Exploring the Viability of Synthetic Query Generation for Relevance Prediction

Aditi Chaudhary
Google Research
Mountain View, CA, USA
aditichaud@google.com

Karthik Raman
Google Research
Mountain View, CA, USA
karthikraman@google.com

Krishna Srinivasan
Google Research
Mountain View, CA, USA
krishnaps@google.com

Kazuma Hashimoto
Google Research
Mountain View, CA, USA
kazumah@google.com

Mike Bendersky
Google Research
Mountain View, CA, USA
bemike@google.com

Marc Najork
Google DeepMind
Mountain View, CA, USA
najork@google.com

ABSTRACT

Query-document relevance prediction is a critical problem in Information Retrieval systems. This problem has increasingly been tackled using (pretrained) transformer-based models which are fine-tuned using large collections of labeled data. However, in specialized domains such as e-commerce and healthcare, the viability of this approach is limited by the dearth of large in-domain data. To address this paucity, recent methods leverage these powerful models to generate high-quality task and domain-specific synthetic data. Prior work has largely explored **synthetic data** generation or query generation (QGen) for Question-Answering (QA) and binary (yes/no) relevance prediction, where for instance, the QGen models are given a document, and trained to generate a query relevant to that document. However in many problems, we have a more fine-grained notion of relevance than a simple yes/no label. Thus, in this work, we conduct a detailed study into how QGen approaches can be leveraged for nuanced relevance prediction. We demonstrate that – contrary to claims from prior works – current QGen approaches fall short of the more conventional cross-domain transfer-learning approaches. Via empirical studies spanning three public e-commerce benchmarks, we identify new shortcomings of existing QGen approaches – including their inability to distinguish between different grades of relevance. To address this, we introduce label-conditioned QGen models which incorporates knowledge about the different relevance. While our experiments demonstrate that these modifications help improve performance of QGen techniques, we also find that QGen approaches struggle to capture the full nuance of the relevance label space and as a result the generated queries are not faithful to the desired relevance label.

ACM Reference Format:
Aditi Chaudhary, Karthik Raman, Krishna Srinivasan, Kazuma Hashimoto, Mike Bendersky, and Marc Najork. 2023. Exploring the Viability of Synthetic Query Generation for Relevance Prediction. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR eCom'23).* ACM, New York, NY, USA, 10 pages.

SIGIR eCom'23, July 27, 2023, Taipei, Taiwan

1 INTRODUCTION

The task of modeling how relevant a document is to a query is among the most central problems in Information Retrieval, and a key component of many IR systems. The e-commerce domain is no exception, with improved relevance models leading to higher consumer engagement and user satisfaction [7]. That said, the e-commerce domain offers additional challenges for relevance modeling – specifically due to its fluidity, with new products appearing every day coupled with the ever-evolving interests of the user base.

The advent of Large Language Models (LLMs) such as GPT [24], T5 [25], PaLM [8] and LLaMa [29], has unlocked new opportunities for potent relevance modeling. However leveraging LLMs comes with a key requirement: data! As in other IR verticals, e-commerce (relevance) labeled training datasets – that are large enough to train these LLMs – are rare[1]. The proprietary nature of user logs, coupled with the increasing privacy expectations of users and the exorbitant costs of collecting high-quality relevance ratings, limit the availability of such data. To tackle this issue, the predominant solution in the IR community has been to leverage large-scale general-purpose IR datasets and perform (zero-shot / few-shot) transfer learning. In particular the MS-MARCO [20] dataset – mined from Bing search logs – is the largest publicly available dataset (with millions of query-document pairs labeled) and most commonly used to train LLMs to understand query-document relevance.

Recently, an alternative paradigm has emerged to overcome the lack of query logs – **synthetically generated** query logs *i.e.,* Query Generation (QGen). Recent works have successfully demonstrated the use of such techniques across different verticals and IR problems, including Question Answering [30], Passage Ranking [1] and Retrieval [9, 19] – with some recent results [9] even outperforming transfer learning from MS-MARCO. Beyond improving relevance prediction (the focus of this paper), these synthetically generated query logs can also be used as a substitute for real logs in different IR technologies and problems. For instance, applications like training query suggestions systems or automatically creating FAQs for consumer-facing applications [5] could all be performed with such logs.

Thus our **first contribution** is providing the first detailed empirical understanding of QGen approaches in the e-commerce domain. Using data from three different e-commerce benchmarks, we

[1]The one notable exception is the recently released ESCI dataset [26] – which we use and discuss later.

