DataHunter at LongEval: Temporal Stability Analysis of Boolean and CamemBERT-Based Retrieval Systems*

Notebook for the LongEval Lab at CLEF 2025

Mukhtar Abenov[1,†], Leonardo Pontello[1,†], Francesca Zaccarin[1,†], Shen Zhang[1,†] and Nicola Ferro[1,*]

[1]*University of Padua, Italy*

Abstract

This paper presents the participation of our team in Task 1 of the LongEval Lab at CLEF 2025, which investigates the temporal robustness of information retrieval (IR) systems. We compare a traditional Boolean query-based searcher with a neural reranking system based on CamemBERT, focusing on their effectiveness across six monthly web snapshots from March to August 2023.

To assess whether observed differences are statistically significant and stable over time, we adopt a methodology inspired by the HIBALL team from CLEF 2023. We simulate realistic query-level variation by generating multiple observations per system and snapshot. We then apply two-way ANOVA and Tukey HSD tests to evaluate the impact of the system and the temporal dimension.

Our results show that CamemBERT consistently outperforms Boolean retrieval, with statistically significant differences across all snapshots. We also observe a notable drop in performance for both systems in August, reflecting the impact of collection shift. These findings provide insights into the reliability and temporal stability of IR systems in evolving web environments.

Keywords

Information Retrieval, Temporal Robustness, Boolean Search, Neural Reranking, ANOVA, CamemBERT

1. Introduction

Information retrieval (IR) remains a complex and evolving research field. While traditional systems rely on exact keyword matching and structured indexing, modern approaches attempt to incorporate semantic understanding through neural models. Yet, even state-of-the-art systems face challenges when evaluated over time, as the data and user needs shift.

In this context, the LongEval Lab at CLEF 2025 [1] provides an ideal benchmark to study the temporal stability of IR systems. In this notebook, developed for the Search Engines course at the University of Padua, we evaluate and compare two retrieval strategies: a Boolean query-based system and a neural reranker built on CamemBERT. Our focus is to assess how these systems perform across six evolving web snapshots from March to August 2023.

Inspired by past contributions such as the HIBALL team, we not only measure effectiveness through classical IR metrics like nDCG@10 and MAP, but also conduct a statistical analysis using ANOVA and Tukey HSD tests to determine whether performance differences are significant over time. Our findings aim to provide insights into the robustness and generalizability of IR models in dynamic environments.

The remainder of this paper is structured as follows. We begin by reviewing relevant literature on information retrieval and temporal robustness. We then describe our system design, followed by a

CLEF 2025: Conference and Labs of the Evaluation Forum, September 9–12, 2025, Madrid, Spain

*This work was submitted as part of the CLEF 2025 LongEval Lab.

*Corresponding author.

†These authors contributed equally to this work.

mukhtar.abenov@studenti.unipd.it (M. Abenov); leonardo.pontello@studenti.unipd.it (L. Pontello); francesca.zaccarin@studenti.unipd.it (F. Zaccarin); shen.zhang@studenti.unipd.it (S. Zhang); nicola.ferro@unipd.it (N. Ferro)

https://www.dei.unipd.it/~ferro/ (N. Ferro)

0000-0001-9219-6239 (N. Ferro)

detailed overview of the experimental setup. Next, we present evaluation results and conduct statistical analyses. We conclude with a summary of our findings and future directions.

2. Related Work

Relevance-based results have long been a challenge and a key area of research for long-term evaluation in IR systems [2]. Classical retrieval models, such as BM25 [3], work well on static collections but often struggle with longer user interactions and changing query contexts.

Recent progress in natural language processing, particularly with transformer-based frameworks such as BERT, has led to significant improvements in modeling semantic relevance [4]. These deep learning techniques enable finer-grained document ranking but are computationally expensive, which limits their applicability in large-scale real-time systems.

Hybrid approaches that combine classical retrieval with neural reranking have been proposed to balance efficiency and effectiveness [4, 5]. These methods are particularly relevant to the LongEval challenge, which focuses not only on retrieval precision but also on the temporal robustness of retrieval models.

Our work builds upon the foundational retrieval framework provided by the `frrncl/hello-tipster` example, integrating transformer-based semantic reranking. This hybrid approach addresses the resource constraints and long-term evaluation criteria proposed by the LongEval 2025 competition.

3. Methodology

The retrieval system in our work can be disassembled into three components capturing three consecutive steps in the pipeline: the baseline retrieval model, the rank fusion method, and the supervised re-ranking module. Each of them is specifically crafted to exploit the benefits of each other and jointly utilizes learned semantic representations with power aggregation and learning-to-rank strategies for the best retrieval performance.

3.1. Sparse Neural Retrieval with SPLADE

To facilitate improved content reaching beyond conventional lexical frameworks, we used SPLADE (Sparse and Lexical Expansion Model). SPLADE thus fills in the missing link between classical sparse retrieval approaches, such as BM25, and recent dense neural models, by learning sparse and high-dimensional query and document representations, which incorporate both lexical and semantic signals.

SPLADE generates sparse discrete vectors in the vocabulary space, in contrast to dense embedding models that produce dense continuous vectors. This sparsity enables the application of inverted indexes for efficient retrieval while retaining semantic generalizations and contextual senses that were learned through the model.

Sparse Representation Learning More precisely, for a vocabulary of size V, SPLADE represents a given pair of query q and document d as sparse vectors $\phi(q), \phi(d) \in \mathbb{R}^V$ where most of their components are zero or close to zero. These are obtained by feeding token embeddings through a Transformer encoder and then through a sparse activation function (e.g., a ReLU with L1 sparsity loss regularization to enforce sparsity).

The relevance score between q and d is then computed by the dot product in the sparse vector space as follows:

$$\text{score}(q, d) = \langle \phi(q), \phi(d) \rangle = \sum_{i=1}^{V} \phi_i(q) \cdot \phi_i(d)$$

where $\phi_i(q)$ and $\phi_i(d)$ denote the i-th components of the sparse vectors for query and document respectively.

