Precision by Design: RM3 and Fusion in Product Search

Georgios Arampatzis*[1], Symeon Symeonidis[2], and Avi Arampatzis[1]

[1]Democritus University of Thrace, Dept. of Electrical and Computer Engr., Xanthi, Greece
[2]Democritus University of Thrace, Dept. of Production and Management Engr., Xanthi, Greece
[1]{geoaramp,avi}@ee.duth.gr, ssymeoni@pme.duth.gr

Abstract

In this work, we present the fully lexical and reproducible system developed by the **DUTH team** for the *TREC 2025 Product Search & Recommendation* track, aiming to improve performance on task-oriented e-commerce queries. Such queries (e.g., *home office makeover*, *birthday party essentials*) often perform poorly in purely lexical retrieval systems because they express high-level user intents rather than concrete product attributes.

Our system indexes approximately 1.08M products using Lucene/Pyserini, retrieves with BM25 (tuned to k_1=0.9, b=0.4), and bridges the intent–metadata gap through carefully calibrated RM3 pseudo-relevance feedback. For the interactive setting, we automatically generate four PRF-based query reformulations per topic and aggregate complementary signals using weighted Reciprocal Rank Fusion.

The system requires neither neural re-ranking nor external resources, runs efficiently on a single CPU node, and produces standard six-column TREC runs with strict de-duplication. Official evaluation results confirm that RM3 and fusion yield consistent improvements over the BM25 baseline across task completion nDCG, MAP, and Essential Recall@1000. These findings highlight that thoughtful lexical reformulation, classical PRF, and simple fusion strategies remain strong and efficient baselines for task-oriented product search.

Keywords: Product Search, Task-Oriented Queries, Lexical Retrieval, BM25, Pseudo-Relevance Feedback, RM3, Rank Fusion

1 Introduction

Task-oriented queries in e-commerce—such as *home office makeover* or *birthday party essentials*—often underperform in purely lexical systems. Users articulate high-level intents rather than concrete attributes, creating a vocabulary and knowledge gap between intent and catalog metadata. The TREC 2025 Product Search & Recommendation Track targets this gap through query reformulation: given a user task, the system must produce one (or a small set of) reformulated queries that make implicit requirements explicit, thereby enabling strong retrieval with standard term-matching engines (e.g., BM25) [1].

In this paper we present a fully lexical, scalable pipeline that requires no neural re-ranking. We index approximately 1.08M products (titles and descriptions) with Lucene/Pyserini and adopt BM25 as our backbone (k_1=0.9, b=0.4). To bridge the intent–metadata gap, we leverage pseudo-relevance feedback (PRF) via RM3 [2] to mine salient expansion terms from top-ranked documents while controlling drift through the original-query weight. Beyond a conventional configuration (fb_terms=80, fb_docs=8, oqw=0.3), we explore parameter sweeps up to $(120, 10, 0.3)$ to probe the recall–precision trade-off under task-oriented queries. We further instantiate an interactive PRF variant that automatically generates four reformulations per query and execute retrieval for each reformulation with BM25. Finally, we aggregate complementary ranking signals using Reciprocal Rank Fusion (RRF) [3], including a weighted variant that places additional emphasis on the strongest RM3 sweep.

Our design goals are efficiency, robustness, and reproducibility. Efficiency follows from the exclusive use of commodity lexical infrastructure; robustness is achieved through calibrated PRF and rank fusion; reproducibility is supported by a minimal set of transparent hyperparameters and standard tooling. All submissions are delivered as six-column TREC run files (`qid, reformulated_query, product_id, rank, score, runid`) with strict per-topic de-duplication. Official evaluation results, reported in Section 4, indicate consistent gains of RM3 and fusion over the BM25 baseline across task completion nDCG, AP, and Essential Recall@1000, while preserving efficient runtime characteristics. Overall, our findings underscore that careful lexical reformulation, classic PRF, and simple rank fusion remain competitive baselines for task-oriented

*Corresponding author.

product search at scale.

