CIR at LongEval 2026: Ad-Hoc Scientific Retrieval, Topic Extraction From Query Logs, and User Simulation

Can Bakirci[1,†], Alexander Brückner[1,†], Joshua Matthew Christian[1,†], Nadja Kahsai Debrezion[1,†], Carmelo Heinrich Antonino Di Pino[1,†], Abdelilah Imam[1,†], Kübra Kartal[1,†], Aaron Benjamin Klatt[1,†], Robin Klinkhammer[1,†], Yannick Mählmann[1,†], Lennard Michel[1,†], Emirhan Sahin[1,†], Leon Damian Thies[1,†], Vico Timmer[1,†], Andreas Konstantin Kruff[1,*] and Jüri Keller[1,*]

[1]TH Köln – University of Applied Sciences, Claudiusstraße 1, Cologne, 50678, Germany

Abstract

In this paper, we describe the CIR contributions to the LongEval Lab 2026. We participated in three subtasks, namely LongEval-Sci, LongEval-TopEx, and LongEval-USim. For the LongEval-Sci ad-hoc search task, we experimented with the citation data and developed a variety of relevance signals that are used directly to boost documents or as features for a learning-to-rank approach. For the topic generation in the LongEval-TopEx task, several efficient in-context learning approaches based on tf-idf keywords were proposed. For the LongEval-USim task, the formalized topics were reused to predict the next query. Furthermore, we experimented with different persona-based and reasoning-based strategies for the next query prediction.

In total, we submitted four runs to Task 1, six runs to Task 2, and 25 runs to Task 3. The evaluation of the ad-hoc search task on test data shows that the temporal signals improve the retrieval performance, whereas the citation signals provide an observable benefit only on the last snapshot. The synthetic labels based on the topics generated for Task 2 show a low agreement with the click-based labels. For Task 3, we report both the pre-evaluation results of the teams, which served as a proxy during the development phase and reflect how the teams optimized their approaches using the training data, and the test results provided by the organizers, which enable a standardized comparison of the performance of the different approaches. All submitted runs made extensive use of LLMs. However, larger models did not consistently outperform smaller ones, with performance depending more on the prompting strategy than on model size. Reasoning-based approaches did not yield substantial improvements, while persona-based prompting showed mixed results across teams. Overall, incorporating more contextual information generally led to better performance.

Keywords

Learning to Rank, Ad-hoc retrieval, Cranfield, Information Need, Chain-of-Thought, User simulation, Next Query Prediction

1. Introduction

The LongEval lab at CLEF focuses on evaluating various aspects of information access systems in dynamically evolving scenarios. For further descriptions of the different tasks and the evaluation outcomes, we refer to the overview papers [1, 2]. In this lab notebook, we summarize our submissions to three tasks of the 2026 iteration of the lab. Specifically, we participated in the LongEval-Sci, LongEval-TopEx, and LongEval-USim tasks with new and extended approaches. The submissions are the result of a students' project course with the Cologne Information Retrieval (CIR) group at TH Köln - University of Applied Sciences in Cologne, Germany. Five teams participated in the course. The implementations, including code and prompt designs from all teams, are publicly available in a corresponding GitHub repository.[1]

CLEF 2026 Working Notes, 21–24 September 2026, Jena, Germany

*Corresponding author.

†These authors contributed equally.

[1]https://github.com/irgroup-classrooms/LongEval26-CIR-classrooms

2. LongEval-Sci: Ad-Hoc Scientific Retrieval

Two teams participated in the first task, LongEval Ad-Hoc Scientific Retrieval. They experimented with publication recency and citation data. Various document boosts were derived from the document metadata and used to re-rank the BM25 baseline over the abstract collection.

2.1. Approaches and Implementation

Team Time4Citation The Time4Citation team created static, additive boosts for timely and cited documents. A standard BM25 baseline serves as the first-stage ranker, retrieving the top 1,000 documents, which are subsequently re-ranked. The approaches are implemented in PyTerrier [3], and the data is interfaced through the `ir_datasets_longeval` integration [4, 5]. The citation data was extracted from the OpenCitation dump provided by the lab.

The evolving state-of-the-art in scientific research continuously introduces new publications, which are indexed across progressing snapshots of the collection. This dynamic environment not only fosters new information needs but can also alter the relevance of existing documents and their utility for future research. Crucial to scientific literature, new publications inherently reference prior work. Although citations and references do not provide a direct measure of immediate reception, they serve as valuable metadata and can be leveraged as ranking signals in various ways.

The first approach focuses on the date of the newest citation as a proxy for recency. This is motivated by the hypothesis that publications continuing to receive citations retain higher recency and stay relevant compared to those whose citation activity has ceased. This date is compared to a reference date to boost more recent documents. For simplicity, the corpus age, measured as the average publication year of all documents in the corpus (2017), is used as a reference point. If the newest citation is newer than the corpus age, a boosting factor of 2.5 is applied; if it is older, the BM25 score is only boosted by 1.5. Publications that were never cited receive no boosting. By this, a citation of any date is used as an additional quality indicator.