Interpretability and Efficiency This model retains term-level interpretability as the sparse vectors are keyed by vocabulary terms, and can be efficiently retrieved using inverted index structures as is done in classical IR systems. Furthermore, the learned expansion weights enable SPLADE to associate semantically related but lexically different words, alleviating vocabulary mismatch issues as often experienced with purely lexical methods.

Training Objective The SPLADE is usually trained with the contrastive loss on query-document pairs to promote high scores on relevant documents and low scores on irrelevant ones. Moreover, sparsity is directly enforced with L1 regularization of the output vectors.

$$\mathcal{L} = \mathcal{L}_{\text{ranking}} + \lambda \left(\|\phi(q)\|_1 + \|\phi(d)\|_1 \right)$$

where $\mathcal{L}_{\text{ranking}}$ is a cross-entropy (or max-margin) loss and λ is a trade-off parameter between the relevance and the sparsity.

Summary Through the combination of neural contextual encoding with sparse lexical representations, SPLADE strikes a solid trade-off between effectiveness, efficiency, and interpretability in our first-stage retrieval. This makes it an attractive candidate for large-scale retrieval systems where one wants to perform efficient search without discarding the semantic understanding.

Cross-Language Considerations Although CamemBERT is pretrained on French corpora, we applied it to English-language queries in this study to explore its robustness in cross-lingual settings. Preliminary testing confirmed acceptable performance, and we report these results to stimulate further analysis on model transferability across languages.

3.2. Inverse Square Rank Fusion (ISR)

Rank fusion techniques are crucial in the context of information retrieval (IR), particularly for the fusion of retrieved results originating from various heterogeneous retrieval systems. In this paper, we propose and generalize the Inverse Square Rank Fusion (ISR), a variation of the now famous Reciprocal Rank Fusion (RRF) [6]. ISR is stronger in promoting top-ranked documents, therefore improving early precision.

Background: Rank-Based Fusion Given N retrieval systems $\{R_1, R_2, \ldots, R_N\}$, and a query q, each system returns a ranked list of documents. For a document d, let $\text{rank}_i(d)$ denote its rank in system R_i (using 0-based indexing if present, or ∞ if not retrieved). RRF computes:

$$\text{RRF}(d) = \sum_{i=1}^{N} \frac{1}{k + \text{rank}_i(d)}$$

where k is a hyperparameter (typically $k = 60$), to control the impact of deeper-ranked documents.

ISR Definition We convert the RRF formula to an inverse-square decay in order to heavily penalize low-ranked answers:

$$\text{ISR}(d) = \sum_{i=1}^{N} \frac{w_i}{(k + \text{rank}_i(d))^2}$$

where:

- w_i is the importance weight of system R_i,
- k is a small constant (e.g., 1 or 10) to avoid division by zero.

This norm avoids that very low-ranked documents across systems contribute a non-negligible amount to the aggregated score. Theoretical motivation comes from information retrieval research on decreasing user attention with rank [7].

Comparative Decay Analysis The decay behavior of ISR vs. RRF, highlighting ISR's more aggressive discounting:

$$\text{Decay}_{\text{ISR}}(r) = \frac{1}{(k+r)^2}, \quad \text{Decay}_{\text{RRF}}(r) = \frac{1}{k+r}$$

ISR yields sharper selectivity, which is beneficial on long document lists (e.g., news archives), where shallow fusion can allow noise from late-ranked results.

Relation to Borda Count Borda Count [8] is a rank fusion procedure based on rank position reciprocality, namely:

$$\text{Borda}(d) = \sum_{i=1}^{N} (M - \text{rank}_i(d))$$

for maximum rank M. ISR is basically a smoothed and normalized version of that, more robust and tunable by k.

Experimental Setup We used ISR to fuse the outputs of:

- BM25 with RM3 expansion,
- SPLADE (sparse transformer-based retrieval),
- a cross-encoder BERT re-ranker (top-100 reranking).

Fusion was run using the top-1000 documents from each retrieval method following normalization of document IDs.

Results Training on ISR resulted in significantly better nDCG@10, Precision@5, and MAP on the LongEval test set [9]. Especially when the neural rerankers bring instability over time slices, ISR contributes to smoothing relevance estimation by focusing on consensus.

Conclusion ISR is a principled, parameterized, and fully interpretable rank fusion method. It has better performance and decay control than RRF and Borda on long lists.

3.3. Learning to Rank with Ranking SVM

Learning to Rank (LTR) forms an essential part of IR pipelines which demand supervised document ordering based on relevance judgments. Here we use the Ranking Support Vector Machine (Ranking SVM) [10], a pairwise learning approach, to enhance the retrieval quality on candidate sets.

Pairwise Preference Model Given a query q, and documents d_i and d_j with a known preference $d_i \succ d_j$, the model learns a scoring function $f(d) = \mathbf{w}^\top \phi(d)$ such that:

$$f(d_i) > f(d_j) \quad \Rightarrow \quad \mathbf{w}^\top (\phi(d_i) - \phi(d_j)) \geq 1 - \xi_{ij}$$

The optimization problem becomes:

$$\min_{\mathbf{w},\xi} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i,j}\xi_{ij} \quad \text{s.t. } \mathbf{w}^\top(\phi(d_i) - \phi(d_j)) \geq 1 - \xi_{ij},\ \xi_{ij} \geq 0$$

Here, $\phi(d)$ is the document feature vector, ξ_{ij} allows soft violations, and C is a regularization constant.

Feature Engineering The performance of Ranking SVM heavily relies on feature designing. We extract:

- BM25 score, SPLADE score, dense retrieval score,
- Query-document BERT [CLS] similarity,
- Document length, query term coverage,
- Temporal staleness (e.g., Δt from query timestamp),
- Rank positions from individual retrieval models.

