CFDA & CLIP Labs at TREC'23 Product Search Track

Jia-Huei Ju[1], Chung-Kang Lo[1*], Yao-Cheng Lu[1*], Kuan-Lin Lai[1*],
Cheng-Wei Huang[1*], Wei-Hsin Chiu[1*], Ming-Feng Tsai[2], Chuan-Ju Wang[1]
[1]Research Center for Information Technology Innovation, Academia Sinica
[1]Department of Computer Science, National Chengchi University

ABSTRACT

In this notebook, we present our pipeline approach for the product search track. We utilize both product textual data and images to enhance retrieval diversity. Our experiments also demonstrate the good generalization capability of a few off-the-shelf retrieval models. Additionally, we adopt retrieval fusion and consider it an efficient method to integrate text and images for product search.

KEYWORDS

zero-shot retrieval, multi-modal retrieval, retrieval-fusion, multi-stage pipeline

ACM Reference Format:
Jia-Huei Ju[1], Chung-Kang Lo[1*], Yao-Cheng Lu[1*], Kuan-Lin Lai[1*],, Cheng-Wei Huang[1*], Wei-Hsin Chiu[1*], Ming-Feng Tsai[2], Chuan-Ju Wang[1] . 2018. CFDA & CLIP Labs at TREC'23 Product Search Track. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The first year of the product search track benchmarks the product search problem for IR research. The organizers introduce the (textual) product ranking and multi-modal product ranking tasks. For these tasks, we experiment with various well-developed retrieval methods using either sparse or dense representations. We also explore different ways to utilize product images in product search. Furthermore, we combine these diverse methods via *retrieval fusion* and adapt it to the common two-stage pipeline (i.e., retrieval-then-rerank) as a cascaded approach.

In our empirical evaluation, we believe the off-the-shelf first-stage retrieval can perform decently in a zero-shot manner. Our experiments also show that sparse retrieval (SR) may outperform dense retrieval (DR) somehow, aligning with observations in recent out-of-domain IR research [10]. We also see retrieval fusion as a simple yet effective way to integrate diverse retrieval methods, effectively increasing recall. Finally, we perform the second-stage re-ranking to obtain the results for our submitted runs.

[*]These authors contributed equally to this work.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

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

2 OUR CASCADED APPROACH

2.1 A Two-Stage Pipeline with Retrieval Fusion

In this track, we adapted the common multi-stage pipeline to a product search scenario. The pipeline mainly comprises a retrieval stage and a re-ranking stage. we denoted them as S_{RT} and S_{RR}, and formulate the pipeline as:

$$\bar{\mathcal{P}}_i \leftarrow S_{\text{RT}}(q, p \in \mathcal{P} | m_i); \tag{1}$$

$$\mathcal{R} \leftarrow S_{\text{RR}}(q, p \in \bar{\mathcal{P}}), \text{ where } \bar{\mathcal{P}} \leftarrow f(\bar{\mathcal{P}}_1, \bar{\mathcal{P}}_2, \dots | w). \tag{2}$$

Here, q represents a user query, and $\mathcal{P}$ represents the full collection of products. While $\bar{\mathcal{P}}_i$ is a small subset of product candidates retrieved using different methods m_i. Moreover, before re-ranking, we fuse multiple product candidate sets $\bar{\mathcal{P}}_i$ into the fused one $\bar{\mathcal{P}}$ by a fusion f with hyperparameters w. Finally, we perform re-ranking with the fused set and obtain the product ranklist R.

2.2 The First-Stage Retrieval Methods

In the following sections, we will report several retrieval methods adopted in the stage S_{RT}: improved sparse retrieval (Section 2.2.1), learned sparse retrieval (Section 2.2.2), text dense retrieval (Section 2.2.3), and image dense retrieval (Section 2.2.4).

2.2.1 Improved Sparse Retrieval. We use the BM25 [9] and the inverted indices (Pyserini toolkit[1] [6]) as the vanilla sparse retrieval. We retrieve top-1000 relevant products based on the estimated relevance scores of a query q and the product title (as the passage) in the collection $\mathcal{P}$. To further improve the vanilla sparse retrieval, we follow the doc2query approach [7] to enrich the collection with predicted queries. Here, we fine-tune a text generator[2] on released training query-product pairs (`train.qrels`.). Consequently, we concatenate each product title with 10 predicted queries from the generator. Then, we can retrieve using the indices re-built with the expanded collection.

2.2.2 Learned Sparse Retrieval. Recent expansion-based learned sparse retrieval, SPLADE [3], have shown the strong capability of out-of-domain generalization [1, 2]. Thus, in addition to explicitly expansion (Section 2.2.1), we use the newly proposed SPLADE++ model [2] and its checkpoint[3] to retrieve top-1000 relevant products with the zero-shot setting. It is worth noting that we use product full content as the passage p.