The second approach relies on the absolute citation counts to favor documents with many citations. Because citation distributions are highly skewed, with a few publications accumulating exponentially more citations than others, a logarithmic transformation is applied to the raw citation count c. To account for documents with zero citations, we add 1 to c to ensure the function remains defined. To control the impact of this boost, a free tuning parameter, α, scales the log-transformed citation count:

$$s_{\text{final}} = s_{\text{BM25}} + \alpha \cdot \ln(1 + c) \tag{1}$$

The best results were achieved with an α of 0.3. The boosting factors and α values were determined by a grid search on the training dataset.

The final approach simplifies the citation boost by focusing solely on whether citations are present. This uses the citation data as a direct document quality indicator. Therefore, the BM25 score of documents that receive at least one citation is increased by a static boost. This boost was also determined by a grid search on the training dataset, testing boosts between 0.2 and 2. The best results were achieved by a boost of 0.9.

Team DenseFusion The second team implemented a multi-stage Learning-to-Rank (LTR) pipeline using LightGBM Gradient Boosting Decision Tree (GBDT) [6]. For the first retrieval stage, BM25 is used, of which the first 500 results are subsequently re-ranked. The cutoff at 500 documents was chosen for efficiency reasons. These candidates are then enriched with a diverse set of features comprising textual, temporal, and bibliographic information. For the textual features, the initial BM25 score, the absolute query length, the length of the abstract, and the exact token match between the query and the abstract were used. Additionally, the cosine similarity between query and abstract embeddings is used. The publication's absolute age in months is used as a temporal signal to represent recency. Since the metadata of the publications was found to be incomplete in some cases, the age could not be calculated at all times. In such cases, a very high age of 999 was used to ensure a score discount that treats the

Table 1
LTR features used for the `Dense-Fusion-Run` and their importance as measured on the training split.

Feature	Category	Description	Gain
`bm25-score`	Lexical	Original BM25 score of the document for the query.	3558.02
`cos-score`	Semantic	Cosine similarity between query and document embeddings.	5835.59
`query-len`	Query	Length of the query in characters.	568.34
`doc-len`	Document	Length of the document in characters.	1376.88
`term-overlap`	Query-Document	Fraction of query terms in the abstract.	977.18
`title-match`	Query-Document	Fraction of query terms in the title.	1263.47
`recency`	Time	Age of the document in months.	7028.65
`log-citation-count`	Citation	Log of the absolute citation count.	793.68

age and completeness of the metadata record as an additional quality indicator of the publication. The citation count of a publication is also used as an LTR feature, where it is log-transformed using $\ln(1 + c)$ to mitigate skewness and ensure numerical stability. A comprehensive overview of features is provided in Table 1.

The document and query embeddings were created with Sentence-Transformers [7] and the all-MiniLM-L6-v2 model [8]. LambdaRank is implemented with LightGBM and nDCG@10 optimization and early stopping to prevent overfitting. In the final inference stage, the system combines the original BM25 score with the new re-ranked score from the LTR model through a weighted linear combination controlled by an α value. The ranking model was trained on the training split of the LongEval-Sci-2026 test collection, which was split again into a train-test split (70/30). This was also used to determine the best α value of 0.4, which slightly emphasizes the original BM25 score. Across all tested values, the scores were rather similar and only a slight drop from 0.4 to 0.5 was measured (nDCG@10 of 0.3845 compared to 0.383).

The analysis of feature importance on the training data using the gain metric of LightGBM showed that the `recency` feature (7028.65) and the `cos-score` feature (5835.59) are the most important predictors of document relevance, indicating that temporal and semantic signals provide more utility than traditional `bm25-score` (3558.02) within this specific temporal collection. The `log-citation-count` had among the lowest importance (793.68). Table 1 provides a full overview.

2.2. Submitted Runs

In total, both teams submitted four runs. The team Time4Citation submitted the three boost configurations and the team DenseFusion submitted the best approach, including all features.

Timely-Boost-V2 The run applies the recency boost to promote publications that were recently cited by 2.5 and publications that were ever cited by 1.5.

Log-boost-03 This run applies the citation boost to promote publications with more citations. The natural logarithm of the citation count of a document is used as a boost. The boost is controlled by an alpha value of 0.3.

binary-boost-09 This run uses the citation information as a binary quality signal by boosting a document if it contains at least one citation by adding 0.9 on top of the baseline BM25 score.

Dense-Fusion-Run This run used the LTR system with all features, trained on the LongEval-Sci collection from 2026. The LTR re-ranker and the BM25 score are weighted by an α value of 0.4.

2.3. Results

The official results on the test snapshots are presented in Table 2 and contrasted with the official BM25 and Qwen3 baselines. All results are measured by the Document Click Through Rate (DCTR) qrels set with nDCG@10. On these qrels, BM25 outperforms Qwen3, most likely due to the similarity of BM25 to the original CORE system. Overall, the `Dense-Fusion-Run` achieves slightly improved effectiveness. The boosting approaches do not always outperform Qwen3 and only sometimes outperform BM25. The `Dense-Fusion-Run` achieves the highest effectiveness on snapshots 1 and 2 and is only marginally weaker at snapshot 3 compared to `Log-boost-03` (0.170 vs. 0.171). The `Log-boost-03` approach achieves the highest effectiveness among the boosting approaches of team Time4Citation, followed by the `binary-boost-09` approach. Only the `Log-boost-03` approach outperforms the BM25 baseline in the later snapshots 2 and 3. These results show that the citation information can improve the effectiveness, but the recency boost alone is not a clear relevance signal in this setting.