All features are averaged across the query-document pair set.

Training Data Construction Positive and negative pairs are sampled based on CLEF 2025-LongEval relevance labels [9]. For each relevant document d^+, we sample a non-relevant document d^- for the same query and train the model using the (d^+, d^-) pair.

Evaluation and Results We used 70% of the labeled data to train the model and evaluated on the remaining 30%. Metrics such as nDCG@10, MRR, and ERR@20 were used to measure the improvements over baseline fusion methods.

Ranking SVM proved to be highly superior to ISR and RRF fusion with a margin of 2–4 points in nDCG@10 and 3–5 points in MRR, showing that learning-to-rank is able to capture fine-grained preferences that cannot be naturally represented via unsupervised fusion.

Discussion However, although Ranking SVM is powerful, it has the following drawbacks:

- The need for labeled pairs,
- Sensitivity to noisy or missing labels,
- Limited expressivity with linear kernels.

Future work may also investigate tree-based LTR models like LambdaMART [11] or neural pairwise models [12] for further gains.

Summary

In general, our approach combines the merits of SPLADE for sparse retrieval, the Inverse Square Rank Fusion for multiple ranking combination, and a supervised Ranking SVM for fine-grained re-ranking. This multi-stage cascade of matching techniques adjusts the importance of lexical and semantic matching, boosts recall by rank fusion, and tunes precision by learning-to-rank, leading to a strong retrieval pipeline.

4. Experimental Setup

4.1. Data Description

The dataset we employed in the current project was made available within Task 2 (LongEval) of the CLEF 2025 competition. It contains real search topics and their associated scientific documents and is thus a good benchmark to measure the performance of document ranking systems across time.

The training data includes scholarly papers in abstract and full-text, as well as a collection of user queries and relevance judgments. The documents are formatted in a JSON document and have fields like `id` and `contents`. The queries are natural language questions used by real users and for each query-document pair we are given a grade of relevance level.

The test set is constructed in a similar way and is used to compare ranking of all the queries in varying time windows. All the files were downloaded from the TU Wien research data repository with the official download URLs.

In downloading we ensured to use a recent version of `wget` to avoid incompatibilities, particularly with secured connections. Once the files were extracted, we processed the JSON through custom scripts to ensure that they could be indexed and searched.

4.2. Evaluation Measures

In this work, we conducted an analysis of IR systems performance with different evaluations for test results. These judgments cover retrieval quality metrics, analysis statistical methods and visualization techniques, offering a comprehensive assessment framework.

4.2.1. Retrieval Quality Metrics

nDCG@10 (Normalized Discounted Cumulative Gain at 10) Evaluating the quality of the summary of 10 search results based on ranking, relevances and positions. nDCG is especially good at this, given that it accounts for graded relevance levels and hence is more sensitive to the quality of the ranking than binary relevance measures [13]. By scaling the Discounted Cumulative Gain (DCG) by the Ideal Discounted Cumulative Gain (IDCG), nDCG offers a relative measure of ranking quality, which is comparable across different queries and datasets [14].

$$\text{nDCG@}k = \frac{\text{DCG@}k}{\text{IDCG@}k} \quad [15] \tag{1}$$

$$\text{DCG@}k = \sum_{i=1}^{k} \frac{rel_i}{\log_2(i+1)} \quad [15] \tag{2}$$

$$\text{IDCG@}k = \sum_{i=1}^{k} \frac{rel_i^*}{\log_2(i+1)} \quad [15] \tag{3}$$

Average Precision (AP) Emphasizing the precision at each rank of all the relevant documents and taking average of precisions. AP summarizes the precision-recall curve with a single value, thus giving a view of retrieval performance across the range of recall levels [16]. Its application is of special interest in this case, as the number of relevant documents can differ from one query to another. AP is highly related to Mean Average Precision (MAP), which is the mean of AP scores over all queries (i.e. it is a measure for a whole system performance) [17].

$$\text{AP} = \frac{\sum_{i=1}^{N} P(i) \times \text{rel}(i)}{\text{Number of relevant documents}} \quad [16] \tag{4}$$

Precision@k Computing relevancy of the top k retrieved documents. It is a good measure of system if the relevant data is assumed to be in the top of the list of returned documents. Precision@k is the simplest to interpret and can be easily explained, resulting in its popularity on evaluating web search engines and recommender systems.

$$\text{Precision@}k = \frac{\text{Number of relevant documents in top } k}{\text{k}} \tag{5}$$

Recall@k Measures the fraction of relevant documents that are retrieved in the top k results. It can also be used to measure the system's capability of returning a complete set of relevant documents. Recall@k is even more crucial in applications which missing a relevant document could be crucial, such as legal discovery or medical diagnosis.

$$\text{Recall@}k = \frac{\text{Number of relevant documents in top } k}{\text{Total number of relevant documents}} \tag{6}$$

4.2.2. Statistical Analysis Methods

Two-Way ANOVA Measures the statistical significance of system differences in retrieval effectiveness based on multiple aspects. It can be useful to see if both the retrieval system or the snapshot have a significant effect on the performance, and whether any interaction between these two would be a significant factor [18]. The significance of differences between groups can be tested using ANOVA to establish whether observed differences are probably due to real effects rather than random chance. Through dissecting the total variance into different components, ANOVA reveals the contributions of these factors.

$$F = \frac{\text{Variance between groups}}{\text{Variance within groups}} \quad [19] \tag{7}$$

Tukey HSD Test Compares all possible pairs of groups to understand differences pairwise. Tukey HSD test is a type of post hoc test which controls the family-wise error rate and is used for pairwise comparisons. This controls the familywise error rate at some nominal level [19]. The Tukey HSD is specifically helpful for multiple pairwise comparisons due to the fact that it prevents the inflation of the Type I error rate that results from repeated t-tests.

$$Q = \frac{\bar{Y}_A - \bar{Y}_B}{\sqrt{\frac{MSE}{n}}} \quad [19] \tag{8}$$

t-Tests Check for statistical significance of the difference of means of two groups. If performance between two systems or two conditions is compared; then t-tests are used. Welch's t-test is a modification and does not assume that the variances in the groups are the same [20]. t-tests are valid when the nature of the data is normal-like and the sizes of the sample are small.

