European language family, whereas Turkish is part of the Altaic language family. Given these language characteristics, we predict that the performance of Spanish prompts will be slightly lower than that of English prompts but still relatively close. In contrast, using Turkish prompts is expected to decrease the performance of the LLM-based recommender systems. The Spanish and Turkish prompts were generated using machine translation through the ChatGPT interface with the GPT-3.5 model. We used a command to translate all prompts into Spanish and Turkish. The command used and an example translation are shown in Table 2. In the experiments, translated prompts were input into a pretrained model, and recommendation performance across different languages was compared. The performance comparisons on ML1M, LastFM and Amazon-Beauty datasets are presented in Table 3, Table 4 and Table 5, respectively. The tables show the evaluation metrics results and the average changes across all metrics compared to the English prompts.

[2] https://github.com/agiresearch/OpenP5

Table 5
Performance on Amazon-Beauty dataset with [Random (R), Sequential (S)] indexing, [seen (s), unseen (u)] prompts for [Sequential, Straightforward] recommendation in [EN: English, ES: Spanish, TR: Turkish] (H: Hitrate, N: NDCG)

Indexing (Prompt)	Lng	Sequential					Straightforward				
		H@5	N@5	H@10	N@10	Avg%	H@5	N@5	H@10	N@10	Avg%
R (s)	EN	0.0318	0.0226	0.0463	0.0273	-	0.0231	0.0168	0.0316	0.0195	-
	ES	0.0288	0.0204	0.0427	0.0249	-8.93	0.0190	0.0141	0.0284	0.0172	-13.94
	TR	0.0262	0.0178	0.0383	0.0217	-19.16	0.0207	0.0151	0.0292	0.0179	-9.08
R (u)	EN	0.0314	0.0222	0.0457	0.0268	-	0.0222	0.0164	0.0312	0.0193	-
	ES	0.0251	0.0167	0.0380	0.0209	-20.92	0.0203	0.0150	0.0286	0.0177	-8.43
	TR	0.0207	0.0152	0.0289	0.0178	-33.99	0.0213	0.0153	0.0297	0.0180	-5.58
S (s)	EN	0.0456	0.0335	0.0622	0.0389	-	0.0318	0.0239	0.0439	0.0278	-
	ES	0.0403	0.0293	0.0563	0.0345	-11.24	0.0295	0.0219	0.0409	0.0256	-7.59
	TR	0.0369	0.0261	0.0513	0.0307	-19.94	0.0301	0.0227	0.0406	0.0261	-6.00
S (u)	EN	0.0454	0.0332	0.0614	0.0385	-	0.0314	0.0237	0.0436	0.0276	-
	ES	0.0379	0.0276	0.0524	0.0322	-16.10	0.0267	0.0197	0.0382	0.0234	-14.87
	TR	0.0280	0.0215	0.0390	0.0250	-36.28	0.0300	0.0228	0.0411	0.0263	-4.68

Observations When we analyze different prompt types (seen or unseen), recommendation types (sequential or straightforward), and datasets (ML1M, LastFM, Amazon Beauty), we observe that the behavior of random indexing is inconsistent and varies across datasets. For example, with seen prompts, the average change in evaluation metrics for Spanish and Turkish prompts remains relatively stable $(\pm 5\%)$ on the ML1M and LastFM datasets. However, on the Amazon Beauty dataset, these changes are more noticeable, ranging from -9% to -19%. In contrast, when using unseen prompts, performance with straightforward recommendations remains more stable compared to sequential recommendations across all datasets. We attribute this behavior to the characteristics of the random indexing. As discussed in OpenP5 [19, 14], random indexing assigns item ids randomly and due to the way tokenizers used by LLMs, these ids might be split into sub-chunks. For example, item ids "2048" and "2049" could be split into ["20", "48"] and ["20", "49"], leading to artificial similarities between unrelated items. When analyzing sequential indexing, we observe a performance decline across all datasets for both Spanish and Turkish prompts, with the decline being more pronounced for Turkish prompts. For example, on the LastFM dataset for sequential recommendations, there is an approximately 15% drop for Spanish prompts compared to a 29% drop for Turkish prompts. This difference is expected due to the distinct linguistic characteristics of Turkish compared to English. Overall, the tables demonstrate that using non-English prompts in an already trained LLM-based recommender system affects the recommendation performance negatively, especially for languages which have different characteristics than English.

