Reinforcement Learning for Adversarial Query Generation to Enhance Relevance in Cold-Start Product Search

Akshay Jagatap
Amazon
ajjagata@amazon.com

Neeraj Anand
Amazon
neeranan@amazon.com

Sonali Singh
Amazon
ssonl@amazon.com

Prakash Mandayam Comar
Amazon
prakasc@amazon.com

Abstract

Accurate mapping of queries to product categories is crucial for efficient retrieval and ranking of relevant products in e-commerce search. Conventionally, such query classification models rely on supervised learning using historical user interactions, but their effectiveness diminishes in cold-start scenarios, where new categories or products lack sufficient training data. This results in poor query-to-category mappings, negatively affecting retrieval and ranking. Synthetic query generation has emerged as a promising solution by augmenting training data; however, existing methods do not incorporate feedback from the query relevance model, limiting their ability to generate queries that enhance product retrieval. To address this, we propose an adversarial reinforcement learning framework that optimizes an LLM-based generator to expose weaknesses in query classification models. The generator produces synthetic queries to augment the classifier's training set, ultimately improving its performance. Additionally, we introduce a structured reward signal to ensure stable training. Experiments on public datasets show an average PR-AUC improvement of +1.82% on benchmarks and +3.26% on a proprietary dataset, demonstrating the framework's effectiveness in enhancing query classification and mitigating cold-start challenges.

1 Introduction

The cold-start problem is a critical challenge in e-commerce, particularly for new products and emerging categories. This issue arises due to multiple factors: (a) Bias in ranking models—ranking algorithms often prioritize established products and categories with a high volume of historical interactions, leading to skewed relevance estimation (Lesota et al., 2021; Ning et al., 2024); (b) Category-specific relevance—the definition of relevance varies across product categories. For instance, in electronics, attributes such as brand and RAM specifications are crucial, whereas in pharmacy, active ingredient composition and dosage strength play a more significant role. These factors make it difficult to effectively rank and surface relevant products for queries related to new or underrepresented categories (Jansen and Booth, 2010; Mateos and Bellogín, 2024). Hence, an essential step in product recommendations is determining the category of a given product, which allows for the up-ranking or down-ranking of products within a specific category. This classification is typically performed in the first-stage ranker, as recommendation systems often employ a two-stage ranking process to refine product relevance and improve retrieval effectiveness (Covington et al., 2016).

Typically, query classification models are trained in a supervised manner, leveraging labeled data derived from customer interactions such as clicks, cart additions, and purchases (Jagatap et al., 2023). However, in new or low-interaction categories, reliance on historical data exacerbates the cold-start problem, as limited user engagement leads to poor classification performance and sub-optimal ranking of products. Conventionally, this issue is addressed by allowing time for new products to accumulate interactions or by inferring relevance through correlations with existing products (Guan et al., 2024). With recent advancements in generative models, synthetic query generation has gained prominence as a viable approach to simulating queries for new products and categories (Chaudhary et al., 2024; Jagatap et al., 2024). This technique provides essential training signals to downstream models, helping to address the cold-start challenge more effectively. While these approaches use generative models to produce synthetic queries for improving downstream classification performance, they do not leverage feedback from the classifier to guide query generation. Specifically, they do not account for whether the generated queries induce high model uncertainty or leads to frequent misclassification

errors. We attempt to address these challenges in our work. The key contributions of our paper are as follows:

1. Adversarial RL-Based Query Generation Framework. We introduce a reinforcement learning framework that establishes a feedback loop between the LLM generator and the classifier, akin to a generative adversarial networks (GAN). The generator is trained to generate synthetic queries that are particularly challenging for the classifier, helping it learn to distinguish difficult edge cases where classification is uncertain. As the generator improves, it produces more effective adversarial queries, which are then used to augment the classification model's training data, leading to a more robust model that mitigates cold-start issues in product search.

2. Reward-Based Guardrails. Such generative adversarial frameworks are often unstable, making training challenging. To address this, we design the reward function to induce stability in the generator while also guiding it toward producing queries that are both challenging for the classifier and meaningful for training. This ensures that the generator does not collapse to producing irrelevant or nonsensical queries, maintaining effectiveness of the adversarial training process.

