Large Language Model Augmented Narrative Driven Recommendations

Shesera Mysore
smy sore@cs.umass.edu
University of Massachusetts Amherst
USA

Andrew McCallum
mccallum@cs.umass.edu
University of Massachusetts Amherst
USA

Hamed Zamani
hzamani@cs.umass.edu
University of Massachusetts Amherst
USA

ABSTRACT

Narrative-driven recommendation (NDR) presents an information access problem where users solicit recommendations with verbose descriptions of their preferences and context, for example, travelers soliciting recommendations for points of interest while describing their likes/dislikes and travel circumstances. These requests are increasingly important with the rise of natural language-based conversational interfaces for search and recommendation systems. However, NDR lacks abundant training data for models, and current platforms commonly do not support these requests. Fortunately, classical user-item interaction datasets contain rich textual data, e.g., reviews, which often describe user preferences and context—this may be used to bootstrap training for NDR models. In this work, we explore using large language models (LLMs) for data augmentation to train NDR models. We use LLMs for authoring synthetic narrative queries from user-item interactions with few-shot prompting and train retrieval models for NDR on synthetic queries and user-item interaction data. Our experiments demonstrate that this is an effective strategy for training small-parameter retrieval models that outperform other retrieval and LLM baselines for narrative-driven recommendation.

CCS CONCEPTS

• Information systems → Recommender systems; Users and interactive retrieval; • Computing methodologies → Natural language generation.

ACM Reference Format:

1 INTRODUCTION

Recommender systems personalized to users are an important component of several industry-scale platforms [16, 17, 46]. These systems function by inferring users’ interests from their prior interactions on the platform and making recommendations based on these inferred interests. While recommendations based on historical interactions are effective, users soliciting recommendations often start with a vague idea about their desired target items or may desire recommendations depending on the context of use, often missing in historical interaction data (Figure 1). In these scenarios, it is common for users to solicit recommendations through long-form narrative queries describing their broad interests and context. Information access tasks like these have been studied as narrative-driven recommendations (NDR) for items ranging from books [5] and movies [18], to points of interest [1]. Bogers and Koolen [5] note these narrative requests to be common on discussion forums and several subreddits1, but, there is a lack of support for these complex natural language queries in current recommenders.

However, with the emergence of conversational interfaces for information access tasks, support for complex NDR tasks is likely to become necessary. In this context, recent work has noted an increase in complex and subjective natural language requests compared to more conventional search interfaces [13, 34]. Furthermore, the emergence of large language models (LLM) with strong language understanding capabilities presents the potential for fulfilling such complex requests [9, 33]. This work explores the potential for re-purposing historical user-item recommendation datasets, traditionally used for training collaborative filtering recommenders, with LLMs to support NDR.

Specifically, given a user’s interactions, \(D_u\), with items and their accompanying text documents (e.g., reviews, descriptions) \(D_u = \{d_i\}_{i=1}^{N_u}\), selected from a user-item interaction dataset \(I_i\), we prompt InstructGPT\(^1\), a 175B parameter LLM, to author a synthetic narrative query \(q_u\) based on \(D_u\) (Figure 2). Since we expect the query \(q_u\) to be noisy and not fully representative of all the user reviews, \(D_u\) is filtered to retain only a fraction of the reviews based on a language-model assigned likelihood of \(q_u\) given a user document, \(d_i\). Then, a pre-trained LM based retrieval model (110M parameters) is fine-tuned for retrieval on the synthetic queries and filtered reviews.

Our approach, which we refer to as Mint\(^2\), follows from the observation that while narrative queries and suggestions are often made in online discussion forums, and could serve as training data, the number of these posts and the diversity of domains for which they are available is significantly smaller than the size and diversity of passively gathered user-item interaction datasets. E.g. while Bogers and Koolen [5] note nearly 25,000 narrative requests for books on the LibraryThing discussion forum, a publicly available user-item interaction dataset for Goodreads contains interactions with nearly 2.2M books by 460k users [43].

