Ask Me What You Need: Product Retrieval using Knowledge from GPT-3

Su Young Kim[1], Hyeonjin Park[1], Kyuyong Shin[1,2], Kyung-Min Kim[1,2]
{suyoung.kim1,hyeonjin.park.ml,ky.shin,kyungmin.kim.ml}@navercorp.com
[1] NAVER CLOVA, [1,2] NAVER AI Lab
South Korea

ABSTRACT

As online merchandise become more common, many studies focus on embedding-based methods where queries and products are represented in the semantic space. These methods alleviate the problem of vocab mismatch between the language of queries and products. However, past studies usually dealt with queries that precisely describe the product, and there still exists the need to answer imprecise queries that may require common sense knowledge, *i.e.*, 'what should I get my mom for mother's day.' In this paper, we propose a GPT-3 based product retrieval system that leverages the knowledge-base (KB) of GPT-3 for question answering; users do not need to know the specific illustrative keywords for a product when querying. Our method tunes prompt tokens of GPT-3 to prompt knowledge and render answers that are mapped directly to products without further processing. Our method shows consistent performance improvement on two real-world and one public dataset, compared to the baseline methods. We provide an in-depth discussion on leveraging GPT-3 knowledge into a question answering based retrieval system.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; • **Computing methodologies** → *Knowledge representation and reasoning*.

KEYWORDS

product retrieval, pretrained language models

ACM Reference Format:
Su Young Kim[1], Hyeonjin Park[1], Kyuyong Shin[1,2], Kyung-Min Kim[1,2]. 2022. Ask Me What You Need: Product Retrieval using Knowledge from GPT-3. In *Proceedings of DLP-KDD (Conference acronym 'XX).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Product search engines have emerged as a key factor for online e-commerce platforms. They allow users to find the best set of products offered by an online merchandise that match the search query. Early work investigating product retrieval focused on improving a lexical matching engine that quantifies the similarity between languages of query and product [4, 9]. However, such methods often suffered from an upper bound in the semantic information they can learn. More recent studies introduced neural product search approaches, where latent representations for queries and products are modelled using a deep neural network [1, 14]. State-of-the-art product retrieval models utilize pre-trained large language models to enhance text embedding and further bridge the vocabulary gap. Wu et al. [12] modelled query and product representations using fine-tuned BERT, and yield an end-to-end learning framework for product search. Lu et al. [7] used a BERT-based query encoder and a graph attention based retrieval network for e-commerce search.

In this paper, we deal with a particular challenge of modelling *intent* queries to retrieve relevant products. These intent queries are different from keyword queries that explicitly describe the desired products. Users may not know the relevant item but only know the search intent, *i.e.*, 'What should I get my son for his birthday?' Inspired by the properties of recent large language models, we expect to obtain such extra common sense knowledge from GPT-3.

GPT-3 has shown great success in Natural Language Processing (NLP) domains such as question answering (QA) [2, 5, 8] and knowledge retrieval [8, 13]. Instead of requiring an explicit knowledge base (KB), our product retrieval engine uses an *implicit* KB that is stored in GPT-3, along with p-tuning [6] to better prompt the stored knowledge. Our main contributions are summarized as follows.

(1) We present a GPT-3 based product retrieval system. To the best of our knowledge, this is the first use of GPT-3 in a product retrieval task.
(2) We conduct ablation studies on how GPT-3 size and tuning methods affect the performance of the product retrieval system.
(3) We show that GPT-3 based product retrieval system is more effective in solving the cold-start problem than other baselines.

2 METHODOLOGY

First, we define the notations used in this paper. Let $Q = \{q_1, \ldots, q_n\}$ be a set of n queries and $C = \{c_1, \ldots, c_m\}$ be a set of m categories. Each query q_i has a purchased product $p_j \in P$ and the product's corresponding categorical information $c_k \in C$. The triplet of q_i, p_j and c_k appear in the data logs, *i.e.*, a user had a search intent q_i and purchased p_j belonging in category c_k.