The robustness is assessed by the Relative Change (RC) [9, 10, 11] and the Delta Relative Improvement (DRI) [12, 13]. An RC of 0 means that the effectiveness stays the same, while a negative RC indicates improving effectiveness. For all systems, the effectiveness deteriorates. Both citation-based boosts show a lower RC compared to all other approaches and baselines. The approach `Log-boost-03` has the lowest change of 0.069 in the second snapshot. In the third snapshot, `Timely-Boost-V2` outperforms all other approaches. DRI measures the delta between the effectiveness of the experimental system and a Pivot baseline at both snapshots and compares these deltas subsequently across time. Similar to the RC before, a perfectly robust DRI would yield a score of 0, meaning that the difference between the experimental and the Pivot system is the same at the later snapshot. If DRI > 0, the improvement over the Pivot system is decreased, and if DRI < 0, it is increased. This measure is especially interesting for dynamic approaches that change over time. The `Dense-Fusion-Run` shows a generally low DRI, and the lowest in the second snapshot (-0.045). Potentially, this is the case because the LTR model is trained only once and remains static across subsequent snapshots. The boosting approaches of the team Time4Citation show a stronger increasing effectiveness compared to the baseline (cf. the lower DRI). This can be attributed to the weak effectiveness at snapshot one and the accumulating citation data over the later snapshots that influence the strength of the boost.

2.4. Conclusion

In total, two teams submitted four runs to the LongEval 2026 Ad-Hoc Scientific Retrieval task. The citation data is used as an LTR feature and direct ranking signal. Furthermore, it is used in conjunction with the publication date as a recency feature. The LTR approach achieved the highest effectiveness, outperforming both baselines on the DCTR qrels-set. The boosting approaches could outperform the baselines only in the last snapshot. Future work should investigate how retraining the LTR approach on the new click data affects the retrieval effectiveness, and especially its robustness. Regarding the citation data, a meaningful benefit could be observed only in the later snapshot. A deeper analysis of the citation data provided could provide more insights into this and potentially what other citation-based indicators could be useful relevance signals.

3. LongEval-TopEx: Topic Extraction From Query Logs

Task 2 required participants to formalize retrieval topics based on the queries from Task 1. Previous work demonstrated that synthetically formalized information needs can enhance synthetic relevance judgments when formal information needs are absent Keller et al. [14]. While applicable to the LongEval setting, where queries are sourced from the CORE search engine logs, the extent to which these synthetic judgments correlate with click-based or human labels remains an open research question. The submitted approach aims to improve the efficiency of the topic formalization by relying solely on the logged query and important keywords as generation context. The generated topics were also used to predict the next query in the LongEval-USim task.

Table 2
Evaluation for Task 1 of LongEval. We report the Average Retrieval Performance (ARP), Relative Change (RC), and Delta Relative Improvement (DRI) as primary measures. The effectiveness is measured by nDCG@10 for based on the DCTR qrels.

	ARP			RC		DRI	
Snapshot	1	2	3	2	3	2	3
baseline-bm25	0.279	0.254	0.150	0.089	0.462	-	-
baseline-qwen3-4b	0.234	0.201	0.133	0.139	0.432	0.047	-0.047
Dense-Fusion-Run	0.326	0.274	0.182	0.158	0.441	0.089	-0.045
Log-boost-03	0.263	0.253	0.182	0.038	0.305	-0.052	-0.274
Timely-Boost-V2	0.130	0.106	0.108	0.184	0.169	0.049	-0.254
binary-boost-09	0.257	0.241	0.168	0.064	0.345	-0.025	-0.200

3.1. Approaches and Implementation

Team JOINorDIE This team submitted a total of six runs for Task 2. A selection of in-context prompting strategies and language models was evaluated for the task. The proposed approaches primarily relied on pseudo-relevance feedback, selecting either the top-3 ranked documents or documents ranked between positions 100 and 102 in a BM25-based PyTerrier index using the provided queries. The titles and abstracts of the retrieved documents were used to compute TF-IDF scores, from which the five highest-ranked terms were extracted and incorporated into the prompts as positive, negative, or contrastive feedback signals for topic generation. The decision to use five terms was based on preliminary experiments indicating that this configuration achieved the best performance. For the topic generation step, a locally hosted llama3.1:8b model and GPT-4o mini were tested, enabling a comparison between open-source and closed-source approaches.

3.2. Results

The application of synthetic retrieval topics is multifaceted, and so are the dimensions they need to be evaluated on. For this preliminary evaluation, we focus on the alignment of the synthetic judgments created by using the topics to the click-based DCTR qrels. Table 3 provides a comprehensive overview together with the baseline that only uses the logged query.

