LLM-Rec: Personalized Recommendation via Prompting Large Language Models

Hanjia Lyu[1], Song Jiang[2], Hanqing Zeng[3], Yinglong Xia[3], Qifan Wang[3], Si Zhang[3], Ren Chen[3], Christopher Leung[3], Jiajie Tang[3], Jiebo Luo[1]
[1]University of Rochester [2]UCLA [3]Meta AI
hlyu5@ur.rochester.edu, jluo@cs.rochester.edu

Abstract

Text-based recommendation holds a wide range of practical applications due to its versatility, as textual descriptions can represent nearly any type of item. However, directly employing the original item descriptions may not yield optimal recommendation performance due to the lack of comprehensive information to align with user preferences. Recent advances in large language models (LLMs) have showcased their remarkable ability to harness commonsense knowledge and reasoning. In this study, we introduce a novel approach, coined LLM-REC, which incorporates four distinct prompting strategies of text enrichment for improving personalized text-based recommendations. Our empirical experiments reveal that using LLM-augmented text significantly enhances recommendation quality. Even basic MLP (Multi-Layer Perceptron) models achieve comparable or even better results than complex content-based methods. Notably, the success of LLM-REC lies in its prompting strategies, which effectively tap into the language model's comprehension of both general and specific item characteristics. This highlights the importance of employing diverse prompts and input augmentation techniques to boost the recommendation effectiveness of LLMs.

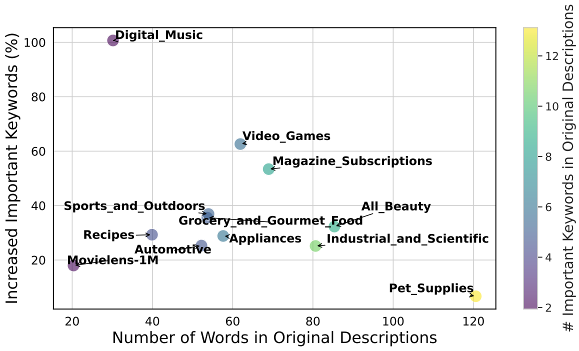

Figure 1: LLM-REC enhances original item descriptions by prompting LLMs to augment important keywords (*e.g.*, adjectives). It applies to various domains and is not limited to datasets with rich textual content.

1 Introduction

Text-based recommendation systems exhibit a broad spectrum of applications, spanning across diverse domains and industries. This versatility mainly stems from the capability of natural language to effectively describe nearly *any* type of items, encompassing not only products, movies, and books but also news articles and user-generated content, including short videos and social media posts (Pazzani and Billsus, 2007; Javed et al., 2021; Poirier et al., 2010; Bai et al., 2022; Wu et al., 2020; Oppermann et al., 2020; Chen et al., 2017; Gupta and Varma, 2017; Wang et al., 2018). Nonetheless, these text-based recommendation systems are frequently challenged by the inherent limitation of **incomplete or insufficient information within item descriptions**, which hinders the task of accurately *aligning* item characteristics with user preferences (Perez et al., 2007; Dumitru et al., 2011). The incompleteness may arise from two sources: a limited comprehension of the items themselves and an insufficient understanding of the users for whom recommendations are generated.

This challenge is not confined only to domains with well-defined and categorized items (*e.g.*, movies), but also extends to domains characterized by novel, unclassified, or less categorically structured items, as observed in the case of user-generated content. In the context of movie recommendations, a movie's description usually include the main actors, and a brief plot summary. However, this limited information may not capture crucial elements like genre, tone, cinematography style, or thematic depth, resulting in less effective recommendations. As for user-generated content, imagine a social platform where users regularly post recipes which are often accompanied with brief textual descriptions like the name of the dish and a few ingredients, but limited details regarding

Findings of the Association for Computational Linguistics: NAACL 2024, pages 583–612

preparation time, dietary restrictions, or flavor profiles. Consider a user who follows a vegan diet and is interested in discovering new plant-based recipes. Since the user-generated content often lacks comprehensive dietary information and may not explicitly mention terms like "vegan", "plant-based", or "vegetarian", in this scenario, the recommendation system, relying solely on the incomplete descriptions, may struggle to discern the vegan-friendliness of the recipes.

The recent advances in the development of large language models (LLMs) underscore their exceptional capacity to store comprehensive world knowledge (Peters et al., 2018; Goldberg, 2019; Tenney et al., 2019; Petroni et al., 2019), engage in complex reasoning (Wei et al., 2022; Zhou et al., 2022), and function as versatile task solvers (Zhao et al., 2023; Ouyang et al., 2022; Kaplan et al., 2020). In light of this advancement and recognizing the challenge posed by incomplete item descriptions, our study introduces the LLM-REC framework. This approach is designed to exploit various prompting strategies to *enrich input text* with the intrinsic capabilities of LLMs for personalized recommendations. By leveraging LLMs, which have been fine-tuned on extensive language datasets (Ouyang et al., 2022; Touvron et al., 2023a), our goal is to unlock their potential in generating input text that is not only contextually aware but also of high quality, thereby elevating the overall recommendation quality.

Through comprehensive empirical experiments, we evaluate the effectiveness of the LLM-REC framework. We find that integrating the augmented text as the new input achieves comparable or even superior recommendation performance compared to more advanced content-based recommendation approaches that rely solely on the original item descriptions. Further in-depth analyses reveal that the devised prompting strategies prompt LLMs to generate words that represent both general and specific item characteristics. It is applicable in a diverse range of domains and is not limited to datasets with rich textual information (Figure 1). Our study provides insights into the impact of different prompting strategies on recommendation performance and sheds light on the potential of leveraging LLMs for personalized recommendation.

2 Related Work

LLM-REC closely aligns with two research directions: (1) augmentation in text-based recommendation, and (2) LLM for recommendation. A comprehensive discussion is provided in Appendix C.

Augmentation in Text-based Recommendation. Text-based recommendation systems leverage natural language processing and machine learning techniques to provide personalized recommendations to users based on textual information (Lops et al., 2019; Qiang et al., 2020). However, the performance of such systems can be compromised when dealing with incomplete or insufficient textual information. To address this limitation, several studies have suggested strategies for enhancing textual information. For instance, Li et al. (2010) proposed to extract contextual cues from online reviews, leveraging these narratives to uncover users' preferences and underlying factors influencing their choices (Sachdeva and McAuley, 2020). Other approaches infer linguistic attributes from diverse sources, including emotion, sentiment, and topic, to refine the modeling of both items and users (Sun et al., 2015; Sailunaz and Alhajj, 2019; Ramage et al., 2010; Chen et al., 2010). Furthermore, some works explore the integration of external knowledge bases to enrich the contextual understanding of items (Di Noia et al., 2012; Musto et al., 2018). In a more recent development, Bai et al. (2022) introduced an approach that employs pre-trained language models to generate additional product attributes, such as product names, to augment item contextual information. Diverging from these prior approaches, our contribution is the LLM-REC framework, which employs large language models to enhance input text, providing a versatile solution for recommendations. A more detailed discussion on the distinctions between LLM-REC and these related work can be found in Section 5.

LLM for Recommendation. Due to LLMs' remarkable text generation ability, many studies have leveraged LLMs as a data augmentation tool (Dai et al., 2023a; Li et al., 2022). Liu et al. (2023a) used an LLM to produce multimodal language-image instruction-following datasets. Through a process of instruction tuning using this generated data, their proposed framework demonstrated an impressive aptitude in advancing vision and language comprehension. There have also been efforts to use LLMs to augment the input side of personalized recommendation. For instance, Chen (2023) incorporated user history behaviors, such as clicks, purchases, and ratings, into LLMs to generate user profiles. These profiles were then combined with the history interaction sequence and candidate items to

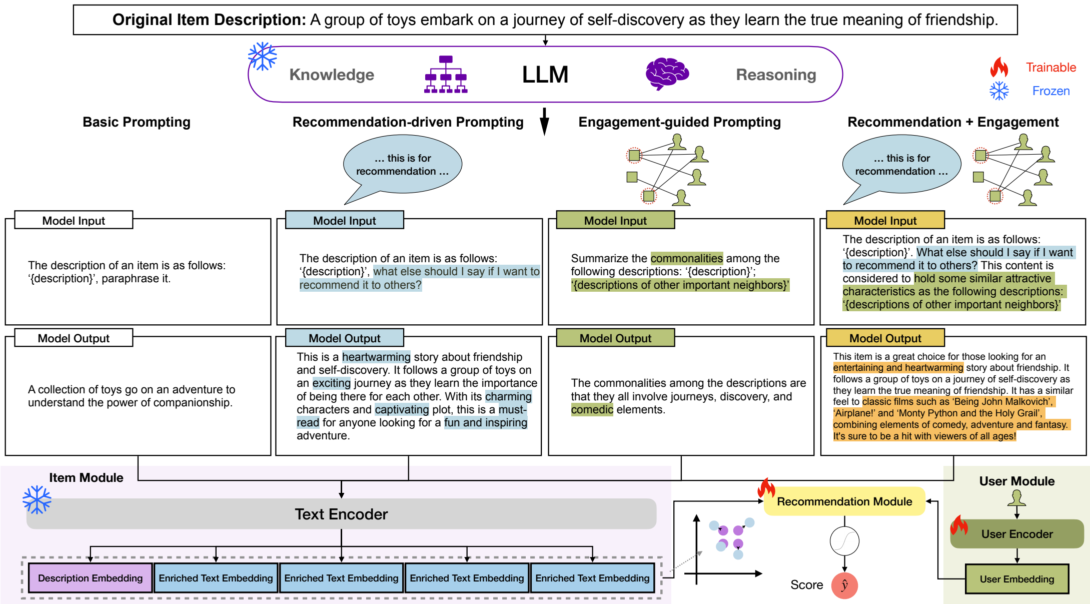

Figure 2: LLM-REC employs four prompting strategies to augment the original item descriptions which often contain incomplete information for recommendation. The augmented text is then concatenated to form the new input for the following recommendation module. LLM-REC plays a crucial role in enabling large language models to provide relevant context and help better align with user preferences. Prompts and augmented texts are highlighted.

construct the final recommendation prompt. LLMs were subsequently employed to predict the likelihood of user-item interaction based on this prompt. Xi et al. (2023) introduced a method that leverages the reasoning knowledge of LLMs regarding user preferences and the factual knowledge of LLMs about items. However, our study focuses specifically on using LLMs' knowledge and reasoning ability to generate augmented input text that better captures the characteristics and nuances of items, leading to improved recommendation performance.

3 LLM-Rec

When composing a summary for recommendation purposes, it is customary to infuse it with specific emphases grounded in the author's *comprehension* of the movie. This might involve accentuating the movie's distinctive attributes that set it apart from other movies. For instance, one may opt to incorporate genre information as a crucial element for classifying the movie. However, the decision to leverage the concept of genre for enhancing the summary is predicated on the author's understanding that the genre is a meaningful construct, effectively aligning the summary with the preferences and expectations of the intended audience. This paper aims to explore the potential of large language models when prompted to generate informative item descriptions and subsequently how to leverage this augmented text for enhancing personalized recommendations. Figure 2 shows the diagram of LLM-REC. Specifically, our study focuses on investigating *four* distinct LLM prompting strategies for description enrichment, namely basic prompting, recommendation-driven prompting, engagement-guided prompting, and the combination of recommendation-driven and engagement-guided prompting. The enriched text is then fed into the final recommendation module.

Basic Prompting. The concept of basic prompting closely resembles the task of crafting a general movie summary. Within this scope, we consider three basic prompting variants and refer to them as p_{para}, p_{tag}, and p_{infer}, respectively in the following experiments. p_{para} instructs LLMs to paraphrase the original item description, emphasizing the objective of maintaining the same information without introducing any additional details. Given the original content description, the prompt we use is *"The description of an item is as follows* '`{description}`', *paraphrase it."* p_{tag} aims to guide LLMs to summarize the content description by using tags, striving to generate a more concise overview that captures key information. The corresponding prompt is *"The description of an item is as follows* '`{description}`', *summarize it with tags."* p_{infer} instructs LLMs to deduce the characteristics of the original content description and provide a categorical response that operates at a broader, less detailed level of granularity. We use the following prompt in the experiments: *"The description of an item is as follows* '{description}', *what kind of emotions can it evoke?"*

Recommendation-driven Prompting. This prompting strategy is to add a recommendation-driven instruction, into the basic prompting, resembling the task of creating a paragraph intended for making recommendations. We refer to the three recommendation-driven prompting as p_{para}^{rec}, p_{tag}^{rec}, and p_{infer}^{rec}, respectively in the following experiments, aligning with their counterparts in the basic prompting strategy. p_{para}^{rec} represents the prompt: *"The description of an item is as follows* '{description}', *what else should I say if I want to recommend it to others?"* The prompt for p_{tag}^{rec} is *"The description of an item is as follows* '{description}', *what tags should I use if I want to recommend it to others?"* The prompt for p_{infer}^{rec} is *"The description of an item is as follows* '{description}', *recommend it to others with a focus on the emotions it can evoke."*

Engagement-guided Prompting. As previously elucidated, the deficiency in item descriptions can also emanate from a limited comprehension of the user cohort for whom the recommendations are being generated. Typically, item descriptions are initially formulated for broad, general purposes, devoid of specific targeting toward particular user groups. As a result, they often fall short in capturing the intricate nuances of items required for a more fine-grained alignment with individual user preferences. The goal of the engagement-guided prompting strategy is to leverage user behavior, specifically the interactions between users and items (*i.e.*, user-item engagement) to devise prompts with the intention to steer LLMs towards a more precise comprehension of the attributes within the items, thus generating more insightful and contextually relevant descriptions that align more closely with the preferences of intended users. We refer to this variant as p^{eng}. To create the engagement-guided prompt, we combine the description of the target item, denoted as d_{target}, with the descriptions of T **important** neighbor items, represented as $d_1, d_2, \cdots, d_T$. The importance is measured based on **user engagement**. More details can be found in Appendix A.6. The exact prompt of this prompting strategy is *"Summarize the commonalities among the following descriptions:* '{description}'; '{descriptions of other important neighbors}'."

Recommendation-driven + Engagement-guided Prompting. It intends to incorporate both the recommendation-driven and engagement-guided instructions, which we denote as $p^{rec+eng}$: *"The description of an item is as follows:* '{description}'. *What else should I say if I want to recommend it to others? This content is considered to hold some similar attractive characteristics as the following descriptions:* '{descriptions of other important neighbors}'."