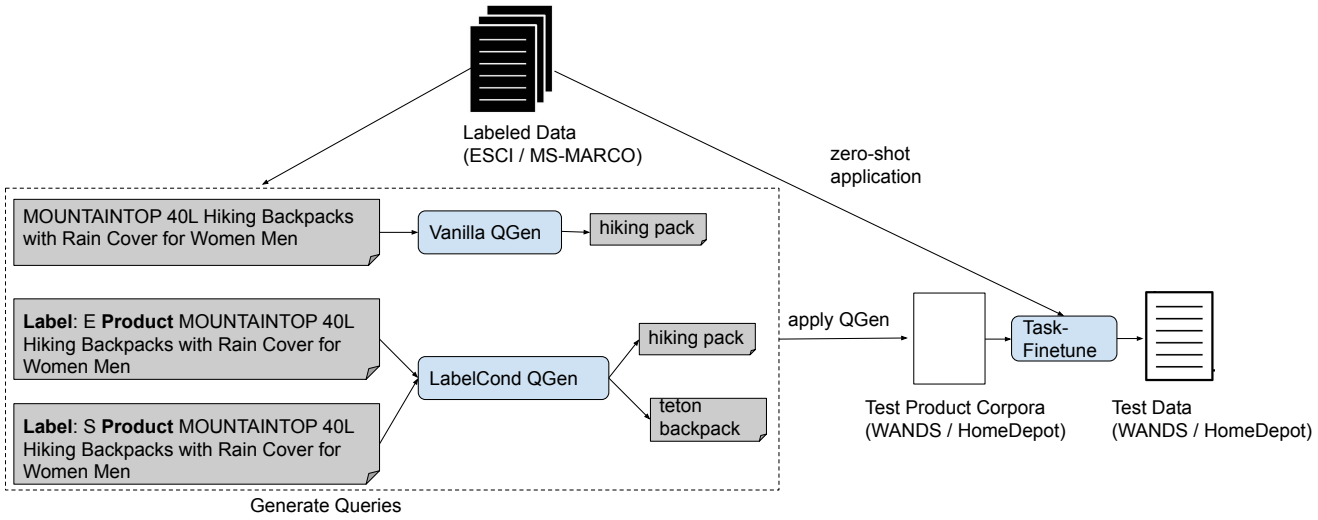

Figure 1: System overview of different approaches we examine for improving relevance prediction on WANDS and HomeDepot. *zero-shot application* is a popular transfer learning strategy where labeled data from existing datasets (ESCI (in-domain) or MS-MARCO (out-of-domain) can be used directly to train downstream task model. *Generate Queries* approach uses labeled data to train Query Generation (QGen) models – *vanilla QGen* which most existing works use, where document information, in our case shopping-product information, is used to generate a relevant query and *LabelCond QGen* which we introduce to generate queries by additionally conditioning on all relevance labels. The generated queries are then used to finetune the downstream model.

study performance of the two major families of QGen approaches (finetuning-based vs. prompt-based) popular in the literature. Our results also demonstrate that models trained using smaller *in-domain* labeled datasets can outperform larger *general-purpose* datasets, thus reinforcing the promise of generating high-quality in-domain synthetic data.

Our **second contribution** involves experiments and analyses that demonstrate that (unlike claims reported in prior works) QGen approaches are outperformed by the more conventional (cross-domain) transfer learning style approaches. Via detailed analyses, we identify a set of key reasons (that we have not seen discussed – or perhaps identified – in prior works) explaining why QGen approaches fall short. For example, we observe that the best existing QGen baseline produces at least one problematic (from the lens of faithfulness / correctness) query for 80+% of products.

Per our study, a key reason responsible for the shortcomings of existing QGen techniques, is their simplification of the label space. More specifically, QGen techniques simplify the problem of query-document (product) relevance into a simple binary one *i.e.,* relevant or not. In fact, most existing approaches only use the relevant query-document pairs, by training the model to produce the associated (relevant) query given the document. This yes/no binarization is unfortunately a gross over-simplification of the complex relationship between queries and documents. For example, TREC relevance judgments are often rated on a 4-point Likert scale. Thus ignoring this nuance seems sub-optimal – as evidenced in our results. Additionally, as noted in Reddy et al. [26], nuanced relevance judgements are important for training a high quality product ranker for a better user search experience. For instance, they define four-class relevance judgements ranging from highly relevant to not relevant. A high quality product ranker should be able to rank the highly relevant product over the next relevance class and so on. Binarizing this would lead to a loss in nuance and thereby the ranking quality. Thus as our **third contribution**, we present modifications to both families of existing QGen approaches (finetuning-based and prompt-based) that recognize and leverage the nuance in the relevance label space. Interestingly, while the finetuning variant leads to the overall best QGen models, we find that the prompt-only methods struggle to understand nuance – indicating potential for future improvements in these pretained prompt models.

2 BACKGROUND: VANILLA QGEN

Powerful transformed-based models like GPT [24], T5 [25], have shown their prowess in generating high-quality text, owing to their ability to attend to even a large context. These models have now become a starting point for generating synthetic data for training further downstream models. In this work, we explore two existing paradigms of QGen approaches – *Finetune-Based* where a QGen model is trained on a subset of training data, and *Prompt-Based* where a large language model (LLM) is leveraged using only few-shot examples. We refer to these existing approaches as *Vanilla* QGen variants as they use information from only the highest relevance label. Below, we briefly describe them.

Finetune-Based. Typically, such a QGen model [1, 18] is given an input text d_i (e.g. passage or document for question generation) and is trained to generate a output question q_i which is relevant to that passage or document. Throughout the paper, the terms 'product, document and, passage' are used interchangeably, but they all refer to an input context which is used for generating the query. Only

Dataset	QGen Input Format	QGen Output Format
ESCI	**Label**: E **Product**: Korean Skin Care K Beauty . **Description**: Seoul Ceuticals CE Ferulic Serum. ..	**Query**: vitamin c serum without hyaluronic acid
	Label: S **Product**: Korean Skin Care K Beauty **Description**: Seoul Ceuticals CE Ferulic Serum. ...	**Query**: indie skincare brand
	Label: C **Product**: Korean Skin Care K Beauty **Description**: Seoul Ceuticals CE Ferulic Serum. ..	**Query**: gundry dark spot diminisher
	Label: I **Product**: Korean Skin Care K Beauty **Description**: Seoul Ceuticals CE Ferulic Serum. ..	**Query**: victim without a face
MS-MARCO	**Label**: Relevant **Document**: Denier (measure): Wikis Thread count or threads per inch (TPI) is a measure of ...	**Query**: thread count definition
	Label: Irrelevant **Document**: Denier (measure): Wikis Thread count or threads per inch (TPI) is a measure of ...	**Query**: perle cotton thread definition

Table 1: Example QGen input/output formats for the ESCI and MS-MARCO dataset. The bolded words is the template we use for constructing the input and output text strings – input is Label: <label> Product: <product title> Description: <product description> and output is Query: <query>.

the relevant query-document pairs from these datasets (e.g. MS-MARCO, Yahoo Answers, Stack Exchange) are used for training such a QGen model. The QGen model is then applied to documents (from the task of interest) to generate synthetic relevant questions. For training QA models, these new question-document pairs are directly used for data augmentation [1, 15, 30]. For training neural retrieval models, an additional retriever (e.g. BM25) is then used to retrieve negative documents for every synthetic relevant question [19, 23].