3.3. Impact of English and Non-English Prompts on LLM Recommender Fine-Tuning

In OpenP5, the model was fine-tuned with English prompts to create OpenP5-T5 model. In this section, we fine-tune the same model with English, Spanish, and Turkish prompts and compare its performance across these languages. For the analysis, LastFM dataset with sequential indexing is used. The training is ran for ten epochs with the setup described in the experimental setup section. The performance comparison is presented in Table 6. In the table, the performance from the original OpenP5-T5 model and retrained model with multi-language prompts are presented.

Observations When we evaluate the performance of the retrained model, we observe a decline in performance on English prompts. For instance, the Hitrate@10 metric drops from 0.0587 to 0.0413 when the prompts are seen and the goal is to make a sequential recommendation. Similarly, it decreases from 0.0679 to 0.0523 when the prompts are unseen and the goal is a straightforward recommendation. However, with the retrained model, the performance across all languages becomes closer. For example, the Hitrate@10 values for English, Spanish, and Turkish prompts are 0.0413, 0.0413, and 0.0459, respectively, when the prompts are seen and the goal is sequential recommendation. For unseen prompts aimed at straightforward recommendation, the Hitrate@10 values are 0.0523, 0.0505, and 0.0523 for English, Spanish, and Turkish prompts, respectively. Overall, the table demonstrates that retraining the model with prompts from different languages affects its performance across these languages.

4. Analysis and Discussion

Large language models (LLMs) have become the dominant tools for various natural language processing tasks, and their integration into recommender systems is a growing trend. This integration empowers users to leverage the capabilities of LLMs in everyday interactions. However, despite efforts to create multilingual LLMs, these models are predominantly trained on English texts. This bias leads to better performance in high-resource languages like English, while lower-resource languages face significant challenges. As LLM-based recommenders evolve, it is crucial to consider these linguistic disparities, alongside other known issues such as position bias, popularity bias, and hallucinations.

Traditionally, recommender systems often function as black boxes for end-users, who simply request recommendations and receive items without insight into the underly-

Table 6
Performance of the retrained model on LastFM dataset with Sequential (S) indexing, [seen (s), unseen (u)] prompts for [Sequential, Straightforward] recommendation in [EN: English, ES: Spanish, TR: Turkish] (H: Hitrate, N: NDCG)

Indexing (Prompt)	Model Type	Lang	Sequential				Straightforward			
			H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10
S (s)	Retrained	EN	0.0229	0.0155	0.0413	0.0214	0.0376	0.0234	0.0523	0.0281
		ES	0.0183	0.0136	0.0413	0.0211	0.0404	0.0281	0.0523	0.0320
		TR	0.0257	0.0163	0.0459	0.0226	0.0367	0.0243	0.0523	0.0293
	Original	EN	0.0395	0.0262	0.0587	0.0323	0.0376	0.0259	0.0661	0.0350
		ES	0.0321	0.0215	0.0523	0.0280	0.0330	0.0212	0.0578	0.0293
		TR	0.0275	0.0167	0.0459	0.0226	0.0312	0.0221	0.0523	0.0289
S (u)	Retrained	EN	0.0248	0.0155	0.0431	0.0213	0.0385	0.0244	0.0523	0.0288
		ES	0.0266	0.0169	0.0431	0.0221	0.0376	0.0261	0.0505	0.0302
		TR	0.0367	0.0260	0.0514	0.0309	0.0340	0.0229	0.0523	0.0288
	Original	EN	0.0403	0.0265	0.0606	0.0331	0.0403	0.0282	0.0679	0.0370
		ES	0.0358	0.0230	0.0532	0.0288	0.0376	0.0239	0.0551	0.0297
		TR	0.0275	0.0195	0.0468	0.0256	0.0330	0.0212	0.0505	0.0269

ing processes. However, we anticipate that these systems will evolve enabling users to interact through natural language in addition to traditional methods like clicks and page-scrolls. We propose four phases in this evolution, as illustrated in Figure 1: (i) Initial Phase: Users interact with recommender via clicks and scrolls, while the recommender system relies on traditional black-box models. (ii) LLM Integration: LLMs are integrated into these systems, using prompts that are independent of the user's language. As the result, the prompts can be in one of the high-resource languages, such as English. (iii) Prompt Template Interaction: Users begin to interact with LLM-based recommenders by filling out prompt templates in their native languages, alongside traditional interactions. (iv) Natural Language Interaction: Users engage with recommenders through natural language requests alongside traditional interactions.

