# Enhancing E-commerce Product Search through Reinforcement Learning-Powered Query Reformulation

Sanjay Agrawal
sanjagr@amazon.com
Amazon.com Inc.
Bengaluru, India

Srujana Merugu
smerugu@amazon.com
Amazon.com Inc.
Bengaluru, India

Vivek Sembium
viveksem@amazon.com
Amazon.com Inc.
Bengaluru, India

## ABSTRACT

Query reformulation (QR) is a widely used technique in web and product search. In QR, we map a poorly formed or low coverage user query to a few semantically similar queries that are rich in product coverage, thereby enabling effective targeted searches with less cognitive load on the user. Recent QR approaches based on generative language models are superior to informational retrieval-based methods but exhibit key limitations: (i) generated reformulations often have low lexical diversity and fail to retrieve a large set of relevant products of a wider variety, (ii) the training objective of generative models does not incorporate a our goal of improving product coverage. In this paper, we propose **RLQR** (**R**einforcement **L**earning for **Q**uery **R**eformulations), for generating high quality diverse reformulations which aim to maximize the product coverage (number of distinct relevant products returned). We evaluate our approach against supervised generative models and strong RL-based methods. Our experiments demonstrate a 28.6% increase in product coverage compared to a standard generative model, outperforming SOTA benchmarks by a significant margin. We also conduct our experiments on an external Amazon shopping dataset and demonstrate increased product coverage over SOTA algorithms.

## CCS CONCEPTS

• **Information systems** → **Web searching and information discovery**; **Query reformulation**; • **Computing methodologies** → **Natural language generation**.

## KEYWORDS

Query Reformulation, Natural Language Generation, Reinforcement Learning, E-Commerce, Product Search

**ACM Reference Format:**
Sanjay Agrawal, Srujana Merugu, and Vivek Sembium. 2023. Enhancing E-commerce Product Search through Reinforcement Learning-Powered Query Reformulation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3583780.3615474


*CIKM '23, October 21–25, 2023, Birmingham, United Kingdom*

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3615474

## 1 INTRODUCTION

Product search forms the bedrock of the rapidly expanding e-commerce landscape of emerging markets. However, these markets present a unique set of challenges due to the low English proficiency of the user base. Product search queries often tend to be code-mixed and consist of English words mixed with words from other languages (e.g., "women payal" meaning "women anklet"). Misspelled queries and frequent use of regional jargon further exacerbate the complexity resulting in few and low quality product results. A typical approach to address these challenges is via query reformulation (QR) where the user query is mapped to other semantically similar queries that are then used to fetch results that are collectively presented to the user. For instance, given a query "luggage with wheels", we source additional products from queries such as "trolley bags" and "rolling suitcase" that have similar intent. This approach ensures retrieval of a larger number of relevant products from the catalog while preserving the intent of the user. In online QR, user queries are mapped to reformulated queries in real-time but latency and compute constraints often limit the complexity of matching algorithm resulting in lower matching accuracy and coverage. Therefore in addition to online QR, most e-commerce sites also host offline QR systems with much more sophisticated matching models. In this paper, we focus on offline QR scenario where the task is to map a large corpus of historical user queries to semantically similar reformulated variants that can yield superior product coverage, i.e., fetch a large number of relevant products.

Earlier QR approaches primarily relied on Information Retrieval (IR) methods based on semantic matching models [35] where the input user queries and a pre-curated list of candidate reformulations are all mapped to vector embeddings in a shared semantic space with top reformulations chosen based on the similarity of the embeddings. These approaches have two many limitations. Firstly, IR-based methods are trained solely based on the semantic match of intent between queries without accounting for the efficacy of the queries in terms of the product coverage. Secondly, effective reformulations often tend to share vocabulary with that of product descriptions and IR-methods often fail to retrieve these the good reformulations, due to the "semantic gap" [13] [21] between the vocabulary of user queries and that of product descriptions. Recent QR methods [25] [23] based on Natural Language Generation (NLG) involve training generative models on <query, reformulated query> pairs in a supervised fashion and utilize the model to generate multiple reformulations of an input query using beam search. While these methods can bridge the semantic gap to some extent and provide superior performance over IR baselines, these models still suffer from two key deficiencies. First, the reformulations generated with beam search for a given input query are often very similar

