JBNU at TREC 2025 Product Search and Recommendations Track

Seong-Hyuk Yim[†], Jae-Young Park[†], Woo-Seok Choi, Gi-Taek An[*], Kyung-Soon Lee[*]

Department of Computer Science & Artificial Intelligence,
Jeonbuk National University, Republic of Korea
{castle_h0326, jyp795133, ccwwsss, gt, selfsolee}@jbnu.ac.kr

[†] These authors contributed equally as co-first authors.
**Corresponding authors.*

Abstract

This paper presents the JBNU team's participation in the TREC 2025 Product Search and Recommendations Track. For the Search Task, we develop two complementary query reformulation strategies: an LLM-driven method that generates structured Lucene-style reformulations to reduce query ambiguity, and a multimodal approach that leverages a vision–language model (VLM) to extract additional semantic cues from web-sourced images. For the Recommendation Task, we adopt a two-stage architecture in which neural retrieval models (dense and learned sparse) generate candidate products, and relation classification—performed by either an LLM or a fine-tuned BERT model—reranks them as substitutes or complements, with final lists refined through weighted score aggregation. Experimental results show that both LLM-based query reformulation and classification-driven reranking consistently improve effectiveness across tasks. Overall, the study demonstrates that lightweight LLM components, when strategically integrated into retrieval and recommendation pipelines, provide a scalable and robust approach to product understanding in the TREC setting.

1. Introduction

We participated in both subtasks of the TREC 2025 Product Search and Recommendations Track:
(1) the Search Task[1] on query reformulation for product retrieval, and
(2) the Recommendation Task[2] on identifying substitute and complementary products.

Search Task. Conventional lexical retrieval methods such as BM25 struggle with abstract, goal-oriented, or contextually nuanced queries in which user intent is implicit rather than tied to explicit product attributes (e.g., "complete home office makeover"). To overcome this limitation, we introduce two complementary reformulation strategies that generate Lucene-compatible structured queries[3]:

- **LLM-based semantic reformulation.** This strategy infers latent user intent from the task description and constructs a structured Lucene query, optionally enriching it with synonyms or semantically related terms to better capture implicit product requirements.
- **Multimodal query reformulation.** This strategy augments the textual query with visual semantics: the query is issued to Google, the top three images are collected, and the Qwen2.5-VL[4] vision–language model is prompted to infer salient visual concepts. These inferred concepts are then incorporated as additional Lucene-compatible reformulation terms.

Recommendation Task. For identifying substitutes and complements, we adopt a two-stage retrieval–classification architecture.

- **Candidate retrieval.** A top-K candidate pool is retrieved using ColBERTv2[5] or SPLADE[6]. For ColBERTv2, relation signals are explicitly incorporated through triplet training samples—

(query, Substitute+, Substitute−) and (query, Complement+, Complement−)—where "+" denotes a true related item and "−" an unrelated one.

- **Classification and reranking.** Retrieved candidates are then reranked via relation classification using either AWQ-quantized Qwen3-14B prompting[7] with few-shot instructions or a fine-tuned ModernBERT classifier[8] predicting C/S/I (Complement/Substitute/Irrelevant). Qwen3-14B was selected for its stable instruction-following behavior; AWQ was applied to reduce inference cost. To improve label robustness, the classifier is run three times per candidate, and the majority-voted label is used for reranking. Final scores combine retrieval and classification outputs, with Complement/Substitute predictions upweighted by multiplying the retrieval score by 10.

2. Submitted Runs

2.1 Search Task

We submitted four runs, all evaluated using the official TREC tools with Pyserini[9] as the backend. Each run applies Lucene-operator query reformulation to provide directional control over retrieval. Runs **jbnu-s01–jbnu-s03** employ **Llama3-8B**[10] with different reformulation strategies, while **jbnu-s04** leverages **Qwen2.5-VL** for multimodal enhancement.

- **jbnu-s01**: Chain-of-Thought (CoT) reasoning for user-intent inference with synonym and semantically similar term augmentation.
- **jbnu-s02**: Type-aware reformulation based on structural characteristics of the query (no synonym augmentation).
- **jbnu-s03**: Synonym-driven term expansion aimed at minimally altering the query while improving recall.
- **jbnu-s04**: Multimodal expansion using three web-searched images and visual concept extraction via Qwen2.5-VL.