Growing interest in efficient and transparent lexical retrieval has motivated the development of pipelines that remain competitive without relying on heavy neural components. Building on this perspective, the present work adapts such principles to the domain of task-oriented e-commerce search. The remainder of this paper is organized as follows. Section 2 reviews prior research on lexical retrieval, pseudo-relevance feedback, and rank-fusion methods in product and ad-hoc search. Section 3 describes the dataset, indexing pipeline, retrieval configurations, and methodological framework used in all submitted runs for the TREC 2025 Product Search & Recommendation track. Section 4 reports the official evaluation results released by the organizers, accompanied by a detailed analysis of system behavior across RM3 configurations and fusion variants. Finally, Section 5 summarizes the main findings and outlines directions for future work, emphasizing opportunities to incorporate lightweight neural components, structured attributes, and learned fusion strategies within an otherwise transparent lexical backbone.

2 Related Work

Classical lexical retrieval remains a strong baseline for ad hoc search. BM25, a central instance of the probabilistic relevance framework, models term saturation and document length normalization and continues to anchor competitive systems across domains [1]. To mitigate the vocabulary gap between user intent and textual evidence, pseudo-relevance feedback (PRF) expands the query using statistics from top-ranked documents. Early PRF methods include Rocchio's vector-space formulation [4], while Relevance Models (RM) cast expansion as generative term sampling from an inferred relevance distribution [5]. In practice, the RM3 variant—which interpolates the expanded model with the original query to control drift—has proven to be a robust, high-performing approach for query expansion in large collections [6].

Beyond single-run retrieval, rank fusion aggregates evidence from multiple rankings to improve robustness and recall. Reciprocal Rank Fusion (RRF) is particularly attractive due to its simplicity and consistently strong performance across heterogeneous runs [7]. RRF's reciprocal discounting of ranks emphasizes consensus near the top of lists while remaining resilient to noise, making it a reliable post-retrieval ensembling strategy.

Query reformulation spans automatic, interactive, and learning-based paradigms. Automatic reformulation often leverages PRF or corpus statistics to expose implicit attributes or subgoals [5]. Interactive reformulation foregrounds user agency and iterative control, showing that guided term suggestion and alternate query views can materially improve effectiveness, especially for exploratory or task-oriented needs [8, 9]. More recently, neural approaches (e.g., expansion via sequence models or reranking with deep encoders) have advanced the state of the art; however, lexical methods remain competitive when carefully tuned and fused, particularly under efficiency constraints and limited training signals.

From a tooling perspective, reproducible lexical baselines over Lucene have been extensively studied through the Anserini/Pyserini stack, which provides standardized indexing and search APIs on top of mature inverted-index infrastructure [10, 11]. Our system builds directly on this line of work: we employ BM25 as the backbone, use RM3 for calibrated PRF, and apply (weighted) RRF to combine complementary lexical signals into a single robust ranking.

In addition to advances in classical retrieval and pseudo-relevance feedback, our work aligns with the DUTH team's broader research agenda on reproducible, transformer-based, and hybrid NLP pipelines. In previous shared-task participations such as SemEval–2023 [12], EXIST–2025 [13], JOKER–2025 [14], and SimpleText–2025 [15], we developed modular frameworks emphasizing transparency, parameter control, and multilingual generalization. These studies—spanning affective analysis, bias detection, text simplification, and hallucination evaluation—share a methodological focus on interpretability and reproducibility. The current TREC Product Search system follows the same design principles, applying them to fully lexical retrieval pipelines and demonstrating that rigorous calibration and controlled fusion remain effective strategies for building competitive, efficient baselines without neural components.

3 System Description

Describe the datasets, indexes, topics/queries, and any external resources (including LLMs, pretraining corpora). Mention license and availability where applicable.