**Table 1: Search queries with samples of reformulations generated using different methods**

| User query | Normal Beam Search | Beam Search with Diversity | | RL with Diversity |
|---|---|---|---|---|
| | | Good Quality | Quality Drift | |
| green saree chanderi silk | sarees green color<br>chanderi silk sare<br>silk saree | Copper Zari Silk Chanderi Saree<br>Chanderi Cotton Silk Blend Saree<br>sari silk 250-300 plain | party wear saree<br>saree for women latest design<br>sarees below 1000 rupees | Chanderi **Cotton and** Silk Saree<br>Chanderi **Handloom** Cotton & Silk Saree<br>**Banarasi** Chanderi Cotton Silk Saree |
| aashirvaad atta 1kg | atta 1kg<br>aashirvaad atta<br>aashirvaad 1kg | Aashirvaad MP Atta 1kg<br>Aashirvaad Select Sharbati Atta 1kg<br>aashirvaad atta 2kg | ghee 1kg<br>atta 25kg bags<br>pilsburry 5kg wheat | aashirvaad **sugar release control** atta 1kg<br>**superior MP** aashirvaad atta 1kg<br>aashirvaad **organic** atta 1kg |

to each other. For example, in Table 1, we can observe that using normal beam search (NBS) results in reformulations with minor syntactic variations such as dropping or adding words, conversion to plural form, etc. This low level of linguistic diversity does not significantly enhance the product coverage. Diverse decoding algorithms [30] during inference have been shown to partially overcome the disadvantages of NBS, potentially increasing diversity to some extent. Table 1 show how beam search with diversity (DBS) can retrieves keywords with greater linguistic diversity, but often there is also a risk (see Column 4) that the generated reformulations may drift away from the original user search intent. A primary limitation of these generative methods is that similar to the IR-based approaches, there is no explicit alignment with the application objective of improving relevant product coverage.

**Contributions.** In this paper, we focus on the offline QR problem and attempt to generate reformulations that are (i) semantically similar to the input user query, (ii) exhibit high lexical diversity with respect to each other, and (iii) maximize coverage of relevant products. We achieve this by combining large language model-based generation with a reinforcement learning (RL) mechanism to optimize the desired custom objective function. Table 1 depicts examples of our approach that yield diverse and superior reformulations. Our contributions are summarized as follows:

1. Keeping in view requirements of product search, we propose a custom reward function based on the key desiderata for query reformulation, in particular, (i) relevance with respect to the input query, and (ii) number of relevant products in search results.
2. We propose a novel, generic RL-based QR framework that consists of (i) a generative language model paired with diverse beam search to produce a set of varied reformulations, (ii) an RL mechanism to determine the optimal policy for generating each token so as to maximize any arbitrary reward function, and (iii) a reward model that computes the goodness of a set of reformulations.
3. Empirical evaluation of the proposed approach on proprietary and public datasets points to significant benefits in terms of product coverage (+28%) and query diversity relative to multiple SOTA baselines. We also present ablation studies to investigate the relative contributions of the various elements of our approach.

Note that our approach extends beyond product search and can be easily customized to include specific metrics in other applications. For example, in sponsored search, advertisers bid on keywords related to their business in order to have their ads appear alongside organic results, Reformulation suggestions in this case need to be generated based not only on the relevance but also the potential revenue. Furthermore, researchers and practitioners can tailor our RL framework to their specific problem and experiment with a variety of reward designs to achieve the learning objectives they desire since it is agnostic of the reward function.

## 2 RELATED WORK

**Query Reformulations:** Reformulation of queries is an important area of research that involves the idea of performing query to query transformations [8] [9]. These methods reformulate the input search query into multiple semantically similar queries, and then retrieve products from the product catalog while preserving user intent. QR work can be categorized into the following categories (but not limited to these). (i) Term dropping and substitution [3] (ii) Term expansion [36] [34] (iii) Machine Translation [22] (iv) Reinforcement Learning [15] [17] [31] (v) Representation learning [9]. There is also research ongoing in the area of keyword to keyword transformations, known as keyword reformulations (KR) [2] [37], which aims to rewrite less frequent keywords without altering the original search intent. Moreover, direct query-to-keyword reformulations (QKR), based on the seq2seq language models [11] [21], are becoming increasingly common. In this paper, we present a general RL-based framework that can be applied to a wide variety of applications, such as KR and QKR.

**Reinforcement Learning in NLP:** In many NLP applications, language models are used to generate text. These models [25] [23] [21] are trained to predict the next word in a sequence, given the previous words and some context. A lot of NLP literature uses reinforcement algorithms [33] [16] to improve heuristic-based evaluation metrics, like BLEU [19] for machine translation. In such cases, policy gradient algorithms such as REINFORCE [32] are used with baselines to optimize the rewards. A growing focus has been placed on utilizing rewards from trained evaluation models in a variety of tasks, such as Machine Translation [10] and Abstracting Summarization [29]. In [15], an RL approach was proposed to generate query rewrites, with reward derived from a trained evaluation model. DRQR [31] proposes a deep reinforcement learning text generation model for query reformulation to automatically generates new reformulations of the query where the author adopts the recurrent neural network based encoder-decoder [5] framework.