2.2 Recommendation Task

We submitted five runs, all based on a unified two-stage architecture that couples neural retrieval (dense or learned sparse) with classification-driven reranking.

- **jbnu-r01**: Relation-learned ColBERTv2 retrieval + AWQ-quantized Qwen3-14B classification/reranking.
- **jbnu-r02**: Pretrained ColBERTv2 retrieval + AWQ-quantized Qwen3-14B classification/reranking.
- **jbnu-r03**: Pretrained ColBERTv2 retrieval + fine-tuned ModernBERT classification/reranking.
- **jbnu-r04**: Relation-learned ColBERTv2 retrieval + fine-tuned ModernBERT classification/reranking.
- **jbnu-r05**: Pretrained SPLADE retrieval + fine-tuned ModernBERT classification/reranking.

3. Experimental Results

3.1 Search Task

Table 1 presents the Search Task results—Task Completion NDCG, Essential-product Recall@1000, and MAP—computed using the official relevance and gain settings. The minimum, median and maximum baselines represent the corresponding per-query statistics from the official TREC leaderboard. Under the TREC 2025 evaluation scheme, NDCG assigns a gain of **10** to *essential* products and **1** to *highly relevant* products, while *somewhat relevant* and *irrelevant* items receive no gain.

Run	NDCG	Recall@1000	MAP
min	0.0118	0.0087	0.0029
median	0.2832	0.2843	0.1527
max	0.6015	0.6147	0.4019
jbnu-s01	0.2861	0.2618	**0.1658**
jbnu-s02	**0.2911**	**0.2787**	0.1540
jbnu-s03	0.2697	0.2459	0.1545
jbnu-s04	0.2517	0.2478	0.1351

Table 1. Search Task Results

The prompt examples used in the evaluation for the jbnu-s02 run are shown in Figure 1.

```
Following is a product search query. Please expand this query to improve BM25 retrieval performance.
Expand the query by anticipating similar product names, variations, or related keywords that could be relevant.
Use **Lucene query syntax** to structure the expanded query properly.
Follow these Lucene rules:
- Use spaces for OR between terms - Use quotes " " for exact phrases
- Use AND/OR, +, - operators if needed - Use () to group terms
- Use field-specific terms like title:keyword if appropriate
- Use boosting with ^N (e.g., dress^3 blouse^1) to emphasize more important words
- Use wildcards ? and * for flexible matches
- Use fuzzy search with ~ (e.g., keyword~) for approximate matches
- Use proximity search with "term1 term2"~k if helpful
Your result must be only single query. Indicate the start of the query with expanded_query:
Identify if the query refers to:
- Product name - Identifier (ASIN, ISBN, etc.)
- Category/Attribute
Generate 3-5 different results and select one best fit to improve performance.
query {query}
```

Figure 1. Best-performing prompt used for the jbnu-s02 model.

Figure 2 visualizes how jbnu-s02 compares with the minimum, median, and maximum participant NDCG scores for each query. Queries with zero scores are excluded in accordance with the official track guidelines.