2.2.3 Text Dense Retrieval. Dense retrieval is based on bi-encoders architecture, offering an efficient maximum inner-product search

[1]https://github.com/castorini/pyserini/tree/master
[2]https://huggingface.co/DylanJHJ/t5-base-product2query
[3]https://huggingface.co/naver/splade-cocondenser-ensembledistil

Table 1: Evaluation results of sparse retrieval baselines with different product content (as passage).

Baseline conditions		Recall@1K	nDCG@1K
(t)	title only ($k1 = 0.5$, $b = 0.3$)	0.6066	**0.3858**
(s)	simplified ($k1 = 0.9$, $b = 0.4$)	0.5940	0.3506
(f)	full ($k1 = 4.68$, $b = 0.87$)	**0.6286**	0.3753

over embeddings. We use Contriever [4] and its checkpoint [4] fine-tuned on MSMARCO (henceforth, Contriever-MS), to encode query and passage title in a zero-shot manner.

2.2.4 Image Dense Retrieval. We also study the zero-shot image retrieval performance with pre-trained CLIP text and image encoders[5] [8]. They are fine-tuned on text-image contrastive objectives, analogous to dense text retrieval in Section 2.2.3. After building the dense indices of CLIP image embeddings, we can apply the same vector search to retrieve the top 1000 relevant products (images).

To conclude our retrieval stage, we obtain multiple sets of top-k relevant products retrieved by the aforementioned retrieval methods. In particular, except for the improved sparse retrieval (Section 2.2.1), we use the other methods without in-domain fine-tuning. Here, we leave the in-domain fine-tuning (i.e., with relevant query-product pairs) as our future works towards better domain-adaptive retrieval.

2.3 The Second-Stage Re-ranking with Retrieval Fusion

In this section, as formulated in Eq. (2), we adopt the retrieval fusion in advance of computational intensive passage re-ranking. First, we fuse multiple first-stage retrieval results into a fused ranklist $\bar{\mathcal{P}}$ with the top 1000 product candidates. Specifically, we adopt the off-the-shelf weighted sum fusion implemented in ranx library[6]. We also search the hyperparameter W in terms of nDCG@100 with human-judged labels (`Dev Qrels`); other fusion details can be found in the experiments in Section 3.1.2. As for the re-ranking, we adopt the cross-encoder architecture and its checkpoint[7] fine-tuned by Sentence-Transformer. We input the query and product title to calculate the relevance score and finally obtain the re-ranked ranklist of products.

3 EXPERIMENTS

3.1 Settings

3.1.1 Evaluation Data. Instead of using the released development data (`Dev Qrels`), we construct a filtered version of it by discarding the empty and code-like query (e.g., "B07SDGB8XG"), resulting in 8941 queries and 169731 judgments.

3.1.2 Baseline Methods. Our reproduced baselines are vanilla sparse retrieval with BM25 search. We have tried different types of product textual content: (t) title only, (s) simplified (i.e., title+description), and (f) full content. As shown in Table 1, we see "title only" can perform better or on par with the other two longer conditions.

[4]https://huggingface.co/facebook/contriever-msmarco
[5]CLIP-large Laion: https://huggingface.co/laion/CLIP-ViT-L-14-laion2B-s32B-b82K
[6]https://github.com/AmenRa/ranx/tree/master/ranx
[7]https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2

Table 2: The evaluation results of different first-stage retrieval methods.

Cond.	Retrieval	Recall		
		@10	@100	@1K
SR	Baseline BM25	0.1613	0.3818	0.6066
Imp.-SR	T5-base expansion[†]	**0.1712**	**0.4127**	**0.6481**
Imp.-SR	BLIP2-opt caption	0.1679	0.3998	0.6301
LSR	SPLADE++	0.1720	0.4153	0.6526
LSR	SPLADE++ (full)	**0.1819**	**0.4392**	**0.6834**
Text-DR	Contriever-MS	(0.1562)	0.4057	0.6572
Img-DR	CLIP-large	(0.1148)	(0.3275)	(0.5547)

3.2 Evaluation Results

We first investigate different first-stage retrieval methods in terms of recall. Then, we examine the effectiveness of fusing different first-stage retrieval for final delivered results. However, due to time constraints, we did not conduct the re-ranking on development data, only on the submitted results (i.e., testing set).

3.2.1 Recall of the First-stage Retrieval Methods. Table 2 shows the experimental first-stage retrieval methods in our pipeline. In addition to the vanilla sparse retrieval (SR), other methods mentioned in Section 2.2 are also reported, including improved sparse retrieval (Imp.-SR), learned sparse retrieval (LSR), text dense retrieval (Text-DR), and image dense retrieval (Img-DR).