We empirically evaluate Mint in a publicly available test collection for point of interest recommendation: POINTREC [1]. To train

\[ r/ MovieSuggestions, r/ booksuggestions, r/ animeSuggestions \]

\(1\) Mint: Data augmentation with Interaction narratives

---

1 Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

2 Mint: Data augmentation with Interaction narratives.
our NDR models, we generate synthetic training data based on user-item interaction datasets from Yelp. Models (110M parameters) trained with Mint significantly outperform several baseline models and match the performance of significantly larger LLM baselines autoregressively generating recommendations. Code and synthetic datasets are available:

3

2 RELATED WORK

Data Augmentation for Information Access. A line of recent work has explored using language models to generate synthetic queries for data augmentation to train models for information retrieval tasks [7, 8, 15, 23, 31]. Here, given a document collection of interest, a pre-trained language model is used to create synthetic queries for the document collection. An optional filtering step excludes noisy queries, and finally, a bi-encoder or a cross-encoder is trained for the retrieval task. While earlier work of Ma et al. [31] train a custom query generation model on web-text datasets, more recent work has leveraged large language models for zero/few-shot question generation [7, 8, 15, 23]. In generating synthetic queries, this work indicates the effectiveness of smaller parameter LLMs (up to 6B parameters) for generating synthetic queries in simpler information-retrieval tasks [7, 8, 23], and finds larger models (100B parameters and above) to be necessary for harder tasks such as argument retrieval [15, 23]. In generating synthetic queries, this work indicates the effectiveness of smaller parameter LLMs (up to 6B parameters) for generating synthetic queries in simpler information-retrieval tasks [7, 8, 23], and finds larger models (100B parameters and above) to be necessary for harder tasks such as argument retrieval [15, 23]. Similar to this work, we explore the generation of synthetic queries with LLMs for a retrieval task. Unlike this work, we demonstrate a data augmentation method for creating effective training data from sets of user documents found in recommendation datasets rather than individual documents. Other work in this space has also explored training more efficient multi-vector models from synthetic queries instead of more expensive cross-encoder models [39] and generating queries with a diverse range of intents than the ones available in implicit feedback datasets to enhance item retrievability [35].

Figure 1: An example narrative query soliciting point of interest recommendations. The query describes the users preferences and the context of their request.

Figure 2: The format of the prompt used in Mint for generating synthetic narrative queries from user-item interaction with a large language model.

Besides creating queries for ad-hoc retrieval tasks, concurrent work of Leszczynski et al. [25] has also explored the creation of synthetic conversational search datasets from music recommendation datasets with LLMs. The synthetic queries and user documents are then used to train bi-encoder retrieval models for conversational search. Our work resembles this in creating synthetic queries from sets of user items found in recommendation interaction datasets. However, it differs in the task of focus, creating long-form narrative queries for NDR. Finally, our work also builds on the recent perspective of Radlinski et al. [36] who make a case for natural language user profiles driving recommenders – narrative requests tie closely to natural language user profiles. Our work presents a step toward these systems.

Finally, while our work explores data augmentation from user-item interactions for a retrieval-oriented NDR task, prior work has also explored data augmentation of the user-item graph for training collaborative filtering models. This work has often explored augmentation to improve recommendation performance for minority [12, 47] or cold-start users [11, 28, 45]. And has leveraged generative models [11, 45] and text similarity models [28] for augmenting the user-item graph.

Complex Queries in Information Access. With the advent of performant models for text understanding, focus on complex and interactive information access tasks has seen a resurgence [2, 29, 32, 48]. NDR presents an example of this – NDR was first formalized in Bogers and Koolen [5] for the case of book recommendation and subsequently studied in other domains [3, 4, 6]. Bogers and Koolen [5] systematically examined narrative requests posted by users on discussion forums. They defined NDR as a task requiring item recommendation based on a long-form narrative query and prior-user item interactions. While this formulation resembles personalized search [42] and query-driven recommendation [20], the length and complexity of requests differentiate these from NDR. Other work has also demonstrated the effectiveness of re-ranking initial recommendations from collaborative filtering approaches.
Then, retrieval models are trained on the synthetic dataset [1] formulate the NDR task without access to the prior interactions (Fig-ure 3). Training data for retrieval models by using LLMs as query gener-

Our proposed method, Mint, for NDR, re-purposes readily available user-item interaction datasets commonly used to train collaborative filtering models for narrative-driven recommendation. This is done by authoring narrative queries for sets of items liked by a user with a large language model. The data is filtered with a smaller language model and retrieval models are trained on the synthetic queries and user items.

Fig 3: Mint re-purposes readily available user-item interaction datasets commonly used to train collaborative filtering models for narrative-driven recommendation. This is done by authoring narrative queries for sets of items liked by a user with a large language model. The data is filtered with a smaller language model and retrieval models are trained on the synthetic queries and user items.

based on the narrative query [18]. More recent work of Afzali et al. [1] formulate the NDR task without access to the prior interactions of a user while also noting the value of contextual cues contained in the narrative request. In our work, we focus on this latter for-\mation of NDR, given the lack of focus on effectively using the rich narrative queries in most prior work. Further, we demonstrate the usefulness of data augmentation from LLMs and user-item interaction datasets lacking narrative queries.