In light of these considerations, this paper explores the impact of non-English prompts on the performance of LLM-based recommendation systems in two folds: (i) by using non-English prompts on an already trained LLM-based recommender, and (ii) by incorporating both English and non-English prompts during further training. We selected Spanish and Turkish, which have distinct linguistic characteristics and resourcedness levels, for our experiments. The results revealed that non-English prompts negatively affect performance, especially in languages less similar to English. However, retraining the model with prompts from multiple languages led to a more balanced performance across all tested languages, despite a slight decrease in performance for English prompts.

5. Conclusion

Recommender systems suggest items to users based on their preferences, employing techniques from traditional methods like collaborative and content-based filtering to advanced deep learning approaches. Recently, large language models (LLMs) have been utilized to enhance recommendations through their generative capabilities, offering more personalized suggestions. Most state-of-the-art LLMs are designed for high-resource languages like English, which are well-studied and have abundant datasets. In this work, we investigate the impact of non-English prompts on LLM-based recommenders by (i) applying non-English prompts to an already trained model, and (ii) retraining the model with both English and non-English prompts. Our findings reveal that non-English prompts can reduce performance on an existing LLM-based recommender, particularly for languages that have different characteristics than English. However, retraining the model with these languages results in more balanced performance across them, though with a slight decrease in performance for English prompts.

This preliminary work explores how prompt language affects LLM-based recommender systems. We tested English, Spanish, and Turkish prompts, and plan to investigate additional languages with varying resource levels and characteristics in future research. Developing evaluation sets in these languages could aid other researchers and improve recommendations for users in their native languages, which we aim to address in future work. As newer multilingual models emerge, future research should consider these models for recommendation tasks and evaluate their performance on low-resource languages.

References

[1] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, Q. Yang, One-class collaborative filtering, in: 2008 Eighth IEEE international conference on data mining, IEEE, 2008, pp. 502–511.

[2] M. Ye, P. Yin, W.-C. Lee, Location recommendation for location-based social networks, in: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '10, ACM, New York, NY, USA, 2010, pp. 458–461.

[3] X. Li, G. Cong, X.-L. Li, T.-A. N. Pham, S. Krishnaswamy, Rank-geofm: A ranking based geographical factorization method for point of interest recommendation, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2015, pp. 433–442.

[4] J. He, X. Li, L. Liao, D. Song, W. K. Cheung, Inferring a personalized next point-of-interest recommendation model with latent behavior patterns, in: Proceedings

of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA., 2016, pp. 137–143.

[5] M. G. Ozsoy, From word embeddings to item recommendation, arXiv preprint arXiv:1601.01356 (2016).

[6] F. Vasile, E. Smirnova, A. Conneau, Meta-prod2vec: Product embeddings using side-information for recommendation, in: Proceedings of the 10th ACM conference on recommender systems, 2016, pp. 225–232.

[7] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th international conference on world wide web, 2017, pp. 173–182.

[8] C. Musto, T. Franza, G. Semeraro, M. de Gemmis, P. Lops, Deep content-based recommender systems exploiting recurrent neural networks and linked open data, in: Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization, ACM, 2018, pp. 239–244.

[9] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, Z. Li, Drn: A deep reinforcement learning framework for news recommendation, in: Proceedings of the 2018 world wide web conference, 2018, pp. 167–176.

[10] B. Shi, E. Z. Tragos, M. G. Ozsoy, R. Dong, N. Hurley, B. Smyth, A. Lawlor, Dares: an asynchronous distributed recommender system using deep reinforcement learning, IEEE access 9 (2021) 83340–83354.

[11] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, P. Jiang, Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer, in: Proceedings of the 28th ACM international conference on information and knowledge management, 2019, pp. 1441–1450.

[12] W. Fan, Z. Zhao, J. Li, Y. Liu, X. Mei, Y. Wang, J. Tang, Q. Li, Recommender systems in the era of large language models (llms), arXiv preprint arXiv:2307.02046 (2023).

[13] S. Geng, S. Liu, Z. Fu, Y. Ge, Y. Zhang, Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5), in: Proceedings of the 16th ACM Conference on Recommender Systems, 2022, pp. 299–315.

[14] S. Xu, W. Hua, Y. Zhang, Openp5: An open-source platform for developing, training, and evaluating llm-based recommender systems, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 386–394.

[15] H. Ngo, D. Q. Nguyen, Recgpt: Generative pre-training for text-based recommendation, arXiv preprint arXiv:2405.12715 (2024).