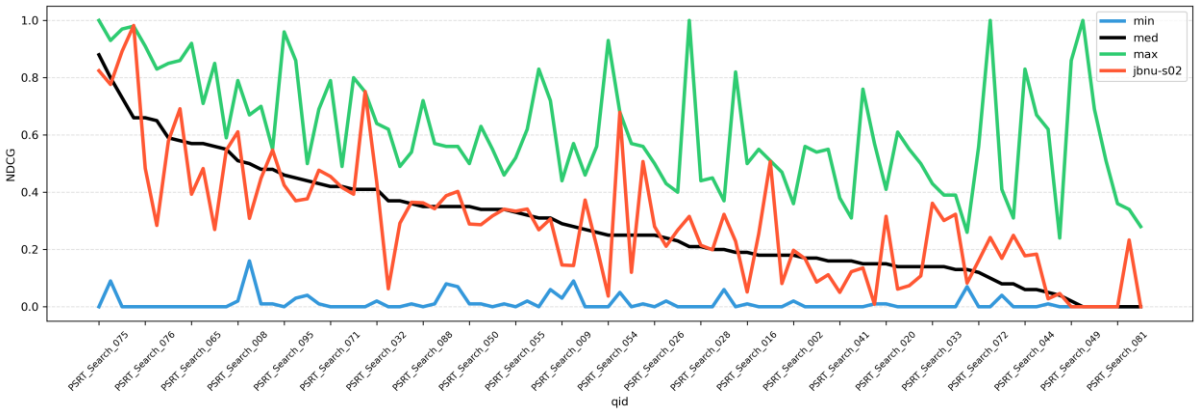

Figure 2. Per-query comparison of TREC 2025 NDCG scores (min/median/max) and jbnu-s02, with queries sorted by the TREC median.

Overall, our submitted runs outperform the participant median, particularly in Task Completion NDCG and MAP. Among the four runs, jbnu-s02 achieves the strongest NDCG by leveraging

structural-pattern detection and type-aware reformulation, whereas jbnu-s01 attains the highest MAP by more effectively modeling user intent and implicit product requirements.

3.2 Recommendation Task

Complement and substitute recommendations were evaluated using NDCG@10, Recall@10, P@10, and diversity, while the Relatedness evaluation uses average NDCG, PoolNDCG@10, and agreement as its primary metrics. Table 2 (complement), Table 3 (substitute), and Table 4 (relatedness) summarize the official results.

Run	NDCG@10	diversity	Recall@10	P@10
min	0.0189	0.7398	-	-
median	0.1000	1.1660	-	-
max	0.2781	1.4704	-	-
jbnu-r01	**0.1864**	0.9782	**0.2384**	**0.1936**
jbnu-r02	0.1370	0.9945	0.1392	0.1532
jbnu-r03	0.1487	1.2178	0.1525	0.1617
jbnu-r04	0.1275	1.2204	0.1479	0.1489
jbnu-r05	0.0237	**1.2538**	0.0200	0.0426

Table 2. Complement Recommendation Results

Run	NDCG@10	diversity	Recall@10	P@10
min	0.2168	0.5004	-	-
median	0.3660	0.7672	-	-
max	0.6138	1.1300	-	-
jbnu-r01	0.4771	0.6777	0.3596	0.4133
jbnu-r02	**0.4927**	0.6883	**0.4174**	**0.4413**
jbnu-r03	0.3778	**0.8979**	0.2920	0.3489
jbnu-r04	0.3566	0.8318	0.2762	0.3277
jbnu-r05	0.3321	0.8964	0.2608	0.2957

Table 3. Substitute Recommendation Results

Run	avg_ndcg	PoolNDCG@10	agreement
min	0.1262	0.3247	-0.4783
median	0.2519	0.4734	-0.2723
max	0.4194	0.6894	0.1532
jbnu-r01	**0.3318**	0.5933	**0.0264**
jbnu-r02	0.3149	**0.5992**	-0.1066
jbnu-r03	0.2632	0.4693	-0.2711
jbnu-r04	0.2421	0.4578	-0.2467
jbnu-r05	0.1779	0.3774	-0.3782

Table 4. Relatedness Results

Figures 3 and 4 present per-query NDCG@10 for complement and substitute recommendation, respectively. Figure 3 compares jbnu-r01, and Figure 4 compares jbnu-r02, against the minimum, median, and maximum participant scores reported by TREC. Queries with zero scores are excluded following the track guidelines.

For complement recommendation, jbnu-r01 produced the strongest overall performance, achieving the highest NDCG@10 (0.1864), Recall@10 (0.2384), and P@10 (0.1936), along with a diversity score of 0.9782.

For substitute recommendation, jbnu-r02 achieved the best results, with NDCG@10 (0.4927), Recall@10 (0.4174), and P@10 (0.4413), and a diversity score of 0.6883.

Overall, under the official evaluation protocol, runs reranked using **LLM-based relation classification** (jbnu-r01, jbnu-r02) generally outperformed those using the **ModernBERT-based classifier** (jbnu-r03–jbnu-r05), although the magnitude of improvement varied across relation types and metrics.