Besides this, a range of work has explored more complex, long-
form, and interactive query formulations for information access; these resemble queries in NDR. Arguello et al. [2] define the tip of tongue retrieval task, a known-item search task where user queries describe the rich context of items while being unable to recall item metadata itself. Mysore et al. [32] formulate an aspect conditional query-by-example task where results must match specific aspects of a long natural language query. And finally, a vibrant body of work has explored conversational critiquing of recommenders where natural language feedback helps tune the recommendations received by users [30, 44, 49].

3 METHOD

3.1 Problem Setup

In our work, we define narrative-driven recommendation (NDR) to be a ranking task, where given a narrative query $q_u$ made by a user $u$, a ranking system $f$ must generate a ranking $R$ over a collection of items $C$. Further, we assume access to a user-item interaction dataset $I$ consisting of user interactions with items $(u, \{d_i\})_{i=1}^{N_u}$. We assume the items $d_i$ to be textual documents like reviews or item descriptions. While we don’t assume there to be any overlap in the users making narrative queries or the collection of items $C$ and the user-items interaction dataset $I$, we assume them to be from the same broad domain, e.g., books, movies, points-of-interest.

3.2 Proposed Method

Our proposed method, Mint, for NDR, re-purposes a dataset of abundantly available user-item interactions, $I = \{(u, \{d_i\})_{i=1}^{N_u}\}$ into training data for retrieval models by using LLMs as query generation models to author narrative queries $q_u$: $\mathcal{D} = \{(q_u, \{d_i\})_{i=1}^{N_u}\}$. Then, retrieval models are trained on the synthetic dataset $\mathcal{D}$ (Figure 3).

3.2.1 Narrative Queries from LLMs. To author a narrative query $q_u$ for a user in $I$, we make use of the 175B parameter InstructGPT4 model as our query generation model QGen. We include the text of interacted items $\{d_i\}_{i=1}^{N_u}$ in the prompt for QGen, and instruct it to author a narrative query (Figure 2). To improve the coherence of generated queries and obtain correctly formatted outputs, we manually author narrative queries for 3 topically diverse users based on their interacted items and include it in the prompt for QGen. The same three few shot examples are used for the whole dataset $I$, and the three users were chosen from $I$. Generating narrative queries based on user interactions may also be considered a form of multi-document summarization for generating a natural language user profile [36].

3.2.2 Filtering Items for Synthetic Queries. Since we expect user items to capture multiple aspects of their interests and generated queries to only capture a subset of these interests, we only retain some of the items present in $\{d_i\}_{i=1}^{N_u}$ before using it for training retrieval models. For this, we use a pre-trained language model to compute the likelihood of the query given each user item, $P_{LM}(q_u|d_i)$, and only retain the top $M$ highly scoring item for $q_u$, this results in $M$ training samples per user for our NDR retrieval models: $\{(q_u, d_i)_{i=1}^{M}\}$. In our experiments, we use FLAN-T5 with 3B parameters [14] for computing and follow Sachan et al. [40] for computing $P_{LM}(q_u|d_i)$. Note that our use of $P_{LM}(q_u|d_i)$ represents a query-likelihood model classically used for ad-hoc search and recently shown to be an effective unsupervised re-ranking method when used with large pre-trained language models [40].

3.2.3 Training Retrieval Models. We train bi-encoder and cross-encoder models for NDR on the generated synthetic dataset — commonly used models in search tasks. Bi-encoders are commonly used as scalable first-stage rankers from a large collection of items. On the other hand, cross-encoders allow a richer interaction between query and item and are used as second-stage re-ranking models. For both models, we use a pre-trained transformer language model architecture with 110M parameters, MPNet, a model similar to BERT [41]. Bi-encoder models embed the query and item independently into high dimensional vectors: $q_u = MPNet(q_u)$, $d_i = MPNet(d_i)$ and rank items for the user based on the minimum L2 distance between