How does LLM-REC affect personalized recommendation? In our experiments, we discover that first and foremost, LLM-REC stands out as a versatile yet simple framework, largely unrestricted by the type of items. Our experimental results on two datasets including the items that are categorically structured and extensively studied to items that are relatively novel and unclassified such as user-generated content, consistently demonstrate the substantial improvement in personalized recommendations. Simple models, such as MLP, can achieve performance on par with, or even better than, more advanced and complex models with the augmented text. This finding underscores the potential of simplified training to address challenges due to more complex models. More importantly, compared to other knowledge-based text augmentation methods, LLM-REC achieves superior recommendation performances and requires considerably less domain expertise compared to prior studies, making it much more accessible for implementation.

Second, LLM-REC contributes to increased recommendation transparency and explainability. The ability to directly investigate the augmented text not only enhances our understanding of the recommendation models but also offers insights into the characteristics of the items. It is invaluable for both users and system designers seeking to comprehend the rationale behind recommendations.

4 Experiments

4.1 Experiment Setup

Datasets and Baslines. Two widely adopted recommendation benchmarks are used, Movielens-1M (Harper and Konstan, 2015) for movie recommendation, and Recipe (Majumder et al., 2019) for recipe recommendation. To assess LLM-REC's efficacy, we compare it against two distinct categories of baselines. The first category includes content-based baselines that takes solely the original item

		Movielens-1M			Recipe		
		Precision@10	Recall@10	NDCG@10	Precision@10	Recall@10	NDCG@10
Item Popularity		0.0426 ±0.0019	0.0428 ±0.0028	0.0530 ±0.0035	0.0116 ±0.0025	0.0274 ±0.0083	0.0201 ±0.0053
MLP		0.2922 ±0.0019	0.2455 ±0.0031	0.3640 ±0.0039	0.0325 ±0.0021	0.0684 ±0.0066	0.0580 ±0.0054
AutoInt (Song et al., 2019)		0.2149 ±0.0078	0.1706 ±0.0075	0.2698 ±0.0092	0.0351 ±0.0032	0.0772 ±0.0102	<u>0.0658</u> ±0.0089
DCN-V2 (Wang et al., 2021)		0.2961 ±0.0050	0.2433 ±0.0057	0.3689 ±0.0033	<u>0.0360</u> ±0.0036	<u>0.0786</u> ±0.0104	0.0653 ±0.0085
EDCN (Chen et al., 2021)		0.2935 ±0.0036	0.2392 ±0.0051	0.3678 ±0.0053	0.0354 ±0.0030	0.0772 ±0.0091	0.0652 ±0.0071
TagGPT (Li et al., 2023a)	LLAMA-2-7B	0.2991 ±0.0017	0.2556 ±0.0038	0.3723 ±0.0023	0.0353 ±0.0024	0.0751 ±0.0067	0.0641 ±0.0057
	GPT-3	0.3001 ±0.0027	0.2569 ±0.0028	0.3747 ±0.0042	0.0356 ±0.0032	0.0752 ±0.0084	0.0637 ±0.0068
KAR (Xi et al., 2023)		0.3056 ±0.0026	0.2623 ±0.0034	0.3824 ±0.0042	0.0298 ±0.0018	0.0611 ±0.0049	0.0525 ±0.0043
- augmented with ground truth		0.3075 ±0.0015	0.2636 ±0.0035	0.3853 ±0.0027	-	-	-
LLM-REC	LLAMA-2-7B	<u>0.3102</u> ±0.0014 (+6.16%)	<u>0.2712</u> ±0.0026 (+10.47%)	<u>0.3867</u> ±0.0027 (+6.24%)	0.0359 ±0.0024 (+10.46%)	0.0770 ±0.0076 (+12.57%)	0.0632 ±0.0052 (+8.97%)
	GPT-3	**0.3150** ±0.0023 (+7.80%)	**0.2766** ±0.0030 (+12.67%)	**0.3951** ±0.0035 (+8.54%)	**0.0394** ±0.0033 (+21.23%)	**0.0842** ±0.0098 (+23.10%)	**0.0706** ±0.0084 (+21.72%)

Table 1: Average recommendation performance between LLM-REC and baseline approaches across five different train/test splits. The best results are highlighted in **bold**, the second-best results are <u>underlined</u>, and relative gains compared to the MLP baseline are indicated in green.

descriptions as input. The second category includes different text augmentation methods. Details including dataset statistics, preprocessing specifics, baselines, model training, hyper-parameter settings and implementation are discussed in Appendix A.

Language Models. Two large language models are selected for experiments. The first is GPT-3 (Brown et al., 2020), particularly its variant `text-davinci-003`. This variant is an advancement over the InstructGPT models (Ouyang et al., 2022). We select this variant due to its ability to consistently generate high-quality writing, effectively handle complex instructions, and demonstrate enhanced proficiency in generating longer form content (Raf, 2023). The second is LLAMA-2 (Touvron et al., 2023b), which is an open-sourced model that has shown superior performance across various external benchmarks in reasoning, coding, proficiency, and knowledge tests. Specifically, we use the LLAMA-2-CHAT variant of 7B parameters.

Evaluation Protocols. We follow the same evaluation methodology of Wei et al. (2019). We randomly divide the dataset into training, validation, and test sets using an 8:1:1 ratio. Negative training samples are created by pairing users and items without any recorded interactions (note that these are pseudo-negative samples). For the validation and test sets, we pair each observed user-item interaction with n items that the user has not previously interacted with. Here we follow the methodology outlined in the previous study (Wei et al., 2019) and set n to $1,000$. It is important to note that there is *no* overlap between the negative samples in the training set and the unobserved user-item pairs in the validation and test sets. This ensures the independence of the evaluation data. We use metrics such as Precision@K, Recall@K and NDCG@K to evaluate the performance of top-K recommendations, where K is set to 10. We report the average scores across five different splits of the testing sets. The recommendation module of LLM-REC is the combination of an MLP model and a dot product.

4.2 Main Results

Integrating the text augmented by LLM-REC boosts recommendation performance. Table 1 shows the average recommendation performance between LLM-REC and baseline approaches across five different splits. In addition to the aforementioned baselines, we include another baseline relies only on item popularity and does not involve any learning process, referred to as Item Popularity. Note that LLM-REC uses the same recommendation module as the MLP baseline. It can be seen that LLM-REC exhibits significant relative gains over the MLP method. For instance, LLM-REC achieves improvements in NDCG@10 from 6.24% to 8.54% in Movielens-1M and from 8.97% to 21.72% in Recipe. These improvements highlight the value of the augmented text in the recommendation process. More importantly, when comparing LLM-REC with the other three content-based methods, we observe that LLM-REC empowers simple MLP models to attain comparable or even superior performance, surpassing other more complex feature-based recommendation methods.

What extra information does LLM-REC incorporate that contributes to its performance improvement? To better understand why LLM-REC is effective, particularly regarding the additional

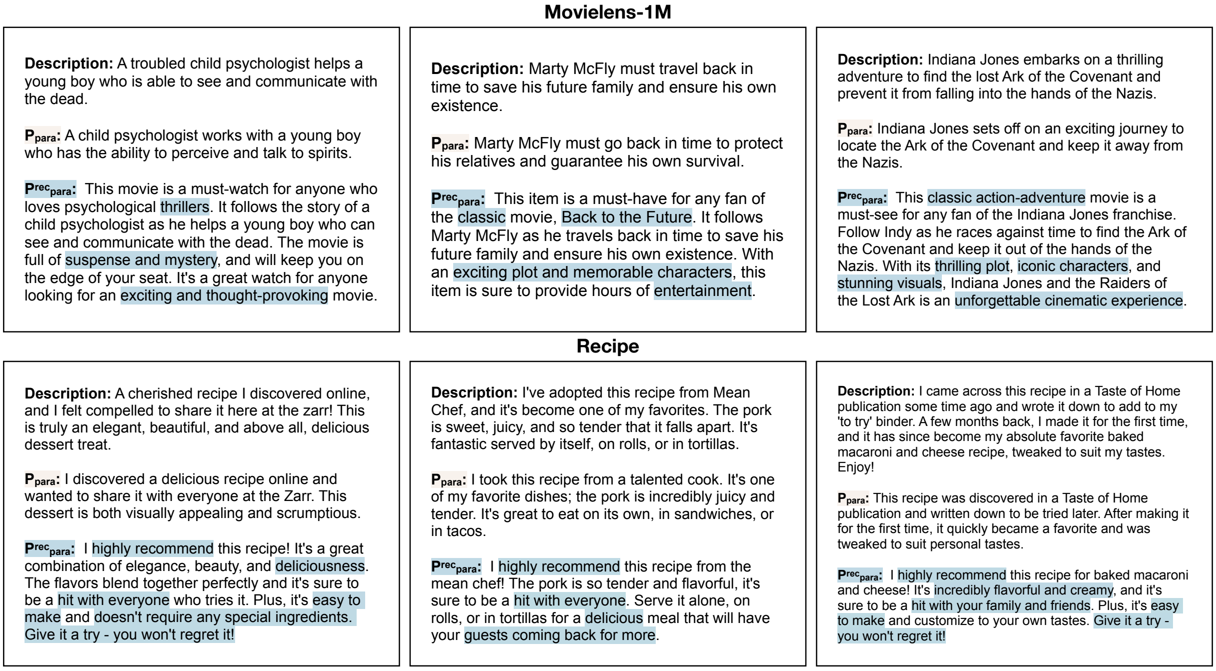

Figure 3: Example responses generated by GPT-3. The additional information augmented via the recommendation-driven prompting is highlighted in blue. Examples generated by LLAMA-2 can be found in Table 13.

information contributed by its prompting approach, we conduct both qualitative and quantitative studies. We find that the augmented content contains more detailed and expressive descriptions, emphasizing item characteristics which helps in understanding items more comprehensively than with their original descriptions and contributing to the performance improvement. Figure 3 shows example responses generated by GPT-3 with p_{para} and p^{rec}_{para}. The first example suggests that the response via p^{rec}_{para} categorizes the movie as a psychological thriller and recommends it as a must-watch for fans of this genre. It also positions the movie as both exciting and thought-provoking, appealing to those looking for more than just entertainment. These distinctive words describe user preferences and item characteristics including genre description, descriptive elements, and viewer recommendation. While Figure 3 might suggest that the LLM-generated text for the Recipe dataset adds only modifiers, these phrases, like "easy to make," actually reflect key characteristics valued in the Recipe dataset, such as simplicity. Some authors may also add #easytomake to their recipe descriptions (Majumder et al., 2019). Consistent patterns are also observed when comparing the responses of p_{tag} with p^{rec}_{tag} (Tables 11 and 14), and p_{infer} with p^{rec}_{infer} (Tables 12 and 15). A more thorough analysis shows that LLM-REC can be applied to diverse item domains and it is not restricted to datasets with rich textual information. Please see Appendix B.2.

We hypothesize that these generated words contribute to improving recommendation performance. To further validate this hypothesis, we design two variants of the response generated by GPT-3, namely p^{mask}_{para} and $p^{keyword}_{para}$. To construct p^{mask}_{para}, we mask the words that appear in the response of p^{rec}_{para} but are absent in the response of p_{para}. To construct $p^{keyword}_{para}$, we append the words that (1) appear in the response of p^{rec}_{para} and (2) are predefined user-preference-related words such as genres to the end of the response of p_{para} (more details in Appendix A.9). These two variants are then fed into MLP models to form baselines. Comparing the performance of p^{rec}_{para} and p^{mask}_{para} in Table 2, we observe a discernible decline in recommendation performance when words unique to the response of p^{rec}_{para} are selectively masked. This outcome highlights the pivotal role played by the supplementary insights introduced through the augmented text. Further, our investigation reveals that the incorporation of vital keywords, as opposed to the inclusion of all response words, can yield even superior recommendation performance. This may be attributed to potential discrepancies or extraneous elements within the response of p^{rec}_{para}.

LLM-REC augmentation outperforms other text augmentation methods for recommenda-

	Movielens-1M	Recipe
p_{para}	0.3746 ±0.0028	0.0611 ±0.0053
$p_{para}^{keyword}$	0.3822 ±0.0049	0.0615 ±0.0060
	(+2.03%)	(+0.65%)
p_{para}^{rec}	0.3777 ±0.0028	0.0646 ±0.0072
p_{para}^{mask}	0.3769 ±0.0040	0.0611 ±0.0066
	(-0.21%)	(-0.52%)

Table 2: Average NDCG@10 across five splits.

tion. We compare LLM-REC with two recent advancements in the field of using LLMs to augment item information, specifically Knowledge Augmented Recommendation (KAR) as proposed by Xi et al. (2023), and TagGPT, as proposed by Li et al. (2023a). KAR introduces a fusion of domain knowledge and prompt engineering to generate factual knowledge pertaining to the items (for detailed implementation information, see Appendix A.7). Since the augmented information may not necessarily be correct, we further implement a variant with ground truth knowledge. It aligns with strategies akin to those introduced by Di Noia et al. (2012), who harnessed external databases to enhance item information. In a manner consistent with this approach, we incorporate genre information into the item descriptions. Note that genre constitutes one of the metadata components in Movielens-1M. Such categorical characteristics are absent in Recipe. Therefore, we only apply this variant to the Movielens-1M dataset.

As shown in Table 1, the incorporation of knowledge-based text augmentation offers significant improvements in recommendation performance for well-classified items, such as movies. However, it becomes evident that this approach faces limitations when applied to items, like user-generated content, that are inherently more novel and dynamic in nature. The example response in Figure 4 shows that one key reason that knowledge augmentation approaches do not yield optimal improvement regarding recommendation performance may lie in the potential **mismatch** between the generated knowledge and the target item. For instance, while the generated ingredient information may be correct for most meatloaf recipes, it could be entirely wrong for a specific recipe without additional context. In contrast to these knowledge augmentation methods, LLM-REC's recommendation-driven prompts provide augmented information that describes the target item at a broader, less granular

Description: Simple and delicious, this dish can be enjoyed hot or cold, and it's perfect sliced cold for picnics or sandwiches. We've relished this basic meatloaf for many years. For a variation, try using your favorite salsa as a topping instead of the traditional one. A medium or hot chipotle salsa adds a fantastic flavor to it.

P^{rec}_{para}: I highly recommend this basic meatloaf! It's easy to make and tastes great hot or cold. It's perfect for a picnic or sandwiches, and you can even switch up the topping for something different. Try brushing on your favourite salsa, like a chipotle medium or hot salsa, for an extra kick of flavour. You won't be disappointed!

KAR: This easy and tasty meatloaf is a great dish to serve either hot or cold. It is perfect for a picnic or sandwiches, and can be enjoyed by everyone.