## 3 PROPOSED METHODOLOGY

We begin with a formal problem statement, the key solution design choices and then present our methodology

**QR for Product Search:** Formally, the offline product search query reformulation problem can be stated as follows: Let $D = \{(Q^i, \tilde{Q}^i)\}_{i=1}^{N}$ denote training data comprising pairs of semantically similar queries where $Q = \{w_1, ...., w_m\}$ is the user query and $\tilde{Q}$=$\{\tilde{w}_1, ...., \tilde{w}_{\tilde{m}}\}$ is the reformulated query. Here, $w$ represents the word or token either in the user query or in the reformulated query. Let $cov(Q)$ denote the number of relevant products (relevance is with respect to $Q$) returned organically by the search engine for the query $Q$. Given a historical set of queries $D_{test}$, for each $Q \in D_{test}$, the goal is to generate multiple (say $B$) reformulations $\{\hat{Q}_j\}_{j=1}^{B}$ that can be issued to the search engine so that the combined results from the reformulations maximise the overall coverage of relevant products with respect to the original query $Q$.

**Solution Design Choices.** Inspired by the relative success of generative LLM based approaches and the need to incorporate the application specific coverage metric, we adopt an RL-based generation operation for the QR task. There are two key questions to address: **Q1**: How do we structure the reward mechanism in a RL setup for QR tasks? **Q2**: What should be the choice of reward function to maximize the product coverage with respect to original query? Since it is desirable to have multiple reformulations per query for better coverage, there are three possible choices for the reward mechanism (**Q1**): (i) Providing independent rewards for each reformulation and considering a simple aggregation (e.g. sum) of these rewards for policy optimization, (ii) Modeling the desired application metric via a more complex non-linear function of the results from the multiple reformulations, e.g., unique relevant product results (iii) Sequential modeling of rewards for the reformulations, i.e., reward for $k^{th}$ query reformulation is based on only the additional utility taking into account the goodness (i.e., already covered relevant products) of the previous $k-1$ reformulations. In our approach, we prefer the first choice as it simplifies the training process, permits parallelization, and makes training faster, but it comes with the downside of lower accuracy since it does not account for overlapping benefits of the various reformulations. Hence, we need additional safeguards to ensure diversity among the reformulations. To address **Q2**, we consider relevancy with respect to the original query, diversity, and the product coverage. We present our reward function in section 3.1.1 which can be adapted for applications beyond product search by replacing product coverage with criteria such as purchases, revenue, downstream impact.

## 3.1 Product Search Reward Model

*3.1.1* ***Reward Function.*** In formulating a reward function, we aim to maximize product coverage (i.e. relevant products returned) using reformulated queries without degrading the quality of reformulations. Our reward function consists of two components: **(i)** $rel(Q, \hat{Q})$, which is the relevance score between the user query ($Q$) and the generated reformulated query ($\hat{Q}$) computed using our trained relevance model (see section 3.1.2) **(ii)** $cov(\hat{Q})$, i.e., number of relevant products returned for a reformulated query ($\hat{Q}$). Incorporating the relevance score $rel(Q, \hat{Q})$ into the reward function allows us to estimate the relevance of results obtained from the query $\hat{Q}$ with respect to the original query $Q$. As shown in equation 1, we define our reward function as follows.

$$r(\hat{Q}) = \begin{cases} rel(Q, \hat{Q}) * cov(\hat{Q}), & \text{if } \beta \leq rel(Q, \hat{Q}) \leq \alpha \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $0 < \alpha, \beta < 1$ and $\beta < \alpha$. Scalar constant $\alpha$ is the upper bound for relevance score, which has not been set to one to prevent RL from generating reformulations that are near duplicates of the original query and have high individual coverage. This safeguard is required since we consider sum of the rewards of the reformulations instead of the unique distinct products to keep the training process tractable. $\beta$ is the lower bound that ensures a certain amount of relevance between the $Q$ and the $\hat{Q}$. We calculate $cov(\hat{Q})$ based on the number of distinct products impressed for each historically logged reformulated query ($\hat{Q}$).

*3.1.2* ***Relevance Model.*** The relevance model, as discussed in the reward function, is essential for controlling the quality of reformulated queries with different intents from those of the original queries. Several recent studies [18] [28] demonstrate poor correlations between heuristics evaluation based on n-gram overlap [19] [3] and word embedding similarity [27] with human judgements on several NLG tasks. We observed that the performance of such heuristic-based metrics in query rewriting is indeed poor. Therefore, also motivated by [15], we create an end-to-end evaluation metric specifically trained to evaluate the quality of the reformulations. We use the bert-base-uncased [7] model and fine-tune it on a dataset containing triplets of the form $(Q, \tilde{Q}, y)$, where $y$ represents a 1/0 label representing relevancy/non-relevancy between query and reformulated query.

## 3.2 Generation Model

There are many publicly available seq2seq language models such as T5, GPT2, ProphetNet [25] [23] [21], etc. We adopt a recently proposed T5 model[1] to generate query reformulations, which has shown to produce state-of-the-art performance on query reformulations generation tasks [14] [4] [20]. We initialize the T5 model with the pre-trained weights and then fine-tune on our <query, reformulated query> training data $D$ in a supervised learning setting. While inference, we provide a user query as an input and predict top B (also called beam size) reformulated queries.