Prompt-Based. Instead of training a full QGen model, recent works such as PROMPTAGAGOR [9] and INPARS [3] leverage large language models (LLMs) as query generator. For instance, PROMPTAGAGOR concatenates 8 relevant question-document pairs $\{(q_0, d_0) \cdots (q_7, d_7)\}$ with the target document of interest (d_t) and prompts the LLM to generate a new question (q_t) that is relevant to d_t. Then, a retriever is used on the generated new query to construct hard negatives to train a new model on the downstream ranking task. INPARS uses 3 question-document pairs followed by a BM25 retriever to train a T5-reranker model.

In this work, we explore the application of these existing QGen approaches to a much harder relevance prediction task, where it has multiple relevance classes which are nuanced as opposed to only binary relevance prediction (e.g in MS-MARCO). In the next section, we describe our adaptations to the above QGen approaches which conditions on all labels.

3 PROPOSED: LABEL-CONDITIONED QGEN FOR FINE-GRAINED RELEVANCE PREDICTION

As mentioned above, the task of relevance prediction for e-commerce entails – given a user-issued query and a product, predict the degree of relevance (e.g. highly relevant, partially relevant, irrelevant) between them. Consider an example from Table 1, where we can see the fine-grained difference in queries across different relevance labels for the same ESCI product. Simply binarizing this task or only considering queries from one relevance label, as done in the above strategies, could risk losing the nuance. Therefore, we extend the above described vanilla QGen techniques to our nuanced relevance prediction task by conditioning the query generation on the relevance label. Below we describe our adaptations:

- **Finetune-Based-LabelCond**: we use the entire training portion of the available data and not just the relevant query-document portion, to train the QGen model. Specifically, each annotated query-document-label triple is transformed such that the label l_i is prepended to the document d_i and the model is trained to the output the query q_i, as shown in Table 1.
- **Prompt-Based-LabelCond**: we follow PROMPTAGATOR and instead of using all 8 examples of just the relevant label, we use 2 labels per each relevance label, where again, the label is prepended to the respective example as shown in Table 1.

As before, the QGen model is applied to the product corpora of the target domain, which generates query-product examples for all labels, on which a downstream task model is then trained. In the next section, we describe in detail this entire process.

4 EXPERIMENTAL SETUP

We conduct experiments for the zero-shot setting, where we assume that we do not have any training data for our dataset of interest. We use two e-commerce datasets as our target, namely, WANDS and HomeDepot, both described below. These datasets were selected to fulfill the following desiderata – a) they provide a significantly-sized test sets in the e-commerce domain and has real-world impact, and b) they have fine-grained nuance in the relevance judgements. To understand the effectiveness of QGen over the more conventional transfer learning approaches, we compare the cross-domain transfer learning approach (which is non-QGen) with two QGen approaches (vanilla vs label-conditioned). For the zero-shot cross-domain transfer learning, we train a downstream relevance prediction model on existing datasets, namely, ESCI and MS-MARCO, where MS-MARCO is more general-purpose while ESCI is e-commerce focussed, albeit much smaller in size. The QGen models are similarly trained on ESCI and MS-MARCO, and applied to the two target datasets to create training data for the downstream task (Figure 1). We now present the datasets in more detail.

4.1 Data

MS-MARCO. Bajaj et al. [2] first introduced the MS MARCO dataset which is constructed from Bing search logs having 8 million passages extracted from general-purpose web documents. Over the years this dataset has been updated and subsets of it have been used for many shared tasks (e.g. TREC[2]). In this paper, we use the same MS-MARCO data as used by Zhuang et al. [36] which comprises of 530,000 queries and a passage corpus of 8 million, each query being annotated with binary relevance judgements (0 for not relevant and 1 for relevant). Furthermore, Zhuang et al. [36] retrieve 35 hard negatives for each relevant query and upsample the relevant examples to match the irrelevant examples, we refer the reader to the paper for more details.

ESCI. Reddy et al. [26] comprises of 2.6 million manually labeled query-product relevance judgements obtained from the Amazon Search pool. To the best of our knowledge, this is the largest shopping queries dataset publicly available which comprises of 130k unique queries covering three languages, namely English, Spanish and Japanese. The query-product pairs are rated for four relevance labels: **Exact (E)** when the product is exactly relevant to the query, **Substitute (S)** when the product is somewhat relevant but it fails to satisfy all requirements of the query (e.g. showing a 'red sweater' product for a 'blue sweater' query), **Complement (C)** when the item doesn't satisfy the query but could be used in combination with the query (e.g. showing 'hydration pack' for a 'hiking bag' query), and **Irrelevant (I)** when the product is completely irrelevant to the central aspect of the query (e.g. 'harry potter book' for a 'telescope' query.

WANDS. Chen et al. [6] is a product-search relevance dataset released by WayFair [3], which primarily focusses on home improvement. It comprises of 233,448 human-annotated relevance judgements comprising of 480 unique queries and 42,994 unique products. Unlike ESCI, WANDS has been labeled with three relevance labels, namely, **Exact-Match** where the product fully matches the user query, **Partial-Match** where the product somewhat matches the query in terms of the target entity but does not satisfy the modifiers, and **Irrelevant** where the product is not relevant to the user query. We consider all 233k examples as our test set for evaluation.[4]

HomeDepot. The *Home Depot Product Search Relevance*[5] released by Home Depot[6] retailer, comprises of 73,789 training examples[7] and 166k test examples, focussing on home improvement e-commerce. However, the relevance labels for the test split are not released publicly so we use the entire train portion for our zero-shot evaluation. These comprise of 54,470 unique products with relevance labels scored between 1 (not relevant) to 3 (highly relevant).