The results indicate a defective Cohen's Kappa correlation between both qrel types across all submitted approaches. The differences between approaches are low and barely distinguishable. On average, across judging Large Language Models (LLMs), `openai-rel-tfidf5` achieves the highest correlation (0.007) and `ollama-rel-5tfidf` achieves the lowest score (-0.012). Regarding the rank correlations based on Tau-AP, the average correlation across all assessor LLMs is moderate and positive. It ranges between 0.42 for `openai-cont-tfidf5` and 0.5 for `ollama-non-rel-5tfid`. Compared to the baseline, similar results could be achieved. Since the baseline also shows only a weak agreement, the results likely indicate the difference between clicks and relevance labels. While a clicking decision at core is based on the presented title and the short snippet, it does not directly indicate relevance as judged by the LLMs on the full abstracts.

4. LongEval-USim: User Simulation

For Task 3, the objective was to predict the final query of a search session. Participants were provided with the session history, including query timestamps, the top-10 search results (SERP) for each query, and user interactions with the retrieved documents, including click types. Sessions from Timeframe 1 (hereafter referred to as Snapshot 1) served as the training data, with the final query provided as ground truth. For Timeframes 2 and 3 (Snapshots 2 and 3), the final query was withheld by the organizers and had to be predicted by the participating systems. Three teams submitted runs for Subtask 3, exploring

Table 3
The alignment between synthetic labels using the generated topics to the DCTR labels (Cohens Kappa) and between the system rankings using both qrel types (Tau-AP).

Method	Cohens Kappa					Tau-AP				
	oss-120b	oss-20b	llama-3	qwen3	mean	oss-120b	oss-20b	llama-3	qwen3	mean
ollama-non-rel-5tfid	-0.019	-0.026	-0.007	0.019	-0.008	0.393	0.497	0.502	0.605	0.499
openai-norel-tfidf5	-0.011	-0.014	0.011	-0.009	-0.006	0.413	0.398	0.474	0.614	0.475
Baseline	-0.008	0.010	-0.018	-0.007	-0.006	0.432	0.494	0.503	0.458	0.472
ollama-rel-5tfidf	-0.008	-0.021	-0.008	-0.014	-0.012	0.446	0.403	0.452	0.518	0.455
o-contrastive-5tfidf	0.008	-0.027	0.037	-0.002	0.004	0.408	0.445	0.509	0.412	0.443
openai-rel-tfidf5	0.031	-0.011	-0.017	0.026	0.007	0.365	0.373	0.551	0.419	0.427
openai-cont-tfidf5	-0.003	0.044	-0.014	-0.002	0.006	0.393	0.445	0.419	0.424	0.420

different persona-based and reasoning-based approaches, as well as leveraging the topic generation pipeline developed for Task 2 to provide these topics as context for the next-query prediction.

4.1. Approaches and Implementation

Team Promptly The main focus of team Promptly was on testing different reasoning strategies, such as Chain-of-Though and Self-Refinement, in combination with prompting strategies that incorporate a persona description into the prompt. In total, six runs were submitted that combine the reasoning strategies iteratively and show the effect of the persona. Three runs were done with and three without the persona. In order to evaluate the effectiveness of the different approaches, six runs (cf. Table 4) were created to test the various setups. The experiments were conducted using a self-hosted Llama3 [15] model via Ollama.[2] The runs can be differentiated between runs 1 - 3, which are variations of the baseline and do not differentiate in the persona, and runs 4 - 6, which use the NLP Tagging to narrow down specific characteristics of the underlying persona in the prompt to make the predicted queries fit better in the session.

The NLP-based user context was implemented to differentiate sessions into two base personas and to assign a base-level classification to each user session in order to better characterize the user's search expertise. This classification distinguishes between an *Advanced Searcher* and a *Non-Advanced Searcher*.

The assignment of the base persona is based on the use of structured query elements such as facets and Boolean operators (e.g., fieldsOfStudy or yearPublished). The presence of such structured filtering behavior is interpreted as an indicator of more advanced search expertise. The underlying assumption is that advanced searchers have a clearer understanding of their information need or may engage in known-item search tasks, enabling them to precisely locate relevant documents by systematically narrowing down the search space using filters.

This definition of the base persona is inspired by session data from Kruff et al. [16], which were used as a foundation for the initial training phase.

Behavioral characteristics used for the NLP-Tagging approach are derived from multiple interaction signals within a session. Query length is used as an indicator of search style: longer queries are associated with a *Conversational* behavior, whereas shorter queries indicate a more keyword-based query formulation pattern. The underlying assumption is that this directly affects the next query, with conversational users being expected to produce longer, more natural-language-like follow-up queries, while keyword-based users are expected to continue issuing short and highly condensed queries.

In addition, the number of queries per session is used to estimate the depth of the search process. Sessions containing four or more queries are classified as *Deep Explorers*, while shorter sessions are labeled as *Quick Fact-Finders*. The assumption here is that Deep Explorers tend to progressively refine their next queries by incorporating additional context and terminology over time, whereas Quick Fact-Finders are expected to converge early and generate only minimal or no further query refinements.

[2]https://github.com/ollama/ollama

The frustration signal distinguishes between users who are able to successfully refine their search and those who are not. Users issuing more than three queries without any document clicks are classified as *Frustrated*, whereas sessions containing at least one click are considered *Successful*. The assumption is that frustrated users are more likely to generate increasingly divergent or unstable next queries due to a lack of satisfactory results, while successful users tend to produce more stable and progressively refined follow-up queries after receiving relevant feedback.