3. Empirical Validation. We demonstrate performance improvements over three public relevance datasets and one industry dataset, showcasing the effectiveness of our approach in enhancing query relevance models. Our adversarial RL-based framework achieves a +1.82% average improvement in PR-AUC across the three public datasets and a +3.26% PR-AUC improvement on a proprietary e-commerce dataset. The deployed model led to a +3.8% increase in purchases within a cold-start category, as validated through A/B testing.

2 Query-Product Relevance Problem

Let $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_0$ represent the product catalog, where $\mathcal{A}_1$ and $\mathcal{A}_0$ correspond to in-category and out-of-category products, respectively. Similarly, let $\mathcal{Q}$ denote the space of all customer text queries. The relevance of a product $a \in \mathcal{A}$ for a query q is denoted by $p^{rel}(a|q)$, allowing us to define a soft classification function for query category membership:

$$p^{true}(y = 1|q) = \frac{\sum_{a \in \mathcal{A}_1} p^{rel}(a|q)}{\sum_{a \in \mathcal{A}} p^{rel}(a|q)}$$

In practice, the true relevance $p^{rel}(a|q)$ is unknown. Instead, we observe interactions shaped by the existing ranking system. Let $p^{seen}(a|q)$ represent the probability of a product being displayed to a customer, factoring in positional biases. Further, the interaction volume $v(a, q)$, capturing customer engagement (e.g., clicks, cart-adds, purchases), follows the relationship: $v(a, q) \propto p^{seen}(a|q)p^{rel}(a|q)$.

Given observed query-product interactions $v_{train}(a, q)$, the existing ranking system $p^{seen}(a|q)$, and product catalog features, our goal is to learn a classification model that predicts query category membership $\hat{p}(y|q)$ to approximate the true probability $p^{true}(y|q)$.

Since true relevance is unknown, we evaluate our model on a test set using an estimated probability $p^{test}(y|q)$, where product relevance is inferred from: $p^{test}(y|q) \propto v^{test}(a, q)/p^{seen}(a|q)$.

While training and test distributions may be similar, learning an accurate query classifier is challenging because training interactions are biased by the ranking system and may not include new products or queries. Offline evaluation on unseen test data provides directional insight, but the true impact of improved classification is best measured through increased customer interactions in an online experiment.

3 Related Works

With the rise of generative LLMs (Naveed et al., 2023) that encode substantial world knowledge, there has been growing interest in utilizing LLMs for synthetic query generation (Chaudhary et al., 2024; Sannigrahi et al., 2024). While most research addresses question-answering and binary relevance, recent work explores query generation for e-commerce products with multi-level relevance, either by fine-tuning LLMs on historical product-query data to generate customer-like queries, which are then used to augment and improve the downstream relevance model (Chaudhary et al., 2023) or have prompted LLMs for query generation implementing feedback loops through Bayesian optimization to refine prompts (Jagatap et al., 2024).

In contrast to these existing methodologies, we propose a reinforcement learning framework that directly incorporates relevance model feedback into the query generation process. The closed-loop system we developed resembles a Generative Adversarial Network (GAN) (De Rosa and Papa, 2021)

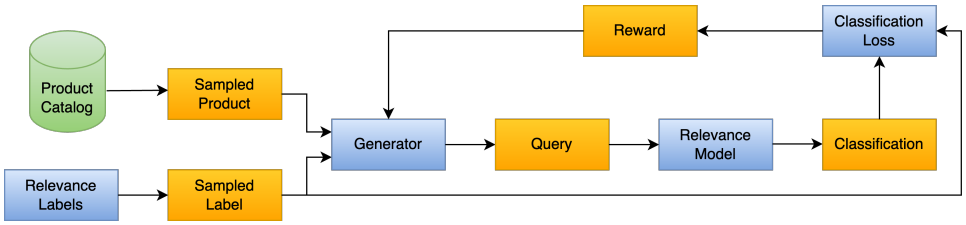

Figure 1: Overview of our reinforcement learning framework for query generation. The generator produces queries conditioned on a sampled product and relevance label. The relevance model evaluates the generated query, providing feedback that is used to compute a reward, which updates the generator through classification loss.

where the relevance model acts as a discriminator providing adversarial rewards, while the generator creates increasingly difficult samples to challenge the discriminator. However, rather than using traditional GANs, we employ reinforcement learning for text generation, building on work by (Yu et al., 2017), who proposed SeqGAN. This approach bridges RL and GANs by treating the generator as an RL agent and using the discriminator to provide rewards.