Table 7 shows some examples for all these datasets and in Table 8 we describe the statistics for each dataset.

[2]https://trec.nist.gov/

[3]https://www.wayfair.com/

[4]There is no designated train/test split provided so for reproducibility we use the entire data as our test set.

[5]https://www.kaggle.com/c/home-depot-product-search-relevance/

[6]http://www.homedepot.com.

[7]The original dataset had 74,068 examples but 279 of those had parsing issues.

4.2 QGen Setup

We use the pretrained mT5-XXL [33] model (13B parameters) as our starting point for all **Finetune-Based*** models, which has been trained for 1M steps on a multilingual corpora, which gives our subsequent models the ability to generate in many languages inherently. For **Prompt-Based*** models ,we use the same setup as PROMPTAGATOR [9] which uses FLAN-137B as the large language model (LLM) [32]. We use the t5x code base https://github.com/google-research/t5x to train all models.

Train QGen. We finetune a QGen model using the training portion of MS-MARCO and ESCI. We use the same train/dev/test splits as provided with the respective datasets and transform the input/output as shown in Table 1. We finetune the QGen model for 100k additional steps with a constant learning rate of 1e-4, Adafactor optimzer, batch size 128, input sequence length 256, target length 32.[8] The best checkpoint for subsequent steps was selected using the performance of BLEU on the validation set.

Apply QGen. Next, we apply the above trained models to generate query-product pairs on WANDS and HomeDepot. For the *label-conditioned* models, the input text is a concatenation of the desired label and the product, whereas for its *vanilla* QGen counterparts the input text is simply the product information. Since ***-LabelCond** QGen models have the ability to generate queries for different relevance labels, unlike their vanilla counterparts which only can generate queries for one relevance label, we generate queries for all relevance labels for a given product. Similar to the training setup, we use input sequence length of 256 with target length to be 32. As an additional filtration step, we remove duplicate queries i.e. if the same query is generated for different labels of the same product, we only retain that query-product-label triple which has the highest model probability.[9]

4.3 Evaluate QGen for Utility

We automatically evaluate the generated synthetic data for its utility to the downstream task. To do that we evaluate the models trained on above-generated data on the respective test sets. We split the resulting filtered QGen data into a train and validation set with a 90:10 ratio such that there is no product overlap across the two sets. We experiment with two styles of downstream models, *classification* and *ranking*.

classification. We use a pretrained mt5-XXL based encoder-only model to perform multi-class classification, and report NDCG.[10] For ESCI-based QGen models, this would become a four-class classification task. We finetune the mt5-encoder for additional 25000 steps with a constant learning rate of 1e-4.[11] We use a batch size of 64 with input sequence length of 608. The reason we chose NDCG, a ranking metric, instead of accuracy is because there is a label mismatch between ESCI and WANDS, which helps avoid an over-simplification by deterministic mapping across the two label sets.

[8]100k steps amount to approximately 8 epochs which we thought were sufficient given the computational and time requirements.

[9]see Table 6 for number of duplicate queries.

[10]We had also tried an encoder-decoder model for the classification task but found encoder-only model to outperform slightly.

[11]We tried two other learning rates of 1e-3 and 5e-5 but found them to be underperforming.

Type	QGen Model	Downstream Model	NDCG@5	NDCG@10	NDCG@20
Baseline	-	Random	0.5776	0.5989	0.6285
	-	zero-shot (ESCI)	**0.8902**	**0.8927**	**0.8987**
	-	zero-shot (MS-MARCO)	0.8642	0.8649	0.8681
Baseline (Vanilla QGen)	PROMPT-BASED (ESCI)	ranking	0.7062	0.7118	0.7217
	FINETUNE-BASED (MS-MARCO)	ranking	0.795	0.7984	0.8015
Ours	PROMPT-BASED-LABELCOND (ESCI)	classification	0.6189	0.6355	0.6561
	FINETUNE-BASED-LABELCOND (ESCI)	classification	0.847	0.8553	0.8661
	FINETUNE-BASED-LABELCOND (MS-MARCO)	ranking	0.8213	0.8318	0.8446

Table 2: Results on WANDS dataset where we report NDCG@5,10,20 (higher the better).

Type	QGen Model	Downstream Model	NDCG@5	NDCG@10	NDCG@20
Baseline	-	Random	0.8939	0.9394	0.9433
	-	zero-shot (ESCI)	**0.9212**	**0.9546**	**0.9575**
	-	zero-shot (MS-MARCO)	0.9144	0.9509	0.9542
Baseline	PROMPT-BASED (ESCI)	ranking	0.9042	0.9445	0.9480
Ours	FINETUNE-BASED-LABELCOND (ESCI)	classification	0.9151	0.9513	0.9544

Table 3: Results on HomeDepot dataset.

This helps evaluate whether the model is correctly ranking exactly-relevant over partially-relevant over irrelevant. In order to compute NDCG, we need a relevance score output for each query-document pair. So, from a downstream model based on the four-way ESCI classification model, we output the prediction probability $p(y_i|x_i)$ where x_i is the concatenation of input query, product title, product description and y_i is the output label (E/S/C/I). We then compute a final score by taking an expectation of the prediction probabilities by multiplying it with the label weight:

$$E_i = \sum_{j=\{E,S,C,I\}} p(y_i^j|x_i) * w_j$$

$$w = \{E = 3.0, S = 2.0, C = 1.0, I = 0.0\}$$

An astute reader may wonder why we go through the trouble of training a multi-class model as opposed to using a ranking model. The reason is that we want the ability to generate new queries for different relevance labels. This is important for search engines where queries/products that have had more user-clicks are often indexed and served with priority (to avoid latency). However, rare or new products often are not covered as they do not have any query associated with them, so having the ability to generate queries across relevance labels for such products becomes crucial to increase coverage. However, for completeness, we do report results from a neural re-ranker and find them to underperforming than the classification model (details in section 5).