Another dimension captures the level of user deliberation during the search process. The average time between query reformulations is used as a proxy for reading and reflection behavior, where short inter-query intervals indicate *Rapid Skimmers*, while longer intervals are associated with *Deep Readers*. The assumption is that this temporal behavior influences the next query formulation, with rapid skimmers expected to make only minor, incremental adjustments, while deep readers are expected to generate more substantial, content-driven changes based on information extracted from previously examined documents.

The combination of these behavioral tags results in a total of 2×3^4 possible persona configurations.

1. For the Baseline run the prompt was provided with the history of the session, meaning that all previous queries were given as context for the LLM to get an understanding of the query style of the user. This configuration corresponds to the History (Baseline) and History + NLP Persona settings in Table 4, and to the runs Promptly_simple-baseline-history-only and Promptly_nlp-persona-history-only reported in later tables.
2. For the Chain-of-Thought (CoT) approach, all queries within a session were provided to the LLM. The model was explicitly encouraged to reason about the common characteristics shared by the previous queries and to leverage these insights for predicting the subsequent query. This configuration corresponds to the History + Hidden CoT and History + NLP Persona + Hidden CoT settings in Table 4, and to the runs Promptly_baseline-cot and Promptly_nlp-persona-cot reported in later tables.
3. In the self-refinement approach, the LLM was tasked with critically reviewing both the reasoning produced by the Chain-of-Thought (CoT) process and the output of the previous run. The model could either validate the prior reasoning or identify weaknesses in it, thereby enabling improved predictions or alternative reasoning strategies. This configuration corresponds to the History + Hidden CoT + Self-Refinement and History + NLP Persona + Hidden CoT + Self-Refinement settings in Table 4, and to the runs Promptly_baseline-cot-refine and Promptly_nlp-persona-cot-refinement reported in later tables.

Team Split n' Simulate Originally, the team aimed to manually identify user characteristics from the provided session files in order to derive a set of personas using a rule-based approach. Based on these characteristics, a classifier was intended to assign the most appropriate persona to each session. However, this initial approach relied primarily on keyword matching within the session strings, which resulted in limited persona coverage. It was found that only 42% of the sessions could be assigned to one of the predefined personas (cf. Table 6, not all versions of the first run were covered, 32 sessions were not covered with annotations).

To address these limitations, the manual rule-based approach was replaced by an LLM-based classifier. For this purpose, Llama3.2:3b was used to classify the users associated with each session into one of seven predefined personas. The LLM was provided with descriptions of the characteristics of the seven predefined personas and was instructed to determine which persona best matched the underlying session based on these descriptions. Whenever the model was unable to confidently assign a session to any specific persona, a fallback category, namely the general researcher, was assigned instead. The resulting label distribution and the defined personas are shown in Table 5. The general researcher persona remained available as a fallback option in the final run, but it was not assigned to any session, as the predefined personas in the prompt were sufficiently well-defined to cover all cases.

The overall methodology was iteratively refined throughout the development process. Initially, sessions were classified in batches of ten, with the assumption that presenting multiple diverse ses-

sions simultaneously would help the model better distinguish between the available persona labels. Versions one and two differed mainly in prompt design, while the underlying batch-classification strategy remained unchanged. However, this setup still produced sessions without persona annotations. Consequently, the approach was modified to classify sessions individually, ultimately resulting in 100% label coverage.

The assigned persona was provided to the LLM in order to predict the last query of the session. The assigned persona was provided to the LLM in order to predict the last query of the session. For the assigned persona, the LLM was also supplied with writing characteristics of the underlying persona, as defined by the team, to enable a more accurate prediction of the next query. The submitted candidates were ranked by the cosine similarity between the candidate query and the second last query in the session, assuming that this query is likely the most similar to the last query in the session.

Team JOINorDIE The team submitted a total of 18 runs for Task 3, with an approach closely related to their submission for Task 2 as described in Section 3. The methodology is based on in-context prompting and employs two different prompt variants. For Prompt A, the model was provided with a TREC-style topic description generated by the LLM based on the user's full query history. In the experimental setup, this generated topic was combined with either the complete query history, the second-to-last query, or the first query of the session to predict the user's next query. Furthermore, three different query reformulation strategies were evaluated within this prompt. For Prompt B, the model received the user's query history together with up to six documents in the contrastive setting or up to three documents in the relevant and non-relevant settings, depending on the respective experimental setting. Each document was represented by its title and abstract, with the combined document content truncated to a maximum of 600 characters. In the relevant setting, these documents were those with which the user had interacted. In the non-relevant setting, the prompt included up to three documents that appeared in the Top-10 retrieval results but with which the user did not interact. The contrastive setting combined both relevant and non-relevant documents. As with Prompt A, the amount of session context varied between using the first query, the second-to-last query, or the complete sequence of previous queries within a session. All runs for both Prompt A and Prompt B were generated using GPT-4o mini.

4.2. Preliminary Results and Discussion

During the implementation phase, each team was free to select its own methods for pre-evaluating performance and identifying potential improvements to further develop and refine their approaches. As a result, the evaluation protocols and measures differed across teams, making the preliminary results not directly or only partially comparable. In the following section, the evaluation measures used by the teams are described, and the corresponding results on the training datasets are reported. The final evaluation scores provided by the organizers were not available at the time of the first submission but will now enable a direct and fair comparison of all submitted approaches under the same evaluation protocol in addition to the preliminary results.