## 3.3 Design Query Reformulation Task as Reinforcement Learning Problem

RL problems consist of two components (i) Environment and (ii) Agent. An RL agent learns to act intelligently when interacting with the environment and receives rewards or penalties based on the outcomes of its actions. We can formulate query reformulation task as an RL problem. When an episode begins, the environment selects the user query as an initial state. Our agent (generative model) then makes an action $\hat{w}_t$ (choose a token) at time t according to the policy $\pi = P(.|\hat{w}_{<t}, Q, \theta)$. Here, the policy $\pi$ is parameterized by the weights of the Generative model. Taking this action, the environment is updated to the new state, and the agent takes another action $\hat{w}_{t+1}$ at time t+1. The agent continues in this manner until the agent returns an end token in response to the action. Then the

[1] pre-trained on 700 GB of C4 Data [24]

environment returns a reward for the generated full reformulated query $\hat{Q}$ which the agent wants to optimize. The objective of an RL agent aims to learn the policy parameter $\theta$ that maximizes the expected reward defined in equation 2. The standard method for solving the maximization problem is Gradient Ascent (or Descent).

$$J(\theta) = E_{\pi_\theta}[r(\hat{Q})] = E_Q E_{\hat{Q}\sim P(.|Q,\theta)} r(\hat{Q}) \tag{2}$$

where $\hat{Q}$=$\{\hat{w}_1, ...., \hat{w}_{\hat{m}}\}$ is generated reformulated query, $\hat{w}_t$ is the sampled token following the policy $\pi_\theta = P(.|\hat{w}_{<t}, Q, \theta)$ at time t, $r(\hat{Q})$ is the reward for $\hat{Q}$. We will now explain the policy gradient method (in section 3.3.1) to find parameters $\theta$ where reward is maximized.

*3.3.1* ***REINFORCE Method***. The REINFORCE algorithm [32] calculates the policy gradient in equation 3 as follows:

$$\nabla J(\theta) = \nabla E_{\pi_\theta}[r(\hat{Q})] = E_{\pi_\theta}[(\sum_{t=1}^{T} r(\hat{Q}) \nabla log(\pi_\theta))] \tag{3}$$

For our query reformulation task where a mini batch of search queries $\{Q^{(i)}\}_{i=1}^n$ and the corresponding reformulations $\{\hat{Q}^{(i)}\}_{i=1}^n$ sampled from the policy $P(.|Q, \theta)$, we can rewrite equation 3 as equation 4 in the following form:

$$\nabla J(\theta) \approx \sum_{i=1}^{n} r(\hat{Q}^{(i)}) \nabla log(P(\hat{Q}^{(i)}|Q^{(i)}, \theta)) \tag{4}$$

*3.3.2* ***REINFORCE with Baseline***. There is one disadvantage of policy gradients methods, which is the high variance that is caused by empirical returns. In order to reduce variance, a baseline b is subtracted from the rewards in the policy gradient. In essence, the baseline serves as a proxy for the expected return, and it should not introduce any bias to the policy gradient. The baseline must be independent of the policy parameters in order to keep the gradient estimate unbiased. In equation 5, we define the new gradient equation as follows:

$$\nabla J(\theta) \approx \sum_{i=1}^{n} (r(\hat{Q}^{(i)}) - b^i) \nabla log(P(\hat{Q}^{(i)}|Q^{(i)}, \theta)) \tag{5}$$

*3.3.3* ***Diverse Beam Search (DBS)*** *[30].* There have been several studies that work on producing diverse reformulations using generative models. In our experiments, we found *Diverse beam search* algorithm to be the most effective with T5 model when compared with other possible decoding algorithms.

## 3.4 Our proposed Approach: RLQR

With the standard RL setup, equation 2 can be optimized by placing a high probability mass on one plausible reformulation and ignoring all other valid reformulation for a given input query. It is always possible to construct an optimal policy $P^*(\hat{Q}|Q)$, which is completely deterministic given an input query Q i.e. it generates only one reformulation per query. The problem lies with the standard RL objective, since it only requires high expected rewards for a single sample (reformulation) from the policy. This problem is significant for query reformulations since our primary objective is to generate as many distinct high-quality reformulations as possible for a given search query. To overcome this issue in the standard RL method,