[16] G. Nicholas, A. Bhatia, Lost in translation: large language models in non-english content analysis, arXiv preprint arXiv:2306.07377 (2023).

[17] J. Zhao, Z. Zhang, Q. Zhang, T. Gui, X. Huang, Llama beyond english: An empirical study on language capability transfer, arXiv preprint arXiv:2401.01055 (2024).

[18] C. F. A. team, The ai language gap, Policy Primer (2024). https://cohere.com/research/papers/the-ai-language-gap.pdf.

[19] S. Xu, W. Hua, Y. Zhang, Openp5: Benchmarking foundation models for recommendation, arXiv preprint arXiv:2306.11134 (2023).

[20] Z. Qiu, X. Wu, J. Gao, W. Fan, U-bert: Pre-training user representations for improved recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, 2021, pp. 4320–4327.

[21] S. Zhang, N. Zheng, D. Wang, Gbert: Pre-training user representations for ephemeral group recommendation, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, pp. 2631–2639.

[22] J. Li, M. Wang, J. Li, J. Fu, X. Shen, J. Shang, J. McAuley, Text is all you need: Learning language representations for sequential recommendation, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 1258–1267.

[23] L. Li, Y. Zhang, D. Liu, L. Chen, Large language models for generative recommendation: A survey and visionary discussions, in: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), 2024, pp. 10146–10159.

[24] A. Vats, V. Jain, R. Raja, A. Chadha, Exploring the impact of large language models on recommender systems: An extensive review, arXiv preprint arXiv:2402.18590 (2024).

[25] S. Dai, N. Shao, H. Zhao, W. Yu, Z. Si, C. Xu, Z. Sun, X. Zhang, J. Xu, Uncovering chatgpt's capabilities in recommender systems, in: Proceedings of the 17th ACM Conference on Recommender Systems, 2023, pp. 1126–1132.

[26] W. Sun, L. Yan, X. Ma, S. Wang, P. Ren, Z. Chen, D. Yin, Z. Ren, Is chatgpt good at search? investigating large language models as re-ranking agents, arXiv preprint arXiv:2304.09542 (2023).

[27] Y. Gao, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, J. Zhang, Chat-rec: Towards interactive and explainable llms-augmented recommender system, arXiv preprint arXiv:2303.14524 (2023).

[28] Z. He, Z. Xie, R. Jha, H. Steck, D. Liang, Y. Feng, B. P. Majumder, N. Kallus, J. McAuley, Large language models as zero-shot conversational recommenders, in: Proceedings of the 32nd ACM international conference on information and knowledge management, 2023, pp. 720–730.

[29] J. Zhang, K. Bao, Y. Zhang, W. Wang, F. Feng, X. He, Is chatgpt fair for recommendation? evaluating fairness in large language model recommendation, in: Proceedings of the 17th ACM Conference on Recommender Systems, 2023, pp. 993–999.

[30] L. Wang, E.-P. Lim, Zero-shot next-item recommendation using large pretrained language models, arXiv preprint arXiv:2304.03153 (2023).

[31] Y. Du, D. Luo, R. Yan, X. Wang, H. Liu, H. Zhu, Y. Song, J. Zhang, Enhancing job recommendation through llm-based generative adversarial networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, 2024, pp. 8363–8371.

[32] Q. Liu, N. Chen, T. Sakai, X.-M. Wu, Once: Boosting content-based recommendation with both open-and closed-source large language models, in: Proceedings of the 17th ACM International Conference on Web Search and Data Mining, 2024, pp. 452–461.

[33] Y. Hou, J. Zhang, Z. Lin, H. Lu, R. Xie, J. McAuley, W. X. Zhao, Large language models are zero-shot rankers for recommender systems, in: European Conference on Information Retrieval, Springer, 2024, pp. 364–381.

[34] J. Liu, C. Liu, P. Zhou, R. Lv, K. Zhou, Y. Zhang, Is

chatgpt a good recommender? a preliminary study, arXiv preprint arXiv:2304.10149 (2023).

[35] J. Tan, S. Xu, W. Hua, Y. Ge, Z. Li, Y. Zhang, Idgenrec: Llm-recsys alignment with textual id learning, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 355–364.

[36] Z. Chu, H. Hao, X. Ouyang, S. Wang, Y. Wang, Y. Shen, J. Gu, Q. Cui, L. Li, S. Xue, et al., Leveraging large language models for pre-trained recommender systems, arXiv preprint arXiv:2308.10837 (2023).