2.1 Retrieval model

Given a query $\hat{q}_i$, the goal of our retrieval model is to select the top-K relevant product categories $\hat{C}$, thus *effectively* reducing the search space for the subsequent ranking model. We choose products' categorical information as the target label for the retrieval stage because using categories as answers makes the template closer to natural language as a human would write it, and the performance of GPT-3 would improve.

Conference acronym 'XX, 2022, Washington, DC

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

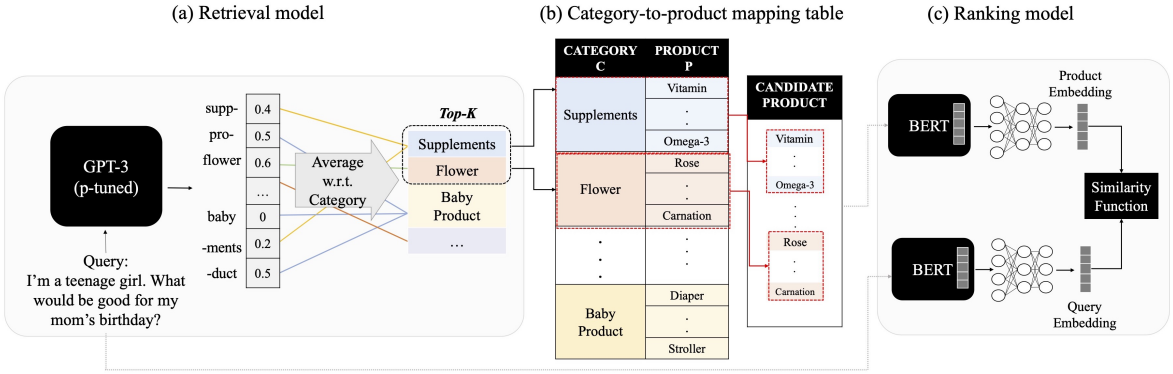

Figure 1: The overview of the proposed product retrieval system. (a) The retrieval model takes a query and selects top-K categories according to the category score. (b) The top-K categories are mapped to candidate products using category-to-product mapping table. (c) Lastly, the candidate products are ranked according to the ranking model.

Training method. To optimize GPT-3 for our downstream task, we use the p-tuning method [6]. We formulate $\tilde{q}_i$ as a concatenation, "[PROMPT$_{1:d}$] [$\hat{q}_i$] [MASK]", in which [PROMPT$_{1:d}$] are the trainable continuous prompt tokens, [$\hat{q}_i$] is the context, and [MASK] is the target. d is the hyperparameter determining the number of prompt tokens. In p-tuning method, only the embeddings for the trainable continuous prompt tokens are updated with the Cross-Entropy loss,

$$L = -\sum_{i=1}^{N} y_i^{\top} log \mathcal{M}(\tilde{q}_i), \tag{1}$$

where $\mathcal{M}$ refers to the GPT-3 model, N is the number of train data and y_i is the one-hot vector of the target category token in the vocabulary of a language model $\mathcal{M}$. In practice, fine-tuning could also be adopted, but many work observed that GPT-style models perform poorly to NLU tasks with fine-tuning [6]. Thus, we use p-tuning in utilizing GPT-3 as an implicit KB for desired knowledge, in other words, finding a relevant product's category for a given query. The comparison between performances of fine-tuning and p-tuning is discussed in Section 4.2.

Inference. To select the top-K relevant product categories $\hat{C}$, we obtain the category score s_i for each category. Let's say the category c_i is 'baby product' and its tokens are T = ['baby', 'pro-', '-duct']. s_i is formulated as,

$$s_i = \sum_{j=1}^{|T|} \alpha_j \mathcal{M}(t_j|\tilde{q}), \tag{2}$$

where t_j denotes j-th token in T. The α_j is the hyperparameter for the weight of each token probability and $\sum_j \alpha_j = 1$. s_i is calculated as the weighted average of the logit scores of tokens in $c_i \in C$, conditioned on the query.