First, in this table, we observe that LSR (SPLADE++) outperforms all the others; it even improved the baseline by 12% recall@1K while using "full" contents as passage, unlike vanilla BM25 search (See Section 3.1.2 and the Table 1 for details). Second, we found that BM25 search with document expansion (i.e., Imp.-SR) is effective on product search as well. Moreover, we use BLIP2 [5], as a zero-shot LLM-enhanced caption generator[8] to expand the product title with captions, offering a little improvement as well. Following these results, we see the expansion as one of the starting points to consider product images in product search. Last, we observe that dense retrieval (DR) methods may not be as robust as SR methods (e.g., Imp.-SR, LSR). In the last two rows, we found Img-DR performs inferior to the baseline a lot (0.5547 vs. 0.6066); Text-DR also performs under our expectation with unstable improvement. Although we hypothesize that the judgments may have some biases towards DR methods.[9]

3.2.2 Effectiveness of Retrieval Fusion. To narrow down our consideration, we select both LSR and Text-DR as baseline settings as they are the best ones among sparse and dense categories, reported in the upper part of Table 3. In the lower part of Table 3, the first row (i.e., None) is the fusion of LSR and Text-DR. This result

[8]https://huggingface.co/Salesforce/blip2-opt-2.7b
[9]The original relevance labels are from sparse retrieval.

Table 3: The full-ranking evaluation results with retrieval fusion. The superscripts indicate the run names of our submissions.

		nDCG			Recall
		@10	@100	@1K	@1K
$\bar{\mathcal{P}}_1$	SPLADE++	0.2921	0.3642	0.4367	0.6834
$\bar{\mathcal{P}}_2$	Contriever-MS	0.2500	0.3243	0.3992	0.6572
Retrieval Fusion: $\bar{\mathcal{P}} \leftarrow f(\bar{\mathcal{P}}_1, \bar{\mathcal{P}}_2, \bar{\mathcal{P}}_3)$					
$\bar{\mathcal{P}}_3$	None	0.3057	0.3817	0.4530	0.7004
	Baseline BM25$^{\text{ER.A}}$	0.3124	0.3888	0.4602	0.7094
	T5-base-expansion$^{\text{ER.B}}$	0.3183	0.3958	0.4654	0.7104
	BLIP2-opt-caption$^{\text{MR.A}}$	0.3165	0.3933	0.4640	0.7111
	CLIP-large$^{\text{MR.B}}$	**0.3240**	**0.4122**	**0.4835**	**0.7443**

proves fusing retrieval results can work effectively, with higher recall and increased nDCG (compared to the upper part). With the simple weighted sum, we found that all of the retrieval methods we experimented with were beneficial. However, to our surprise, the image dense retrieval via CLIP-large can boost the performance to a different level. We believe different retrieval methods provide diverse views, which help to aggregate into more effective results.

4 CONCLUSION

Our experiments indicate the decent generalization capabilities of the aforementioned retrieval methods. Additionally, we have concluded that sparse retrieval performs more robustly than dense methods. We hypothesize that the product search is distant from ad-hoc passage retrieval scenarios, which have longer and more contextualized query. Furthermore, our findings reveal that learned sparse retrieval, such as SPLADE, holds strong potential in product search scenarios, especially with shorter queries and noisy collections of data. In terms of multi-modal ranking, we explored retrieval fusion and validated its effectiveness as a baseline using product texts and images.

ACKNOWLEDGMENTS

We thank Jheng-Hong (Matt) Yang for valuable discussions and helpful experimental sharing.

REFERENCES

[1] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval. arXiv:2109.10086 [cs.IR]

[2] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From Distillation to Hard Negative Sampling: Making Sparse Neural IR Models More Effective. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 2353–2359. https://doi.org/10.1145/3477495.3531857

[3] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 2288–2292. https://doi.org/10.1145/3404835.3463098

[4] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised Dense Information Retrieval with Contrastive Learning. https://doi.org/10.48550/ARXIV.2112.09118

[5] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597* (2023).

[6] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 2356–2362. https://doi.org/10.1145/3404835.3463238

[7] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. arXiv:1904.08375 [cs.IR]

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proc. of PMLR*, Vol. 139. 8748–8763. https://proceedings.mlr.press/v139/radford21a.html

[9] Robertson, S.E., S. Walker, S. Jones, M.M. Beaulieu, and M. Gatford. 1994. Okapi at TREC-3. In *TREC-3*. p. 109–126.

[10] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. arXiv:2104.08663 [cs.IR]

A TRAINING PRODUCT2QUERY

TBD.