Our improved generator produces diverse synthetic queries that are systematically incorporated into the relevance model's training corpus. The resulting enhancement in relevance model robustness is particularly significant for mitigating cold-start issues (Han et al., 2022) common in product search systems. This methodology also resembles self-training semi-supervised learning paradigms, where an established teacher model trained on extensive datasets generates synthetic labels to enhance a student model's performance and broaden its input distribution coverage (Pace et al., 2024; Shen et al., 2024).

4 Proposed Approach

A standard approach for synthetic data augmentation in query classifiers is fine-tuning a LLM on historical search logs (Jagatap et al., 2024). In this method, the model is trained on a dataset of `(product, query, relevance)` tuples to generate queries conditioned on both product attributes and relevance labels (e.g., `Exact`, `Irrelevant`). This ensures that the generated queries align with specific relevance categories, enhancing their effectiveness for downstream classification tasks. For unseen or sparsely populated product categories, the fine-tuned generator produces synthetic queries to augment the classifier's training set, thereby improving generalization in low-data settings. Despite its effectiveness, `Fine-Tuned` approach presents several limitations. The generator is heavily conditioned on product metadata, resulting in queries that often closely resemble product descriptions rather than capturing the diversity of real-world search behavior (Jagatap et al., 2024).

4.1 Adversarial RL-Based Query Generation

The proposed `Adversarial-RL` framework incorporates reinforcement learning (RL) to address these limitations. The initial steps remain the same as in `Fine-Tuned` approach: the generator is trained to generate queries conditioned on the product and relevance label, and the generated queries augment the classifier's training data. In `Adversarial-RL`, within the RL framework, the generator produces a synthetic query conditioned on a given product and relevance label, which is then evaluated by the relevance model.The classifier's predicted relevance is evaluated against the ground-truth label assigned during generation. A high classification loss indicates a challenging query that effectively probes the classifier's decision boundaries, revealing areas of uncertainty or misclassification. The generator is rewarded for producing challenging queries, encouraging the generation of diverse queries that enhance classifier robustness. This reinforcement mechanism drives the generator to create queries that deviate from product metadata while preserving semantic relevance (see Figure 1). This results in a generator that more effectively augments the downstream classifier, particularly in cold-start scenarios where limited historical data is available for training.

We formulate the training of the LLM generator as a Proximal Policy Optimization (PPO) problem (Stiennon et al., 2022), where the classifier acts as the reward model. The PPO algorithm updates the generator's policy parameters θ by maximizing the following objective function:

$$\mathbb{L}(\theta) = \mathbb{E}[\min\left(r_t(\theta)A_t, \texttt{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)]$$

Here, $\mathbb{E}$ denotes the empirical expectation and

$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio between the new policy π_θ and old policy $\pi_{\theta_{\text{old}}}$. a_t represents the token chosen at position t in the sequence. s_t is the context (all previous tokens) at position t. A_t is the advantage estimate, which in our case is derived from the reward function. ϵ is a hyperparameter that constrains policy updates. The advantage function A_t is calculated using the reward signal from the classifier at the end of the sequence. The clipping operation, controlled by the hyperparameter ϵ, prevents excessive policy updates that could destabilize training.