In our experiment, we heuristically set α_1 to 0.8 and let the rest share the same weight of $\sum_{j=2}^{|T|} \alpha_j = 0.2$. We give the highest weight to the first token because it is the most important in decoding the answer from GPT-3. Finally, category set C is sorted by s_i and the top-K categories $\hat{C}$ are used in the ranking stage.

Table 1: Basic Statistics of Datasets.

Dataset	# of Pair			# of Categories	# of Items
	Train	Valid	Test		
Gift	44,173	5,522	5,522	1,357	41,589
Co-purchase	67,524	8,441	8,440	1,076	64,020
Google LCC	5603	476	476	5	6555

2.2 Ranking Model

We first use the category-to-product mapping table, Figure 1-(b), to prepare the candidate product set. The candidate products are then ranked using the ranking model, which can be any model that leverages embedding-based similarity methods.

In this paper, we use BERT [3] with multi-layer perceptron (MLP) layers as the simple embedding method to leverage flexibility in the architecture. Learning latent representations of queries and products with this embedding method and then calculating a similarity score is shown in Figure 1-(c). Specifically, given the query $\hat{q}$ and the candidate product $\hat{p}_i$, the similarity score is calculated as,

$$S(\hat{q}, \hat{p}_i) = f(E_{\hat{q}}; \theta) \cdot f(E_{\hat{p}_i}; w), \tag{3}$$

where $E_{\hat{q}}$ and $E_{\hat{p}_i}$ represent the BERT embeddings of $\hat{q}$ and $\hat{p}_i$, and f is MLP layers parameterized by θ and w.

We use Binary Cross-Entropy (BCE) loss between the predicted score $S(\hat{q}, \hat{p}_i)$ and ground truth label $\hat{y}$ which is defined as,

$$\hat{y} = \begin{cases} 1, & (\hat{q}, \hat{p}_j) \in D^{tr} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where D^{tr} is the set of query and product pairs (q_i, p_j) in the data log. We also use the weighted loss to handle the class imbalance problem between positive and negative samples. The weight update happens both in the BERT parameters and MLP parameters.

3 EXPERIMENTS

3.1 Datasets and Evaluation Metrics

In this section, we conduct experiments on our e-commerce datasets, where the purchase log is transformed into query-product pairs. These queries do not describe the products, but instead, contain search intents that require the system to exhibit natural language understanding (NLU) ability. We additionally test our method on a public dataset. Detailed statistics of the datasets are described in Table 1.

Gift dataset. We retrieved reviews that contain the word 'gift' from one year of shopping review log on our e-commerce platform, spanning from May 20, 2020, to May 25, 2021. We subsampled a total of 55, 217 review logs, then involved human resources to form natural language queries from the review logs, to produce query-product pairs. Since the log contains user information, we could compute Toppop (age or gender) for users asking the queries, as baselines.

Co-purchase dataset. We sampled 45, 234 purchase logs from our e-commerce platform, spanning from September 01, 2021, to September 5, 2021. For each purchase log, we randomly picked a product as an anchor and formed a query with its category information, *i.e.*, if the anchor product is a type of vitamin, the query is 'What can be co-purchased with vitamins?' The positive sample is a randomly picked product from the same purchase log. Additionally, to formulate it into a more complex NLU task, we asked 82B GPT-3 [5] for the intention of the co-purchase to include it in the query. [1]

Google LCC To verify the generality of our proposed method and allow others to reproduce our results, we additionally test on a public dataset shared by Google LCC. This dataset consists of 6, 555 English question-answer pairs, which are categorized as one of 'stackoverflow', 'culture', 'technology', 'science', or 'life arts' based on the nature of the question. In this paper, we viewed questions as queries, answers as products, and categories as product categories.

Evaluation metrics. We evaluate model performance on Gift, Co-purchase and Google LCC dataset in terms of HR@K, a simple yes/no metric that looks at whether any of the top-K recommended products include the ground truth product,

$$\text{HR@}K \text{ for a query} = \max_{i=1,..,K} \begin{cases} 1, & r \in \mathcal{T}_i \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where r is the ground truth product, and $\mathcal{T}_i$ is the set of recommended products up to i-th rank. We get HR@K across queries and compute the average.