**Algorithm 1** A training algorithm for learning generator parameters using a RL approach

**Require:** Train Dataset $T$, Batchsize bs, Diverse Beam Search *DBS*, Beam Size $B$, relevant products returned $cov(\hat{Q})$, Scalar constants $\alpha$ & $\beta$ , Relevance score $rel(Q, \hat{Q})$
**Ensure:** Learn Generative Model Parameters $\theta$
**for** Number of Training Iterations **do**
  $\{Q_{bs=1}, ..., Q_{bs=last}\} \in T$
  **for** i ← 1 to bs=last **do**
    $\{\hat{Q}_1^{(i)}, ...., \hat{Q}_B^{(i)}\} \leftarrow DBS(P, \theta, Q^{(i)}, B)$
    **for** j ← 1 to B **do**
      **if** $\beta \le rel(Q^{(i)}, \hat{Q}_j^{(i)}) \le \alpha$ **then**
        $r(\hat{Q}_j^{(i)}) \leftarrow cov(\hat{Q}_j^{(i)}) * rel(Q^{(i)}, \hat{Q}_j^{(i)})$
      **else**
        $r(\hat{Q}_j^{(i)}) = 0$
      **end if**
    **end for**
    **for** j ← 1 to B **do**
      $b_j^{(i)} \leftarrow \frac{1}{B-1} \sum_{k=1,k\neq j}^{B} r(\hat{Q}_k^{(i)})$
    **end for**
  **end for**
  Updating parameters $\theta$ by ascending its gradient $\nabla J(\theta)$: $\sum_{bs=i} \sum_{j=1}^{B} ((r(\hat{Q}_j^{(i)}) - b_j^{(i)}) \nabla log(P(\hat{Q}_j^{(i)}|Q^{(i)}, \theta)))$
**end for**

we use diverse beam search *DBS* in our RL approach. DBS uses as input, the generator model $P(.|Q, \theta)$ with parameter $\theta$, Query $Q$, and returns B diverse samples $\{\hat{Q}_1, \hat{Q}_2, ....., \hat{Q}_B\}$ from the policy $\pi$. Our RL approach aims to maximize the expected total rewards (described in equation 1) obtained from B diverse reformulations sampled from the policy using the *DBS* decoding algorithm. We define our final RL training objective as follows after considering *DBS* as our decoding algorithm:

$$J(\theta, DBS) = E_Q J(\theta, Q, DBS) \tag{6}$$

$$J(\theta, Q, DBS) = \sum_{j=1}^{B} E_{\hat{Q}_j \sim DBS(P,\theta,Q)} r(\hat{Q}_j) \tag{7}$$

Diverse beam search can control how much diversity is desired in the reformulations, while our reward maximizes product coverage. We optimize the revised objective (in equation 8) using the REINFORCE with baseline algorithm as follows:

$$\nabla J(\theta, DBS) = \sum_{i=1}^{n} \sum_{j=1}^{B} ((r(\hat{Q}_j^{(i)}) - b_j^{(i)}) \nabla log(P(\hat{Q}_j^{(i)}|Q^{(i)}, \theta))) \tag{8}$$

$$b_j^{(i)} = \frac{1}{B-1} \sum_{k=1,k\neq j}^{B} r(\hat{Q}_k^{(i)}) \tag{9}$$

$\{Q^{(i)}\}_{i=1}^n$ is a mini-batch of user queries while $\{\hat{Q}_1^{(i)}, ....., \hat{Q}_B^{(i)}\}_{i=1}^n$ are the corresponding reformulations generated using $DBS(P, \theta, Q^{(i)})$. We briefly describe our proposed RL approach in Algorithm 1 and show our proposed architecture in Figure 1.

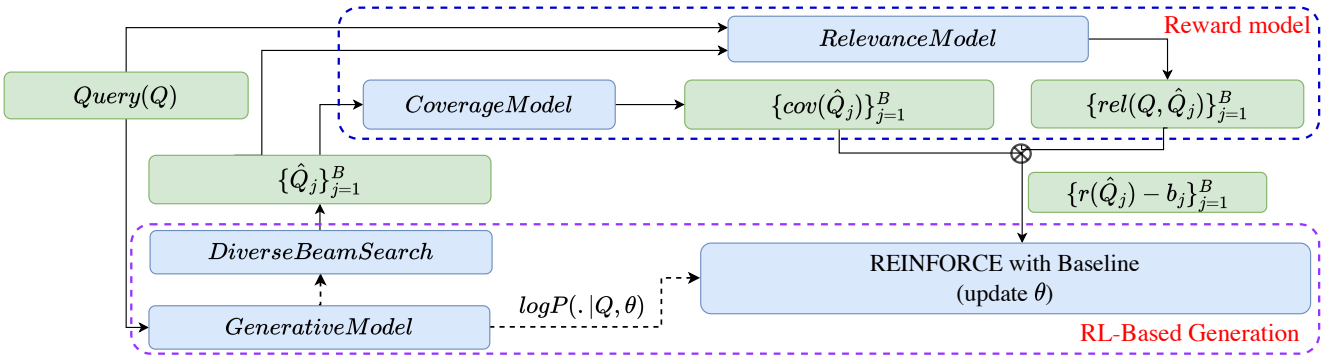


**Figure 1: RLQR framework comprising RL-based generation and a reward model. Coverage model is a look-up on search logs.**

## 4 EMPIRICAL EVALUATION

We present our findings on the benefits of using reinforcement learning for query reformulation tasks. Our discussion begins with a description of the datasets and the experimental setup.