Parameterized Reward Function: Since the generator is trained to produce queries in an adversarial manner and is explicitly rewarded for generating challenging samples, it may unintentionally be guided to generate semantically incorrect queries. For example, when prompted to generate a `relevant` query for a pharmacy product, the LLM might incorrectly generate the query *"washing machine"*. While the classifier correctly predicts it as *irrelevant*, the generator, rewarded for confusing the classifier, would receive a high reward despite the query being incorrect. To mitigate this issue, we initialize the generator from a fine-tuned model and impose a KL divergence penalty to restrict deviations from its learned distribution. Our structured reward function is defined at each token position as the generator sequentially generates text: For each token position $t < T$ (before the end-of-sequence (`EOS`) token): $R(t) = -\beta \cdot D_{\text{KL}}(\pi_\theta||\pi_{\text{FT}})$. For the final token position $t = T$ (at `EOS`):

$$R(T) = \alpha \cdot L_{\text{cls}} - (1-\alpha) \cdot log P_{\text{gen}} - \beta \cdot D_{\text{KL}}(\pi_\theta||\pi_{\text{FT}})$$

The term D_{KL} represents the KL divergence (Kullback and Leibler, 1951) between the current and fine-tuned policies at each token position, ensuring that the generator does not deviate excessively from the pre-trained distribution. The classifier's cross entropy loss over the complete sequence is denoted by L_{cls}, guiding the generator to produce queries that effectively challenge the classifier. The term P_{gen} captures the generation probability, which is incorporated into the reward to stabilize learning. If the generator confidently produces a challenging query, it receives a reward proportional to P_{gen}, encouraging the exploration of difficult yet meaningful queries rather than generating random noise. The hyperparameters α and β control the balance between these reward components, ensuring that the generator optimizes for both adversarial and semantically valid query generation.

4.1.1 Training Schedule

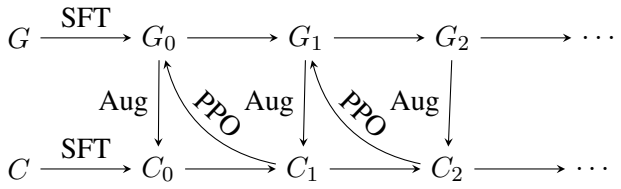

Figure 2: Illustration of the iterative reinforcement learning framework for improving the generator G through PPO feedback and enhancing the classifier C via synthetic data augmentation.

As shown in Figure 2, our training process begins with both the generator and classifier undergoing Supervised Fine-Tuning (SFT) on customer data, yielding G_0 and C_0. The training then follows an automated reinforcement learning cycle consisting of two steps. In the first step, **Data Augmentation**, the generator G_N generates synthetic queries using metadata and relevance labels as input of new or unseen products. This newly generated synthetic data, denoted as D_N^{syn}, is combined with the original classifier training dataset D_0 to create an augmented dataset: $D_N = D_N^{\text{syn}} \cup D_0$. The classifier C_N is then trained on D_{N-1}, meaning C_N represents the classifier trained with the augmented dataset D_{N-1}, which was generated using the generator G_{N-1} from the previous cycle.

Next, in **PPO Training**, the updated classifier C_{N+1} provides PPO rewards to the generator G_N. Using these rewards, the generator is fine-tuned for 2 epochs, resulting in an improved generator G_{N+1}. This iterative process of data augmentation followed by PPO-based optimization constitutes a single training cycle. The training is repeated for a total of 4 cycles, progressively refining both models. These hyperparameters: PPO training epochs (2), classifier training epochs (5), and the number of training cycles (4) are fixed and can be adjusted based on validation performance.

5 Experiments

5.1 Datasets

Ecom-Pharma is an internal dataset sampled from real customer interactions on an e-commerce pharmacy platform. The dataset is partitioned into temporally disjoint sets: train (`Sep 2024–Nov 2024`) and test (`Dec 2024`). To construct the dataset, we

start with the pharmacy catalog (ground truth list of products) and identify "weak pharmacy intent queries" that have led to at least 5% clicks on pharmacy products. For each query, we retrieve all clicked products (set A) and classify them as pharmacy or non-pharmacy. We then expand the query set by retrieving all queries associated with products in set A. Each query is mapped to a binary label (pharmacy/non-pharmacy) based on interaction volume and used to train the query classifier. For generator fine-tuning, we use product-query pairs from set A, weighted by interaction volume.