[37] K. Bao, J. Zhang, Y. Zhang, W. Wang, F. Feng, X. He, Tallrec: An effective and efficient tuning framework to align large language model with recommendation, in: Proceedings of the 17th ACM Conference on Recommender Systems, 2023, pp. 1007–1014.

[38] F. Yang, Z. Chen, Z. Jiang, E. Cho, X. Huang, Y. Lu, Palr: Personalization aware llms for recommendation, arXiv preprint arXiv:2305.07622 (2023).

[39] X. Li, Y. Zhang, E. C. Malthouse, Pbnr: Prompt-based news recommender system, arXiv preprint arXiv:2304.07862 (2023).

[40] J. Zhang, R. Xie, Y. Hou, W. X. Zhao, L. Lin, J.-R. Wen, Recommendation as instruction following: A large language model empowered recommendation approach, arXiv preprint arXiv:2305.07001 (2023).

[41] J. Ji, Z. Li, S. Xu, W. Hua, Y. Ge, J. Tan, Y. Zhang, Genrec: Large language model for generative recommendation, in: European Conference on Information Retrieval, Springer, 2024, pp. 494–502.

[42] Z. Wang, Y. Du, Z. Sun, H. Chua, K. Feng, W. Wang, J. Zhang, Re2llm: Reflective reinforcement large language model for session-based recommendation, arXiv preprint arXiv:2403.16427 (2024).

[43] L. Wu, Z. Qiu, Z. Zheng, H. Zhu, E. Chen, Exploring large language model for graph data understanding in online job recommendations, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, 2024, pp. 9178–9186.

[44] Y. Wu, R. Xie, Y. Zhu, F. Zhuang, X. Zhang, L. Lin, Q. He, Personalized prompt for sequential recommendation, IEEE Transactions on Knowledge and Data Engineering (2024).

[45] C. Wang, Y. Yu, W. Ma, M. Zhang, C. Chen, Y. Liu, S. Ma, Towards representation alignment and uniformity in collaborative filtering, in: Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining, 2022, pp. 1816–1825.

[46] L. Wu, Z. Zheng, Z. Qiu, H. Wang, H. Gu, T. Shen, C. Qin, C. Zhu, H. Zhu, Q. Liu, et al., A survey on large language models for recommendation, arXiv preprint arXiv:2305.19860 (2023).

[47] R. Azamfirei, S. R. Kudchadkar, J. Fackler, Large language models and the perils of their hallucinations, Critical Care 27 (2023) 120.

[48] J. Tan, Y. Ge, Y. Zhu, Y. Xia, J. Luo, J. Ji, Y. Zhang, User-controllable recommendation via counterfactual retrospective and prospective explanations, in: ECAI 2023, IOS Press, 2023, pp. 2307–2314.

[49] I. Team, Internlm: A multilingual language model with progressively enhanced capabilities, 2023. https://github.com/InternLM/InternLM.

[50] Y. Cui, Z. Yang, X. Yao, Efficient and effective text encoding for chinese llama and alpaca, arXiv preprint arXiv:2304.08177 (2023).

[51] E. C. Acikgoz, M. Erdogan, D. Yuret, Bridging the bosphorus: Advancing turkish large language models through strategies for low-resource language adaptation and benchmarking, arXiv preprint arXiv:2405.04685 (2024).

[52] H. T. Kesgin, M. K. Yuce, E. Dogan, M. E. Uzun, A. Uz, H. E. Seyrek, A. Zeer, M. F. Amasyali, Introducing cosmosgpt: Monolingual training for turkish language models, arXiv preprint arXiv:2404.17336 (2024).

[53] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of machine learning research 21 (2020) 1–67.

[54] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, arXiv preprint arXiv:2307.09288 (2023).

[55] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, arXiv preprint arXiv:2106.09685 (2021).

[56] P. Joshi, S. Santy, A. Budhiraja, K. Bali, M. Choudhury, The state and fate of linguistic diversity and inclusion in the nlp world, arXiv preprint arXiv:2004.09095 (2020).

[57] T. I. Dhamecha, R. Murthy V, S. Bharadwaj, K. Sankaranarayanan, P. Bhattacharyya, Role of language relatedness in multilingual fine-tuning of language models: A case study in indo-aryan languages, arXiv preprint arXiv:2109.10534 (2021).