3.1 Dataset

We use the official corpus and topics released for the *TREC 2025 Product Search & Recommendation* track [16]. The collection comprises product pages with canonical identifiers and core textual fields: `title`, free-text `description`, and lightweight category/attribute strings (e.g., brand, size, color). Unless otherwise noted, we index English text and ignore non-text modalities in this submission. TREC's pooled assessment protocol [17] is used by the organizers to produce final judgments.

Scope and coverage. After format normalization and integrity checks (non-empty fields, UTF-8 validity), the indexable set comprises **1,080,178** products. We retain both raw and tokenized text per record to support exact reproducibility. Where attributes provide useful context,

we flatten them into the `text` field; otherwise, they are preserved in the raw payload for future use.

Topics/queries. Evaluation uses the official 2025 test topics (released August 25, 2025) with identifiers of the form `PSRT_Search_###`. Topics are task-oriented (e.g., "complete home office makeover") and encode implicit requirements rather than explicit product names; we use the organizer IDs verbatim in all submissions.

Relevance assessments and tuning policy. We do not use any 2025 test relevance information for tuning. Prior-year qrels (e.g., 2023) were consulted strictly for pipeline sanity checks (file formats/validators), not for parameter selection. Final judgments are provided by the organizers via pooled assessment and are reported by the track.

Availability and license. The corpus and topics are distributed by the track organizers for research use (see the track's data release and license in Organizers 16). All results reported here are produced on the official release without external data.

Implementation note. We ingest the raw JSON Lines and build a single Lucene index using the Anserini/Pyserini stack [10, 11, 18], storing positions, document vectors, and raw content to ensure reproducible downstream experiments.

Table 1. Dataset summary used in our experiments.

Statistic	Value
Indexable products	1,080,178
Raw corpus size (compressed)	1.03 GB
Indexed fields	title, description (+ flattened attrs)
Test topics (2025)	100
Submission depth (K)	1000 per topic

3.2 Models

Our system is fully lexical and model–minimal: we do not employ neural re-rankers, dense retrieval, LLM prompting, or external pretraining. All models are instantiated via the Anserini/Pyserini stack on top of Lucene [10, 11, 18].

BM25 backbone. We use BM25 [1] as implemented in Pyserini (`LuceneSearcher.set_bm25`). Unless otherwise stated, the parameters are k_1=0.9 and b=0.4. The baseline run `bm25_v1` retrieves the top-K=1000 products per topic from a single Lucene index.

Pseudo-relevance feedback (RM3). To mitigate the vocabulary gap, we apply RM3 [6] (`LuceneSearcher.set_rm3`), interpolating an expansion model estimated from the BM25 top documents with the original query. We submitted three RM3 variants: (i) `rm3_v1` with (fb_terms=80, fb_docs=8, oqw=0.3), (ii) `rm3_f40d5w05` with (40, 5, 0.5) as a more conservative setting, and (iii) `rm3_f120d10w03` with (120, 10, 0.3) as a stronger, recall-oriented sweep with drift control through the original-query weight. All RM3 runs return up to K=1000 unique products per topic.

Interactive PRF reformulations. For the interactive track, `prf_v1` automatically generates four PRF-based reformulations per topic. Candidate terms are mined from the BM25 top-50 (minimum token length, stopword filtering). We construct distinct reformulations (e.g., top-6, top-10, mid-rank slices), execute BM25 for each, and then deduplicate products to a single unique list per topic (submission limit K=1000). No manual curation or human-in-the-loop adjustments are applied.

Table 1: Submitted models and configurations. All runs use the same Lucene index.

Run ID	Component	Configuration
`bm25_v1`	BM25	k_1=0.9, b=0.4
`rm3_v1`	RM3	fb_terms=80, fb_docs=8, oqw=0.3
`rm3_f40d5_w05`	RM3 (conservative)	fb_terms=40, fb_docs=5, oqw=0.5
`rm3_f120d10w3`	RM3 (strong)	fb_terms=120, fb_docs=10, oqw=0.3
`prf_v1`	Interactive PRF	4 reformulations/query; BM25; dedup to K=1000

3.3 Methodology

Our methodology is fully lexical and aims at a precise but reproducible pipeline. We (i) normalize the product corpus into a minimal JSON schema, (ii) build a single Lucene index, (iii) retrieve with BM25 [1] and apply pseudo-relevance feedback (RM3) [6] for query expansion, and (iv) optionally aggregate multiple runs via Reciprocal Rank Fusion (RRF) [7]. All components are instantiated using the Anserini/Pyserini stack [10, 11, 18].