Ingredients: - 2 lbs ground beef - 1/2 cup bread crumbs - 1/4 cup onion, finely chopped - 1/4 cup green pepper, finely chopped - 1 egg - 1/4 cup ketchup - 1 teaspoon Worcestershire sauce - 1 teaspoon garlic powder - 1 teaspoon salt - 1/4 teaspoon black pepper - Optional: your favorite salsa

Figure 4: An example response generated using knowledge augmentation prompts (Xi et al., 2023). The additional information is highlighted in red.

level, especially when compared to KAR. More importantly, LLM-REC does not require domain knowledge throughout the entire process.

In terms of the second text augmentation baseline, TagGPT (Li et al., 2023a), which extracts tags using LLMs, several key observations can be made. First, we note an improvement in recommendation performance using tag generation compared to the baseline methods. Second, the prompts specifically designed within our LLM-REC framework demonstrate superior effectiveness compared to those used in TagGPT.

4.3 Ablation Study

How does each prompting strategy perform? We conduct an ablation study to examine the impact on recommendation performance when models use either original item descriptions alone or a combination of these descriptions with augmented text derived from one of four distinct prompting strategies. The results, presented in Table 3, reveal a noteworthy and consistent enhancement in recommendation performance across various prompting strategies within two benchmark datasets.

We also note variations in the performance of these strategies across different domains, aligning with our expectations. In Movielens-1M, the strategy combining recommendation-driven and engagement-guided approaches yields the best results. Conversely, in Recipe, the recommendation-driven strategy alone proves most effective. This variability suggests that combining multiple objectives in a single prompting strategy does not

	Movielens-1M	Recipe
Item Description	0.3640 ±0.0039	0.0580 ±0.0054
Basic	0.3747 ±0.0042 (**+2.94%**)	0.0644 ±0.0068 (**+11.03%**)
Rec	0.3786 ±0.0041 (**+4.01%**)	**0.0649** ±0.0069 (**+11.90%**)
Eng	0.3801 ±0.0032 (**+4.42%**)	0.0628 ±0.0077 (**+8.28%**)
Rec+Eng	**0.3802** ±0.0037 (**+4.45%**)	0.0625 ±0.0060 (**+7.76%**)

Table 3: Average NDCG@10 across five different splits among different prompting strategies.

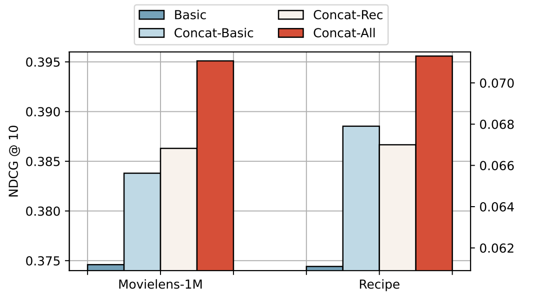

Figure 5: The ablation study shows that recommendation benefits from concatenating the embeddings of the input text augmented by LLM.

	Movielens-1M	Recipe
Item Description	0.3640 ±0.0039	0.0580 ±0.0054
Duplication	0.3567 ±0.0043	0.0590 ±0.0068
Text Concatenation	0.3853 ±0.0027	0.0591 ±0.0065
`Concat-All` (ours)	**0.3951** ±0.0035	**0.0706** ±0.0084

Table 4: Average NDCG@10 across five splits among different methods of extra information integration.

always lead to superior performance. When LLMs are tasked with generating descriptions serving multiple purposes, the balance of information becomes crucial. If neighboring item descriptions vary widely, it can challenge the model's ability to generate useful content, potentially leading to less optimal improvements. To address this, LLM-REC integrates all enriched text and leverages the subsequent recommendation module to effectively model the extra information. Future work can investigate different prompt designs, aiming to effectively achieve multiple objectives simultaneously.

How does concatenating the augmented responses affect recommendation? In Table 1, we show that the MLP model, which combines all augmented text with the original description, outperforms more advanced models that rely solely on the original description as input. Now we take a deeper look at the quality of the combined augmented text. We employ the same recommendation module (*i.e.*, an MLP with a dot product) and evaluate the recommendation performance of various concatenation combinations. As shown in Figure 5, the model denoted as `Basic` uses the embeddings of text augmented through p_{para}. `Concat-Basic` represents the model that concatenates the embeddings of the input text augmented by all basic prompts. Additionally, `Concat-Rec` is the model that employs the concatenation of the embeddings of input text augmented by all recommendation-driven prompts. Lastly, `Concat-All` stands for the model that combines the embeddings of input text augmented by all four prompting strategies. Our findings reveal that concatenating more information *consistently* enhances recommendation performance. This emphasizes the added value of incorporating augmented text as opposed to relying solely on the original content description. Complete results of Figure 5 can be found in Figure 8.

How to effectively integrate the augmented responses to maximize improvement? Table 4 shows the recommendation performances of other concatenation variants: (1) **Duplication**: We duplicate the embeddings of the original item description to match the dimension size of the embeddings of `Concat-All`; (2) **Text concatenation**: Instead of concatenating the embeddings of all response (*i.e.*, `Concat-All`), we concatenate the responses first, and then convert it to embeddings. Through a comparative analysis of the model's performance, contrasting the first variant with `Concat-All`, it becomes evident that the observed improvement in performance is not attributable to changes in embedding size. Further, by comparing the performance of the second variant against `Concat-All`, we discover that in scenarios where the text encoder remains unmodified, the most effective strategy to integrate all enriched information is by first converting the text into embeddings and then concatenating these embeddings. This approach surpasses the method of concatenating text prior to its conversion into embeddings. Future research can explore the potential of modifying the text encoder to further enhance model efficiency and effectiveness.

Does modifying the word choice in the designed prompts significantly affect the augmented output? To investigate this, we construct one variant prompt for each of LLM-REC's prompts, ensuring

Prompting Strategy		Variant
Basic	p_{para}	0.8859 ± 0.0898
	p_{tag}	0.9112 ± 0.1000
	p_{infer}	0.6819 ± 0.1500
Rec	p^{rec}_{para}	0.7011 ± 0.1369
	p^{rec}_{tag}	0.8627 ± 0.1248
	p^{rec}_{infer}	0.8458 ± 0.0652
Eng	p^{eng}	0.6218 ± 0.1012
Rec+Eng	$p^{rec+eng}$	0.8542 ± 0.0802

Table 5: Average cosine similarity between the variant prompt responses and those generated from LLM-REC.

they convey the same meaning but with different word choices. As shown in Table 5, despite variations in wording of the prompts, the responses remain largely similar. The observed lower cosine similarity is primarily attributed to differences in the format of the responses which can be mitigated through various strategies, such as additional fine-tuning of the model or incorporating specific instructions within the prompts.

5 Discussions and Conclusions

In this study, we have investigated the effectiveness of LLM-REC as a simple yet impactful mechanism for improving recommendation through LLMs. Our approach is among those early attempts (Lin et al., 2023; Chen et al., 2023) that leverage LLMs for text augmentation in recommendation. There are three key contributions that distinguish our work from the concurrent ones. First, while previous work, such as KAR (Xi et al., 2023), focuses on design augmentation algorithm for a specific recommendation model, our model focuses on input text augmentation with LLMs, which is suitable for any content-based backbone recommendation models, demonstrating the flexibility of our approach. Second, in addition to our recommendation-driven augmentation using LLMs, we also design engagement-guided prompts to augment the input, which contains more personalized item characteristics. Overall, we conduct comprehensive experiments, with different combinations of prompting strategies, to not only illustrate the superior performance of our approach but also uncover the underlying rationale of the improvements.

We introduce LLM-REC, which enhances recommendation by augmenting the original item descriptions which often contains incomplete information for effective recommendations using large language models. We observed from extensive experiments that combining augmented input text and original item descriptions yields notable improvements in recommendation quality. These findings show the potential of using LLMs and strategic prompting techniques to enhance the accuracy and relevance of recommendation with an easier training process. By incorporating additional context, we enable the recommendation algorithms to capture more nuanced information and generate recommendations that better align with user preferences.

Limitations

In this study, we have investigated the effectiveness of LLM-REC as a simple yet effective mechanism for improving recommendation through large language models. While effective, LLM-REC does have its limitations. First, there is extra computational cost associated with LLM-REC framework. The primary computational load comes from the augmentation phase including the augmented output text length. Our findings indicate that selecting important words for inclusion, rather than incorporating all response words, can lead to improved recommendation performance, as evidenced in Table 2. Our future work will explore the balance between the number of words generated and the resulting performance enhancements. Second, similar to many LLM-based studies, LLM-REC faces challenges in promptly incorporating the latest knowledge. We plan to investigate methods in future work for LLMs to autonomously gather and summarize current knowledge from external sources, thereby improving text augmentation effectiveness.

Acknowledgments

Luo was supported in part by NSF Award #2238208.

References

Xiao Bai, Lei Duan, Richard Tang, Gaurav Batra, and Ritesh Agrawal. 2022. Improving text-based similar product recommendation for dynamic product advertising at yahoo. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2883–2892.

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. Dbpedia-a crystallization point for the web of data. *Journal of web semantics*, 7(3):154–165.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Sergey Brin. 1998. The pagerank citation ranking: bringing order to the web. *Proceedings of ASIS, 1998*, 98:161–172.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Bo Chen, Yichao Wang, Zhirong Liu, Ruiming Tang, Wei Guo, Hongkun Zheng, Weiwei Yao, Muyu Zhang, and Xiuqiang He. 2021. Enhancing explicit and implicit feature interactions via information sharing for parallel deep ctr models. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 3757–3766.

Jilin Chen, Rowan Nairn, Les Nelson, Michael Bernstein, and Ed Chi. 2010. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1185–1194.

Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, et al. 2023. When large language models meet personalization: Perspectives of challenges and opportunities. *arXiv preprint arXiv:2307.16376*.

Ting Chen, Liangjie Hong, Yue Shi, and Yizhou Sun. 2017. Joint text embedding for personalized content-based recommendation. *arXiv preprint arXiv:1706.01084*.

Zheng Chen. 2023. Palr: Personalization aware llms for recommendation. *arXiv preprint arXiv:2305.07622*.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, et al. 2023a. Auggpt: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*.

Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023b. Uncovering chatgpt's capabilities in recommender systems. *arXiv preprint arXiv:2305.02182*.

Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker. 2012. Linked open data to support content-based recommender systems. In *Proceedings of the 8th international conference on semantic systems*, pages 1–8.

Horatiu Dumitru, Marek Gibiec, Negar Hariri, Jane Cleland-Huang, Bamshad Mobasher, Carlos Castro-Herrera, and Mehdi Mirakhorli. 2011. On-demand feature recommendations derived from mining public product descriptions. In *Proceedings of the 33rd international conference on software engineering*, pages 181–190.

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*.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 299–315.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Shashank Gupta and Vasudeva Varma. 2017. Scientific article recommendation by using distributed representations of text and graph. In *Proceedings of the 26th international conference on world wide web companion*, pages 1267–1268.

F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19.

Oktie Hassanzadeh and Mariano P Consens. 2009. Linked movie data base. In *LDOW*.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182.

Umair Javed, Kamran Shaukat, Ibrahim A Hameed, Farhat Iqbal, Talha Mahboob Alam, and Suhuai Luo. 2021. A review of content-based and context-based recommendation systems. *International Journal of Emerging Technologies in Learning (iJET)*, 16(3):274–306.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Chen Li, Yixiao Ge, Jiayong Mao, Dian Li, and Ying Shan. 2023a. Taggpt: Large language models are zero-shot multimodal taggers. *arXiv preprint arXiv:2304.03022*.

Chunyuan Li, Haotian Liu, Liunian Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Houdong Hu, Zicheng Liu, Yong Jae Lee, et al. 2022. Elevater: A benchmark and toolkit for evaluating language-augmented visual models. *Advances in Neural Information Processing Systems*, 35:9287–9301.

Ruyu Li, Wenhao Deng, Yu Cheng, Zheng Yuan, Jiaqi Zhang, and Fajie Yuan. 2023b. Exploring the upper limits of text-based collaborative filtering using large language models: Discoveries and insights. *arXiv preprint arXiv:2305.11700*.

Yize Li, Jiazhong Nie, Yi Zhang, Bingqing Wang, Baoshi Yan, and Fuliang Weng. 2010. Contextual recommendation based on text mining. In *Coling 2010: Posters*, pages 692–700.

Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, et al. 2023. How can recommender systems benefit from large language models: A survey. *arXiv preprint arXiv:2306.05817*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning.

Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. 2023b. Is chatgpt a good recommender? a preliminary study. *arXiv preprint arXiv:2304.10149*.

Peter Lofgren. 2015. *Efficient algorithms for personalized pagerank*. Stanford University.

Pasquale Lops, Dietmar Jannach, Cataldo Musto, Toine Bogers, and Marijn Koolen. 2019. Trends in content-based recommendation: Preface to the special issue on recommender systems based on rich item descriptions. *User Modeling and User-Adapted Interaction*, 29:239–249.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. 2019. Generating personalized recipes from historical user preferences. *arXiv preprint arXiv:1909.00105*.

Cataldo Musto, Pierpaolo Basile, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2017. Introducing linked open data in graph-based recommender systems. *Information Processing & Management*, 53(2):405–435.

Cataldo Musto, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2018. Semantics-aware recommender systems exploiting linked open data and graph-based features. In *Companion Proceedings of the The Web Conference 2018*, pages 457–460.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197.

Michael Oppermann, Robert Kincaid, and Tamara Munzner. 2020. Vizcommender: Computing text-based similarity in visualization repositories for content-based recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):495–505.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web: methods and strategies of web personalization*, pages 325–341. Springer.

Luis G Perez, Manuel Barranco, and Luis Martinez. 2007. Building user profiles for recommender systems from incomplete preference relations. In *2007 IEEE International Fuzzy Systems Conference*, pages 1–6. IEEE.

Matthew E Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting contextual word embeddings: Architecture and representation. *arXiv preprint arXiv:1808.08949*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Damien Poirier, Isabelle Tellier, Françoise Fessant, and Julien Schluth. 2010. Towards text-based recommendations. In *RIAO 2010: 9th international conference on Adaptivity, Personalization and Fusion of Heterogeneous Information*, pages 0–0.

Jipeng Qiang, Zhenyu Qian, Yun Li, Yunhao Yuan, and Xindong Wu. 2020. Short text topic modeling techniques, applications, and performance: a survey. *IEEE Transactions on Knowledge and Data Engineering*, 34(3):1427–1445.