\[\text{https://platform.openai.com/docs/models/gpt-3, text-davinci-003}\]
Embeddings are obtained by averaging token embeddings from the final layer of MPNet, and the same model is used for both queries and items. Cross-encoder models input both the query and item and output a score to be used for ranking \( s = f_{CB}(q_u, d_i) \), where \( f_{CB} \) is parameterized as \( w^T \text{MPNet}(\cdot) \). We train our bi-encoder model with a margin ranking loss: \( \mathcal{L}_{Bi} = \sum_u \sum_{i=1}^{M} \max(0, \mathcal{L}(q_u, d_i) - \mathcal{L}(q_u, d_j) + \delta, 0) \) with randomly sampled negatives \( d' \) and \( \delta = 1 \). Our cross-encoders are trained with a cross-entropy loss: \( \mathcal{L}_{Cr} = \sum_u \sum_{i=1}^{M} \log(\frac{e^{\text{MPNet}(q_u)} \cdot e^{\text{MPNet}(d_i)}}{\sum_{j=1}^{M} e^{\text{MPNet}(d_j) + \delta}}) \). For training, 4 negative example items \( d' \) are randomly sampled from ranks 100-300 from our trained bi-encoder. At test time, we retrieve the top 200 items with our trained bi-encoder and re-rank them with the cross-encoder - we evaluate both these components in experiments and refer to them as BiEnc-MINT and CrEnc-MINT.

## 4 EXPERIMENTS AND RESULTS

Next, we evaluate MINT on a publicly available test collection for NDR and present a series of ablations.

### 4.1 Experimental Setup

#### 4.1.1 Datasets

We perform evaluations on an NDR dataset for point-of-interest (POI) recommendation POINTREC [1]. POINTREC contains 112 realistic narrative queries (130 words long) obtained from discussion forums on Reddit and items pooled from baseline rankers. The items are annotated on a graded relevance scale by crowd-workers and/or discussion forum members and further validated by the dataset authors. The item collection \( C \) in POINTREC contains 700k POIs with metadata (category, city) and noisy text snippets describing the POI obtained from the Bing search engine. For test time ranking, we only rank the candidate items in the city and request category (e.g., “Restaurants”) of the query available in POINTREC - this follows prior practice to exclude clearly irrelevant items [1, 26]. We use user-item interaction datasets from Yelp to generate synthetic queries for training.\(^5\) Note also that we limit our evaluations to POINTREC since it presents the only publicly available, manually annotated, and candidate pooled test collection for NDR, to our knowledge. Other datasets for NDR use document collections that are no longer publicly accessible [24], contain sparse and noisy relevance judgments due to them being determined with automatic rules applied to discussion threads [18, 24], lack pooling to gather candidates for judging relevance [18, 24], or lack realistic narrative queries [21]. We leave the development of more robust test collections and evaluation methods for NDR to future work.

#### 4.1.2 Implementation Details

Next, we describe important details for MINT and leave finer details of the model and training to our code release. To sample user interactions for generating synthetic queries from the Yelp dataset, we exclude POIs and users with fewer than ten reviews to ensure that users were regular users of the site with well represented interests. This follows common prior practice in preparing user-item interaction datasets for use [27]. Then we retain users who deliver an average rating greater than 3/5 and with 10-30 above-average reviews. This desirably biases our data to users who commonly describe their likings (rather than dislikes). It also retains the users whose interests are summarizable by QGen. In the Yelp dataset, this results in 45,193 retained users. Now, 10,000 randomly selected users are chosen for generating synthetic narrative queries. For these users, a single randomly selected sentence from 10 of their reviews is included in the prompt (Figure 2) to QGen, i.e., \( N_u = 10 \). After generating synthetic queries, some items are filtered out (§3.2.2). Here, we exclude 40% of the items for a user. This results in about 60,000 training samples for training BiEnc-MINT and CrEnc-MINT. These decisions were made manually by examining the resulting datasets and the cost of authoring queries. The expense of generating \( q_u \) was about USD 230.

#### 4.1.3 Baselines

We compare BiEnc-MINT and CrEnc-MINT models against several standard and performant retrieval model baselines. These span zero-shot/unsupervised rankers, supervised bi-encoders, unsupervised cross-encoders, and LLM baselines. BM25: A standard unsupervised sparse retrieval baseline based on term overlap between query and document, with strong generalization performance across tasks and domains [38]. Contriver: A BERT-base bi-encoder model pre-trained for zero-shot retrieval with weakly supervised query-document pairs [22]. MPNet-1B: A strong Sentence-Bert bi-encoder model initialized with MPNet-base and trained on 1 billion supervised query-document pairs aggregated from numerous domains [37]. BERT-MSM: A BERT-base bi-encoder fine-tuned on supervised question-passage pairs from MSMARCO. UPR: A two-stage approach that retrieves items with a Contriver bi-encoder and re-ranks the top 200 items with a query-likelihood model using a FlanT5 model with 3B parameters [14, 40]. This may be seen as an unsupervised “cross-encoder” model. Grounded LLM: A recently proposed two-stage approach which autoregressively generates ten pseudo-relevant items using an LLM (175B INSTRUCTGPT) prompted with the narrative query and generates recommendations grounded in C by retrieving the nearest neighbors for each generated item using a bi-encoder [19]. We include one few-shot example of a narrative query and recommended items in the prompt to the LLM. We run this baseline three times and report average performance across runs. We report NDCG at 5 and 10, MAP, MRR, and Recall at 100 and 200. Finally, our reported results should be considered lower bounds on realistic performance due to the unjudged documents (about 70% at \( k = 10 \)) in our test collections [10].