3.3.1 Preprocessing. We ingest the raw JSON Lines files and normalize each product into three required fields: a unique `docid` (string), a `title` (string), and a `text` (string) that concatenates the product description with select flattened attributes when they add salient context (e.g., brand, size). Empty or malformed entries are discarded. Text is kept both as raw payload and as tokenized content to guarantee reproducible indexing. No neural annotations or external resources are introduced.

3.3.2 Retrieval / Generation. We construct a single Lucene index with Pyserini's `JsonCollection`,

storing positions, document vectors, and raw content (`--storePositions --storeDocvectors --storeRaw`). Unless noted, we use Lucene's default English analyzer and stopword list. Retrieval is performed with BM25 using `LuceneSearcher.set_bm25` and hyperparameters k_1=0.9, b=0.4. To bridge the intent–metadata gap in task-oriented queries, we adopt RM3 via `LuceneSearcher.set_rm3`, which interpolates a relevance model estimated from the BM25 top results with the original query. We evaluate three RM3 configurations: a conventional setting (fb_terms=80, fb_docs=8, oqw=0.3), a conservative setting $(40, 5, 0.5)$, and a stronger sweep $(120, 10, 0.3)$ that favors recall while controlling drift through the original-query weight.

For the interactive track, we auto-generate four PRF-based reformulations per topic. Candidate terms are mined from the BM25 top-50 after stopword removal and length filtering; we then build distinct expansions (e.g., top-6, top-10, and mid-ranked slices) and run BM25 for each reformulation. The resulting lists are deduplicated at the `(qid, product_id)` level and re-ranked to produce a single unique list per topic with depth K=1000.

3.3.3 Reranking / Fusion. We do not employ neural re-ranking. Instead, we fuse complementary lexical runs using RRF [7]. Given input run set $\mathcal{R}$ and rank $r(d)$ of document d in a run, the fused score is $\mathrm{RRF}(d) = \sum_{R\in\mathcal{R}} \frac{1}{k+r_R(d)}$ with the standard k=60. RRF emphasizes consensus near the top of lists while remaining robust to noise and idiosyncrasies across inputs. We also consider a weighted variant that assigns larger weight to the strongest RM3 sweep $(120, 10, 0.3)$ and unit weight to other inputs; fusion is strictly post-retrieval and leaves the index unchanged.

3.3.4 Prompting and Parameters. We do not use LLM prompting or any generative components. All hyperparameters are enumerated explicitly (BM25 k_1, b; RM3 fb_terms, fb_docs, oqw; RRF k and optional weights). Parameters are shared across all topics, and tuning does not rely on 2025 test judgments. Prior-year qrels are consulted only for sanity checks of file formats and validators.

Reproducibility. All experiments operate on the same Lucene index; submitted runs are in the six-column TREC format (`qid, reformulated_query, product_id, rank, score, runid`) with strict deduplication per topic and tab-separated output (no headers). Scripts for indexing, BM25/RM3 retrieval, interactive PRF, and RRF fusion are available to ensure end-to-end replication.

3.4 Evaluation Measures

We follow the official guidance of the *TREC 2025 Product Search & Recommendation* track for effectiveness reporting. Unless stated otherwise, results are computed on the pooled judgments produced by the organizers [17] and are reported at the cutoffs specified by the track (@10, @20, @100, @1000). We rely on `trec_eval/pytrec_eval` for implementation consistency.