### 4.1 Experimental Setup

**Dataset Generation** We collected a sample of ~2 MM human audited query pairs from IN marketplace, categorized as relevant or irrelevant. Supervised training of T5 models is achieved by using relevant pairs that represent 80% of the total set. Training data for RL is collected from anonymous user behavior data, including a sample of ~1MM user queries from IN marketplace. To evaluate the generator models, we curated a set of 80k search queries.

**Experimental Details** We use T5 as our generative model which uses t5-base with 220 million trainable parameters. We utilize a pre-trained checkpoint and train for 5 epochs on Q-Q data in a supervised setup. We use the Adam optimizer with a batch size of 64 and a learning rate of $1^{*}10^{-4}$. The maximum length of both the source and the target text is set to 20. T5 model performs a particular task by adding a prefix to its input sequence. We use the prefix 'summarize' as our prefix in order to train the model. After training T5 in a supervised manner, we then fine-tune T5 using our RL method for another epoch. As part of our RL setup, we train the model with a batch size of 32. In equation 1, we set $\alpha$ and $\beta$ to 0.95 and 0.5, respectively. The rest of the RL setup is similar to the supervised setup. We conducted all our experiments on 8 GPUs in an AWS p3.16xlarge EC2 instance. The hyperparameters were chosen empirically based on the experiments performed in this study. Our experiments are repeated multiple times by changing the random seed in order to determine statistical significance.

**Why use T5 as a generative model?** Besides T5, many pre-trained seq2seq generative models, such as GPT2 [23], GPT2-medium [6], ProphetNet [21], are publicly available for generating text, such as answering questions, reformulating, summarizing, etc. As part of our experiments, we trained GPT2, GPT2-medium, and ProphetNet models on Q-Q data in a supervised setup. However, the generated texts did not yield satisfactory inference results for our QR tasks.

**Algorithm Baselines** For the purpose of evaluating the efficacy of our proposed method, we take following baseline measurements.

**(i) Normal Beam Search (NBS) [25]:** We use a T5 model, which is trained in a supervised manner and uses a normal beam search for inference. This baseline represents the state of the art performance achieved by recent generative models.

**(ii) Diverse Beam Search (DBS) [30]:** We use the same trained supervised T5 model, but we perform an inference using a DBS. This baseline represents the use of decoding algorithms in generative models to generate a greater degree of diversity.

**(iii) Task-Oriented QR [17]:** Author introduces a RL-based query reformulation system that rewrites a query to maximize the number of relevant products returned. In contrast to our approach, here, a search engine retrieves some products corresponding to a user query and uses them to reformulate that user query.

**(iv) DRQR [31]:** DRQR is an RL-based QR method where the reward is the weighted sum of two components, F1 and QPP[2].

**(v) CLOVER [15]:** CLOVER introduces a diversity-driven RL based algorithm to generate both high-quality and diverse reformulations by optimizing for human assessment of reformulations quality.

**Evaluation Metric** Our objective is to generate reformulations that maximize product coverage with respect to original query. Hence, we use this criterion to evaluate our method against the baselines. Note that product coverage is calculated based on the number of distinct relevant products by all historically logged reformulated queries. Here, Precision@k and Recall@k are not used since labeling millions of query-product pairs is not feasible .

### 4.2 Main Results

In Table 2, we compare the average relevant products returned per user query between supervised and RL-based generations. We use a generative model T5 with beam size 10. Our experiments show that RLQR achieves the highest product coverage among all generation methods, including supervised and RL-based algorithms. In addition, we find that among the supervised baselines, the DBS decoding algorithm improves products returned over normal beam search, but is inferior to most RL-based methods. Among RL-based methods, CLOVER has been optimized to generate diverse reformulations, but returned products are still lower than our method RLQR, indicating that maximizing diversity alone is not likely to optimize product coverage. We also demonstrate our proposed method's efficacy on a publicly available Amazon shopping dataset (Aicrowd)

[2]QPP is based on the query performance predictors like Inverse Document Frequency

**Table 2: Comparison of the number of relevant products returned per user query using various approaches against internal Amazon datasets and external ones. The supervised NBS method has been taken as a baseline, so the gain(%) is 0.**

| Fine-tuning Approach | Decoding Algorithm | Number of Relevant Products Returned /Gain (in %) | |
|---|---|---|---|
| | | Amazon Data | Aicrowd |
| *Supervised Baselines* | | | |
| Supervised | NBS | 2820/0% | 111/0% |
| Supervised | DBS | 3327/17.8% | 122/9.9% |
| *RL Baselines* | | | |
| Task Oriented QR | DBS | 3120/10.6% | 114/2.7% |
| DRQR | DBS | 3390/20.2% | 130/17.1% |
| CLOVER | DBS | 3478/23.3% | 132/18.9% |
| **RLQR** | DBS | **3626/28.6%** | **145/30.6%** |