Our experiments also utilize three public datasets: **WANDS** (Chen et al., 2022), **Home Depot** (Home Depot, 2016), and **Amazon ESCI** (Reddy et al., 2022), all of which consist of product-query pairs annotated with relevance labels. The **WANDS** dataset focuses on product search relevance in the home improvement domain, categorizing relevance into ExactMatch, PartialMatch, and Irrelevant. The **Home Depot** dataset also provides product-query relevance annotations but assigns real-valued relevance scores, which we discretize into three categorical levels—Irrelevant, PartialMatch, and ExactMatch—using the 33rd and 66th percentile thresholds. Lastly, the **Amazon ESCI** dataset is a large-scale collection of product search queries with four relevance levels: Exact, Substitute, Complement, and Irrelevant.

5.2 Algorithms & Metrics

Since the generator is used only during training, its size does not impact inference latency. At inference, we prioritize efficiency, opting for a smaller relevance model. As the generator operates in an offline setup, we prioritize generation quality over latency, leveraging `FLAN-T5-XL` for the `Ecom` dataset and `FLAN-T5-Large` for public datasets. For all datasets, the classifier is built on the `FLAN-Small` encoder with a classification head.

Classification Metrics: To assess the performance of our classifier model, we measure PR-AUC (Davis and Goadrich, 2006) for the entire test set.

Generation Metrics: We compute BERTScore (Zhang et al., 2020), which measures the semantic similarity between the generated queries and the target queries.

Ranking Metrics: On external datasets where class labels are ordered, we evaluate ranking performance using the approach in prior work. For each query-product pair, we compute the score: $E_i = \sum_{j \in \{E,P,I\}} p(y_j|x_i) \cdot w_j$ where E, P, and I denote `ExactMatch`, `PartialMatch`, and `Irrelevant`, respectively. The weight values are set as: $w_j = \{E = 2.0, P = 1.0, I = 0.0\}$. We then compute NDCG@10 by ranking products based on E_i.

5.3 Results & Discussion

In this section, we analyze the impact of different training strategies on downstream relevance model performance across multiple datasets. We further investigate the impact of generator size on downstream model performance. Additionally, we explore how parameterization choices and reward design influence RL training stability and downstream performance.

Strategy	PR-AUC	BERT-score
`Prompted`	+0.30%	83.58%
`Fine-tuned`	+2.38%	92.51%
`Adversarial RL`	**+3.26%**	91.94%

Table 1: Improvement in performance using different strategies on the `Ecom-Pharma` dataset. We show the relative improvement in performance over the base classifier.

RQ1. Does RL improve downstream relevance model performance?

Table 1 presents the relative improvements in PR-AUC and BERT-score across different training strategies on the `Ecom-Pharma` dataset. While `Fine-Tuning` based augmentation significantly enhances classification performance over the base model, `Adversarial-RL` based augmentation achieves the highest PR-AUC gain of +3.26%, demonstrating its effectiveness in refining query generation to improve retrieval performance. However, the slight drop in BERT-score compared to `Fine-Tuning` suggests that adversarial training may prioritize generating diverse queries that deviate from observed data.

Further, we evaluated our approach across three public e-commerce benchmarks: `WANDS`, `Home Depot`, and `Amazon ESCI`. Table 2 demonstrates that our `Adversarial-RL` approach consistently outperforms `Fine-Tuning` in PR-AUC, micro-averaged across the multiple relevance labels. We also observe an improvement in ranking effectiveness (NDCG@10). Notably, the `Amazon ESCI` dataset shows the highest gain in PR-AUC (+3.42%) and NDCG@10 (+2.22%) when using adversarial RL. The BERT-Score metric indicates that `Fine-Tuning` generates queries which are sim-

ilar to the ones we observe in the test data, while adversarial RL introduces slight variations due to reinforcement learning optimizing for diversity.

Strategy	PR-AUC (micro)	NDCG@10	BERT-score
WANDS			
None	85.69%	96.42%	-
Fine-Tuning	86.21%	96.88%	96.52%
Adv. RL	**86.63%**	**97.40%**	96.35%
Home Depot			
None	48.38%	93.32%	-
Fine-Tuning	48.46%	93.45%	91.46%
Adv. RL	**49.49%**	**94.69%**	91.13%
Amazon ESCI			
None	63.70%	96.12%	-
Fine-Tuning	65.28%	97.15%	94.86%
Adv. RL	**67.12%**	**98.34%**	94.03%

Table 2: Impact on classification and ranking performance basis different data augmentation strategies across public datasets.