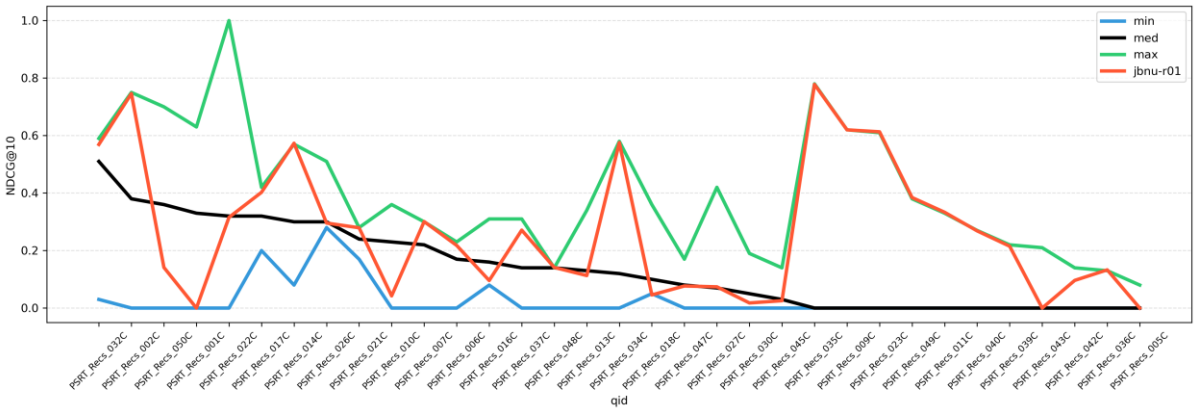

Figure 3. Complement recommendation: per-query comparison of TREC 2025 NDCG@10 (min/median/max) and jbnu-r01, with queries sorted by the TREC median.

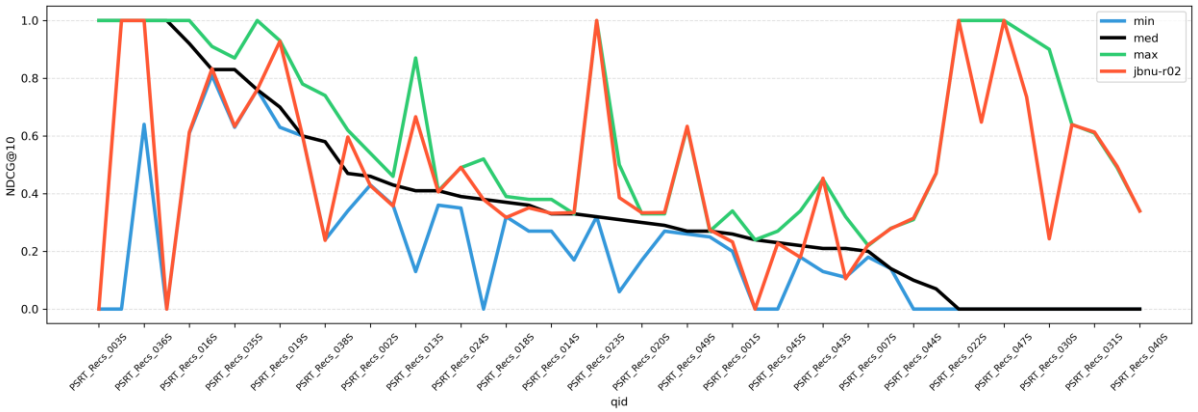

Figure 4. Substitute recommendation: per-query comparison of TREC 2025 NDCG@10 (min/median/max) and jbnu-r02, with queries sorted by the TREC median