Raf. 2023. How do text-davinci-002 and text-davinci-003 differ? *OpenAI*.

Daniel Ramage, Susan Dumais, and Dan Liebling. 2010. Characterizing microblogs with topic models. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 4, pages 130–137.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Noveen Sachdeva and Julian McAuley. 2020. How useful are reviews for recommendation? a critical review and potential improvements. In *proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 1845–1848.

Kashfia Sailunaz and Reda Alhajj. 2019. Emotion and sentiment analysis from twitter text. *Journal of Computational Science*, 36:101003.

Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1161–1170.

Jianshan Sun, Gang Wang, Xusen Cheng, and Yelin Fu. 2015. Mining affective text to improve social media item recommendation. *Information Processing & Management*, 51(4):444–457.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lei Wang and Ee-Peng Lim. 2023. Zero-shot next-item recommendation using large pretrained language models. *arXiv preprint arXiv:2304.03153*.

Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable recommendation via multi-task learning in opinionated text data. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 165–174.

Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, pages 1–7.

Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the web conference 2021*, pages 1785–1797.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM international conference on multimedia*, pages 1437–1445.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. MIND: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606, Online. Association for Computational Linguistics.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, Rui Zhang, and Yong Yu. 2023. Towards open-world recommendation with knowledge augmentation from large language models. *arXiv preprint arXiv:2306.10933*.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 974–983, New York, NY, USA. Association for Computing Machinery.

Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Recommendation as instruction following: A large language model empowered recommendation approach. *arXiv preprint arXiv:2305.07001*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Additional Details of Experiment Setup

A.1 Datasets

Dataset	# Interaction	# Item	# User
MovieLens-1M	1,000,209	3,706	6,040
Recipe	132,246	4,125	2,589

Table 6: Benchmark Statistics.

MovieLens-1M (Harper and Konstan, 2015) is a highly recognized benchmark dataset commonly used for evaluating item recommendation systems.[1] It contains a vast collection of 1,000,209 ratings provided by 6,040 MovieLens users, covering 3,900 movies. Each user has at least 20 ratings. Following He et al. (2017), we convert the rating data into implicit feedback. More specifically, each entry is marked as 0 or 1 indicating whether the user has rated the corresponding item. The original movie data only contain movie titles and genres. We employ GPT-3 (`text-davinci-003`) to generate the content description of each movie using the following prompt: *"Summarize the movie* `{title}` *with one sentence. The answer cannot include the movie title."* The response from GPT-3 is used as the item description. `Temperature` is set at 0 to generate more focused and deterministic responses. Note that inclusion of the movie title is entirely *optional*. We opt not to include the title intentionally, as our design for LLM-Rec emphasizes its role as a general prompting framework. This versatility is important, as it is intended to function across a wide array of item types, including those that may not possess pre-defined titles, such as short videos.

Recipe (Majumder et al., 2019) is another benchmark dataset we use to assess the recommendation performance. This dataset consists of recipe details and reviews sourced from `Food.com`.[2] The metadata includes ratings, reviews, recipe names, descriptions, ingredients, directions, and so on. For instance, an example recipe description is *"All the delicious flavors of mac n' cheese transformed into a warm, comforting bowl of soup!"*. In our evaluation, we employ the recipe descriptions as item descriptions for the four prompting strategies. Similar to the MovieLens-1M dataset, we apply filtering criteria, excluding users with fewer than 20 ratings and items with fewer than 30 ratings. Note that all original descriptions presented as examples in this paper have been paraphrased to protect user privacy.

The selection of these benchmarks is mainly motivated by two factors. First, we select movies and recipes as they represent two distinct types of content. Movies, being more categorically organized and widely researched, contrast sharply with recipes, which are diverse, user-generated content from social media platforms, often lacking a strict categorical structure and presenting more novelty. Second, the nature of their descriptions differs significantly: movie descriptions typically comprise narrative summaries, whereas recipe descriptions are instructional. Evaluating our model on these varied datasets allows for a comprehensive analysis of how different prompting strategies affect recommendation outcomes, providing valuable insights into their effectiveness across diverse content types.

A.2 Baselines

To assess LLM-Rec's efficacy, we compare it against two distinct categories of baselines. The first category includes baselines that takes solely the original item descriptions as input. This includes models from MLP to more complex content-based approaches. Specifically, we choose three more advanced, content-based recommendation models. AutoInt is a multi-head self-attentive neural network with residual connections designed to explicitly model feature interactions within a low-dimensional space (Song et al., 2019). DCN-V2 represents an enhanced version of DCN and incorporates feature crossing at each layer (Wang et al., 2021, 2017). Lastly, EDCN (Chen et al., 2021) introduces a bridge module and a regulation module to collaboratively capture layer-wise interactive signals and learn discriminative feature distributions for each hidden layer in parallel networks, such as DCN. The purpose of this comparison is to evaluate the added value of augmented text in improving recommendation outcomes.

The second category includes different text augmentation methodologies. Here, LLM-Rec is evaluated against two recent advancements in the field of using LLMs to augment item information. The first method is Knowledge Augmented Recommendation (KAR) as proposed by Xi et al. (2023). KAR introduces a fusion of domain knowledge and prompt engineering to generate factual knowledge pertaining to the items (for implementation details, see Appendix A.8) In contrast to KAR's approach,

[1]License: `https://files.grouplens.org/datasets/movielens/ml-1m-README.txt`

[2]License: `https://www.kaggle.com/datasets/shuyangli94/food-com-recipes-and-user-interactions`

LLM-REC places a particular emphasis on the innate common-sense reasoning capabilities of large language models and notably does not mandate domain expertise. Since the augmented information may not necessarily be correct, we further implement a variant with ground truth knowledge. It aligns with strategies akin to those introduced by Di Noia et al. (2012), who harnessed external databases to enhance item information. The second method, TagGPT, proposed by Li et al. (2023a), extracts tags using LLMs, similar to one of our prompting strategies for item descriptions.

Although collaborative Filtering (CF) is another widely used technique in recommendation systems, given our primary focus on addressing the issue of incomplete item descriptions, we do not conduct experiments under CF settings (Li et al., 2023b). Instead, we concentrate on comparing our method with other input augmentation approaches.

A.3 Item and User Modules

We use Sentence-BERT (Reimers and Gurevych, 2019) to derive the textual embeddings from the original item description and augmented text. The embedding model is `all-MiniLM-L6-v2`. We directly apply it to convert the natural language to embeddings without fine-tuning it, freezing only the text-encoder and not the user encoder to avoid the additional computational cost associated with training and fine-tuning the sentence transformer model. For users, we employ an embedding table to convert user ID into latent representations. For both MovieLens-1M and Recipe datasets, the output dimension is set at 128. We have considered using embeddings derived from the LLMs. However, as our goal is to propose a general framework that can leverage both open-source and proprietary models, we do not pursue further exploration of this aspect in the current study.

A.4 Model Training

To facilitate the model training process, we employ the binary cross-entropy loss, expressed as:

$$L = -\sum_{(u,i)\in Y} [y_{u,i} \cdot \log \hat{y}_{u,i} + (1 - y_{u,i}) \cdot \log(1 - \hat{y}_{u,i})] \quad (1)$$

where (u, i) represents the user-item pair, and Y denotes the set that contains all positive and negative samples. The variable $y_{u,i}$ serves as a label, with a value of 1 indicating that user u has engaged with item i, and 0 representing the absence of interaction. The prediction score $\hat{y}_{u,i}$, ranging from 0 to 1, reflects the likelihood of user u interacting with item i. In our dataset, each instance of user-item interaction is considered a positive sample. Alongside these positive samples, we incorporate negative samples by randomly pairing users with items that lack any prior recorded interactions. To mitigate the risk of overfitting and enhance training efficiency, we implement an early stopping mechanism. The window size and evaluation frequency are both configured to be 5. It is noteworthy that we have also explored the viability of employing the Bayesian Personalized Ranking (BPR) Loss (Rendle et al., 2012) within our framework. However, subsequent experimentation reveals that the BPR Loss does not offer superior performance when compared to the binary cross-entropy loss. Consequently, we opt to use the binary cross-entropy loss as our loss function.

A.5 Hyper-parameter Settings

Large Language Models. We perform experiments with two large language models. For GPT-3 (`text-davinci-003`), `temperature`, `max_token`, `top_p`, `frequency penalty`, and `presence penalty` are set as 0, 512, 1, 0.0, and 0.6, respectively. For LLAMA-2 (7B LLAMA-2-CHAT), we set `do_sample` to be `True`, `top_k` 10, and the `num_return_sequences` 1. LLAMA-2's generation is conducted on 8 NVIDIA GeForce RTX 2080 Ti GPUs, each equipped with 11 GB of memory.

Recommendation Modules We initialize the model parameters randomly, following a Gaussian distribution. To optimize the framework, we employ the AdamW algorithm (Loshchilov and Hutter, 2017) with a weight decay value of 0.0005. For the MLP model, the hyper-parameter grids for the learning rate and dropout rate are $\{0.0001, 0.0005, 0.001\}$ and $\{0.1, 0.3, 0.5\}$, respectively. For AutoInt (Song et al., 2019), the hyper-parameter grids for the learning rate, dropout rate, hidden layer size, number of attention layers, and attention heads are $\{0.001, 0.005, 0.01\}$, $\{0.1, 0.3, 0.5\}$, $\{16, 32, 64, 128\}$, $\{1, 2\}$, and $\{1, 2\}$, respectively. For DCN-V2 (Wang et al., 2021), the learning rate, dropout rate, hidden layer size, and number of cross layers are searched in $\{0.001, 0.005, 0.01\}$, $\{0.1, 0.3, 0.5\}$, $\{16, 32, 64, 128\}$, $\{1, 2\}$, and $\{1, 2\}$, respectively. Since the network structure of EDCN (Chen et al., 2021) is similar with DCN-V2 (Wang et al., 2021), we apply the hyper-parameter settings of DCN-V2 to EDCN. The performance is evaluated every five epochs, and the early stop mechanism is configured to have a patience of 5. We set the batch size to 4096 for all baselines except for AutoInt which is 1024 due to the memory limitation. Settings that achieve the highest Recall@K on the validation set are chosen for the evaluation on the testing set.

A.6 Importance Measurement for Engagement-guided Prompting

In our study, we show an example of using Personalized PageRank (PPR) (Brin, 1998) score as the metric to find the important neighbor items. PPR is a widely employed technique for finding significant neighbors in recommendation systems (Ying et al., 2018). In particular, we first construct the user-item bipartite graph $G = (V, E)$. In this notation, G represents the bipartite graph, E denotes the set of nodes, and E represents the set of edges. There are two types of nodes including users $V_{user} \subset V$ and items $V_{item} \subset V$ ($V_{user} \cup V_{item} = V, V_{user} \cap V_{item} = \varnothing$). An edge $e \in E$ between a user node $v \in V_{user}$ and an item node $i \in V_{item}$ is created if this user interacts with this item.

Next, we proceed by calculating the Personalized PageRank (PPR) score for each item node. The PPR value $\pi(s, t)$, where s is the source node and t is the target node, signifies the probability that a random walk initiated from node s concludes at node t. This value offers a quantified measure of their relative importance from the standpoint of an individual node (Lofgren, 2015). For every item node, we construct a set of significant neighboring items. By identifying the top T item nodes with the highest PPR scores, we pinpoint important neighbor items guided by user engagement. The rationale behind this approach lies in the observation that when users frequently engage with two items, there tends to be a greater similarity between these two items through the lens of user preferences. By incorporating this information, we aim to capture user preferences more effectively, leading to enhanced performance in content recommendation. For both datasets, we set $T = 3$. For the Movielens-1M dataset, we find the important neighbor items that share the same genre as the target item.

A.7 Implementation Details

Our methods are implemented and experiments are conducted using PyTorch. The computation of PPR scores is facilitated by the use of the `torch-ppr` library. Each experiment is run on one NVIDIA A100 GPU with 80 GB of memory at a time. Further, we adapt the codes of the DeepCTR[3] and DeepCTR-Torch[4] repositories to implement AutoInt (Song et al., 2019), DCN-V2 (Wang et al., 2021), and EDCN (Chen et al., 2021). Table 7 summarizes all prompts of LLM-REC.

A.8 KAR Augmentation Details

In KAR, Xi et al. (2023) applied a specific prompt to elicit factual knowledge about movies of the Movielens-1M dataset (Harper and Konstan, 2015). The prompt instructed the model to: *"Introduce movie* `{item description}` *and describe its attributes precisely (including but not limited to* `scenario-specific factors`*)"*. In their study, the `item description` was the movie titles. Human experts were enlisted to refine the answers generated by LLMs in response to the question: *"List the importance factors or features that determine whether a user will be interested in a movie."* These refined factors were then considered as the `{scenario-specific factors}`, including *genre, actors, directors, theme, mood, production quality, and critical acclaim*. Because the responses generated using these prompts were not publicly released, we re-implement the same methodology, employing LLMs to generate the factual knowledge of items. In the case of the Recipe dataset (Majumder et al., 2019), we use recipe description as the `item description`. The same approach is then adopted to identify `scenario-specific factors`. Initially, the prompt is adapted to: *"List the importance factors or features that determine whether a user will be interested in a recipe."* Subsequently, the answers generated by CHATGPT are validated (see Table 8). The resulting set of `scenario-specific factors` for Recipe comprises a diverse range of attributes, including *dietary preferences, ingredients, cuisine type, cooking time, nutritional value, allergies, taste preferences, skill level, occasion, cost, health and wellness goals, food allure, reviews and ratings, cooking equipment, personal experience, season and weather, cultural or ethical considerations, creativity and variety, simplicity, popularity and trends*. These prompts are then employed to enrich the factual knowledge of both movies and recipes using GPT-3 (`text-davinci-003`). For illustrative examples of the responses, please refer to Table 9.

[3] https://github.com/shenweichen/DeepCTR
[4] https://github.com/shenweichen/DeepCTR-Torch

p_{para}	*"The description of an item is as follows* '{description}', *paraphrase it."*
p_{tag}	*"The description of an item is as follows* '{description}', *summarize it with tags."*
p_{infer}	*"The description of an item is as follows* '{description}', *what kind of emotions can it evoke?"*
p^{rec}_{para}	*"The description of an item is as follows* '{description}', *what else should I say if I want to recommend it to others?"*
p^{rec}_{tag}	*"The description of an item is as follows* '{description}', *what tags should I use if I want to recommend it to others?"*
p^{rec}_{infer}	*"The description of an item is as follows* '{description}', *recommend it to others with a focus on the emotions it can evoke."*
p^{eng}	*"Summarize the commonalities among the following descriptions:* '{description}'; '{descriptions of other important neighbors}'."
$p^{rec+eng}$	*"The description of an item is as follows:* '{description}'. *What else should I say if I want to recommend it to others? This content is considered to hold some similar attractive characteristics as the following descriptions:* '{descriptions of other important neighbors}'."