ranking. In the neural re-ranker setup, we use the RankT5 model [36] which uses T5 encoder with pointwise ranking loss wherein the loss for each query-document pair is independently computed. The authors train the RankT5 model on MS-MARCO which has binary relevance judgements. We follow the same modeling setup as them, with the main difference being that we use mt5-XL as our starting point instead of T5-Large, as used by them.[12] Input sequence length is 256 with constant learning rate of 1e-4. This ranking model is used in the FINETUNE-BASED and PROMPT-BASED QGen baselines to evaluate the downstream performance. In these baselines, as you recall, the QGen models are trained to generate only relevant queries. To create training data for the ranking downstream model, we need to create negative query-document pairs as well i.e. documents which are not relevant to a query. For this, we use a dual-encoder T5-based retriever [21, 22][13] to retrieve top-35 documents for every generated query. We use all 35 as our hard negative query-document pairs and upsample the relevant documents to have a equal label distribution and train a RankT5 model. This model ranks the target query-product pairs so we directly use that to compute NDCG.

Below, we briefly summarize all the model variants we experiment with.

4.4 All Model Variants

First, we describe the baselines which do not use QGen:

- **Random** where for the target datasets the documents for a given query are randomly ranked.
- **zero-shot (ESCI)** where we train a downstream model for multi-class classification on all of the ESCI training data and apply it directly to WANDS and Homedepot test data.
- **zero-shot (MS-MARCO)** where we train a ranking model using RankT5 [36] with the pointwise loss function on the MS-MARCO training data and apply it directly to the WANDS and Homedepot test data.

Next, we describe the baselines which use existing QGen approaches:

[12] Due to hardware restrictions we could not train the mt5-XXL model variant with their code setup.

[13] We used the mt5-BASE model finetuned with the unsupervised objective proposed by Izacard et al. [14], based on the t5x-retrieval code base: https://github.com/google-research/t5x_retrieval.

Desired Labels	PROMPT-BASED-LABELCOND (ESCI)	FINETUNE-BASED-LABELCOND (ESCI)	FINETUNE-BASED-LABELCOND (MS-MARCO)
Label: E	bed frame with storage plans free	wood bed frame	-
Label: S	acacia wood bed	king bed frame with drawers	-
Label: C	how much does a queen size mattress cost	adjustable bed frame king	-
Label: I	what is the best platform bed frame	headboard with lights and frame	-
Label: Relevant	-	-	what kind of wood is used for bed frames
Label: Irrelevant	-	-	what is a king size bed frame
Gold Label: Exact	hardwood beds		
Gold Label: Partial	geralyn upholstered storage platform bed, floating bed, beds that have leds		
Gold Label: Irrelevant	jordanna solid wood rocking		

Table 4: Some examples of generated queries from different QGen models for product "solid wood platform bed good , deep sleep can be quite difficult to have in this busy age . fortunately , there ' s an antidote to such a problem : a nice , quality bed frame like the acacia kaylin . solidly constructed from acacia wood , this bed frame will stand the test of time and is fit to rest your shoulders on for years and years . its sleek , natural wood grain appearance provides a pleasant aesthetic to adorn any bedroom , acting both as a decorative piece as well as a place to give comfort after a hard day of work . our bed frame is designed to give ample under-bed space for easy cleaning and other usages , with a headboard attached to further express the craftiness . it can be used with other accessories such as a nightstand or bookcase headboard and is compatible with many types of mattresses including memory foam , spring , or hybrid ones . there ' s nowhere better to relax than your own home , and with this bed frame that feeling of homeliness will even be more emphasized . rest comfortably and in style . ". For reference, we also provide the gold queries from the WANDS test data for the reader.

- **Prompt-Based (ESCI)** where we randomly sample 8 query-product pairs from ESCI having *Exact (E)* relevance label and similar to PROMPTAGATOR prompt FLAN-137B to generate one relevant query for a new WANDS/Homedepot product. For the downstream application, we follow the **ranking** setup described in subsection 4.3.
- **FineTune-Based (MSMARCO)** where we finetune the QGen model on only those query-passage pairs from MS-MARCO that have *Relevant* label. For every new target product, we generate one relevant query and use the retriever to retrieve 35 documents as negative examples following the **ranking** setup.

Finally, we describe our adaptations of the above QGen approaches:

- **Finetune-Based-LabelCond (ESCI)** where we finetune the QGen model on all ESCI examples, and for every new target product generate queries for all four relevance labels. For the subsequent downstream model, we initialize it with the multi-class classification model trained on all ESCI data (which we had used in our **zero-shot** setting), and further finetune it on the synthetic data, following the **classification** setup.
- **FineTune-Based-LabelCond (MSMARCO)** where we finetune the QGen model on MS-MARCO examples which and for every new product generate two queries for each of the two relevance labels. We use the **ranking** setup to train the downstream model and initialize it with the MS-MARCO-finetuned-ranking model (used in the **zero-shot** setting).
- **Prompt-Based-LabelCond (ESCI)** where we prompt FLAN-137B with 8 ESCI examples comprising of 2 examples per each relevance label. For every new target product, we generate queries for all four relevance labels, and follow **ranking** setup to train the downstream model.

5 RESULTS AND DISCUSSION

In this section, we present the results of two major QGen families (finetune-based vs prompt-based) comparing them with cross-domain transfer learning approach. Since WANDS is a more recent and challenging dataset in comparison to HomeDepot[14], we focus on WANDS for our discussion. We report results for WANDS in Table 2 and for HomeDepot in Table 3. Here are our main findings:

Zero-shot Transfer Learning wins over any QGen! Overall, we find that zero-shot transfer learning outperform all QGen approaches, both vanilla and label-conditioned. This is unlike what existing works such as INPARS and PROMPTAGATOR where QGen approaches give the best downstream performance. This could be attributed to the difficulty of the downstream task, which in this case is a nuanced relevance prediction task, while the existing works focus on binary relevance which is much simpler.