Team Promptly For the pre-evaluation of the submitted approaches, the ground truth queries were compared against either the top-1 candidate aggregated across all sessions or the average score computed over all candidates and sessions. As evaluation metrics, ROUGE-L and BERTScore [17] were selected. ROUGE-L captures the overlap in longest common subsequences between the predicted and the original query and therefore primarily reflects exact token-level matching. To complement this, BERTScore was used to account for semantic similarity between predictions and ground truth queries, thereby capturing meaning-preserving but lexically diverse outputs.

Surprisingly, the experiments show an inverse relationship between prompt architecture complexity and performance in next-query prediction. While more sophisticated prompting strategies such as Chain-of-Thought, Self-Refinement, and persona-based roleplay are often beneficial in generative or reasoning-intensive settings, they consistently degrade performance in this task. As shown in Table 4,

Table 4
Overview of the different run files of Team Promptly. All measures are averaged across sessions.

Approach	Top-1 Rouge-L	Max Rouge-L	Top-1 BERTScore	Max. BERTScore
History (Baseline)	0.3663	0.4827	0.3267	0.4436
History + Hidden CoT	0.3233	0.4273	0.2816	0.4001
History + Hidden CoT + Self-Refinement	0.2946	0.4205	0.2743	0.3868
History + NLP Persona	0.2325	0.3723	0.2006	0.3338
History + NLP Persona + Hidden CoT	0.1898	0.3440	0.1943	0.3139
History + NLP Persona + Hidden CoT + Self-Refinement	0.2298	0.3576	0.1862	0.3063

BERTScore decreases across all evaluated configurations when prompt complexity increases, indicating a systematic drop in semantic similarity to the reference outputs.

This trend is particularly pronounced for persona-based prompts: all runs incorporating personas consistently achieve substantially lower scores than the corresponding baseline runs without personas. This indicates that the additional persona framing systematically degrades performance rather than providing any benefit in this short-text retrieval setting.

An exception to this pattern is observed only in the final persona-based run, where applying Chain-of-Thought followed by Self-Refinement yields a slight improvement in ROUGE-L. However, this gain is isolated and does not translate to BERTScore, which remains lower compared to simpler baselines. Overall, the results suggest that increased prompt complexity tends to hurt rather than help in this task setting, reinforcing the effectiveness of simpler, more direct prompting strategies for next-query prediction.

The observed degradation may also be partially attributable to model scale limitations, as smaller LLMs may not reliably support reasoning or self-reflective prompting strategies such as CoT and Self-Refinement.

Team Split 'n Simulate Classification coverage was significantly reduced across the different iteration steps of the implementation, and the fallback persona *General Researcher* was no longer required in the final setup. While most personas are relatively evenly distributed, the *Student* persona remains dominant with 50.6%. Since no ground truth information about CORE user groups is available, these distributions are difficult to validate.

Performance also varies notably across personas, ranging from an average cosine similarity of 0.3071 for the *Sustainability & Climate Researcher* to 0.4028 for the *Policy & Security Analyst*. This suggests that persona design and prompt formulation have a substantial impact on simulator performance. More fine-grained personas and better knowledge of the underlying user population could further improve the approach. Be aware that for the calculation in Table 5 not all candidates were averaged for every session, but the best was picked and averaged over all sessions.

For evaluation, cosine similarity between the ground-truth query and all generated candidate queries was computed using TF-IDF vector representations and subsequently averaged across candidates and sessions, thereby primarily capturing lexical overlap. In addition, SBERTScore was used to capture semantic similarity. This setup is aligned with Team Promptly, which also combines lexical and semantic measures for pre-evaluation.

Across four iterations, performance improved from 0.2100 to 0.3591, alongside improved persona coverage. While cosine similarity mainly reflects lexical overlap, SBERTScore yields an average of 0.8 in the overall best run, indicating strong semantic alignment with ground truth queries.

Additional experiments with smaller LLMs, such as Phi-3, did not improve performance but instead produced overly verbose predictions. The final setup also incorporated contextual inputs such as SERP result lists, clicked documents, and adjusted temperature to increase output diversity due to redundancy penalties in the task design.

Nevertheless, as shown in Table 6, these modifications slightly decreased participant-defined evaluation measures. This reflects a trade-off inherent in the task, in which redundancy is penalized, encouraging diversity at the expense of marginal performance on the metric.

Table 5
Persona distribution in the training dataset and best-performing configurations based on average cosine similarity between the top-ranked candidate and the ground-truth query.

Persona	Percentage	Avg. Top-1 Cosine Sim.
Student Researcher	50,6 %	0.3589
Engineering & Tech Developer	11,1 %	0.3681
Sustainability & Climate Researcher	9,4 %	0.3071
Healthcare & Life Science Researcher	4,4 %	0.3161
Business & Economics Researcher	12,2 %	0.3784
Policy & Security Analyst	7,8 %	0.4028
Academic Researcher	4,4 %	0.3623

Table 6
Improvements across the different iterations of the approaches of Team Split n' Simulate

Approach	Avg. Cosine	Unknown sessions
Rule-based Classification	0.2100	>30
LLM Based Batch Classification v1	0.3100	22
LLM Based Batch Classification v2	0.3531	31
LLM Based Single Session Classification	0.3591	0
LLM Based Single Session Classification + Model change for prediction	0.2915	0
LLM Based Single Session Classification + Click Information + Temperature	0.3579	0

A key limitation of the approach is the use of relatively small available models, which likely constrained overall performance and the effective use of richer contextual signals.