Table 7: LLM-REC prompts.

KAR is also composed of a preference reasoning prompt for user information augmentation. Since we only focus on the item side, only the item factual prompt is implemented to examine how different focuses on LLMs' ability between LLM-REC and KAR affect recommendation performance.

A.9 Keywords Construction

The keyword generation process differs between the Movielens-1M and Recipe datasets. For Movielens-1M, the keywords are derived from genre labels, which are intrinsic components of the dataset's metadata. In the case of Recipe, the process involves multiple steps. Initially, we compile a list of unique words found in the responses generated through the recommendation-driven prompting strategy. Subsequently, we filter out stopwords and proceed to construct unigrams and bigrams using the NLTK package. Following this, a manual review is conducted to identify phrases that appear at least five times in the corpus. These phrases are then scrutinized to determine whether they contain words relevant for categorizing recipes. The final list of keywords for Recipe contain "easy", "homemade", "baking", "health", "healthy", "dessert", and "dinner". These keywords collectively serve as indicative descriptors for recipes within the dataset.

The factors or features that determine whether a user will be interested in a recipe can vary from person to person, but some important factors commonly include: **1. Dietary Preferences:** Whether the recipe aligns with the user's dietary restrictions, such as vegetarian, vegan, gluten-free, or keto. **2. Ingredients:** The availability and appeal of the ingredients used in the recipe. **3. Cuisine Type:** Whether the recipe falls within a cuisine the user enjoys, like Italian, Mexican, or Asian. **4. Cooking Time:** The user's available time for cooking, as some may prefer quick and easy recipes, while others enjoy longer cooking processes. **5. Nutritional Value:** Users may consider the nutritional content of the recipe, such as calorie count, protein, or fiber. **6. Allergies:** Any known food allergies that need to be accommodated in the recipe. **7. Taste Preferences:** Whether the flavors and taste profile of the recipe match the user's preferences. **8. Skill Level:** Some users may look for recipes that match their cooking skills, such as beginner-friendly or gourmet. **9. Occasion:** If the recipe is suitable for a particular occasion, like a holiday, party, or a regular weekday meal. **10. Cost:** The affordability of the ingredients and whether it fits within the user's budget. **11. Health and Wellness Goals:** Recipes that align with specific health or weight management goals, like low-carb or heart-healthy. **12. Food Allure:** The visual appeal and presentation of the dish through images and descriptions. **13. Reviews and Ratings:** User reviews and ratings on the recipe, which can indicate its popularity and quality. **14. Cooking Equipment:** Whether the user has the necessary kitchen tools and appliances for the recipe. **15. Personal Experience:** Previous experiences with similar recipes or dishes may influence interest. **16. Season and Weather:** Seasonal ingredients or dishes that match the current season or weather. **17. Cultural or Ethical Considerations:** Recipes that respect cultural, ethical, or environmental values. **18. Creativity and Variety:** The desire for trying something new and innovative in cooking. **19. Simplicity:** Some users may prefer straightforward, minimal-ingredient recipes. **20. Popularity and Trends:** Trending recipes or those recommended by friends and family may pique a user's interest. These factors can significantly impact a user's decision to explore a recipe, and personalized recommendation systems often consider several of these factors to suggest recipes tailored to individual preferences.

Table 8: Response generated by CHATGPT about the question *"List the importance factors or features that determine whether a user will be interested in a recipe."*

Prompt: Introduce movie Batman Forever and describe its attributes precisely (including but not limited to genre, actors, directors, theme, mood, production quality, and critical acclaim) **Response:** Batman Forever is a 1995 American superhero film directed by Joel Schumacher and produced by Tim Burton. It is the third installment in the Batman film series, based on the DC Comics character of the same name. The film stars Val Kilmer as Bruce Wayne/Batman, alongside Tommy Lee Jones, Jim Carrey, Nicole Kidman, Chris O'Donnell, Michael Gough, and Pat Hingle. The film follows Batman as he battles Two-Face and The Riddler, both of whom are determined to wreak havoc in Gotham City. The film's theme is the struggle between good and evil, and the power of redemption. The mood of the film is dark and brooding, with a hint of humor. The production quality of Batman Forever is high, with impressive special effects and cinematography. The film was well-received by critics, who praised its action sequences, performances, and visual style. It was a box office success, grossing over $336 million worldwide.
Prompt: Introduce recipe 'Numerous Amish bread recipes have been shared, but none seem to include the starter. Therefore, I thought it would be helpful to provide this. Happy baking! Here's the Amish Friendship Bread and its starter.' and describe its attributes precisely (including but not limited to dietary preferences, ingredients, cuisine type, cooking time, nutritional value, allergies, taste preferences, skill level, occasion, cost, health and wellness goals, food allure, reviews and ratings, cooking equipment, personal experience, season and weather, cultural or ethical considerations, creativity and variety, simplicity, popularity and trends) **Response:** Amish Friendship Bread is a sweet, yeasty bread that is made with a starter that is passed from person to person. It is a traditional Amish recipe that has been around for generations. Dietary Preferences: Vegetarian, Gluten-Free (if using gluten-free flour) Ingredients: All-purpose flour, sugar, active dry yeast, milk, butter, eggs, vanilla extract, cinnamon, and salt. Cuisine Type: American Cooking Time: 10 minutes to prepare the starter, 1 hour to prepare the dough, and 1 hour to bake the bread.Nutritional Value: High in carbohydrates, low in fat and protein. Allergies: Contains dairy, eggs, and wheat. Taste Preferences: Sweet, yeasty, and slightly spiced. Skill Level: BeginnerOccasion: Breakfast, brunch, or snack. Cost: Low Health and Wellness Goals: A healthier alternative to store-bought breads. Food Allure: Comfort food that is easy to make. Reviews and Ratings: Generally positive reviews. Cooking Equipment: Large bowl, measuring cups and spoons, wooden spoon, loaf pan, and oven. Personal Experience: Delicious and easy to make. Season and Weather: Can be made year-round. Cultural or Ethical Considerations: Traditional Amish recipe. Creativity and Variety: Can be adapted to include different flavors and ingredients. Simplicity: Easy to make with basic ingredients. Popularity and Trends: Popular among home bakers.

Table 9: Response generated via the item factual prompt proposed by Xi et al. (2023).

B Additional Analysis

B.1 Additional Discussions on Augmented Text

Overall, we observe that LLM-REC effectively elicits LLMs to enrich item descriptions with additional information. This augmentation results in descriptions that are not only more detailed but also more expressive, enhancing the original item descriptions. Tables 11 and 12 show example responses generated by GPT-3 comparing the recommendation-driven and basic promptings in terms of `tag` and `infer`. Tables 13, 14, and 15 show example responses generated by LLAMA-2 comparing the recommendation-driven and basic promptings. Responses from the recommendation prompting strategies provides several additional pieces of information and context compared to the responses from the basic prompting strategies. Take the responses in Table 11 as an example, the recommendation prompting strategy introduces new themes like "Supernatural", "Paranormal", and "Psychological Thriller", which are not present in the responses from the basic promptings. These themes suggest a broader and more specific context for the story, indicating not just communication with the dead, but also elements of horror and suspense. The term "Troubled Child" adds a new dimension to the "Young Boy" mentioned in the first sentence, suggesting that the child's character may face internal conflicts or challenges.

What extra information does engagement-guided strategy prompt LLMs to augment? Consistent with our previous experiments, we curate exemplary responses obtained from p^{eng} for closer examination (Figure 6). Our analysis reveals a distinct pattern compared to what we have observed with recommendation-driven prompting. There are primarily two scenarios to consider. First, if the descriptions of the important neighbor items and the target items exhibit high similarity, the impact of p^{eng} resembles that of p_{para}, as exemplified in the second Recipe example in Figure 6. Second, p^{eng} guides LLMs to generate additional information, which may be derived from the descriptions of the important neighbor items. Consequently, how the engagement-guided strategy influences LLMs' text generation—whether it aligns with one of the behaviors we have described, both of them, or even other unexplored patterns—largely depends on the composition of the important neighbor items. This composition, in turn, is contingent on the neighbor sampling method which is out of the scope of our study. We leave a more in-depth exploration of this topic to future research endeavors.

Interestingly, the recommendation-driven + engagement-guided prompting strategy is able to generate text that shares similar characteristics with both sub-strategies. How they quantitatively form the final generation remains an open question. Examples can be found in Table 17.

Table 16 shows example responses of LLAMA-2 to the engagement-guided prompting strategy. Table 17 shows example responses of GPT-3 to the recommendation-driven and engagement-guided prompting strategy. Overall, the 7B LLAMA-2-CHAT performs poorly compared to GPT-3. In some cases, there is no generated content as we have also observed in Appendix D.

B.2 Additional Discussions on Applicable Item Domains and Available Textual Information

The applicability of LLM-REC beyond movies and recipes, particularly in domains with sparse textual information, remains a question. To address this, we conduct an analysis to determine LLM-REC's efficacy in enriching text across various domains and text lengths. We use item descriptions from ten distinct domains in the Amazon review dataset (Ni et al., 2019), which includes product metadata like descriptions, category, price, brand, and image features. The selected domains are all beauty, appliances, automotive, digital music, grocery and gourmet food, pet supplies, sports and outdoors, video games, magazine subscriptions, and industrial and scientific. For each domain, we sample 50 items and prompt GPT-3 using p_{para}^{rec} (*i.e.*, recommendation-driven prompting).

Our previous discussions highlight that the most valuable information for improving recommendation performance typically aligns with expressive words pertinent to item characteristics. While no single metric directly quantifies this added information, we use the increase in the number of adjectives as a proxy. Additionally, the total word count serves as a straightforward metric to approximate the volume of augmented information. By comparing the augmented texts with the original item descriptions, we calculate the percentage increase in the number of adjectives. Note that the adjective increase is computed as a ratio of the difference in adjective count to the original word count.

As Figure 1 demonstrates, LLM-REC effectively enriches item descriptions across multiple domains, including those lacking in rich textual content. For instance, the average word counts in movie and digital music descriptions are only 20.34 and 30.18 words, respectively. LLM-REC enhances expressiveness, with a notable increase in the use of adjectives.

B.3 Additional Experiments on Applying LLM-Rec to Other Baselines

We extend the application of LLM-REC to other text-based recommendation systems and replicate the experiments. The results, as presented in Table 10, indicate that LLM-REC can be easily adapted to various text-based recommendation systems and generally enhances recommendation performance compared to using the original text.

B.4 Additional Discussions on Integration Process

In our setup, the text encoder is frozen (not fine-tuned), with a fixed output dimension for all vectors. The fundamental difference between the `Concat-All` and Text Concatenation methods lies in their processing sequence. The `Concat-All` method initially transforms individual text segments into embeddings and subsequently concatenates these embeddings. In contrast, the Text Concatenation method first concatenates the text segments and then converts this combined text into a single embedding.

The observed superiority of the `Concat-All` method can be attributed to how these processes handle information density. When lengthy text segments are concatenated before encoding, there is a higher likelihood of **information loss**, particularly given the constraints of a frozen text encoder. This encoder, **not** being fine-tuned for the specific nuances of our data, may struggle to effectively capture and retain crucial information from longer text inputs. Therefore, processing shorter text segments individually before concatenation (as in `Concat-All`) may help in preserving important features and nuances in the embeddings.

B.5 Additional Discussions on Prompt Design

To investigate whether modifying the word choice in the designed prompts significantly affects the augmented output, we construct one variant prompt for each of LLM-REC's prompts, ensuring they convey the same meaning but with different word choices. Take p_{para} as an example, p_{para} is *"The description of an item is as follows '{*`description`*}', paraphrase it."*. One variant is *"Summarize the given item description, '{*`description`*}', using different words."* Next, we randomly sample 50 items from Movielens-1M, and prompt GPT-3 with these variants. The cosine similarity between the responses generated from the variant prompt and LLM-REC's prompt is computed and shown in Table 5. Tables 18-25 shows the example responses.

B.6 Additional Discussions on Dynamic Prompts

The concept of dynamic prompts in recommendation systems is an intriguing area that holds the potential for enhancing personalization. By incorporating descriptions of a user's most recently interacted items into prompts, the system can generate item descriptions on-the-fly that are more closely aligned with the user's current interests and preferences. This approach could lead to more precise and tailored recommendations, as the generated descriptions would reflect the user's evolving tastes.

One of the primary considerations is the computational cost associated with generating dynamic prompts. Each user interaction would require real-time processing to update the prompt, which could be resource-intensive, especially for large-scale systems with many users and items.

To mitigate computational costs, several strategies can be employed. Developing efficient algorithms for prompt generation and item description generation can help mitigate computational costs. Implementing caching mechanisms for frequently accessed data can reduce the processing time required for updating prompts. Instead of completely regenerating prompts after each interaction, the system could employ incremental updates to modify prompts based on recent changes in user behavior.

While the implementation of dynamic prompts presents several challenges, it also offers a promising avenue for enhancing personalization in recommendation systems. With careful consideration, this approach has the potential to cater more effectively to individual user needs.