### 4.2 Results

Table 1 presents the performance of the proposed method compared against baselines. Here, bold numbers indicate the best-performing model, and superscripts indicate statistical significance computed with two-sided t-tests at \( p < 0.05 \).

Here, we first note the performance of baseline approaches. We see BM25 outperformed by Contriver, a transformer bi-encoder model trained for zero-shot retrieval; this mirrors prior work [22]. Next, we see supervised bi-encoder models trained on similar passage (MPNet-1B) and question-answer (BERT-MSM) pairs outperform a weakly supervised model (Contriver) by smaller margins. Finally, the Grounded LLM outperforms all bi-encoder baselines, indicating strong few-shot generalization and mirroring prior results [19]. Examining the MINT models, we first note that the BiEnc-MINT sees statistically significant improvement compared to BM25...

\(^5\)https://www.yelp.com/dataset
and outperforms the best bi-encoder baselines by 11-13% on precision measures and 5-7% on recall measures. Specifically, we see a model trained for question-answering (BERT-MSM) underperform BiEnc-Mint, indicating the challenge of the NDR task. Further, BiEnc-Mint, trained on 5 orders of magnitude lesser data than MPNet-1B, sees improved performance – indicating the quality of data obtained from Mint. Furthermore, BiEnc-Mint also performs at par with a 175B LLM while offering the inference efficiency of a small-parameter bi-encoder. Next, we see CrEnc-Mint outperform the baseline bi-encoders, BiEnc-Mint, UPR, and Grounded LLM by 4-21% on precision measures and 7-13% on recall measures – demonstrating the value of Mint for training NDR models.

4.3 Ablations

In Table 2, we ablate various design choices in Mint. Different choices result in different training sets for the BiEnc and CrEnc models. Also, note that in reporting ablation performance for CrEnc, we still use the performant BiEnc-Mint model for obtaining negative examples for training and first-stage ranking. Without high-quality negative examples, we found CrEnc to result in much poorer performance.

No item filtering. Since synthetic queries are unlikely to represent all the items of a user, Mint excludes user items \{d_i\}_{i=1}^{N_u} which have a low likelihood of being generated from the document (§3.2.2). Without this step, we expect the training set for training retrieval models to be larger and noisier. In Table 2, we see that excluding this step leads to a lower performance for BiEnc and CrEnc, indicating that the quality of data obtained is important for performance.

6B LLM for QGen. Mint relies on using an expensive 175B parameter INSTRUCTGPT model for QGen. Here, we investigate the efficacy for generating \(q_u\) for \(\{d_i\}_{i=1}^{N_u}\) with a 6B parameter INSTRUCTGPT model (text-curie-001). We use an identical setup to the 175B LLM for this. In Table 2, we see that training on the synthetic narrative queries of the smaller LLM results in worse models – often underperforming the baselines in Table 1. This indicates the inability of a smaller model to generate complex narrative queries while conditioning on a set of user items. This necessity of a larger LLM for generating queries in complex retrieval tasks has been observed in prior work [15, 23].

6B LLM for Item Queries. We find a smaller 6B LLM to result in poor quality data when used to generate narrative queries conditioned on \(\{d_i\}_{i=1}^{N_u}\). Here we simplify the text generation task using a 6B LLM to generate queries for individual items \(d_i\). This experiment also mirrors the setup for generating synthetic queries for search tasks [7, 15]. Here, we use 3-few shot examples and sample one item per user for generating \(q_u\). Given the lower cost of using a smaller LLM, we use all 45,193 users in our Yelp dataset rather than a smaller random sample. From Table 2, we see that this results in higher quality queries than using smaller LLMs for generating narrative queries from \(\{d_i\}_{i=1}^{N_u}\). The resulting BiEnc model underperforms the BiEnc-Mint, indicating the value of generating complex queries conditioned on multiple items as in Mint for NDR. We see that CrEnc approaches the performance of CrEnc-Mint – note, however, that this approach uses the performant BiEnc-Mint for sampling negatives and first stage ranking. We leave further exploration of using small parameter LLMs for data augmentation for NDR models to future work.