**Table 3: Average unique quality reformulations and lexical diversity for various algorithms**

| Fine-tuning Approach | Decoding Algorithm | Unique High Quality Query reformulations | Distinct Unigram | Distinct Bigram |
|---|---|---|---|---|
| Supervised | NBS (I) | 2.79 | 0.393 | 0.550 |
| Supervised | DBS (II) | 3.54 | 0.463 | 0.624 |
| RL (RLQR) | DBS (III) | **4.05** | **0.469** | **0.668** |
| *Ensemble Results* | | | | |
| Ensemble (I) | | 2.79 | - | - |
| Ensemble (I + II) | | 4.66 (+1.87) | - | - |
| Ensemble (I + II + III) | | **7.75 (+3.09)** | - | - |

[26]. This dataset is available in three languages: English, Japanese, and Spanish. Since we use EN trained pre-trained models, we consider only EN data when training and validating our models. Our results on public dataset demonstrate that among all baselines, RLQR performs best, outperforming supervised and RL-based baselines. Note that the products returned for public dataset is not as high as Amazon's internal dataset because the former is quite limited whereas the latter is based on huge search logs.

## 4.3 Ablation Study

In Table 3 , we examine our methods' effectiveness in generating Unique High-Quality Reformulations and degree of diversity by calculating the number of distinct unigrams and bigrams [12]. **(i) High Quality Reformulations:** *Unique High-Quality Query Reformulations* refers to the total number of reformulations (average value per input query) retrieved after applying a relevance model threshold. With this relevance model threshold, we ensure that reformulations with the corresponding query meet a high standard of quality. The results demonstrate that RLQR is capable of generating more high quality reformulations than supervised baselines. Additionally, in Table 3, we present ensemble results combining multiple approaches to discover net high quality reformulations as well as net new additions using a new algorithm. Ensemble (I + II) produces a total of 4.66 high quality reformulations where Diverse Beam Search (II) yields 1.87 new reformulations out of 3.54 and the rest were already present in Normal Beam Search (I). In ensemble (I + II + III), we obtain a total of 7.75 high quality reformulations, while our proposed method (III) yields a total of 3.09 net new reformulations. The net new addition of 3.09 new high quality reformulations shows that our RL based approach is able to generate a good amount of more new pairs which were not part of other two approaches. **(ii) Distinct n-gram statistics [12]:** As proposed in [15], we report degree of diversity by calculating the number of distinct unigrams and bigrams in generated reformulations. It counts the total number of distinct n-grams in the generated reformulations combined for an input query. The value is then divided by the total number of tokens generated in the reformulations to avoid favoring long sentences. As shown in Table 3, diverse beam search (II) significantly improved distinct unigram and bigram statistics over normal beam search (I), while RLQR further surpassed diverse beam search.

## 4.4 Deployment Considerations

The key idea of query reformulation is to surface relevant products for a customer query from the products corresponding to the queries $Q = \{q_1, q_2.., q_k\}$ , where $q_1..q_k$ are query reformulations. QR inference pipeline consists of the following broad steps: **HCQC (High Coverage Query Cache)**: We curate these queries based on the high number of products retrieved in the past. **LCQC (Low Coverage Query Cache)**: We also curate a list of head and torso queries, which are mostly repeatable and cover most of the query coverage. Then, we use our RL model to infer over every query in LCQC, generating B reformulations using a prefix trie [1] constraint to ensure that all generated reformulations are from HCQC. This process is followed offline, i.e., ahead of time, and refreshed hourly/daily depending on the inference speed of the model, which is highly dependent on factors like hardware, software optimizations, and implementation. Specifically, whenever a real-time query is requested, a lookup to QR cache is performed to collect all reformulations, and the resulting reformulated queries are then passed to the search index in order to return relevant products within a very low (in ms) latency limit.

# 5 CONCLUSION

An important challenge in the offline QR system is generating high-quality, diverse reformulations that maximize the product coverage. The reformulations generated by the existing generative seq2seq models are often very similar to one another. The low linguistic diversity of these reformulations makes them unsuitable for retrieving more products. Furthermore, due to disparities in training objectives and desired outcomes, existing generative seq2seq models do not produce satisfactory results. As a result of our RL-based approach, we are able to overcome these limitations and maximize product coverage by generating multiple diverse high-quality reformulations.

# REFERENCES


[1] Jun-Ichi Aoe, Katsushi Morimoto, and Takashi Sato. 1992. An efficient implementation of trie structures. *Software: Practice and Experience* 22, 9 (1992), 695–721.

[2] Javad Azimi, Adnan Alam, and Ruofei Zhang. 2015. Ads keyword rewriting using search engine results. In *Proceedings of the 24th International Conference on World Wide Web*. 3–4.