C Extended Related Work

Augmentation in Text-based Recommendation. Traditionally, researchers have advocated the augmentation of item descriptions through the incorporation of external knowledge sources (Di Noia et al.,

	Movielens-1M			Recipe		
	Precision@10	Recall@10	NDCG@10	Precision@10	Recall@10	NDCG@10
AutoInt (Song et al., 2019)	0.2149 ±0.0078	0.1706 ±0.0075	0.2698 ±0.0092	0.0351 ±0.0032	0.0772 ±0.0102	0.0658 ±0.0089
- Basic	0.2590 ±0.0038	0.2193 ±0.0049	0.3224 ±0.0052	0.0361 ±0.0030	0.0797 ±0.0097	0.0664 ±0.0076
- Rec	0.2593 ±0.0035	0.2197 ±0.0068	0.3242 ±0.0059	0.0357 ±0.0029	0.0794 ±0.0096	0.0660 ±0.0079
- Eng	0.2323 ±0.0011	0.1857 ±0.0032	0.2899 ±0.0031	0.0349 ±0.0028	0.0764 ±0.0090	0.0642 ±0.0072
- Rec+Eng	0.2620 ±0.0021	0.2230 ±0.0037	0.3270 ±0.0022	0.0349 ±0.0029	0.0759 ±0.0099	0.0647 ±0.0075
DCN-V2 (Wang et al., 2021)	0.2961 ±0.0050	0.2433 ±0.0057	0.3689 ±0.0033	0.0360 ±0.0036	0.0786 ±0.0104	0.0653 ±0.0085
- Basic	0.3083 ±0.0033	0.2601 ±0.0051	0.3842 ±0.0054	0.0365 ±0.0028	0.0802 ±0.0093	0.0658 ±0.0084
- Rec	0.3062 ±0.0031	0.2572 ±0.0049	0.3831 ±0.0041	0.0362 ±0.0035	0.0794 ±0.0108	0.0670 ±0.0095
- Eng	0.2990 ±0.0024	0.2496 ±0.0020	0.3725 ±0.0021	0.0356 ±0.0032	0.0786 ±0.0094	0.0647 ±0.0076
- Rec+Eng	0.3114 ±0.0021	0.2641 ±0.0038	0.3882 ±0.0028	0.0357 ±0.0034	0.0793 ±0.0104	0.0654 ±0.0083

Table 10: Average recommendation performance by applying LLM-REC to other text-based recommendation modules across five different train/test splits.

2012; Musto et al., 2018; Sachdeva and McAuley, 2020). Notably, Di Noia et al. (2012) harnesse data from external databases such as `DBpedia` (Bizer et al., 2009), `Freebase` (Bollacker et al., 2008), and `LinkedMDB` (Hassanzadeh and Consens, 2009) to gather comprehensive information pertaining to movies, including details about actors, directors, genres, and categories. This approach aimed to enrich the background knowledge available to movie recommender systems. The explicit semantics embedded in these external knowledge sources have demonstrated a discernible enhancement in recommendation performance (Musto et al., 2017). However, this process necessitates a profound domain expertise to effectively and efficiently select and leverage the precise database, ensuring the incorporation of genuinely valuable information into item descriptions (Dumitru et al., 2011).

LLM for Recommendation. The use of large language models in recommender systems has garnered significant attention in recent research (Lin et al., 2023; Chen et al., 2023). Many studies have explored the direct use of LLMs as recommender models. The underlying principle of these approaches involves constructing prompts that encompass the recommendation task, user profiles, item attributes, and user-item interactions. These task-specific prompts are then presented as input to the LLMs, which is instructed to predict the likelihood of interaction between a given user and item (Dai et al., 2023b; Gao et al., 2023; Geng et al., 2022; Li et al., 2023b; Liu et al., 2023b; Zhang et al., 2023). For instance, Wang and Lim (2023) designed a three-step prompting strategy to directly guide LLMs to capture users' preferences, select representative previously interacted items, and recommend a ranked list of 10 items. While these works demonstrate the potential of LLMs as powerful recommender models, the focus primarily revolves around utilizing the LLMs directly for recommendation purposes. However, a potential issue of these methods is that LLMs may generate predictions merely from memorizing training samples which poses a challenge for conducting effective evaluations. In this study, we approach the problem from a different perspective. Rather than using LLMs as recommender models, this study explores diverse prompting strategies to *augment input text* with LLMs for personalized content recommendation. The actual recommendation process still relies on existing recommendation methodologies.

D Example Responses

Tables 29 and 30 show example responses by GPT-3 and the 7B LLAMA-2-CHAT on Movielens-1M (Harper and Konstan, 2015) and Recipe (Majumder et al., 2019). Augmented components are highlighted (recommendation-driven: blue; engagement-guided: green; rec+eng: orange). In summary, both GPT-3 and LLAMA-2 exhibit the capability to enrich item descriptions with supplementary information. Nevertheless, the LLAMA-2-CHAT model with its 7B parameters demonstrates comparatively poorer performance, which could be attributed to its limited parameter scale. This limitation offers insight into the diminished recommendation quality when using LLAMA-2 responses in contrast to GPT-3. Future research endeavors should focus on optimizing the LLM-REC framework, particularly concerning the selection of different large language models as backbones, to enhance recommendation outcomes.

Description	A troubled child psychologist helps a young boy who is able to see and communicate with the dead.
p_{tag}	#Psychology #Child #YoungBoy #Dead #Communication
p_{tag}^{rec}	Tags: Child Psychology, Troubled Child, Young Boy, Communicate with Dead, Supernatural, Paranormal, Ghost Story, Psychological Thriller
Description	This unique Tex-Mex recipe was created by a distant relative who was stationed at the Alamo almost 160 years ago. Family stories say that he was searching for quality tomatoes when Santa Ana launched an attack on the now-iconic Texas landmark. Fortunately, our ancestor had hidden this recipe in his coon-skin cap's lining, ensuring that both he and his avocado dip survived to see another day. Years later, as he lay dying from a chihuahua bite infection, he gazed deeply into his son's eyes and, with his final breath, whispered the memorable words, "Remember... the... guacamole."
p_{tag}	#TexMex #Alamo #AvocadoDip #CoonSkinCap #SantaAna #Chihuahua #Guacamole
p_{tag}^{rec}	Tags: Tex-Mex, Alamo, Santa Ana, Avocado Dip, Chihuahua, Guacamole, Recipe, Family Lore

Table 11: Example responses to p_{tag} and p_{tag}^{rec} generated by GPT-3. The additional information augmented via the recommendation-driven prompting is highlighted in blue.

Description	A former Roman General is forced to become a gladiator and seek revenge against the corrupt emperor who murdered his family.
p_{infer}	This description can evoke a range of emotions, including anger, sadness, revenge, determination, and courage.
p_{infer}^{rec}	This item is sure to evoke strong emotions in its viewers. It tells the story of a former Roman General who is forced to become a gladiator and seek revenge against the corrupt emperor who murdered his family. It is a tale of tragedy, courage, and justice that will leave you feeling inspired and moved.
Description	Our garden's potatoes are ripe and ready, and this recipe is always the first one I go for! They're incredibly delicious!
p_{infer}	This description can evoke feelings of excitement, anticipation, joy, and satisfaction.
p_{infer}^{rec}	This recipe is sure to bring a smile to your face! The potatoes are fresh from the garden, and the flavor is unbeatable. Enjoy the feeling of accomplishment as you savor the deliciousness of these potatoes. You won't regret it!

Table 12: Example responses to p_{infer} and p_{infer}^{rec} generated by GPT-3. The additional information augmented via the recommendation-driven prompting is highlighted in blue.

Description	Two agents from a secret organization protect the Earth from alien threats while maintaining the public's ignorance of extraterrestrial life.
p_{para}	" Two government agents keep Earth safe from intergalactic dangers while concealing the truth about alien life from the public."
p_{para}^{rec}	"If you're a fan of action-packed sci-fi with a dash of humor, then this is the show for you! The agents are hilarious and the alien threats they face are so creative and thrilling. Plus, the way they manage to keep the public in the dark about alien life is just genius. It's like a sci-fi comedy version of 'Men in Black' with a lot more explosions. Give it a try and let me know what you think!"
Description	Indulge in something unique tonight. It's tasty and relatively simple to make. Plus, I usually have all the ingredients on hand.
p_{para}	Sure! Here's an example paraphrase of the original description
p_{para}^{rec}	If you're looking for something different and easy to make, try this recipe! It's good and has everything you need. I usually have these ingredients on hand, so it's a great option when you're feeling spontaneous. Give it a try and let me know what you think!

Table 13: Example responses to p_{para} and p_{para}^{rec} generated by LLAMA-2. The additional information augmented via the recommendation-driven prompting is highlighted in blue.

Description	An ant embarks on a journey to save his colony from the oppressive rule of a group of grasshoppers.
p_{tag}	Sure! Here is an example of how you could summarize the description of the item with tags
p_{tag}^{rec}	One tag you could use to recommend this book to others is "insects" or "animals in fiction." This tag would help others find the book if they are interested in stories that feature insects or animals as main characters.
Description	I adore this crock-pot chicken recipe for a couple of reasons: first, my family can't get enough of it, and second, it's incredibly simple to prepare! My sister shared this recipe with me. She has two kids, and they're big fans as well! Plus, if there are any leftovers, they're the best you'll ever have!
p_{tag}	Tags
p_{tag}^{rec}	If you want to recommend the recipe to others, you could use tags such as "chicken recipe", "easy recipe", "family-friendly recipe", "leftovers recipe", and "sister's recipe". Explanation

Table 14: Example responses to p_{tag} and p_{tag}^{rec} generated by LLAMA-2. The additional information augmented via the recommendation-driven prompting is highlighted in blue.

Description: A young police officer must race against time to save a bus full of passengers from a bomb planted by a vengeful criminal. p_{infer}: Suspense Explanation p_{infer}^{rec}: I would recommend this movie to others because it evokes a sense of urgency and suspense that is both thrilling and emotionally resonant. The scene where the young police officer races against time to defuse the bomb is heart-pumping and intense, and the emotional stakes are high as the officer works to save the passengers. The movie also explores themes of justice and revenge, making it a thought-provoking and emotionally impactful viewing experience. Overall, I would highly recommend this movie to anyone looking for a gripping and emotional thriller.
Description: Bored with the same old ground beef recipes? Give this spicy dish a try! If you prefer it extra saucy, feel free to double the sauce amount. Update: Since hoisin sauce is quite sweet, you might want to start with a small amount and gradually increase it to your taste! p_{infer}: 1. Desire p_{infer}^{rec}: "Hey fellow foodies! Have you tried this Spicy Beef and Noodles recipe? It's a game changer! The hoisin sauce adds a depth of flavor that will leave you wanting more, and the spicy kick is perfect for those who like a little heat in their meals. I made it for my family last night and they all loved it! Give it a try and let me know what you think! #foodie #delicious #homemade"

Table 15: Example responses to p_{infer} and p_{infer}^{rec} generated by LLAMA-2. The additional information augmented via the recommendation-driven prompting is highlighted in blue.

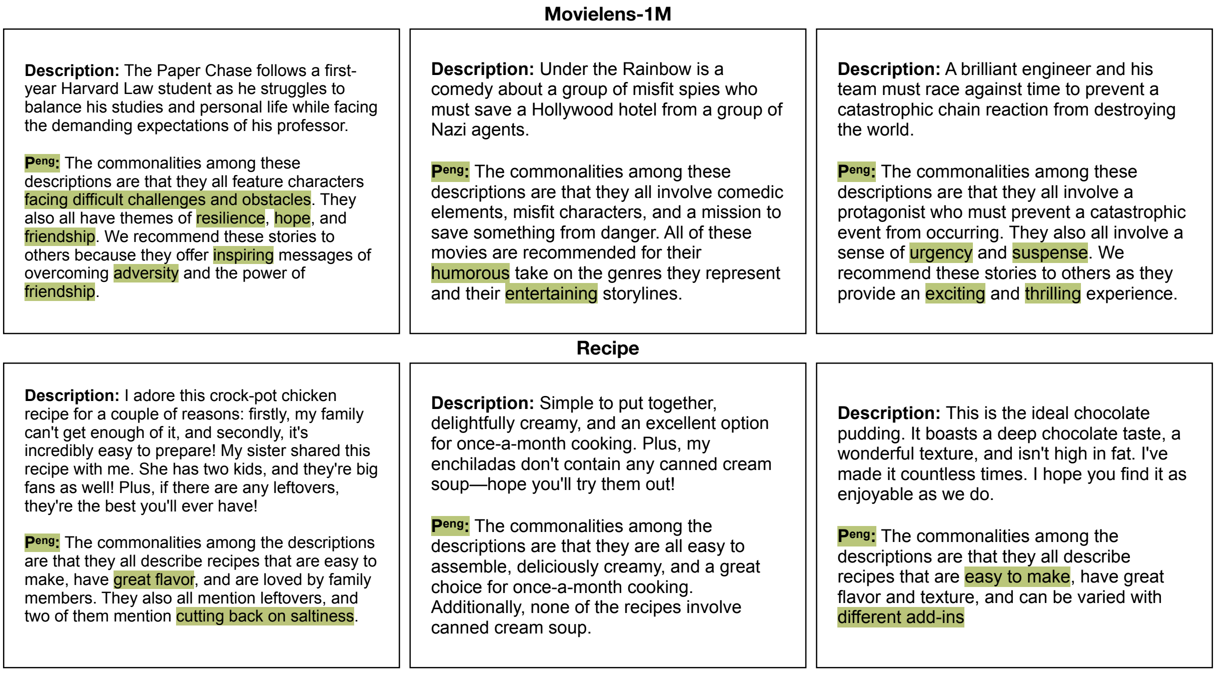

Figure 6: Example responses generated by GPT-3. The additional information augmented via the engagement-guided prompting is colored green. We choose the example responses generated by GPT-3 for illustration. Examples generated by LLAMA-2 can be found in Appendix B.1.

Description: A troubled child psychologist helps a young boy who is able to see and communicate with the dead. p^{eng}: Despite their different subjects and genres, the descriptions share a common theme of personal transformation and empowerment. Each story revolves around a character's journey towards self-discovery and self-acceptance, as they navigate challenges and obstacles that help them grow and develop as individuals.

Table 16: Example responses to p^{eng} generated by LLAMA-2. The additional information augmented via the engagement-guided prompting is highlighted in green.

Description: The Paper Chase follows a first-year Harvard Law student as he struggles to balance his studies and personal life while facing the demanding expectations of his professor.
$p^{rec+eng}$: If you're looking for a movie that will keep you on the edge of your seat, The Paper Chase is a great choice. It follows a first-year Harvard Law student as he navigates the pressures of his studies and personal life while facing the expectations of his professor. It's a story of determination and resilience in the face of adversity, and it will leave you feeling inspired and motivated.
Description: Our garden's potatoes are ripe and ready, and this recipe is always the first one I go for! They're incredibly delicious!
$p^{rec+eng}$: This potato recipe is a must-try! It's packed with flavor and so easy to make. Plus, you don't need to add any extra salt - the ranch dressing mix does the trick. It's a great way to get your family to enjoy potatoes in a new way. Plus, it's a great way to use up leftovers if you have any. Give it a try - you won't be disappointed!

Table 17: Example responses to $p^{rec+eng}$ generated by GPT-3. The additional information augmented via the engagement-guided prompting is highlighted in orange.

Description: In this Cold War-era thriller, a Soviet submarine captain attempts to defect to the United States with his state-of-the-art vessel.
p_{para}: The description of an item is as follows: '{description}', paraphrase it. **Response:** In this suspenseful movie set during the Cold War, a captain of a Soviet submarine attempts to defect to the US with his advanced submarine.
Variant Prompt: Summarize the given item description, '{description}', using different words. **Response:** This suspenseful movie from the Cold War period follows a Soviet submarine commander who attempts to defect to America with his advanced submarine.