3.2 Methods Compared

We compare against a conventional baseline (TopPop), a traditional web retrieval baseline (BM25 [11]), and an transformer-based baseline that is widely used for NLP modelling (BERT [3]). All these baselines are formed as a 2-stage retrieval system, where the retrieval model follows each baseline method but the same ranking model as ours is shared across all baselines. Note that the top 10 categories were retrieved for the gift and co-purchase dataset, whereas the top one category was retrieved for the Google LLC dataset.

[1] We exclude TopPop for Co-purchase since this dataset is collected without demographic information

Table 2: Results on the two product retrieval datasets.

Retrieval Models	Gift dataset		Co-purchase dataset	
	HR@300	HR@500	HR@300	HR@500
TopPop (age)	0.0541	0.0630	-	-
TopPop (gender)	0.0590	0.0670	-	-
BM25	0.0060	0.0063	0.0065	0.0075
BERT	0.1317	0.1492	0.1135	0.1609
Ours (13B)	**0.1514**	**0.1699**	**0.1494**	**0.2121**

Table 3: Results the Google LCC dataset.

Retrieval Models	Google LCC dataset	
	HR@300	HR@500
BM25	0.0945	0.1176
BERT	0.0840	0.1450
Ours (6.7B)	**0.3004**	**0.4685**

TopPop. Toppop baseline retrieves categories according to the category popularity. We test toppop on two levels, age and gender of the user asking the query.

BM25. Overall, BM25 remains a strong baseline for zero-shot text retrieval [11]. BM25 is a bag-of-words (BOW) information retrieval model that relies on an exact lexical match between a query and documents (categories).

BERT-based similarity search. The current effective approaches integrate BERT [3] as an embedding generation component in the retrieval model, with the input of a concatenated string of query and candidate texts. BERT and a simple nonlinear model are then trained with BCE loss where incorrect pairs get penalized.

4 RESULTS

4.1 Retrieval Performance

4.1.1 Product Retrieval. We have our main comparison table in Table 2. We see superior performance of our proposed method on both datasets, across all metrics. BM25 [11] retrieval model showed the worst performance, as the tasks are formed as QA-tasks, therefore the lexical matching method is significantly suboptimal. TopPop baselines also failed to retrieve relevant categories to the query. BERT [3]-based similarity search was the most comparable considering it is a transformer-based pre-trained language model, however, our proposed retrieval system showed superior performance of 15.0% and 31.6% compared to BERT. The higher performance of our method comes from the ability to carefully consider the scores of each token of the category.

4.1.2 Public Dataset. Our method on Google LLC dataset showed superior performance compared to BERT [3] and BM25 [11], proving that our method can be generalized to not only product retrieval

Table 4: Performance comparison of different tuning methods on Gift dataset.

Retrieval Models	HR@300	HR@500
Zero-shot (1.3B)	0.0076	0.0130
Fine-tuned (1.3B)	0.0092	0.0158
Ours (137M)	0.0570	0.0954
Ours (1.3B)	0.0628	0.0996
Ours (13B)	**0.1514**	**0.1699**

* GPT-3 (13B) is too large for fine-tuning.

Table 5: The cold-start performance on Gift dataset.

Retrieval Models	HR@300	HR@500
BM25	0.0002	0.0009
BERT	0.0141	0.0235
Ours (13B)	**0.0262**	**0.0410**

but also to general informational retrieval cases. The more conspicuous performance increase results from the nature of the dataset. The product retrieval dataset has many possible answers for one ground truth product, whereas in the Question-Answering scenario the right answer is apparent, thus the ranking model could perform better.

4.2 Effective Tuning Method for Knowledge Retrieval

Interestingly, fine-tuning GPT-3 shows only a slight performance improvement compared to the zero-shot approach (Table 4). We conjecture that fine-tuning the model parameters triggered catastrophic forgetting, which subsided the knowledge GPT-3 gained from pre-training. Liu et al. [6] empirically demonstrated that pre-trained language models with properly optimized p-tuning can capture far more knowledge than fine-tuning, and show that such is true across various model scales for NLU tasks. We observe the same trend in our KB-based product retrieval system. Table 4 shows that our proposed method with 137 million parameters significantly outperforms the other tuning methods, which are 1.3 billion zero-shot and 1.3 billion fine-tuned models.

