JBNU at TREC 2024 Product Search Track

Gi-taek An, Seong-Hyuk Yim, Jun-Yong Park, Woo-Seok Choi, Kyung-Soon Lee*

Department of Computer Science & Artificial Intelligence,
CAIIT, Jeonbuk National University, Republic of Korea
{gt, castle_h0326, pjy010608, ccwwsss, selfsolee}@jbnu.ac.kr
**Corresponding author*

Abstract

This paper describes the participation of the jbnu team in the TREC 2024 Product Search Track. This study addresses two key challenges in product search related to sparse and dense retrieval models. For sparse retrieval models, we propose modifying the activation function to GELU to filter out products that, despite being retrieved due to token expansion, are irrelevant for recommendation based on the scoring mechanism. For dense retrieval models, product search document indexing data was generated using the generative model T5 to address input token limitations. Experimental results demonstrate that both proposed methods yield performance improvements over baseline models.

1. Introduction

The TREC 2024 Product Search Track[1] aims to identify relevant products within an Amazon dataset for given queries, with the goal of optimizing retrieval methods and improving evaluation metrics. Our team submitted 12 run files employing both the Learned Sparse Retrieval and Dense Retrieval approaches.

In the context of Learned Sparse Retrieval, SPLADE++[2] utilizes BERT's MLM(Masked Language Model) task to expand tokens and extend input sentences for search. However, due to the ReLU activation function in its scoring mechanism, the model may fail to adequately capture the influence of certain key terms within similar phrases. For example, in the query *'my hero academia kids shirt*,' while the term *'shirt*' is crucial, products labeled as '*Mens'* or *'Girls Boys'* should be ranked lower. However, the model fails to properly account for this distinction." To address this issue, we modified the model's activation function to GELU[3], allowing for the incorporation of negative weights. Compared to ReLU, GELU facilitates a smoother transition in weighting functions, mitigating abrupt cutoff effects and improving ranking adjustments.

To address input token limitations in dense retrieval models, we summarized and indexed product information using the T5[4] model. To evaluate the impact on product search performance, we used the TAS-B[5] and ColBERTv2[6] models.

2. Submitted Runs

In our experiments, we evaluated each model individually and conducted comparative analyses of the combined results from four models using the Ranx[7] library. For the Product Ranking Track, we submitted

12 runs, consisting of five single models and seven fusion models, with the latter leveraging Ranx. The combination weights were determined based on the development data.

Single Model:

- **jbnu01**: Modification of the SPLADE++ model's activation function to GELU (without training)
- **jbnu02**: SPLADE++ model is trained on product search data, with product title and T5-generated summarized data indexed
- **jbnu03**: TAS-B zeroshot model, with product title and T5-generated summarized data indexed
- **jbnu04**: ColBERTv2 zeroshot model, with product title and T5-generated summarized data indexed
- **jbnu09**: jbnu01 model, using product titles and summaries for indexing

Fusion Model:

- **jbnu05**: Fusion of jbnu01 and jbnu03
- **jbnu06**: Fusion of jbnu01 and jbnu04
- **jbnu07**: Fusion of jbnu02 and jbnu03
- **jbnu08**: Fusion of jbnu02 and jbnu04
- **jbnu10**: Fusion of BM25 model, after data preprocessing, with jbnu04 model
- **jbnu11**: Fusion of jbnu09 and jbnu03
- **jbnu12**: Fusion of jbnu09 and jbnu04

3. Experimental Results

The experimental results were evaluated based on the metrics provided by TREC 2024, including the median values, NDCG, NDCG@100, P@100, and MAP. The performance of the jbnu01 model was compared against the median values, as illustrated in Figures 1, 2, and 3. Overall, the results for each query exceeded the average performance. The performance metrics for each model are summarized in Table 1. According to the experimental findings, jbnu01 and jbnu02 demonstrated strong performance as single models, while jbnu05 and jbnu10 achieved competitive results among the fusion models.

Figure 1. Comparison of TREC2024 min, median, and max values with jbnu02's NDCG@20 results

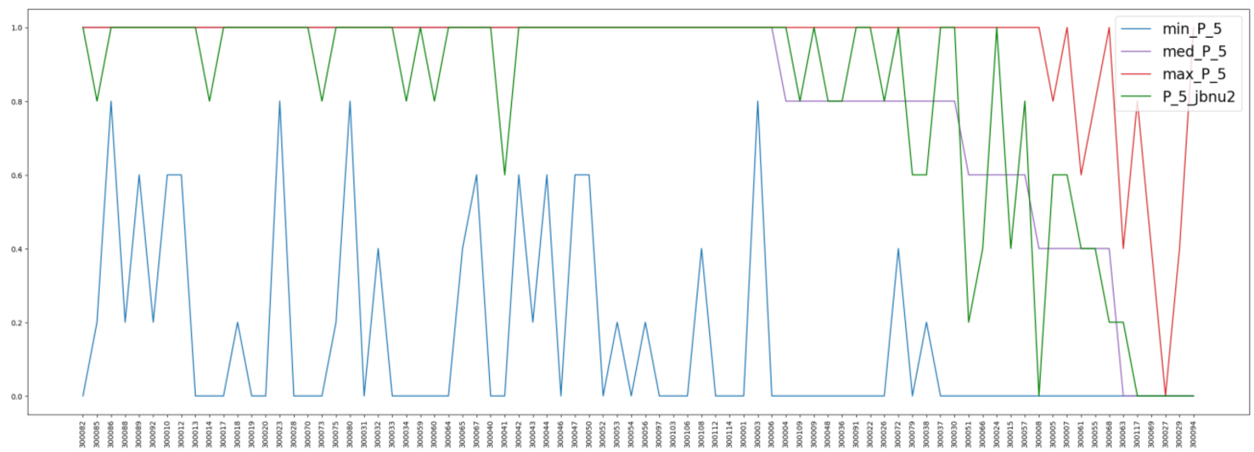

Figure 2. Comparison of TREC2024 min, median, and max values with jbnu02's P@5 results