Team JOINorDIE Team JOINorDIE followed a similar evaluation approach to the other teams by computing similarity between ground-truth and generated candidate queries as described above. Specifically, both average cosine similarity and average SBERT score were used as evaluation measures. Cosine similarity is based on TF-IDF vector representations and therefore primarily captures lexical overlap, whereas SBERT provides a complementary measure that accounts for semantic similarity between queries. This setup aligns with the strategies used by both Team Promptly and Team JOINorDIE, as both rely on a combination of lexical and semantic similarity-based metrics for pre-evaluation, albeit with differences in aggregation and implementation details.

While providing relevant or non-relevant feedback in the form of the top five TF-IDF terms leads to relatively similar SBERT scores, the non-relevant feedback consistently performs slightly better across both prompt variants. However, the most notable observation is that contrastive feedback consistently outperforms both alternative setups across nearly all configurations for Prompt B, with the only exception being the run `openai_first_B_contrastive`, with the non-relevant feedback being the second-best run. This might be explainable due to the fact, that there is only a small amount of click interaction for the provided session, leading to the problem that most relevant feedback configurations will end up with only one document or no context, while the contrastive feedback and the non-relevant will always end up with atleast three documents for context. For Prompt A interestingly the three different prompt variation scenarios show similar results. With the "contrastive" scenario performing best and the "non-relevant" scenario second.

Another striking observation is the large standard deviations for both the SBERT scores and, in particular, the cosine similarity scores. The standard deviations of the cosine similarity are consistently within a similar range as the average scores themselves, indicating that the approaches struggle substantially, depending on the session, to generate predictions with lexical overlap or term-level similarity to the ground-truth queries. Although the standard deviations for SBERT are comparatively lower compared to the actual scores, they still remain relatively high. This suggests that the difficulty is not limited to predicting lexically similar queries, but rather indicates that certain sessions are inherently

Table 7
Overview of the results for JOINorDIE's submission to Task 3

Approach	Cosine Mean	Cosine Std	SBERT Mean	SBERT Std
openai_first_A_scenario_1	0.1606	0.1586	0.3918	0.2559
openai_first_A_scenario_2	0.1579	0.1602	0.3959	0.2534
openai_first_A_scenario_3	**0.1686**	0.1690	0.3965	0.2597
openai_last_A_scenario_1	0.1601	0.1619	0.3941	0.2607
openai_last_A_scenario_2	0.1608	0.1687	0.3944	0.2558
openai_last_A_scenario_3	0.1651	0.1742	0.3952	0.2615
openai_all_A_scenario_1	0.1564	0.1626	0.3885	0.2597
openai_all_A_scenario_2	0.1588	0.1606	0.3932	0.2560
openai_all_A_scenario_3	0.1635	0.1689	**0.3979**	0.2630
openai_first_B_rel	0.1895	0.1929	0.4450	0.2552
openai_first_B_non_rel	0.2000	0.1886	0.4541	0.2516
openai_first_B_contrastive	0.2016	0.1923	0.4480	0.2539
openai_last_B_rel	0.1948	0.1968	0.4477	0.2555
openai_last_B_non_rel	0.2000	0.1868	0.4562	0.2498
openai_last_B_contrastive	0.2106	0.1948	0.4601	0.2483
openai_all_B_rel	0.1889	0.1855	0.4408	0.2507
openai_all_B_non_rel	0.2113	0.1943	0.4601	0.2522
openai_all_B_contrastive	**0.2116**	0.1943	**0.4615**	0.2471

more challenging to predict than others, even on a semantic level.

4.3. Test Results and Discussion

To keep the tables in this section reasonably compact and because these metrics are less informative on their own, the results for Top-1 and best-average cosine similarity as well as SERP overlap are only briefly described in the text. Instead, we focus on the relative change values for RDS, while for cosine similarity and SERP overlap we report the differences between Top-1 and best-of-five performance in order to gain insights into ranking quality and the effectiveness of the applied re-ranking strategies.

Rank-Diversity Score (RDS) Detailed quantitative results for all approaches and snapshots are provided in Table 8, while the following paragraphs summarize the key findings.

Overall, the Prompt B configurations with the entire query history of the JOINorDIE team performed particularly well for the first snapshot. The generated query suggestions exhibited high diversity while simultaneously maintaining a high semantic similarity to the ground truth, resulting in the best RDS scores. Notably, these approaches also relied on GPT-4o mini, the largest language model used across all submitted systems. However, despite their strong initial performance, these approaches experienced a substantial decline in later snapshots, resulting in one of the largest relative performance drops between snapshots 1 and 2 as well as between snapshots 1 and 3.

In contrast, the Split 'n Simulate approach achieved competitive performance already in the first snapshot while maintaining a considerably more stable performance across all snapshots. As a result, it obtained the best relative change scores for both snapshot transitions (1–2 and 1–3). The combination of persona-based query generation and a high sampling temperature appears to have achieved its intended goal of increasing query diversity, which is particularly beneficial for the RDS metric.