Label-conditioned QGen wins over vanilla QGen! Within the QGen approaches, we find that our adaptation of conditioning on all relevance labels outperforms the vanilla versions which do not. From the results of FINETUNE-BASED and FINETUNE-BASED-LABELCOND trained on MS-MARCO, we find that exposing the QGen models to all labels (in the case of MS-MARCO they are binary) performs better by +3.3 NDCG@10 points. Therefore, we finetune with all labels on a related dataset (ESCI) for WANDS and find that it outperforms even the MS-MARCO-based QGen models. For prompt-based QGen models, we find that its label-conditioned counterpart underperforms its vaniila variant. However, the prompt-based vanilla variant is far behind (-8.3 NDCG@10 points) the finetune-based vanilla variant to begin with.

In-domain training is important! We find that for both transfer learning and QGen approaches, transferring from a related domain is important in downstream performance. For instance, within the transfer learning models, the model trained on ESCI (zero-shot (ESCI))

[14] We refer the reader to Chen et al. [6] for more information.

gives the best downstream performance, even outperforming the model trained on MS-MARCO (zero-shot (MS-MARCO)), which is trained on nearly 10 times larger training data than ESCI. This again emphasizes that having a related dataset to transfer from is essential for downstream performance, similar to Gururangan et al. [13]. Similarly, within QGen approaches, the label-conditioned model trained on ESCI (Finetune-Based-LabelCond (ESCI)) outperforms its MS-MARCO counterpart. Clearly, relatedness of the target dataset to the training dataset is also important for the QGen model training.

Below we discuss the probable reasons for the shortcomings of QGen approaches. We inspect three QGen models which have been trained with all labels, namely, PROMPT-BASED-LABELCOND (ESCI), FINETUNE-BASED-LABELCOND (ESCI) and FINETUNE-BASED-LABELCOND (MS-MARCO), for the number of duplicate queries generated by the model. Specifically, a duplicate query here refers to the QGen model producing the same query across different relevance labels for the same product. In Table 6 we report the results for WANDS. As you recall, for each of the WANDS 42,994 products, the QGen models trained on ESCI, generated 171,976 queries, one for each of the four relevance labels. For QGen models using MSMARCO, we generate 85,988 queries, one of each of the two relevance classes. In Table 6 we find that the FINETUNE-BASED-LABELCOND (ESCI) QGen model produces duplicate queries for 81% of the products, which suggests that simply prepending label information in the input context is insufficient for the model to learn how to generate discriminative queries. We would also like to highlight the fact that this is happening despite exposing the QGen model to the entire ESCI training data which is 1.6 million examples, of which only 5 of the 1.1 million products had duplicate queries. In Table 5 we report the distribution of generated queries across different labels, after applying the filtration step (described in subsection 4.2) where we remove the duplicate queries. Clearly, noise in the synthetic queries causes errors in the subsequent downstream models. Interestingly, despite PROMPT-BASED-LABELCOND (ESCI) and FINETUNE-BASED-LABELCOND (MS MARCO) models having more number of valid queries, they still underperform FINETUNE-BASED-LABELCOND (ESCI).

The reason why FINETUNE-BASED-LABELCOND (MS-MARCO) is underperforming its ESCI counterpart could be attributed to a) the difference in domain and, b) the style of queries. For instance, queries from MS-MARCO-trained-QGen models are more formal *what*-style questions, while queries from ESCI-trained-QGen models are more informal and similar in style to the gold queries. Although, PROMPT-BASED-LABELCOND (ESCI) has far fewer duplicate queries it severely underperforms probably because of poor overall quality. In Table 4 we present the generated queries from different QGen model for a product. We also provide the user-issued or gold query from the WANDS test set for the same product.[15] For PROMPT-BASED-LABELCOND (ESCI) we see that the query for the highest relevance label i.e. 'E' focuses on the entity bed frame with free storage plans, while from the product description we know that it is mainly about a bed frame which is made from acacia wood and additionally has storage. Nowhere does the product talk about storage plans. In fact, the query for the next relevance label 'S' is more relevant than the one for 'E'. Clearly, exposing the models to only 8 examples, as proposed by PROMPTAGATOR [9] is insufficient, in comparison to the 1.6 million examples used by FINETUNE-BASED-LABELCOND (ESCI), especially for the WANDS dataset. On the other hand, PROMPTAGATOR work had found that exposing the models to only 8 task-specific examples for QGen had outperformed finetuned models which were trained on $O(100k)$ MS-MARCO examples. We would like to note that the Dai et al. [9] also apply an additional consistency filtration step to the generated queries, wherein they only retain those queries which are answerable from the passage from which it was generated. They find that that adding this round-trip consistency adds 2.5 points (avg.) but for smaller datasets it negatively impacts the downstream performance. Therefore, we experimented with round-trip consistency for the FINETUNE-BASED-LABELCOND (ESCI) model for WANDS, which is the best among all QGen variants. Specifically, we use the downstream relevance prediction model trained on ESCI (i.e. the model used for **zero-shot** transfer learning) and re-label the generated queries.[16] We first find that the predicted label of 49% of the generated queries do not match the label which was used to generate the query (i.e. the desired label). We then use the predicted label as the final label for that query and train a downstream model as before. We find this results in only +1 point improvement.[17]

This highlights that even though QGen techniques offer a promising solution for adapting models to new domains, they need further investigation and analyses to make them more effective across different tasks.