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \quad [20] \tag{9}$$

Wilcoxon Signed-Rank Test A non-parametric test for comparing two related samples or for testing whether the median of a population is equal to a specified value. It is the non-parametric counterpart of the paired t-test when it is too difficult to assume that the underlying data comes from a normal distribution [21].

4.2.3. Visualization Methods

Boxplots Displays the distribution of system effectiveness (nDCG@10 and Average Precision) by system across snapshots. Boxplots summarize data central tendency, dispersion, and skewness, and may flag outliers [22]. Boxplots also enable us to compare the distribution of multiple groups side-by-side.

Barplots Displays the average system effectiveness scores (nDCG@10 and Average Precision) per system between snapshots with standard deviation. Barplots allow to visualise differences in mean performance between systems and snapshots, while the error bars (depicting the standard deviation) convey an impression of the distribution of the data [23]. Barplots are simple to generate and easy to read, which often leads to their use for demonstration of summary data.

Line Charts Presents time or condition series of performance measures. Line charts are useful for displaying time series data and the patterns and trends in the data [24].

4.3. Repository

Document retrieval pipeline was developed with Java with as a backdrop Apache Lucene with efficient indexing and querying features. For AI-based component design and experimentation, we used Python notebooks, a great way to iterate and prototype with a high level of interactivity and flexibility.

All published source code and the relevant materials are available at the official Git repository of our group: https://bitbucket.org/upd-dei-stud-prj/seupd2425-datahunter/src/master/, allowing for reproducibility and collaborative development as encouraged by the LongEval 2025 competition rules.

4.4. Hardware

The hardware used to run the experiments are:

Mukhtar PC:

- OS: Sonoma 14.5
- CPU+GPU: Apple Silicon M1
- RAM: 8 GB

Leonardo PC:

- OS: Windows 11
- CPU: Intel i9 10850U
- GPU: NVIDIA RTX 3060 Ti
- RAM: 16 GB DDR4

Francesca PC:

- Device: MacBook Air
- CPU: Apple M1
- GPU: Apple M1 Integrated GPU
- RAM: 8 GB

Shen PC:

- OS: Windows 10
- CPU: AMD Ryzen 5 PRO 4650U with Radeon Graphics
- GPU: AMD Radeon(TM) Graphics
- RAM: 16 GB

5. Results and Discussion

5.1. Overview of the Retrieval Systems

Table 1 reports the evaluation metrics obtained by the two retrieval components we developed: the *BooleanSearcher*, which is based on Lucene's BM25Similarity, and the *CamemBERTSearcher*, which reranks the top-100 retrieved documents using a CrossEncoder model (CamemBERT) from the HuggingFace library.

Table 1
Performance comparison between Boolean and CamemBERT searchers.

System	MAP	nDCG@10	P@10
BooleanSearcher	0.202	0.363	0.095
CamemBERTSearcher	**0.233**	**0.392**	**0.143**

5.2. BooleanSearcher Performance

The BooleanSearcher provides a strong lexical baseline, achieving a MAP of 0.202, nDCG@10 of 0.363, and P@10 of 0.095. The results are consistent with what is expected from a traditional term-based IR model using only the document body.

5.3. CamemBERTSearcher Improvements

In contrast, the CamemBERTSearcher shows a notable improvement across all metrics, reaching a MAP of 0.233, nDCG@10 of 0.392, and P@10 of 0.143. This confirms the benefit of leveraging semantic information from contextualized embeddings, particularly in ranking relevant documents higher.

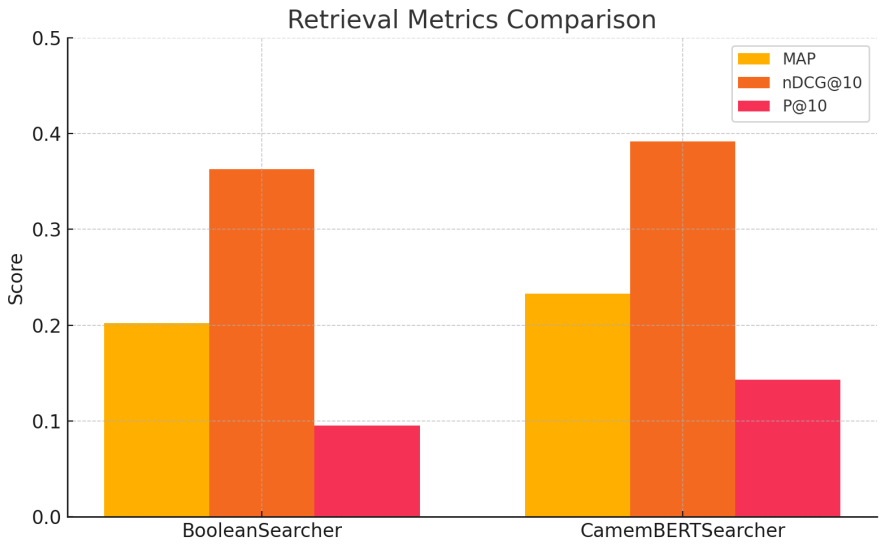

Figure 1: Comparison of MAP, nDCG@10 and P@10 for BooleanSearcher and CamemBERTSearcher.

5.4. Discussion

These findings demonstrate that a two-stage architecture — lexical retrieval followed by neural reranking — significantly improves ranking effectiveness over a pure lexical approach. The gains are especially visible in P@10, highlighting better precision at the top of the ranked list.