[3] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 65–72.

[4] Jerry Zikun Chen, Shi Yu, and Haoran Wang. 2020. Exploring Fluent Query Reformulations with Text-to-Text Transformers and Reinforcement Learning. *arXiv preprint arXiv:2012.10033* (2020).

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[6] Avisha Das and Rakesh M Verma. 2020. Can machines tell stories? A comparative study of deep neural language models and metrics. *IEEE Access* 8 (2020), 181258–181292.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[8] Jianfeng Gao, Shasha Xie, Xiaodong He, and Alnur Ali. 2012. Learning lexicon models from search logs for query expansion. In *Proceedings of EMNLP*.

[9] Mihajlo Grbovic, Nemanja Djuric, Vladan Radosavljevic, Fabrizio Silvestri, and Narayan Bhamidipati. 2015. Context-and content-aware embeddings for query rewriting in sponsored search. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 383–392.

[10] Julia Kreutzer, Shahram Khadivi, Evgeny Matusov, and Stefan Riezler. 2018. Can neural machine translation be improved with user feedback? *arXiv preprint arXiv:1804.05958* (2018).

[11] Mu-Chu Lee, Bin Gao, and Ruofei Zhang. 2018. Rare query expansion through generative adversarial networks in search advertising. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*. 500–508.

[12] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055* (2015).

[13] Yijiang Lian, Zhijie Chen, Jinlong Hu, Kefeng Zhang, Chunwei Yan, Muchenxuan Tong, Wenying Han, Hanju Guan, Ying Li, Ying Cao, et al. 2019. An end-to-end Generative Retrieval Method for Sponsored Search Engine–Decoding Efficiently into a Closed Target Domain. *arXiv preprint arXiv:1902.00592* (2019).

[14] Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Query reformulation using query history for passage retrieval in conversational search. *arXiv preprint arXiv:2005.02230* (2020).

[15] Akash Kumar Mohankumar, Nikit Begwani, and Amit Singh. 2021. Diversity driven Query Rewriting in Search Advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3423–3431.

[16] Khanh Nguyen, Hal Daumé III, and Jordan Boyd-Graber. 2017. Reinforcement learning for bandit neural machine translation with simulated human feedback. *arXiv preprint arXiv:1707.07402* (2017).

[17] Rodrigo Nogueira and Kyunghyun Cho. 2017. Task-oriented query reformulation with reinforcement learning. *arXiv preprint arXiv:1704.04572* (2017).

[18] Martha Palmer, Rebecca Hwa, and Sebastian Riedel. 2017. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

[19] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[20] Gustavo Penha, Arthur Câmara, and Claudia Hauff. 2022. Evaluating the robustness of retrieval pipelines with query variation generators. In *European Conference on Information Retrieval*. Springer, 397–412.

[21] Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063* (2020).

[22] Yiming Qiu, Kang Zhang, Han Zhang, Songlin Wang, Sulong Xu, Yun Xiao, Bo Long, and Wen-Yun Yang. 2021. Query Rewriting via Cycle-Consistent Translation for E-Commerce Search. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2435–2446.

[23] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[24] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv e-prints* (2019). arXiv:1910.10683

[25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* 21, 140 (2020), 1–67.

[26] 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. arXiv:2206.06588

[27] Vasile Rus and Mihai Lintean. 2012. An optimal assessment of natural language student input using word-to-word similarity metrics. In *Intelligent Tutoring Systems: 11th International Conference, ITS 2012, Chania, Crete, Greece, June 14-18, 2012. Proceedings 11*. Springer, 675–676.

[28] Ananya B Sai, Akash Kumar Mohankumar, Siddhartha Arora, and Mitesh M Khapra. 2020. Improving dialog evaluation with a multi-reference adversarial dataset and large scale pretraining. *Transactions of the Association for Computational Linguistics* 8 (2020), 810–827.

[29] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems* 33 (2020), 3008–3021.

[30] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424* (2016).

[31] Xiao Wang, Craig Macdonald, and Iadh Ounis. 2020. Deep reinforced query reformulation for information retrieval. *arXiv preprint arXiv:2007.07987* (2020).

[32] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3 (1992), 229–256.

[33] Yuxiang Wu and Baotian Hu. 2018. Learning to extract coherent summary via deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[34] Jinxi Xu and W Bruce Croft. 2017. Quary expansion using local and global document analysis. In *Acm sigir forum*, Vol. 51. ACM New York, NY, USA, 168–175.

[35] Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006* (2015).

[36] Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. 2020. BERT-QE: contextualized query expansion for document re-ranking. *arXiv preprint arXiv:2009.07258* (2020).

[37] Hao Zhou, Minlie Huang, Yishun Mao, Changlei Zhu, Peng Shu, and Xiaoyan Zhu. 2019. Domain-constrained advertising keyword generation. In *The World Wide Web Conference*. 2448–2459.