Task Completion nDCG (primary). The primary evaluation measure is a task-centric, graded nDCG, where gains reflect the annotation schema: `Essential` = 3, `Highly` = 2, `Somewhat` = 1, `Not` = 0. For a ranked list $\langle d_1, \ldots, d_K\rangle$ with per-rank gain g_i, Discounted Cumulative Gain is

$$\mathrm{DCG@}K = \sum_{i=1}^{K} \frac{2^{g_i}-1}{\log_2(i+1)},$$

and the normalized score is $\mathrm{nDCG@}K = \mathrm{DCG@}K/\mathrm{IDCG@}K$ with respect to the ideal ordering [19].[1]

Essential Product Recall@K. To measure task completion directly, we report recall over the set of *Essential* ($g = 3$) products:

$$\mathrm{Recall}^{\mathrm{Ess}}@K = \frac{|\{d \in \text{top-}K : g(d) = 3\}|}{|\{d : g(d) = 3\}|}.$$

This complements nDCG by focusing solely on required items.

Product Coverage Score. Coverage captures how well the retrieved set spans different task requirements/aspects. Let A be the set of aspects and R_a the set of relevant products to aspect $a \in A$. A generic coverage surrogate is the macro-average aspect recall:

$$\mathrm{Coverage@}K = \frac{1}{|A|}\sum_{a\in A} \frac{|\{d \in \text{top-}K : d \in R_a\}|}{|R_a|}.$$

Aspect definitions follow the track's annotations.

Average Precision (AP). We also report AP as a standard single-number summary of early precision and depth-sensitivity [20]. AP assumes binary relevance; we map graded labels to binary by treating $g > 0$ as relevant (track default unless otherwise specified). For ranks $1..K$,

$$\mathrm{AP} = \frac{1}{R}\sum_{i=1}^{K} P(i)\cdot \mathrm{rel}(i),$$

[1] We denote this variant as *Task Completion nDCG* since the track emphasizes retrieving the full set of essential products for accomplishing the task.

where $P(i)$ is precision at rank i, $\text{rel}(i) \in \{0, 1\}$, and R is the number of relevant items.

Diversity Measures. To account for multi-aspect tasks, we use intent-aware diversity. We report α-nDCG@K with $\alpha = 0.5$ unless otherwise noted [21], which discounts repeated gain from the same aspect and rewards breadth across aspects.

Statistical Significance. For pairwise system comparisons we apply the paired two-tailed randomization test recommended for IR evaluation [22], at $\alpha = 0.05$. We report significance markers only after verifying identical pools and topic sets across runs.

Reporting. All evaluation measures are macro-averaged across topics. When the track provides official scripts or cutoffs, we adopt those verbatim for primary tables and relegate alternative cutoffs to the appendix for completeness.

3.5 Implementation and Environment

Implementation. All experiments were conducted on a single CPU node using a fully lexical Lucene-based retrieval stack (Anserini/Pyserini). The system requires no GPUs, no external model checkpoints, and no network access at run time. Indexing and retrieval are implemented in Python (CPython 3.10) with OpenJDK 21 for the underlying Lucene components.

Determinism. The system is fully deterministic: BM25 scoring, RM3 term estimation, and reciprocal rank fusion contain no stochastic components. Repeated executions on the same index reproduce byte-identical run files without requiring random seeds.

Reproducibility. We provide scripts and entrypoints that rebuild all submitted runs end-to-end, from the official corpus to final six-column TREC-formatted output files. The pipeline has no external dependencies and enables exact reproduction of all reported results.

4 Results

Table 2 reports the official evaluation results for the TREC 2025 Product Search and Recommendation task, as obtained from the TREC Evalbase. Performance is evaluated using three complementary metrics: task completion nDCG, mean average precision (MAP), and Essential Recall@1000. Together, these metrics capture ranking quality, early precision, and coverage of task-critical products.

4.1 Task Completion nDCG

Task completion nDCG is the primary evaluation metric of the Product Search task. Relevance labels are importance-weighted (Essential=3, Highly=2, Somewhat=1), reflecting the contribution of retrieved products to successful task completion rather than topical relevance alone.