6 RELATED WORK

Synthetic Question Generation has come a long way from relying on simple but rigid heuristics [28] to using neural-network approaches, specifically seq-to-seq model [11, 12, 27, 35], to now even leveraging large language models (LLMs) through prompting [9]. Much of the work in this area has focussed on question generation in the context of QA systems. Below we describe some of the representative works in this area.

QGen for QA. Pre-transformer era had seq-to-seq models trained with attention to read an input sentence and generate a question with respect to an answer which is contained in that sentence e.g. for factiod QA [12, 35] Du and Cardie [11] go beyond using single sentence context (as Du et al. [12] note that 30% of SQuaD questions span answers beyond single sentence) for generating questions. Transformers [31] changed the game subsequently with their power of attention to refer to specific parts of text – the QGen models have further improved. For instance, Lopez et al. [18] use a GPT-2 [24] language model to train a question-generation model using the passage as input. They also train an answer-aware variant where they mark start and end of the answer span with special tokens in the context. However, they find the answer-aware variant to be under-performing for question generation (in terms of BLEU metric) than the answer-unaware model. They hypothesize that this is

[15] Note that all products in the test set do not have queries for each relevance label, we simple sampled from those products which have for qualitative evaluation purposes.

[16] Dai et al. [9] use the downstream model trained on synthetic data instead we use a model trained on good quality ESCI data

[17] Given that, for WANDS in Table 2 the PROMPT-BASED-LABELCOND (ESCI) is almost 21 points behind its finetuned counterparts, we did not apply this additional step which would require additional model training.

QGen Model	Desired Label	Generated Query Distribution
PROMPT-BASED-LABELCOND (ESCI)	Label: E	32816
	Label: S	33271
	Label: C	34998
	Label: I	33455
	All	134540
FINETUNE-BASED-LABELCOND (ESCI)	Label: E	37573
	Label: S	28969
	Label: C	26986
	Label: I	23651
	All	117179
FINETUNE-BASED-LABELCOND (MS-MARCO)	Label: Relevant	42790
	Label: Irrelevant	40933
	All	83723

Table 5: Total number of generated queries (post filtering) for WANDS. Total number or products: 42994.

Number of Products	PROMPT-BASED-ALLLABELS (ESCI)	FINETUNE-BASED-ALLLABELS (ESCI)	FINETUNE-BASED-ALLLABELS (MS-MARCO)
at least 1 duplicate	2825	35126	2335
duplicate query for E and S	543	13805	NA
duplicate query for S and C	595	14105	NA
duplicate query for C and I	571	15687	NA

Table 6: Measuring for how many times does the QGen model produce the same or duplicate query across different relevance labels for the same product. For MS-MARCO-QGen model, remember we only had binary relevance labels, hence rows 2–4 are not applicable for this model. Total number or products: 42994.

because there is no explicit mechanism to inform the model on how to use the answer information, somewhat similar to what we find in our label-conditioned models as well where they seem to not use the label information effectively. Ünlü Menevşe et al. [30] explore question-generation for Spoken QA task. More recently, Cao and Wang [4], Chakrabarty et al. [5], Ko et al. [15] propose approaches to generate more open-ended questions, whose answers often span multiple sentences and could be long-form. Cao and Wang [4] create a question-type ontology to guide the model to generate a particular type of question. They essentially concatenate the question-type with the multi-sentence input to generate the question. In the hope of controlling the question generation, they train it jointly with question focus prediction which uses semantic graphs. In principle the question focus and label conditioning are related as in our case, question focus is the conditioned label, however, their main goal of work is to generate questions which are diverse and illicit complex reasoning or curiosity [15]. It is not evaluated on improving any downstream tasks.

Label-Conditioned QGen. Some previous work have looked at label conditioning in QGen models for classificication tasks. Kumar et al. [16], Yang et al. [34] find that pre-pending class-labels to input text is quite effective in class-conditional test generation and thereby data augmentation. They show the effectiveness of this approach for classification tasks (e.g. SST-2 with binary relevance, SNIPS with 7 intents, and TREC with six-classes, SNLI, commonsense reasoning) across different pretrained LMs including auto-encoder LM (BERT Devlin et al. [10]), auto-regressive LM (GPT-2 Radford et al. [24]) and pretrained seq-2-seq LM (BART [17]). In this work, we look at fine-grained relevance prediction, where the task is difficult in that the multiple classes have an inherent ordering, and therefore it is harder for QGen models to produce discrimnative queries across such fine-grained labels.

7 LIMITATIONS AND NEXT STEPS

From the above results, it is apparent that QGen approaches, although offering a promising direction especially for zero-shot settings, need considerable work to outperform transfer learning. Clearly, simply adding label information in the input context does not provide a sufficient signal for the model to generate discriminative queries. We need to explicitly enforce this signal throughout the QGen training process. In this work, we only generate one query, but using beam search we could generate multiple queries for a given product-label combination, resulting in a diverse collection. Another challenge in working with QGen approaches is that the typical strategy for evaluating the synthetic data is to evaluate it on a downstream task, requiring two additional steps after training a QGen model: applying the QGen model for generating queries and then training a downstream task model, to understand the effect of synthetic data. So if a researcher wanted to experiment with multiple QGen models, they would have to run three times the number of experiments to understand which QGen model is the best one, which is a waste of resources and time. This means that we need to come up with an intrinsic evaluation metric that correlates well the downstream task performance. Our next steps are focused on addressing these issues.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for the valuable feedback and suggestions. We would also like to thank Honglei Zhuang and Rolf Jagerman for helping us run and adapt RankT5 model for our experiments.

REFERENCES

[1] Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA Corpora Generation with Roundtrip Consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 6168–6173. https://doi.org/10.18653/v1/P19-1620

[2] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. MS Marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* (2016).

[3] Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Data augmentation for information retrieval using large language models. *arXiv preprint arXiv:2202.05144* (2022).