RQ2. What is the impact of generator size on the relevance model performance? A larger generator is expected to encode more world knowledge, enabling it to generate more diverse and informative queries when properly guided. As shown in Table 3, scaling from FLAN-T5-Large to FLAN-T5-XL for WANDS dataset, enhances both classification performance (PR-AUC) and ranking effectiveness (NDCG@10). The Fine-Tuning approach achieves a +4.89% gain in PR-AUC and +1.31% in NDCG@10, while Adversarial-RL further improves PR-AUC by +5.44%. However, the NDCG@10 gain is comparatively lower (+0.53%), suggesting that while increasing generator capacity significantly enhances classification, its impact on ranking is positive but relatively smaller.

Strategy	FLAN-T5-Large → FLAN-T5-XL	
	Δ PR-AUC (micro)	Δ NDCG@10
Fine-Tuning	+4.89%	+1.31%
Adv. RL	+5.44%	+0.53%

Table 3: Relative improvement in classification and ranking when scaling from FLAN-T5-Large to FLAN-T5-XL for WANDS dataset.

RQ3. How do the weights in parameterization impact the downstream performance? The choice of reward weighting parameters plays a crucial role in determining downstream classifier performance during Adversarial-RL. Figure 3 illustrates the impact of α and β on PR-AUC performance computed across Amazon ESCI dataset.

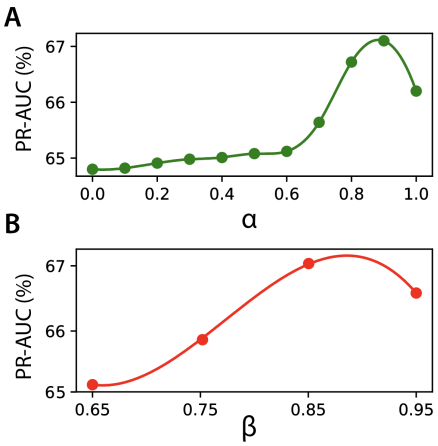

Figure 3: Effect of reward weighting parameters (A) α and (B) β on final classification model performance on ESCI dataset. Data points represent actual observations, while the curve represents a smoothing spline fit.

In Figure 3A, we observe that increasing α enhances PR-AUC, demonstrating that prioritizing classification loss as a reward signal improves the downstream classifier's performance. However, beyond $\alpha = 0.9$, performance degrades, as the diminishing contribution of the generation probability term (completely absent when $\alpha = 1.0$) leads to instability during training. In Figure 3B, we examine the impact of β, which controls the contribution of KL penalty to the reward. Classifier performance improves as β increases up to approximately 0.85, suggesting that lower values allow the adversarial reward to dominate, leading to the generation of semantically irrelevant queries. However, beyond this threshold, performance slightly declines, indicating that excessive regularization limits beneficial exploration.

6 Conclusion

In this work, we propose an adversarial reinforcement learning framework to enhance search query relevance by jointly optimizing query generation and classification using classifier feedback as a reward signal. Empirical results on e-commerce datasets show improved classification and ranking performance over fine-tuning-based augmentation. By incorporating structured rewards, KL regularization, and confidence-weighted training, we ensure informative query generation while minimizing incorrect examples. Deploying our approach in the pharmacy category led to a $+13.9\%$ increase in product views and $+3.8\%$ increase in purchases, demonstrating its real-world effectiveness.

Limitations

While our adversarial reinforcement learning framework enhances query generation and classifier robustness, several challenges remain that require further investigation.

Training Stability. Adversarial training can be unstable, requiring careful hyperparameter tuning to prevent degenerate query generation. Future work can explore advanced regularization techniques to mitigate this issue.

Generalizability to Other Domains. Our experiments focused on e-commerce search, but the framework could benefit other retrieval tasks, such as dialogue systems (retrieving relevant responses in conversational AI), code search (enhancing programming assistant recommendations), and information extraction (retrieving structured data from unstructured documents), among others.