Across the DUTH (garamp) submissions, RM3-based runs consistently achieve the highest nDCG scores. The conservative RM3 configuration (`rm3_f40d5_w05`) achieves the highest task completion nDCG value (0.385) among all submitted runs in the official evaluation, while maintaining strong performance across MAP and Essential Recall@1000. Slightly lower but comparable nDCG values are observed for the standard and more aggressive RM3 variants, suggesting that calibrated query expansion improves task completion effectiveness without substantially degrading ranking stability.

4.2 Mean Average Precision

MAP measures early precision by averaging precision at ranks where relevant products are retrieved. Unlike nDCG, MAP is sensitive to ranking consistency across the entire list rather than the relative importance of individual items.

Among the DUTH runs, the standard RM3 configuration (`rm3_v1`) achieves the highest MAP score (0.212), marginally outperforming the conservative and aggressive variants. This result suggests improved precision across ranked results, particularly for moderately relevant products. Differences among RM3 configurations reflect the expected precision–recall trade-off inherent to pseudo-relevance feedback methods.

4.3 Essential Recall@1000

Essential Recall@1000 measures the fraction of essential products retrieved within the top 1000 results, emphasizing coverage and robustness. This metric captures the system's ability to retrieve all task-critical items, even if some are ranked lower.

The highest Essential Recall@1000 (0.465) is achieved by the conservative RM3 configuration, indicating strong coverage of critical products while maintaining ranking quality. All RM3 variants substantially outperform the BM25 baseline on this metric, confirming that pseudo-relevance feedback significantly improves recall for task-oriented product search.

4.4 Cross-metric Interpretation

Taken together, the evaluation metrics reveal a consistent pattern across the DUTH submissions. RM3-based query expansion improves task completion effectiveness by simultaneously enhancing ranking quality (nDCG), early precision (MAP), and coverage of essential products (Recall@1000). Variations among RM3 configurations re-

Table 2: Official results for the TREC 2025 Product Search and Recommendation task. Evaluation is performed using task completion nDCG (Essential=3, Highly=2, Somewhat=1), mean average precision (MAP), and Essential Recall@1000, as reported by the official TREC evaluation server [16].

Run ID	nDCG	MAP	Recall@1000
rm3_f40d5_w05	**0.385**	0.209	**0.465**
rm3_v1	0.379	**0.212**	0.455
rm3_f120d10w3	0.375	0.209	0.453
bm25_v1	0.351	0.167	0.432
prf_v1	0.300	0.148	0.375

flect expected trade-offs between recall amplification and ranking stability.

The BM25 baseline, while competitive given its simplicity, underperforms across all metrics, highlighting the importance of feedback-driven reformulation for bridging the vocabulary gap between user intent and product metadata. Overall, the results demonstrate that a lightweight and fully reproducible lexical pipeline—combining BM25 with calibrated RM3 pseudo-relevance feedback—can achieve strong task-oriented effectiveness without reliance on neural re-ranking or external resources.

5 Conclusion and Future Work

We presented a fully lexical, scalable system for the *TREC 2025 Product Search & Recommendation* track. Our approach indexes ∼1.08M products and relies on BM25 [1] as a strong backbone, RM3 pseudo-relevance feedback [6] for calibrated query expansion, and (weighted) Reciprocal Rank Fusion [7] to aggregate complementary lexical evidence. The system is efficient, robust, and reproducible, implemented on top of the Anserini/Pyserini/Lucene stack [10, 11, 18], requires no neural re-ranking or external pretraining, and produces standard six-column TREC runs with strict per-topic de-duplication.

Official evaluation results indicate consistent gains of RM3 and fusion over the BM25 baseline across task completion nDCG, MAP, and Essential Recall@1000, while the interactive PRF variant yields diverse reformulations suitable for user selection or downstream aggregation.

These results reaffirm that careful lexical reformulation, classic PRF, and simple rank fusion remain competitive baselines for task-oriented product search when efficiency and transparency are primary constraints, even in comparison to heavier neural ranking approaches [1, 6, 7, 23].