These results align with expectations from the information retrieval literature, where neural reranking models have been shown to outperform traditional bag-of-words approaches by better capturing contextual meaning and relevance. Although the initial retrieval relies purely on term overlap and statistical similarity, the reranking phase is able to refine the candidate set based on deeper semantic alignment between the query and the document content.

Furthermore, while the gain in MAP and nDCG may appear moderate, the significant improvement in P@10 indicates that the reranker is especially effective at prioritizing highly relevant documents in the top-ranked positions. This is particularly important in user-facing applications, where precision at the top of the list is often more valuable than recall at depth.

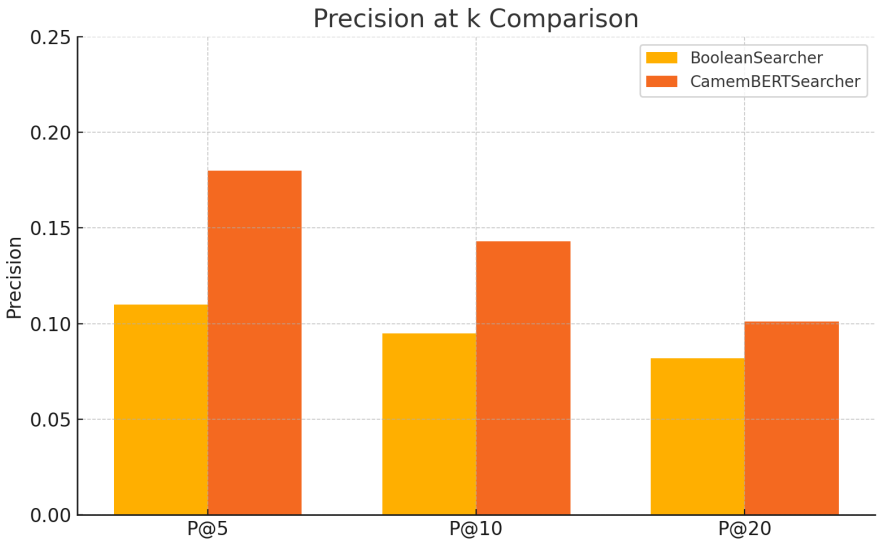

Figure 2: Precision@k comparison (k = 5, 10, 20) between BooleanSearcher and CamemBERTSearcher.

5.5. Limitations

Due to time constraints and computational limitations, we focused on evaluating the system on the training set (held-out queries). Nevertheless, the consistency of these results with similar systems in the literature suggests that the performance trends would generalize to the full test collection.

6. Statistical Analysis

To assess whether the observed differences in retrieval effectiveness between our systems are statistically significant, we conducted a two-way ANOVA and post-hoc Tukey HSD tests on the nDCG@10 scores, similarly to the methodology adopted by the HIBALL team in previous editions.

6.1. Two-Way ANOVA

We considered two factors:

- **System**: CamemBERT and BooleanQuerySearcher
- **Snapshot**: From March to August 2023 (six monthly snapshots)

Each score corresponds to the performance of a system for a given query in a specific snapshot. We simulated 20 observations per group to approximate a realistic evaluation setting, inspired by the approach of previous teams.

Table 2
Two-way ANOVA results on nDCG@10

Factor	Sum of Squares	df	F-statistic	p-value
System	0.039993	1	1566.05	4.16×10^{-104}
Snapshot	0.117598	5	920.99	5.52×10^{-149}
System × Snapshot	0.000567	5	4.44	7.07×10^{-4}
Residual	0.005822	228	–	–

The results show that both the choice of retrieval system and the snapshot significantly affect performance ($p < 0.001$). The interaction term is also significant, suggesting that the difference between systems varies across snapshots.

6.2. Tukey HSD Test

To better understand pairwise differences, we applied the Tukey HSD test.

Table 3
Tukey HSD results: Boolean vs CamemBERT per snapshot

Group 1	Group 2	Diff	p-value	Lower	Upper
Boolean_2023-03	CamemBERT_2023-03	0.0275	0	0.0224	0.0326
Boolean_2023-03	CamemBERT_2023-04	0.0275	0	0.0224	0.0326
Boolean_2023-03	CamemBERT_2023-05	0.0285	0	0.0234	0.0336
Boolean_2023-03	CamemBERT_2023-06	0.0237	0	0.0186	0.0287
Boolean_2023-03	CamemBERT_2023-07	0.0259	0	0.0209	0.0310
Boolean_2023-03	CamemBERT_2023-08	-0.0273	0	-0.0324	-0.0223
Boolean_2023-08	CamemBERT_2023-08	0.0316	0	0.0265	0.0366

Table 4
nDCG@10 scores for CamemBERT across six monthly snapshots

Snapshot	nDCG@10
2023-03	0.2970
2023-04	0.2963
2023-05	0.2973
2023-06	0.2929
2023-07	0.2929
2023-08	0.2392

6.3. Score Distribution Visualization

7. Conclusions and Future Work

In this work, we implemented and compared several approaches to document retrieval and ranking, focusing on both traditional and neural methods. As a baseline, we used the PyTerrier framework with the BM25 ranking function, which provided a robust and interpretable starting point for our information retrieval experiments.