[4] Shuyang Cao and Lu Wang. 2021. Controllable Open-ended Question Generation with A New Question Type Ontology. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 6424–6439. https://doi.org/10.18653/v1/2021.acl-long.502

[5] Tuhin Chakrabarty, Justin Lewis, and Smaranda Muresan. 2022. CONSISTENT: Open-Ended Question Generation From News Articles. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. 6954–6968. https://aclanthology.org/2022.findings-emnlp.517

[6] Yan Chen, Shujian Liu, Zheng Liu, Weiyi Sun, Linas Baltrunas, and Benjamin Schroeder. 2022. WANDS: Dataset for Product Search Relevance Assessment. In *Proceedings of the 44th European Conference on Information Retrieval*. 128–141.

[7] Nurendra Choudhary, Nikhil Rao, Sumeet Katariya, Karthik Subbian, and Chandan K. Reddy. 2022. ANTHEM: Attentive hyperbolic entity model for product search. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*. 161–171.

[8] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311* (2022).

[9] Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755* (2022).

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[11] Xinya Du and Claire Cardie. 2018. Harvesting Paragraph-level Question-Answer Pairs from Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1907–1917. https://doi.org/10.18653/v1/P18-1177

[12] Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to Ask: Neural Question Generation for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1342–1352. https://doi.org/10.18653/v1/P17-1123

[13] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 8342–8360. https://doi.org/10.18653/v1/2020.acl-main.740

[14] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards Unsupervised Dense Information Retrieval with Contrastive Learning. *arXiv preprint arXiv:2112.09118* (2021).

[15] Wei-Jen Ko, Te-yuan Chen, Yiyan Huang, Greg Durrett, and Junyi Jessy Li. 2020. Inquisitive Question Generation for High Level Text Comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6544–6555. https://doi.org/10.18653/v1/2020.emnlp-main.530

[16] Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data Augmentation using Pre-trained Transformer Models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*. 18–26. https://aclanthology.org/2020.lifelongnlp-1.3

[17] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

[18] Luis Enrico Lopez, Diane Kathryn Cruz, Jan Christian Blaise Cruz, and Charibeth Cheng. 2020. Transformer-based end-to-end question generation. *arXiv preprint arXiv:2005.01107* 4 (2020).

[19] Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2021. Zero-shot Neural Passage Retrieval via Domain-targeted Synthetic Question Generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. 1075–1088. https://doi.org/10.18653/v1/2021.eacl-main.92

[20] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 (CEUR Workshop Proceedings, Vol. 1773)*. https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf

[21] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, Dublin, Ireland, 1864–1874. https://doi.org/10.18653/v1/2022.findings-acl.146

[22] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2021. Large Dual Encoders Are Generalizable Retrievers. arXiv:2112.07899 [cs.IR]

[23] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375* (2019).

[24] 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.

[25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html

[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* (2022). arXiv:2206.06588

[27] Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating Factoid Questions With Recurrent Neural Networks: The 30M Factoid Question-Answer Corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 588–598. https://doi.org/10.18653/v1/P16-1056

[28] Noah A. Smith and Michael Heilman. 2011. *Automatic factual question generation from text*. Technical Report CMU-LTI-11-004. Carnegie Mellon University, Language Technologies Institute.

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

[30] Merve Ünlü Menevşe, Yusufcan Manav, Ebru Arisoy, and Arzucan Özgür. 2022. A Framework for Automatic Generation of Spoken Question-Answering Data. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. 4659–4666. https://aclanthology.org/2022.findings-emnlp.342

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[32] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652* (2021).

[33] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 483–498. https://doi.org/10.18653/v1/2021.naacl-main.41

[34] Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. Generative data augmentation for commonsense reasoning. *arXiv preprint arXiv:2004.11546* (2020).

[35] Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and M. Zhou. 2017. Neural Question Generation from Text: A Preliminary Study. In *Natural Language Processing and Chinese Computing*.

[36] Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2022. Rankt5: Fine-tuning T5 for text ranking with ranking losses. *arXiv preprint arXiv:2210.10634* (2022).

Dataset	Query	Label	Product Data
ESCI	calculator texas instruments	E	Title: Texas Instruments TI-84 Plus CE Color Graphing Calculator, Black 7.5 Inch Product Bullet Point: High-resolution, full-color backlit display Rechargeable battery ...
MS-MARCO	what is the role of mast cells in inflammation	0	Document: Cells of the Immune System Cell types with critical roles in adaptive immunity are antigen-presenting cells ...
WANDS	salon chair	Exact	Title: 21.7 " w waiting room chair with ... Description: this is a salon chair , barber chair for a hairstylist ...
Home Depot	jeldwen 24 inch bifold doors	2.33	Title: JELD-WEN Smooth 2-Panel Arch Top Hollow Core Molded Interior Closet ... Description: The 2-Panel Arch Top Interior door from JELD-WEN has a classic design ...

Table 7: Examples from the ESCI, MS-MARCO, WANDS and HomeDepot datasets.

Dataset	Label	Train	Dev	Test
ESCI	E	1,093,105	13,948	-
	S	351,975	4,357	-
	C	45,662	577	-
	I	160,870	2,024	-
MS-MARCO	0	18,646,285	13,960	-
	1	18,646,285	6,492	-
WANDS	Exact	-	-	25,614
	Irrelevant	-	-	61,201
	Partial	-	-	146,633
Home Depot	1st Quartile	-	-	5,095
	2nd Quartile	-	-	6,773
	3rd Quartile	-	-	27,704
	4th Quartile	-	-	34,217

Table 8: Labels Distribution for the ESCI, MS-MARCO, WANDS and HomeDepot datasets. We used ESCI and MS-MARCO only for training (so no test set stats shown) and we used WANDS and Home Depot only for testing as a downstream task (so only test set stats are shown).