4.3 Influence of Language Model Size

Recent studies have shown that training deep neural networks with large parameters leads to significant performance gains in a wide range of tasks [2, 10]. As such, we found that scaling up the size of GPT-3 empowers the ability to solve QA tasks. As presented in Table 4, 13 billion model surpasses other models by a very great margin. It is worth noting that the performance differs even more significantly when the model size varies from 1.3 billion to 13 billion, than from 137 million to 1.3 billion. This implies that increasing the model size can dramatically increase the knowledge probing ability of the language model.

4.4 Performance on Cold-Start Problem

We evaluate the cold-start performance against two other baselines, BM25 and BERT-based search. To properly compare the performance, we take the same train dataset as before but prepare a separate test dataset consisting of search intent and product pairs not seen during training. Our method achieves an increase of 85.8% in HR@300 metric compared to BERT-based search (Table 5). We conclude that the knowledge already encoded in GPT-3 helps retrieve the right products, although a particular query or product's semantic information has not been learned during training. Our product retrieval system overcomes the vocabulary upper bound problem as well as allowing flexibility in query formation.

5 CONCLUSION

We propose a GPT-3 based product retrieval system that can leverage the implicit KB stored in GPT-3 to answer intent queries that may contain out-distribution vocabularies. Our method is non-trivial because it uses GPT-3 in product retrieval tasks while using p-tuning to guide the prompting of knowledge and retrieval of information. We test our method on two real-world and one public dataset, where we see superior performance compared to the competitive baseline, BERT, even in the cold-start setting. In the future, we plan to develop a personalized product retrieval system that integrates individual user behavior logs into our system.

REFERENCES

[1] Keping Bi, Qingyao Ai, and W. Bruce Croft. 2021. *Learning a Fine-Grained Review-Based Transformer Model for Personalized Product Search.*

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, et al. 2020. Language Models are Few-Shot Learners. In *NeurIPS*.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

[4] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in Facebook Search. *SIGKDD* (2020).

[5] Boseop Kim, HyoungSeok Kim, Sang-Woo Lee, et al. 2021. What Changes Can Large-scale Language Models Bring? Intensive Study on HyperCLOVA: Billions-scale Korean Generative Pretrained Transformers. In *EMNLP*.

[6] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. *arXiv* (2021).

[7] Hanqing Lu, Youna Hu, Tong Zhao, Tony Wu, Yiwei Song, and Bing Yin. 2021. Graph-based Multilingual Product Retrieval in E-commerce Search.

[8] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. WebGPT: Browser-assisted question-answering with human feedback. *arXiv* (2021).

[9] Priyanka Nigam, Yiwei Song, Vijai Mohan, Vihan Lakshman, Weitian (Allen) Ding, Ankit Shingavi, Choon Hui Teo, Hao Gu, and Bing Yin. 2019. Semantic Product Search. Association for Computing Machinery. https://doi.org/10.1145/3292500.3330759

[10] Kyuyong Shin, Hanock Kwak, Su Young Kim, Max Nihlen Ramstrom, Jisu Jeong, Jung-Woo Ha, and Kyung-Min Kim. 2021. Scaling Law for Recommendation Models: Towards General-purpose User Representations. *arXiv* (2021).

[11] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models.

[12] Xuyang Wu, Alessandro Magnani, Suthee Chaidaroon, Ajit Puthenputhussery, Ciya Liao, and Yi Fang. 2022. A Multi-Task Learning Framework for Product Ranking with BERT. In *WWW*.

[13] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. 2021. An empirical study of gpt-3 for few-shot knowledge-based vqa. *arXiv* (2021).

[14] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. *Jointly Optimizing Query Encoder and Product Quantization to Improve Retrieval Performance*.