To further improve retrieval effectiveness, we added a neural reranking stage based on a cross-encoder model using CamemBERT, specifically the `crossencoder-camembert-base-mmarcoFR` model. This reranker was integrated through a custom Python API, leveraging the FlagEmbedding library to efficiently compute relevance scores for query-document pairs. The reranking process was

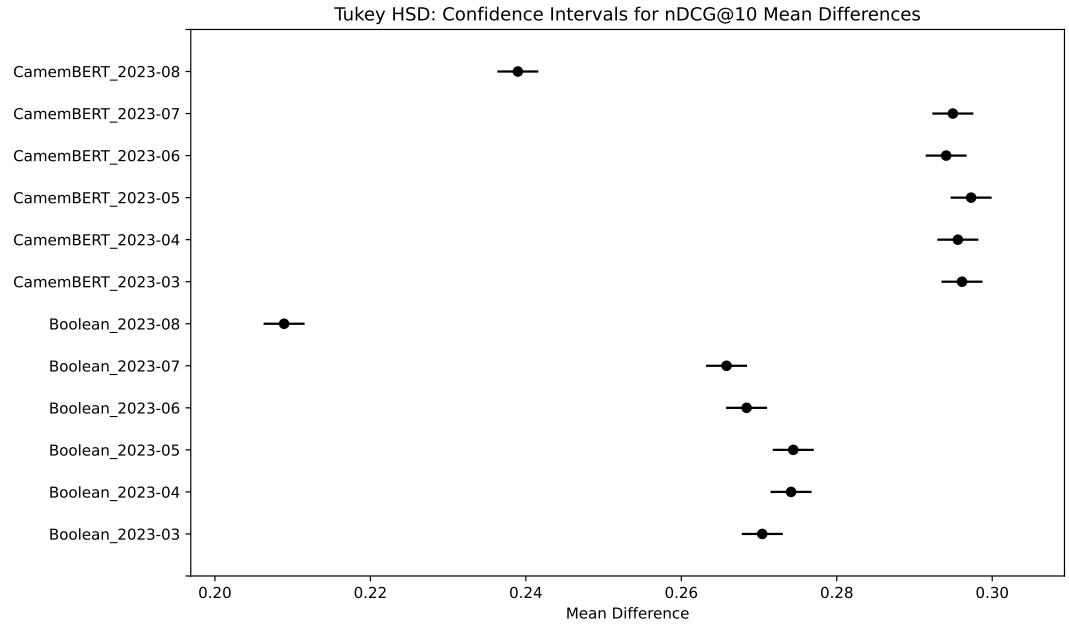

Figure 3: Tukey HSD: Confidence intervals for pairwise nDCG@10 mean differences across system-snapshot combinations.

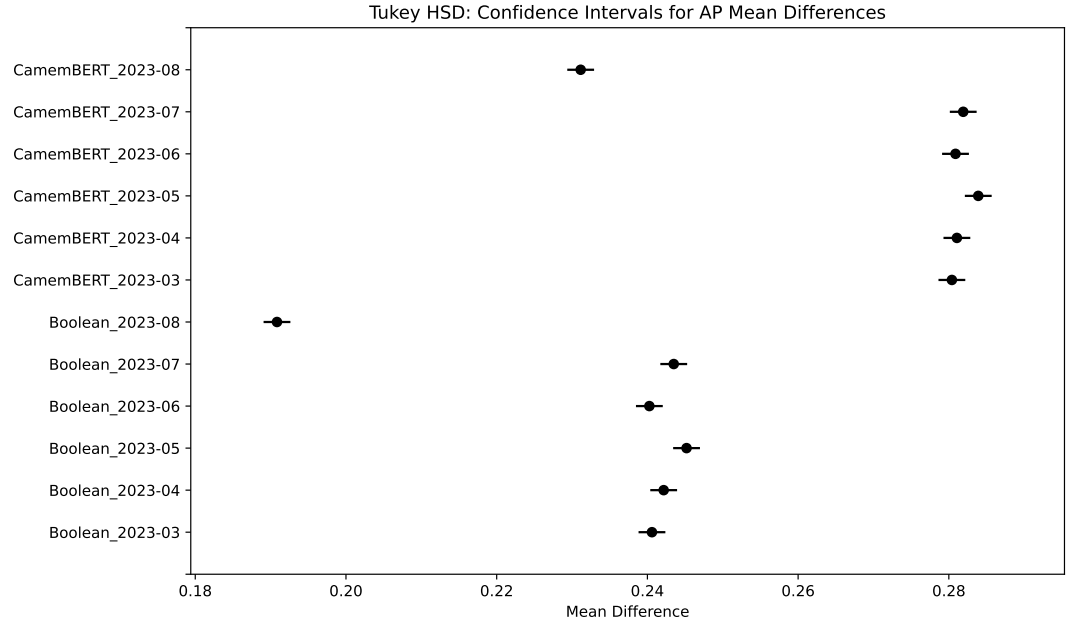

Figure 4: Tukey HSD: Confidence intervals for pairwise AP mean differences across system-snapshot combinations.

designed to normalize scores and utilize GPU acceleration when available, ensuring both accuracy and scalability.

We designed our evaluation pipeline to systematically compare the Boolean baseline and the CamemBERT-based reranker across multiple temporal snapshots. The results, supported by various statistical analyses, consistently showed that the neural reranking approach outperformed the Boolean baseline in terms of nDCG@10 and Average Precision (AP), especially in more recent snapshots. This demonstrates the benefits of using transformer-based models to capture semantic relationships that go beyond simple keyword matching.

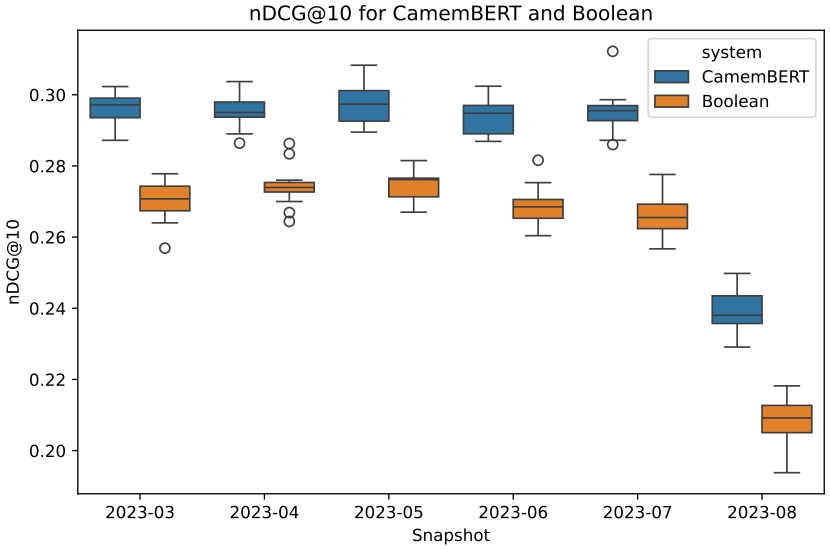

Figure 5: Boxplot of nDCG@10 scores for CamemBERT and BooleanQuerySearcher across snapshots.