5 CONCLUSIONS

In this paper, we present Mint, a data augmentation method for the narrative-driven recommendation (NDR) task. Mint re-purposes historical user-item interaction datasets for NDR by using a 175B parameter large language model to author long-form narrative queries while conditioning on the text of items liked by users. We evaluate bi-encoder and cross-encoder models trained on data from Mint on the publicly available Pointrec test collection for narrative-driven point of interest recommendation. We demonstrate that the resulting models outperform several strong baselines and ablated models and match or outperform a 175B LLM directly used for NDR in a 1-shot setup.

However, Mint also presents some limitations. Given our use of historical interaction datasets for generating synthetic training data and the prevalence of popular interests in these datasets longer, tailed interests are unlikely to be present in the generated synthetic datasets. In turn, causing retrieval models to likely see poorer performance on these requests. Our use of LLMs to generate synthetic queries also causes the queries to be repetitive in structure, likely causing novel longer-tail queries to be poorly served. These limitations may be addressed in future work.
Besides this, other avenues also present rich future work. While MINT leverages a 175B LLM for generating synthetic queries, smaller parameter LLMs may be explored for this purpose - perhaps by training dedicated QGen models. MINT may also be expanded to explore more active strategies for sampling items and users for whom narrative queries are authored - this may allow more efficient use of large parameter LLMs while ensuring higher quality training datasets. Next, the generation of synthetic queries from sets of documents may be explored for a broader range of retrieval tasks beyond NDR given its promise to generate larger training sets – a currently underexplored direction. Finally, given the lack of larger-scale test collections for NDR and the effectiveness of LLMs for authoring narrative queries from user-item interaction, fruitful future work may also explore the creation of larger-scale datasets in a mixed-initiative setup to robustly evaluate models for NDR.

ACKNOWLEDGMENTS

We thank anonymous reviewers for their invaluable feedback. This work was partly supported by the Center for Intelligent Knowledge Base Construction. Any opinions, findings and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect those of the sponsors.

REFERENCES

Table 2: MINT ablated for different design choices on POINTREC.

<table>
<thead>
<tr>
<th>Ablation</th>
<th>NDCG@5</th>
<th>NDCG@10</th>
<th>MAP</th>
<th>MRR</th>
<th>Recall@100</th>
<th>Recall@200</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiEnc-MINT</td>
<td>0.3489</td>
<td>0.3263</td>
<td>0.1890</td>
<td>0.3982</td>
<td>0.5263</td>
<td>0.6221</td>
</tr>
<tr>
<td>− No item filtering</td>
<td>0.2949</td>
<td>0.2766</td>
<td>0.1634</td>
<td>0.3505</td>
<td>0.4979</td>
<td>0.5951</td>
</tr>
<tr>
<td>− 6B LLM for QGen</td>
<td>0.2336</td>
<td>0.2293</td>
<td>0.1125</td>
<td>0.2287</td>
<td>0.426</td>
<td>0.5435</td>
</tr>
<tr>
<td>− 6B LLM for Item Queries</td>
<td>0.3012</td>
<td>0.2875</td>
<td>0.1721</td>
<td>0.3384</td>
<td>0.4800</td>
<td>0.5909</td>
</tr>
<tr>
<td>CrEnc-MINT</td>
<td>0.3725</td>
<td>0.3489</td>
<td>0.2192</td>
<td>0.4317</td>
<td>0.5448</td>
<td>0.6221</td>
</tr>
<tr>
<td>− No item filtering</td>
<td>0.3570</td>
<td>0.3379</td>
<td>0.2071</td>
<td>0.4063</td>
<td>0.5366</td>
<td>0.6221</td>
</tr>
<tr>
<td>− 6B LLM for QGen</td>
<td>0.2618</td>
<td>0.2421</td>
<td>0.1341</td>
<td>0.3118</td>
<td>0.4841</td>
<td>0.6221</td>
</tr>
<tr>
<td>− 6B LLM for Item Queries</td>
<td>0.3792</td>
<td>0.3451</td>
<td>0.2128</td>
<td>0.4098</td>
<td>0.5546</td>
<td>0.6221</td>
</tr>
</tbody>
</table>


Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. 2023. Promptagator: Few-shot...