Table 18: Example responses generated by GPT-3 for p_{para} and its variant that shares the same meaning but differs in word choices.

Description: In this Cold War-era thriller, a Soviet submarine captain attempts to defect to the United States with his state-of-the-art vessel.
p_{tag}: The description of an item is as follows: '{description}', summarize it with tags. **Response:** Cold War, Soviet, submarine, defect, United States, state-of-the-art.
Variant Prompt: Condense the provided item description '{description}' using tags. **Response:** #ColdWar #Thriller #SovietSubmarine #Defect #US #StateOfTheArtVessel

Table 19: Example responses generated by GPT-3 for p_{tag} and its variant that shares the same meaning but differs in word choices.

Description: In this Cold War-era thriller, a Soviet submarine captain attempts to defect to the United States with his state-of-the-art vessel.
p_{infer}: The description of an item is as follows: '{description}', what kind of emotions can it evoke? **Response:** This description can evoke feelings of suspense, tension, and excitement. It can also evoke feelings of patriotism and loyalty as the Soviet submarine captain attempts to defect to the United States.
Variant Prompt: Based on the description '{description}', what emotions might it elicit? **Response:** -Tension-Suspense-Fear-Anxiety-Dread-Unease-Paranoia

Table 20: Example responses generated by GPT-3 for p_{infer} and its variant that shares the same meaning but differs in word choices.

Description: In this Cold War-era thriller, a Soviet submarine captain attempts to defect to the United States with his state-of-the-art vessel.
p^{rec}_{para}: The description of an item is as follows: '{description}', what else should I say if I want to recommend it to others? **Response:** This Cold War-era thriller is a gripping and suspenseful story that will keep you on the edge of your seat. It follows the daring journey of a Soviet submarine captain as he attempts to defect to the United States with his state-of-the-art vessel. With its thrilling plot and captivating characters, this movie is sure to be an unforgettable experience.
Variant Prompt: In addition to the description '{description}', what other details would you include to endorse it to others? **Response:** This Cold War-era thriller follows the story of a Soviet submarine captain, Marko Ramius, who attempts to defect to the United States with his state-of-the-art vessel. Along the way, he must outwit and outmaneuver a relentless CIA analyst, Jack Ryan, who is determined to stop him. With intense action sequences and a gripping plot, this movie will keep you on the edge of your seat as the two men battle for control of the submarine. Featuring an all-star cast including Sean Connery, Alec Baldwin, and Scott Glenn, The Hunt for Red October is a thrilling and suspenseful ride that will leave you wanting more.

Table 21: Example responses generated by GPT-3 for p^{rec}_{para} and its variant that shares the same meaning but differs in word choices.

Description: In this Cold War-era thriller, a Soviet submarine captain attempts to defect to the United States with his state-of-the-art vessel.
p^{rec}_{tag}: The description of an item is as follows: '{description}', what tags should I use if I want to recommend it to others? **Response:** Tags: Cold War, Thriller, Soviet Submarine, Defection, United States, State-of-the-Art Vessel
Variant Prompt: If you aim to recommend '{description}' to others, what tags or terms would you use? **Response:** Tags/Terms: Cold War, Thriller, Soviet Submarine, Defection, United States, State-of-the-Art Vessel.

Table 22: Example responses generated by GPT-3 for p^{rec}_{tag} and its variant that shares the same meaning but differs in word choices.

Description: In this Cold War-era thriller, a Soviet submarine captain attempts to defect to the United States with his state-of-the-art vessel.
p^{rec}_{infer}: The description of an item is as follows: '{description}', recommend it to others with a focus on the emotions it can evoke. **Response:** This Cold War-era thriller will take you on an intense journey of suspense and intrigue. Follow a Soviet submarine captain as he attempts to defect to the United States with his state-of-the-art vessel, and experience the tension and drama of a high-stakes game of cat and mouse. Feel the fear and anticipation as the captain navigates a dangerous course of political and personal risk.
Variant Prompt: Using the description '{description}', endorse this item to others while emphasizing the emotions it can inspire. **Response:** This Cold War-era thriller will have you on the edge of your seat! Follow the gripping story of a Soviet submarine captain as he attempts to defect to the United States with his state-of-the-art vessel. Feel the tension and suspense as you watch this thrilling tale unfold. Don't miss out on this captivating story of courage and determination!

Table 23: Example responses generated by GPT-3 for p^{rec}_{infer} and its variants that shares the same meaning but differs in word choices.

Description: The Paper Chase follows a first-year Harvard Law student as he struggles to balance his studies and personal life while facing the demanding expectations of his professor.

Descriptions of other important neighbors: 'The Paper Chase follows a first-year Harvard Law student as he struggles to balance his studies and personal life while facing the demanding expectations of his professor.'; 'The Shawshank Redemption is a story of hope, friendship, and resilience in the face of despair.'; 'A criminal fakes insanity to escape prison, only to find himself in a mental institution where he must battle the oppressive head nurse.'

p^{eng}: Summarize the commonalities among the following descriptions: '{description}'; '{descriptions of other important neighbors}'.

Response: The commonalities among the descriptions are that they all involve characters facing difficult challenges and having to find a way to overcome them. In The Paper Chase, the protagonist must balance his studies and personal life while facing the demanding expectations of his professor. In The Shawshank Redemption, the characters must find hope, friendship, and resilience in the face of despair. In the other description, the criminal must fake insanity to escape prison and then battle the oppressive head nurse in the mental institution.

Variant Prompt: Outline the shared characteristics of the following descriptions: '{description}' and '{descriptions of other important neighbors}'.

Response: The shared characteristics of the descriptions are: - A protagonist facing a difficult situation - A struggle to balance personal life and studies - Overcoming adversity - Hope, friendship, and resilience - Oppressive forces to be overcome.

Table 24: Example responses generated by GPT-3 for p^{eng} and its variant that shares the same meaning but differs in word choices.

Description: The Paper Chase follows a first-year Harvard Law student as he struggles to balance his studies and personal life while facing the demanding expectations of his professor.

Descriptions of other important neighbors: 'The Paper Chase follows a first-year Harvard Law student as he struggles to balance his studies and personal life while facing the demanding expectations of his professor.'; 'The Shawshank Redemption is a story of hope, friendship, and resilience in the face of despair.'; 'A criminal fakes insanity to escape prison, only to find himself in a mental institution where he must battle the oppressive head nurse.'

$p^{rec+eng}$: The description of an item is as follows: '{description}'. What else should I say if I want to recommend it to others? This content is considered to hold some similar attractive characteristics as the following descriptions: '{descriptions of other important neighbors}'.

Response: If you're looking for a movie that will keep you on the edge of your seat, The Paper Chase is a great choice. It follows a first-year Harvard Law student as he navigates the pressures of his studies and personal life while facing the expectations of his professor. It's a story of determination and resilience in the face of adversity, and it will leave you feeling inspired and motivated.

Variant Prompt: If you wish to recommend '{description}' to others, what other details would you provide? This content possesses certain akin attractive attributes as the following descriptions: '{descriptions of other important neighbors}'.

Response: The Paper Chase is a classic drama that follows the struggles of a first-year Harvard Law student as he navigates the demanding expectations of his professor while trying to balance his studies and personal life. It is a story of perseverance, determination, and resilience in the face of adversity. The film features an all-star cast, including John Houseman, Timothy Bottoms, and Lindsay Wagner, and is sure to captivate viewers with its gripping plot and powerful performances.

Table 25: Example responses generated by GPT-3 for $p^{rec+eng}$ and its variant that shares the same meaning but differs in word choices.

	Movielens-1M			Recipe		
	Precision@10	Recall@10	NDCG@10	Precision@10	Recall@10	NDCG@10
Original Item Description	0.2922 ±0.0019	0.2455 ±0.0031	0.3640 ±0.0039	0.0325 ±0.0021	0.0684 ±0.0066	0.0580 ±0.0054
Duplicating Original Description Embeddings	0.2858 ±0.0020	0.2417 ±0.0032	0.3567 ±0.0043	0.0327 ±0.0030	0.0694 ±0.0082	0.0590 ±0.0068
Text Concatenation	0.3075 ±0.0015	0.2636 ±0.0035	0.3853 ±0.0027	0.0332 ±0.0026	0.0714 ±0.0083	0.0591 ±0.0065
`Concat-All` (ours)	**0.3150** ±0.0023	**0.2766** ±0.0030	**0.3951** ±0.0035	**0.0394** ±0.0033	**0.0842** ±0.0098	**0.0706** ±0.0084

Table 26: Concatenating embeddings of the responses augmented by LLM-REC outperforms concatenating duplicate embeddings of original description. It also achieves a superior performance than concatenating the raw text.

Description: This traditional meatloaf makes a fantastic meal, whether served hot or cold. It's perfect for slicing and enjoying cold at a picnic or in sandwiches. For a different twist, consider brushing your favorite salsa on top instead of the typical topping. A medium or hot chipotle salsa adds an especially tasty kick.
KAR: This easy and tasty meatloaf is a great dish to serve either hot or cold. It is perfect for a picnic or sandwiches, and can be enjoyed by everyone. Ingredients: - 2 lbs ground beef - 1/2 cup bread crumbs - 1/4 cup onion, finely chopped - 1/4 cup green pepper, finely chopped - 1 egg - 1/4 cup ketchup - 1 teaspoon Worcestershire sauce - 1 teaspoon garlic powder - 1 teaspoon salt - 1/4 teaspoon black pepper - Optional: your favorite salsa Cuisine type: American Cooking time: 1 hour Nutritional value: High in protein, low in fat Allergies: None Taste preferences: Mildly seasoned with a hint of sweetness from the ketchup Skill level: Easy Occasion: Perfect for a picnic, lunch, dinner, or snack Cost: Inexpensive Health and wellness goals: Low in fat, high in protein Food allure: Comfort food that is easy to make Reviews and ratings: Highly rated Cooking equipment: Baking sheet Personal experience: Delicious and easy to make Season and weather: Perfect for any season Cultural or ethical considerations: None Creativity and variety: Can be topped with your favorite salsa Simplicity: Easy to make Popularity and trends: A classic dish that is always popular Instructions: 1. Preheat oven to 350F. 2. In a large bowl, combine ground beef, bread crumbs, onion, green pepper, egg, ketchup, Worcestershire sauce, garlic powder, salt, and pepper. Mix until all ingredients are combined. 3. Place mixture into a greased 9x5 inch loaf pan. 4. Bake for 1 hour or until internal temperature reaches 160F. 5. Optional: Brush your favorite salsa on top of the meatloaf before serving. 6. Enjoy!

Table 27: An example response generated via knowledge augmentation prompts (Xi et al., 2023). The additional information is highlighted in red.

	Movielens-1M			Recipe		
	Precision@10	Recall@10	NDCG@10	Precision@10	Recall@10	NDCG@10
Item Description	0.2922 ±0.0019	0.2455 ±0.0031	0.3640 ±0.0039	0.0325 ±0.0021	0.0684 ±0.0066	0.0580 ±0.0054
Basic	0.3001 ±0.0027	0.2569 ±0.0028	0.3747 ±0.0042	0.0356 ±0.0024	0.0754 ±0.0089	0.0644 ±0.0068
Recommendation-driven	0.3025 ±0.0023	0.2577 ±0.0053	0.3786 ±0.0041	**0.0361** ±0.0031	**0.0771** ±0.0086	**0.0649** ±0.0069
Engagement-guided	0.3036 ±0.0020	**0.2608** ±0.0030	0.3801 ±0.0032	0.0348 ±0.0031	0.0732 ±0.0088	0.0628 ±0.0077
Recommendation+Engagement	**0.3038** ±0.0020	0.2603 ±0.0042	**0.3802** ±0.0037	0.0349 ±0.0024	0.0732 ±0.0066	0.0625 ±0.0060

Table 28: Average recommendation performance among different prompting strategies across five different splits. The best performance among the three Basic Prompting and three Recommendation-driven Prompting strategies is reported. The overall best results are highlighted in **bold**.