Benefits Beyond Cold-Start. While our approach is particularly beneficial in low-data settings, further evaluation is needed to determine its impact in high-data regimes. Future work should assess whether adversarial query generation improves performance even when ample training data is available.

By addressing these limitations, we can expand the applicability and robustness of our framework across diverse retrieval tasks.

References

Aditi Chaudhary, Karthik Raman, and Michael Bendersky. 2024. It's all relative! – a synthetic query generation approach for improving zero-shot relevance prediction. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1645–1664, Mexico City, Mexico. Association for Computational Linguistics.

Aditi Chaudhary, Karthik Raman, Krishna Srinivasan, Kazuma Hashimoto, Mike Bendersky, and Marc Najork. 2023. Exploring the viability of synthetic query generation for relevance prediction. *arXiv preprint arXiv:2305.11944*.

Yan Chen, Shujian Liu, Zheng Liu, Weiyi Sun, Linas Baltrunas, and Benjamin Schroeder. 2022. Wands: Dataset for product search relevance assessment. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part I*, page 128–141, Berlin, Heidelberg. Springer-Verlag.

Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 191–198, New York, NY, USA. Association for Computing Machinery.

Jesse Davis and Mark Goadrich. 2006. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 233–240. ACM.

Gustavo H De Rosa and Joao P Papa. 2021. A survey on text generation using generative adversarial networks. *Pattern Recognition*, 119:108098.

Jiewen Guan, Bilian Chen, and Shenbao Yu. 2024. A hybrid similarity model for mitigating the cold-start problem of collaborative filtering in sparse data. *Expert Systems with Applications*, 249:123700.

Cuize Han, Pablo Castells, Parth Gupta, Xu Xu, and Vamsi Salaka. 2022. Addressing cold start in product search via empirical bayes. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3141–3151.

Home Depot. 2016. Home Depot Product Search Relevance Dataset.

Akshay Jagatap, Nikki Gupta, Sachin Farfade, and Prakash Mandayam Comar. 2023. Attribert - session-based product attribute recommendation with bert. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 3421–3425, New York, NY, USA. Association for Computing Machinery.

Akshay Jagatap, Srujana Merugu, and Prakash Mandayam Comar. 2024. Improving search for new product categories via synthetic query generation strategies. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 29–37, New York, NY, USA. Association for Computing Machinery.

Bernard J. Jansen and Danielle Booth. 2010. Classifying web queries by topic and user intent. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, page 4285–4290, New York, NY, USA. Association for Computing Machinery.

S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Oleg Lesota, Alessandro Melchiorre, Navid Rekabsaz, Stefan Brandl, Dominik Kowald, Elisabeth Lex, and Markus Schedl. 2021. Analyzing item popularity bias of music recommender systems: Are different genders equally affected? In *Proceedings of the 15th ACM Conference on Recommender Systems*, RecSys '21, page 601–606, New York, NY, USA. Association for Computing Machinery.

Pablo Mateos and Alejandro Bellogín. 2024. A systematic literature review of recent advances on context-aware recommender systems. *Artificial Intelligence Review*, 58(1):20.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

Wentao Ning, Reynold Cheng, Xiao Yan, Ben Kao, Nan Huo, Nur Al Hasan Haldar, and Bo Tang. 2024. Debiasing recommendation with personal popularity. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 3400–3409, New York, NY, USA. Association for Computing Machinery.

Alizée Pace, Jonathan Mallinson, Eric Malmi, Se bastian Krause, and Aliaksei Severyn. 2024. West-of-n: Synthetic preferences for self-improving reward models.

Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. Shopping queries dataset: A large-scale ESCI benchmark for improving product search.

Sonal Sannigrahi, Thiago Fraga-Silva, Youssef Oualil, and Christophe Van Gysel. 2024. Synthetic query generation using large language models for virtual assistants. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2837–2841.

Jiaming Shen, Ran Xu, Yennie Jun, Zhen Qin, Tianqi Liu, Carl Yang, Yi Liang, Simon Baumgartner, and Michael Bendersky. 2024. Boosting reward model with preference-conditional multi-aspect synthetic data generation. *arXiv preprint arXiv:2407.16008*.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2022. Learning to summarize from human feedback. *Preprint*, arXiv:2009.01325.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.