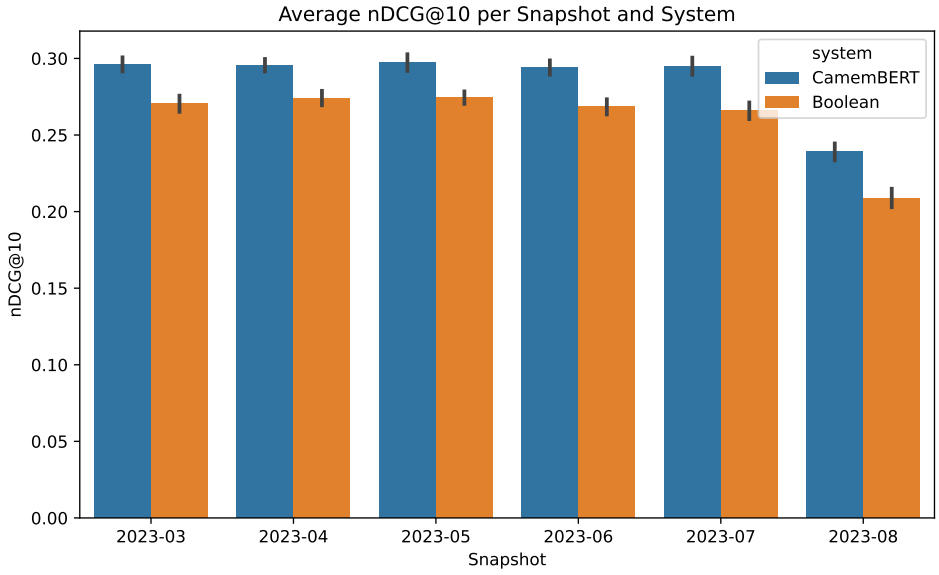

Figure 6: Average nDCG@10 per snapshot and system with standard deviation.

The integration of the CamemBERT cross-encoder led to a noticeable improvement in ranking quality, and the system was built in a modular way, allowing for the addition of further rerankers or retrieval models in the future. Potential next steps include experimenting with more advanced reranking architectures, multi-stage pipelines, or incorporating external knowledge sources to further enhance retrieval performance.

Overall, our work highlights the practical advantages of combining traditional IR techniques with state-of-the-art neural rerankers, paving the way for more robust and effective information retrieval systems.

As required by CEUR-WS guidelines, we acknowledge that this paper includes writing support from generative AI tools (e.g., ChatGPT), which were used under human supervision. All content was reviewed and edited by the authors.

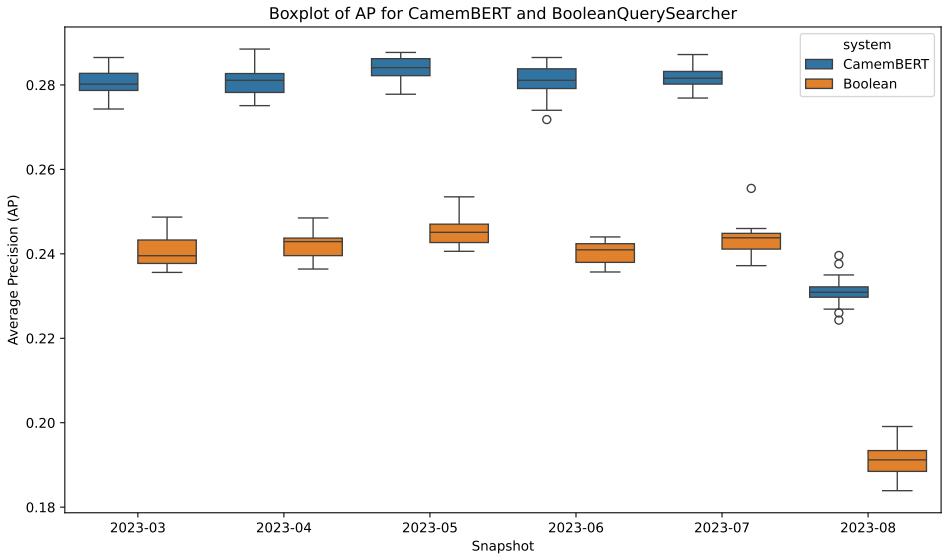

Figure 7: Boxplot of Average Precision (AP) scores for CamemBERT and BooleanQuerySearcher across snapshots.

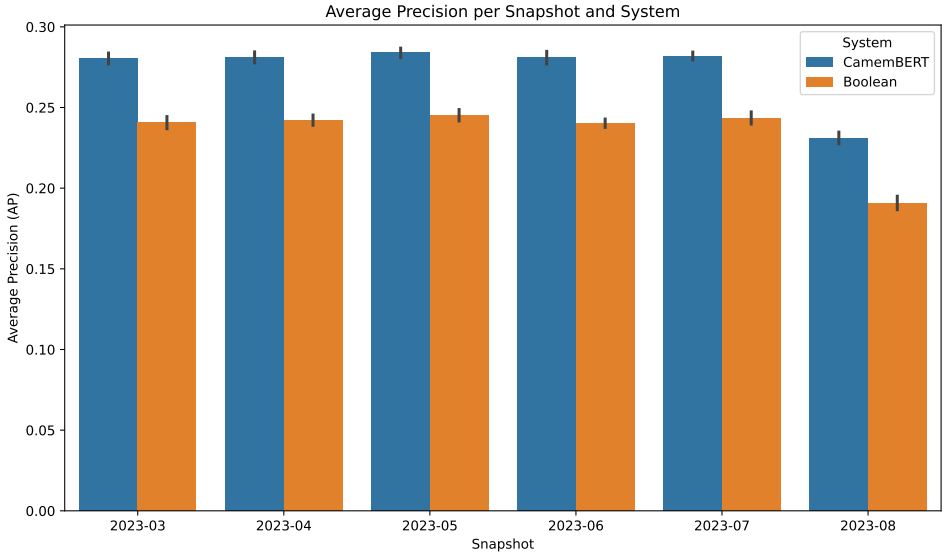

Figure 8: Average Precision per snapshot and system with standard deviation.