Figure 5 presents the prompt template used for Qwen3-14B–based relation classification in the reranking stage.

```
You are an expert AI assistant specializing in e-commerce product relationship analysis.
Your task is to classify how each product document relates to the main query product.
## Classification Categories:
- **I (Irrelevant):** The document describes a product that has no meaningful connection to the query product.
  - It cannot replace or be used with the query product.
  - Lifestyle or unrelated items are always Irrelevant.
- **S (Substitute):** The document describes a product that serves the same or a very similar core function as the query product.
  - Substitutes are alternatives the customer could reasonably buy instead of the query product.
  - Substitutes may differ in brand, model, color, size, or slight features, but they fulfill the same main purpose.
  - Example: iPhone 14 vs Samsung Galaxy S22 → Substitute.
- **C (Complement):** The document describes a product that is typically used together with the query product.
  - Complements include accessories, add-ons, consumables, maintenance products, or peripherals.
  - A complement does not replace the query but increases its usefulness, protection, or convenience.
  - Example: Phone case, charger, cleaning kit, or game for a console → Complement.
## Special Rules:
- If a product could both substitute and complement, **prefer Substitute (S)** because the customer could directly replace the query with it.
- If a product is the exact same item but in a different style, version, or edition, classify as **S (Substitute)**.
- If a product is a bundle that contains the query product along with extras, classify as **C (Complement)**.
- Ignore abstract similarities (e.g., both are "for home use") unless the function overlaps directly.
## Examples:
### Example 1
[Query]: Nintendo 3DS - Flame Red
[Documents]:
1. Wine Country Gift Baskets Gourmet Feast
2. PlayStation Portable Core (PSP 1000)
3. Nintendo 3DS Compatible with 3DS / 3DS XL / 2DS AC Adapter
[Correct Output]:
{{"1": "I", "2": "S", "3": "C"}}
### Example 2
[Query]: Classic Leather Sofa
[Documents]:
1. KitchenAid Stand Mixer
2. Fabric Sectional Couch
3. Leather Cleaner and Conditioner Kit
[Correct Output]:
{{"1": "I", "2": "S", "3": "C"}}
---
## Your Task:
[Query]: {query_text}
[Documents]:
{sentences_text}
## Output Format Rules (STRICT):
- Output ONLY a valid JSON object.
- Keys = document numbers as strings.
- Values = one of 'I', 'S', or 'C'.
- Do NOT include explanations, markdown, or extra text.
[Your Output]:
```

Figure 5. Prompt template for Qwen3-14B relation classification (C/S/I) used in the reranking stage.

4. Conclusions

This paper presented the JBNU team's participation in the TREC 2025 Product Search and Recommendations Track, addressing intent-aware retrieval and relation-aware recommendation in large-scale product catalogs.

For the Search Task, our findings show that LLM-guided Lucene-style query reformulation is an effective strategy for reducing ambiguity in task-oriented queries. Additionally, VLM-based

multimodal enrichment demonstrates promise as a complementary signal for capturing fine-grained semantic cues beyond text.

For the Recommendation Task, a two-stage pipeline—neural retrieval followed by classification-based reranking—consistently achieved above-median performance. Notably, Qwen3-14B prompting-based classification surpassed fine-tuned ModernBERT rerankers, proving highly effective for both substitute and complement prediction.

Overall, the results indicate that pairing structured query reformulation for search with lightweight LLM-based reranking for recommendation provides a powerful and scalable framework for product understanding in the TREC 2025 setting.

Acknowledgment

This research was supported by the Ministry of Science and ICT (MSIT), Korea, under the National Program for Excellence in Software (SW), supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP), in 2025 (Project No. 2022-0-01067).

References

[1] TREC 2025 Product Search and Recommendations: Search Task. https://trec-product-search.github.io/search

[2] TREC 2025 Product Search and Recommendations: Product Recommendations Task. https://trec-product-search.github.io/recommendations

[3] Apache Lucene, *Query Parser Syntax*, 2010. [Online]. Available: https://lucene.apache.org/core/2_9_4/queryparsersyntax.html

[4] Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., ... & Lin, J. (2025). Qwen2.5-vl technical report. arXiv preprint arXiv:2502.13923.

[5] Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., & Zaharia, M. (2022, July). Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 3715-3734).

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

[7] Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., ... & Qiu, Z. (2025). Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

[8] Warner, B., Chaffin, A., Clavié, B., Weller, O., Hallström, O., Taghadouini, S., ... & Poli, I. (2025, July). Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 2526-2547).

[9] Pyserini (2025). An easy-to-use Python toolkit for reproducible information retrieval research. https://github.com/castorini/pyserini

[10] Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., ... & Vasic, P. (2024). The llama 3 herd of models. arXiv preprint arXiv:2407.21783.