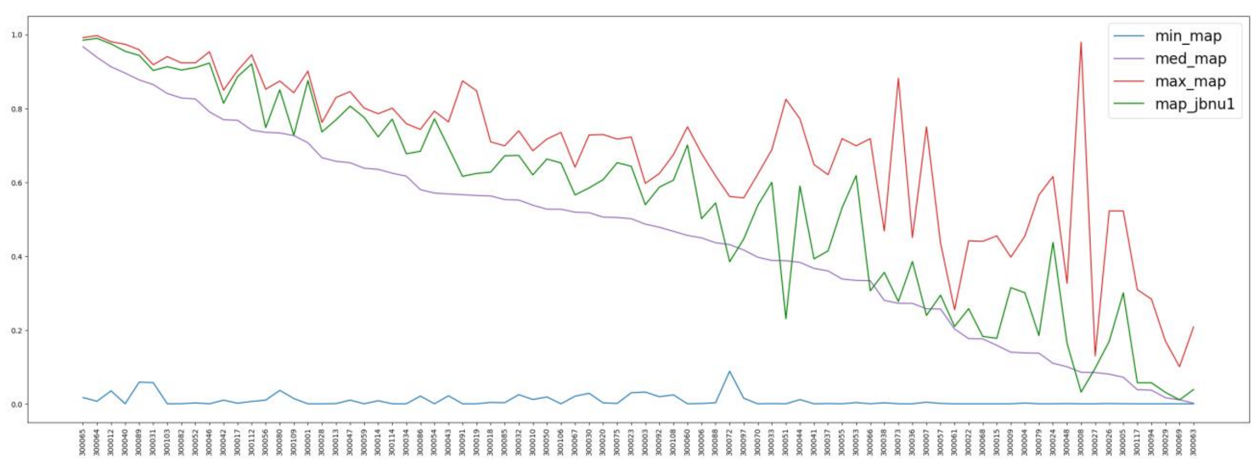

Figure 3. Comparison of TREC2024 min, median, and max values with jbnu01's MAP results

Run Name	NDCG	NDCG@20	NDCG@100	P@5	P@100	MAP
Single model						
jbnu01	**0.7376**	0.6704	**0.6854**	0.8025	**0.7014**	**0.5490**
jbnu02	0.7189	**0.6739**	0.6728	**0.8150**	0.6925	0.5322
jbnu03	0.6389	0.5974	0.5959	0.7750	0.6240	0.4394
jbnu04	0.6952	0.6607	0.6648	0.8050	0.6759	0.5141
jbnu09	0.6887	0.6526	0.6552	0.8000	0.6775	0.4952
Fusion model						
jbnu05	**0.7595**	0.6786	0.6904	0.7975	0.7034	**0.5805**
jbnu06	0.7513	0.6936	0.6960	0.8325	0.6992	0.5752
jbnu07	0.7324	0.6719	0.6763	0.8075	0.6953	0.5474
jbnu08	0.7265	0.6781	0.6835	0.8175	0.6956	0.5485
jbnu10	0.7330	**0.6968**	**0.7003**	**0.8250**	**0.7061**	0.5602
jbnu11	0.7183	0.6614	0.6672	0.8125	0.6819	0.5289
jbnu12	0.7233	0.6739	0.6787	0.8125	0.6866	0.5424

Table 1. Results of Product Ranking Task.

4. Conclusion

The experimental results demonstrated that the model incorporating negative weights outperformed the standard SPLADE++ model. Furthermore, in the Dense Retrieval approach, integrating product summary information generated by a generative model as indexing data led to performance improvements.

Acknowledgment

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the National Program for Excellence in SW) supervised by the IITP(Institute of Information & communications Technology Planing & Evaluation) in 2024(2022-0-01067)

References

[1] TREC2024 Product Search Track(2024). https://trec-product-search.github.io/

[2] Formal, T., Lassance, C., Piwowarski, B., & Clinchant, S. (2021). SPLADE v2: Sparse lexical and expansion model for information retrieval. arXiv preprint arXiv:2109.10086.

[3] Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415.

[4] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, *21*(140), pp.1-67.

[5] Hofstätter, S., Lin, S. C., Yang, J. H., Lin, J., & Hanbury, A. (2021, July). Efficiently teaching an effective dense retriever with balanced topic aware sampling. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 113-122.

[6] Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., & Zaharia, M. (2021). Colbertv2: Effective and efficient retrieval via lightweight late interaction. arXiv preprint arXiv:2112.01488.

[7] Bassani, E., & Romelli, L. (2022, October). ranx. fuse: A Python Library for Metasearch. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management, pp. 4808-4812.