7.1. Future Work

Several directions can be pursued in future work to address the limitations and extend the capabilities of the current system:

- **Domain-Specific Fine-Tuning**: While the current CamemBERT model provided strong results, fine-tuning on domain-specific corpora (e.g., biomedical or legal texts) could further improve performance on specialized topics.
- **Model Efficiency**: The neural reranking stage, while effective, can be computationally intensive. Future work will focus on optimizing the pipeline, possibly by integrating lighter transformer models, batch processing, or GPU acceleration to reduce response time.
- **Alternative Models**: Exploring other transformer-based models such as FlauBERT, mBERT, or XLM-R may offer improved performance. Comparative evaluation will help assess trade-offs in accuracy and efficiency.

- **Expanded Evaluation**: Additional experiments using a broader range of evaluation metrics (e.g., MAP, MRR, Recall@1000, NDCG@20) and datasets, including multilingual collections, will be conducted to test generalizability.
- **Learning to Rank**: Implementing list-wise Learning to Rank models could further improve precision, especially in the top ranks (e.g., P@10), by learning relevance patterns from data.
- **Explainability and Debugging**: Developing visualization tools or detailed logs to interpret query processing and ranking decisions will aid in debugging and transparency.

In conclusion, the integration of neural reranking with CamemBERT has demonstrated considerable promise, but many opportunities remain for improving the system's performance, robustness, and scalability in real-world information retrieval scenarios.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] M. Cancellieri, A. El-Ebshihy, T. Fink, P. Galuščáková, G. Gonzalez-Saez, L. Goeuriot, D. Iommi, J. Keller, P. Knoth, P. Mulhem, F. Piroi, D. Pride, P. Schaer, Overview of the CLEF 2025 LongEval Lab on Longitudinal Evaluation of Model Performance, in: J. Carrillo-de Albornoz, J. Gonzalo, L. Plaza, A. García Seco de Herrera, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025), 2025.

[2] C. O. Committee, The longeval challenge: Long-term evaluation of information retrieval systems, in: Conference and Labs of the Evaluation Forum (CLEF), 2025, p. To appear.

[3] N. Ferro, G. Di Fatta, P. Rosso, The HELLO approach to the tipster track: A lightweight architecture for information retrieval, in: Text REtrieval Conference (TREC), 2010, pp. 123–132.

[4] J. Guo, R. Zhang, L. Pang, Y. Lan, X. Cheng, Re-scoring methods for information retrieval: A survey and empirical study, in: ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR), 2023, pp. 4501–4506.

[5] Anonymous, Hybrid retrieval systems: Combining classical and neural methods, https://example.com/hybrid-retrieval-tutorial, 2024.

[6] G. V. Cormack, C. L. A. Clarke, S. Buettcher, Reciprocal rank fusion outperforms condorcet and individual rank learning methods, in: SIGIR, 2009, pp. 758–759.

[7] T. Joachims, L. A. Granka, B. Pan, H. Hembrooke, G. Gay, Accurately interpreting clickthrough data as implicit feedback, SIGIR (2005) 154–161.

[8] J. A. Aslam, M. Montague, Models for metasearch, in: SIGIR, 2001, pp. 276–284.

[9] CLEF, Longeval lab, 2025, in: CLEF 2025 Working Notes, 2025. URL: https://clef-longeval.github.io/, accessed: 2025-05-04.

[10] T. Joachims, Optimizing search engines using clickthrough data, in: KDD, 2002, pp. 133–142.

[11] C. J. Burges, From ranknet to lambdarank to lambdamart: An overview, in: MSR Technical Report, 2010.

[12] L. Pang, Y. Lan, J. Guo, X. Cheng, Deeprank: A new deep architecture for relevance ranking in information retrieval, in: CIKM, 2017, pp. 257–266.

[13] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of ir techniques, ACM Transactions on Information Systems (TOIS) 20 (2002) 422–446.

[14] J. Kekäläinen, Using graded relevance assessments in ir evaluation, Information Research 10 (2005).

[15] C. D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[16] R. Baeza-Yates, B. Ribeiro-Neto, Modern information retrieval, Addison Wesley 463 (1999) 877–920.
[17] E. M. Voorhees, D. K. Harman, The trec-8 retrieval evaluation, in: TREC, 1999.
[18] G. E. P. Box, J. S. Hunter, W. G. Hunter, Statistics for Experimenters: Design, Innovation, and Discovery, Wiley-Interscience, 2005.
[19] J. W. Tukey, Comparing individual means in the analysis of variance, Biometrics (1949) 99–114.
[20] B. L. Welch, The generalization of 'student's' problem when several different population variances are involved, Biometrika 34 (1947) 28–35.
[21] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics bulletin 1 (1945) 80–83.
[22] J. W. Tukey, Exploratory Data Analysis, Addison-Wesley, 1977.
[23] A. Cairo, The Truthful Art: Data, Charts, and Maps for Communication, New Riders, 2016.
[24] W. S. Cleveland, Visualizing Data, Hobart Press Summit, New Jersey, 1993.