Limitations. Our system is text-only and English-only; images and rich structured attributes are not exploited beyond light textual flattening. Fusion weights are heuristic rather than learned, and we do not optimize directly for aspect coverage or diversity.

Future work. We plan to: (i) explore lightweight neural re-rankers or learning-to-rank on top of strong lexical runs; (ii) investigate LLM-guided reformulation under drift controls; (iii) incorporate structured attributes and images for multimodal expansion; (iv) move from heuristic to learned fusion; (v) optimize directly for coverage/diversity using intent-aware objectives such as α-nDCG [21]; and (vi) extend to multilingual settings and recommendation subtasks (complements/substitutes).

References

[1] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 2009.

[2] N. Abdul-Jaleel, W.B. Croft, F. Diaz, et al. Relevance model 3 (rm3) and pseudo-relevance feedback. In *TREC*, 2004.

[3] J.S. Culpepper and A. Moffat. Efficient measures for online evaluation of search engines. In *ADC*, 2009.

[4] J. J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall, 1971.

[5] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of SIGIR*, pages 120–127, 2001.

[6] Nasreen Abdul-Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah Larkey, et al. Umass at trec 2004: Novelty and hard. In *Proceedings of TREC*, 2004. (Commonly cited as the description of RM3 in practice).

[7] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of SIGIR*, pages 758–759, 2009.

[8] Gary Marchionini. Exploratory search: From finding to understanding. *Communications of the ACM*, 49 (4):41–46, 2006.

[9] Diane Kelly. Interactive information retrieval: A historical perspective. *Foundations and Trends in Information Retrieval*, 1(1):1–58, 2009.

[10] Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Reproducible ranking baselines using lucene. In *Proceedings of SIGIR*, pages 1339–1342, 2017.

[11] Peilin Yang, Sean MacAvaney, and Jimmy Lin. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of SIGIR*, pages 1253–1256, 2018.

[12] Georgios Arampatzis, Vasileios Perifanis, Symeon Symeonidis, and Avi Arampatzis. Duth at semeval-2023 task 9: An ensemble approach for twitter intimacy analysis. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1225–1230, 2023. URL https://aclanthology.org/2023.semeval-1.170/.

[13] Georgios Arampatzis et al. Duth at exist 2025: Multilingual sexism detection with soft labels and transformers. In *CLEF 2025 Working Notes*, 2025. URL https://ceur-ws.org/Vol-4038/.

[14] Georgios Arampatzis and Avi Arampatzis. Duth at clef joker 2025 tasks 2 and 3: Translating puns and proper names with neural approaches. In *CLEF 2025 Working Notes*, 2025. URL https://ceur-ws.org/Vol-4038/.

[15] Georgios Arampatzis and Avi Arampatzis. Duth at clef 2025 simpletext track: Tackling scientific text simplification and hallucination detection. In *CLEF 2025 Working Notes*, 2025. URL https://ceur-ws.org/Vol-4038/.

[16] Track Organizers. Trec 2025 product search & recommendation track overview. In *Proceedings of TREC 2025*, 2025. Official data release, task definitions, and licensing.

[17] Ellen M. Voorhees. The trec experiment and evaluation. In *TREC: Experiment and Evaluation in Information Retrieval*, pages 3–20. MIT Press, 2005.

[18] Jimmy Lin et al. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. arXiv preprint arXiv:2102.10073, 2021. Accessed: insert date.

[19] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM TOIS*, 20(4):422–446, 2002.

[20] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[21] Charles L. A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *SIGIR*, pages 659–666, 2008.

[22] Mark D. Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *CIKM*, pages 623–632, 2007.

[23] Jimmy Lin, Ronak Pradeep, Rodrigo Nogueira, Andrew Yates, and Sean MacAvaney. Pretrained transformers for text ranking: Bert and beyond. In *Proceedings of SIGIR*, pages 1212–1221, 2021.