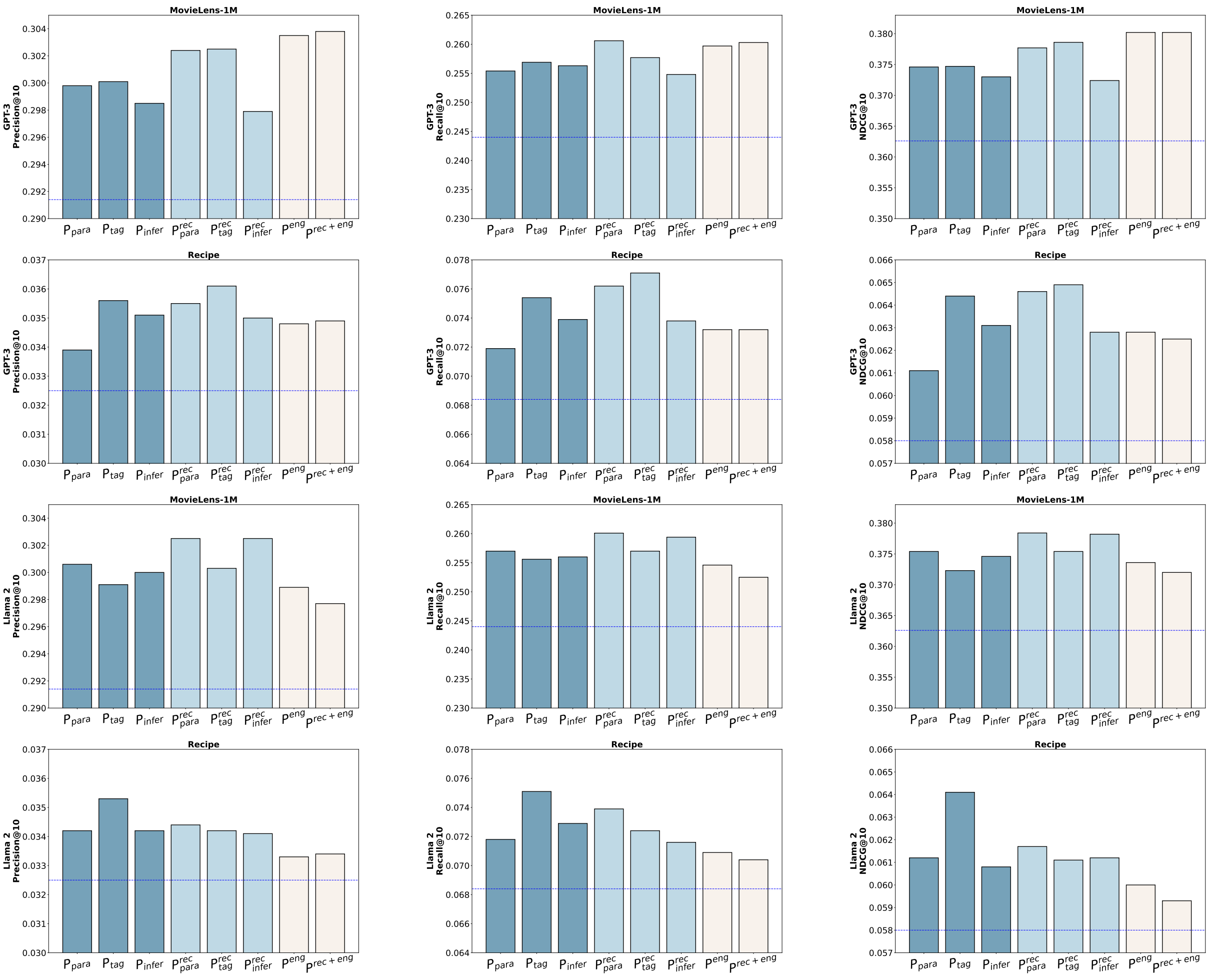

Figure 7: **The ablation study conducted on different prompting strategies shows that augmenting the input text with responses generated by large language models using our proposed prompting strategies enhances recommendation performance. However, the extent of this improvement may vary depending on the characteristics of the datasets used.** The *basic prompting* strategy includes three variants: p_{para} (paraphrase), p_{tag} (tag), and p_{infer} (infer). The *recommendation-driven* versions of these three variants are denoted by p^{rec}_{para}, p^{rec}_{tag}, and p^{rec}_{infer}. In addition, p^{eng} represents the *engagement-guided* prompts, and $p^{rec+eng}$ stands for the *recommendation + engagement* prompts. The blue line in each figure indicates the performance achieved by using only the original description embeddings. Note that in this ablation study, all the baselines use a concatenation of the the original description embeddings and prompt response embeddings as their model input.

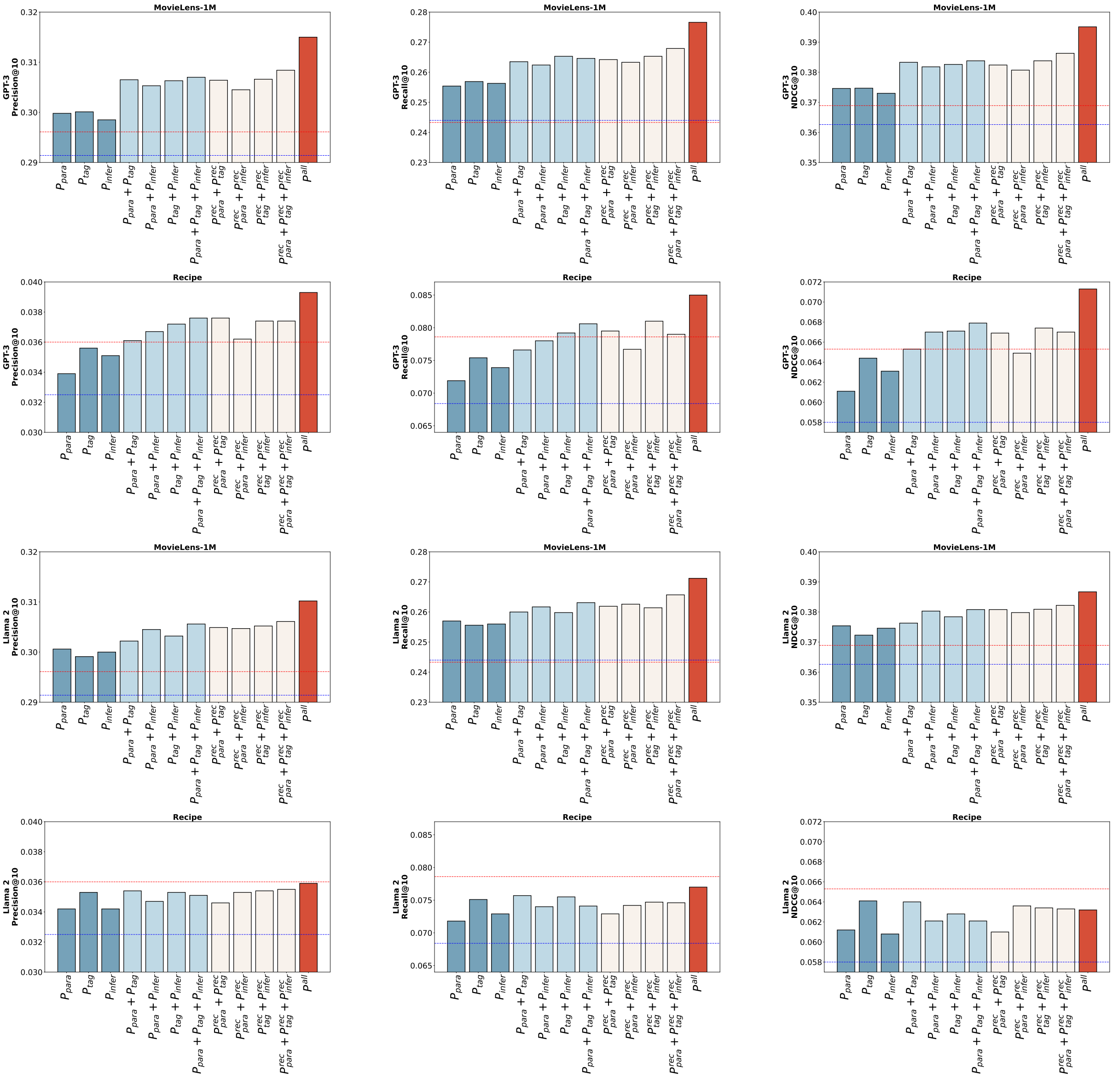

Figure 8: **The ablation study shows that overall, recommendation benefits from concatenating the embeddings of the input text augmented by large language models.** The blue line in each figure indicates the performance achieved by using only the original description embeddings, while the red line represents the performance achieved by DCN-V2 (Wang et al., 2021).

Description: A criminal fakes insanity to escape prison, only to find himself in a mental institution where he must battle the oppressive head nurse.

${p_{para}}^{\text{GPT-3}}$: A criminal attempts to avoid punishment by pretending to be mentally ill, only to end up in a psychiatric hospital where he must confront the tyrannical head nurse.
${p_{para}}^{\text{LLAMA-2}}$: A mentally ill criminal fakes insanity to escape prison and ends up in a mental institution where he must confront the abusive head nurse. Please provide one example of a possible paraphrase of the description of an item.
${p_{tag}}^{\text{GPT-3}}$: #Criminal #Insanity #Escape #MentalInstitution #OppressiveNurse
${p_{tag}}^{\text{LLAMA-2}}$: 1. Drama 2. Thriller 3. Horror 4. Crime 5. Mystery
${p_{infer}}^{\text{GPT-3}}$: This description can evoke a range of emotions, including fear, suspense, anxiety, dread, and uncertainty. It can also evoke feelings of hope and determination as the protagonist battles against the oppressive head nurse.
${p_{infer}}^{\text{LLAMA-2}}$: Fear Explanation
${p_{para}^{rec}}^{\text{GPT-3}}$: This thrilling story follows a criminal who must use his wits to survive in a mental institution. With a gripping plot and intense characters, this story will keep you on the edge of your seat as you watch the protagonist battle the oppressive head nurse. It's a must-read for anyone looking for an exciting and suspenseful read.
${p_{para}^{rec}}^{\text{LLAMA-2}}$: In this scenario, I would recommend the movie to anyone who enjoys dark humor and psychological thrillers. It has an excellent cast, including Jim Carrey in one of his most memorable roles, and the tension and suspense are top-notch. Additionally, the movie tackles some heavy themes in a thought-provoking way, making it a great choice for those interested in exploring the complexities of the human psyche. Overall, it's a must-see for fans of the genre.
${p_{tag}^{rec}}^{\text{GPT-3}}$: Tags: Crime, Mental Health, Prison, Insanity, Oppression, Battle
${p_{tag}^{rec}}^{\text{LLAMA-2}}$: If you want to recommend the movie "Cuckoo's Nest" to others, you could use tags such as "psychological thriller", "prison escape", "mental illness", "oppressive nurse", and "1970s films".
${p_{infer}^{rec}}^{\text{GPT-3}}$: This thrilling story will take you on an emotional rollercoaster as you follow a criminal's journey from prison to a mental institution. You'll feel his desperation as he attempts to fake insanity, and his fear as he faces the oppressive head nurse. It's a gripping tale of courage and resilience that will leave you on the edge of your seat.
${p_{infer}^{rec}}^{\text{LLAMA-2}}$: The movie "Shutter Island" is a psychological thriller that explores the themes of paranoia, suspense, and the blurring of reality. The story follows a U.S. Marshal who is sent to investigate the disappearance of a patient from a remote psychiatric hospital, only to find himself plunged into a web of intrigue and danger. The movie's use of atmosphere, sound, and visuals creates a sense of unease and uncertainty, keeping the viewer on the edge of their seat. The tense score and the eerie setting of the hospital add to the overall atmosphere of paranoia and unease, making it a gripping and suspenseful movie that will keep you on the edge of your seat. | Example Answer | Description | | The movie "Shutter Island" is a psychological thriller that explores the themes of paranoia, suspense, and the blurring of reality. | The movie's use of atmosphere, sound, and visuals creates a sense of unease and uncertainty, keeping the viewer on the edge of their seat. | Please provide answers for the next 3 questions.
${p^{eng}}^{\text{GPT-3}}$: The commonalities among the descriptions are that they all involve a criminal, redemption, and friendship. They also all involve a battle against an oppressive force.
${p^{eng}}^{\text{LLAMA-2}}$: The commonalities among the descriptions are
${p^{rec+eng}}^{\text{GPT-3}}$: This item is a thrilling story of redemption and resilience in the face of adversity. It follows a criminal who fakes insanity to escape prison, only to find himself in a mental institution where he must battle the oppressive head nurse. Along the way, he discovers hope, friendship, and love through a unique relationship with a gifted therapist. This story is sure to captivate readers with its suspenseful plot and inspiring message.
${p^{rec+eng}}^{\text{LLAMA-2}}$: Example Answer

Table 29: Example of responses by GPT-3 and the 7B LLAMA-2-CHAT on Movielens-1M (Harper and Konstan, 2015).

Description: Oh, so refreshing! Ideal for a sweltering summer day and it couldn't be any easier. Just mix together four ingredients, pour into the pie shell, and freeze. You're going to adore this! (Prep time does not include freezing.) Keep a couple of these pies in the freezer for when unexpected guests drop by.

${p_{para}}^{\text{GPT-3}}$: This delicious and easy-to-make pie is the perfect treat for a hot summer day. Simply mix together four ingredients and pour into a pie shell, then freeze. Keep a few of these pies in the freezer for when you have unexpected guests. Prep time does not include freezing.
${p_{para}}^{\text{LLAMA-2}}$: Sure! Here's a paraphrased version of the description
${p_{tag}}^{\text{GPT-3}}$: #refreshing #summer #simple #4ingredients #pie #freezer #unexpectedguests
${p_{tag}}^{\text{LLAMA-2}}$: summer, hot, refreshing, simple, icecream, pie, freezable, surprise guests, unexpected visitors
${p_{infer}}^{\text{GPT-3}}$: This description can evoke feelings of excitement, anticipation, and satisfaction. It can also evoke feelings of relaxation, refreshment, and contentment.
${p_{infer}}^{\text{LLAMA-2}}$: Curiosity - the use of the word'refreshing' and 'perfect for a hot summer day' may pique the reader's interest and make them want to try the recipe.
${p^{rec}_{para}}^{\text{GPT-3}}$: I highly recommend this delicious and refreshing pie! It's so easy to make with only 4 ingredients, and it's perfect for a hot summer day. Plus, you can keep a couple of these pies in the freezer for when unexpected guests show up. Prep time is minimal, so you can enjoy this delicious treat in no time!
${p^{rec}_{para}}^{\text{LLAMA-2}}$: "This frozen lemonade pie is a game-changer! Not only is it incredibly refreshing and perfect for hot summer days, but it's also super easy to make. With only 4 ingredients and a quick stir-together method, you can have this pie ready in no time. Plus, the fact that it's frozen means you can keep it on hand for when unexpected guests arrive. I highly recommend giving it a try - your taste buds (and your guests) will thank you!"
${p^{rec}_{tag}}^{\text{GPT-3}}$: Tags: #summertreat #refreshing #easyrecipe #frozendessert #pie #4ingredients #quickprep #unexpectedguests
${p^{rec}_{tag}}^{\text{LLAMA-2}}$: - desserts - ice cream - pie - refreshing - summer - simple - frozen Answer
${p^{rec}_{infer}}^{\text{GPT-3}}$: This delicious and refreshing pie is the perfect treat for a hot summer day! With only four ingredients and minimal prep time, you can have a delicious dessert ready in no time. Keep a couple of these pies in the freezer for when unexpected guests show up and you'll be sure to make them feel welcome with this delightful treat. Enjoy the feeling of satisfaction that comes with making something so simple yet so delicious!
${p^{rec}_{infer}}^{\text{LLAMA-2}}$: Sure, here's an example answer based on the provided description
${p^{eng}}^{\text{GPT-3}}$: The commonalities among the descriptions are that they all involve recipes that require minimal ingredients, are easy to make, and can be stored in the freezer for unexpected guests. They also all have a positive tone, suggesting that the recipes will be delicious.
${p^{eng}}^{\text{LLAMA-2}}$: simple, refreshing, flavor, and freeze.
${p^{rec+eng}}^{\text{GPT-3}}$: You won't be disappointed with this delicious and easy-to-make pie! With only four ingredients, you can whip up a tasty treat in no time. Plus, it's perfect for those hot summer days when you need something cool and refreshing. Keep a few of these pies in the freezer for when unexpected guests show up - they'll love it! And with minimal prep time, you can enjoy this delicious dessert without any hassle.
${p^{rec+eng}}^{\text{LLAMA-2}}$:

Table 30: Example of responses by GPT-3 and the 7B LLAMA-2-CHAT on Recipe (Majumder et al., 2019).