Interestingly, the baseline approach submitted by Team Promptly also achieved competitive results for snapshots 1 and 2 despite being comparatively simple. Similar to Split 'n Simulate, it relied on small locally hosted language models, demonstrating that competitive performance does not necessarily require very large models.

As already indicated by the preliminary experiments, the reasoning-based approaches of Team Promptly as well as the NLP tagging approaches consistently underperformed across all snapshots.

Table 8
Effectiveness of the simulators for the three snapshots, measured by RDS (Rank-Diversity Score) and RC (Relative Change) the systems are sorted by the RDS based on the first snapshot in decreasing order. The table only contains the two best and worst runs for every team according to the average RDS score across snapshots to provide a concise overview.

Measures	**RDS**			**RC**	
Run / Snapshot	1	2	3	1-2	1-3
JOINorDIE_openai_all_B_non_rel	**0.5914**[1]	**0.3138**[5]	**0.2171**[2]	0.4690	0.6330
JOINorDIE_openai_all_B_contrastive	**0.5434**[2]	**0.3225**[4]	**0.2135**[3]	0.4070	0.6070
Promptly_simple-baseline-history-only	**0.5389**[3]	**0.3349**[2]	0.1638	0.3790	0.6960
JOINorDIE_openai_all_A_scenario_3	**0.5283**[4]	0.2974	0.1772	0.4370	0.6650
Split n' Simulate_Split-n-Simulate-run	**0.5232**[5]	**0.4368**[1]	**0.2547**[1]	**0.1650**[1]	**0.5130**[1]
JOINorDIE_openai_all_A_scenario_1	0.5152	0.2620	0.1887	0.4910	0.6340
JOINorDIE_openai_all_B_rel	0.5133	0.2805	0.1897	0.4540	0.6300
JOINorDIE_openai_last_A_scenario_3	0.5106	0.2830	0.2018	0.4460	0.6050
JOINorDIE_openai_first_A_scenario_1	0.5080	0.2518	0.1713	0.5040	0.6630
JOINorDIE_openai_all_A_scenario_2	0.5046	0.2734	0.1862	0.4580	0.6310
JOINorDIE_openai_first_A_scenario_3	0.5008	0.3025	0.1710	0.3960	0.6590
JOINorDIE_openai_last_B_non_rel	0.4908	0.3043	0.2063	0.3800	0.5800
JOINorDIE_openai_last_A_scenario_2	0.4898	0.2884	0.2030	0.4110	0.5860
JOINorDIE_openai_last_A_scenario_1	0.4863	0.2842	0.1719	0.4160	0.6470
JOINorDIE_openai_first_A_scenario_2	0.4857	0.2629	0.1896	0.4590	0.6100
JOINorDIE_openai_last_B_contrastive	0.4595	**0.3297**[3]	**0.2071**[4]	**0.2820**[3]	**0.5490**[5]
Promptly_baseline-cot	0.4538	0.1476	0.1278	0.6750	0.7180
Promptly_baseline-cot-refine	0.4505	0.1207	0.1171	0.7320	0.7400
JOINorDIE_openai_first_B_non_rel	0.4387	0.2696	**0.2067**[5]	0.3850	**0.5290**[3]
JOINorDIE_openai_first_B_contrastive	0.4347	0.2755	0.1957	**0.3660**[5]	0.5500
JOINorDIE_openai_first_B_rel	0.4127	0.2492	0.1927	0.3960	**0.5330**[4]
JOINorDIE_openai_last_B_rel	0.4045	0.2936	0.1958	**0.2740**[2]	**0.5160**[2]
Promptly_nlp-persona-cot-refinement	0.3178	0.1494	0.0827	0.5300	0.7400
Promptly_nlp-persona-history-only	0.2909	0.1670	0.0959	0.4260	0.6700
Promptly_nlp-persona-cot	0.2202	0.1539	0.0683	**0.3010**[4]	0.6900

However, an interesting observation is that the performance degradation from snapshot 1 to snapshot 2 was noticeably larger for the baseline reasoning approaches than for their NLP-tagging counterparts. Although both approaches eventually converged to similarly low performance levels in later snapshots, the NLP tags appear to provide slightly better robustness over time.

Average Cosine Similarity Detailed quantitative results for all approaches and snapshots are provided in Table 9, while the following paragraphs summarize the key findings.

For the average cosine similarity metric, the JOINorDIE runs using the non-relevant (Scenario 2 for Prompt A) and contrastive strategies (Scenario 3 for Prompt A) performed well for both Prompt A and Prompt B. In contrast to the RDS results, the performance degradation across snapshots was considerably smaller, placing these approaches consistently among the top five systems for snapshots 2 and 3. This suggests that these approaches are effective at generating semantically similar queries but are penalized by the RDS metric due to their comparatively lower diversity.

The Split 'n Simulate approach again demonstrated strong performance across all snapshots and ranked first for snapshots 2 and 3. A similar trend can be observed for the baseline submission of Team Promptly. Across all cosine-based evaluation metrics (Average, Best, and Top-1 Cosine Similarity), the JOINorDIE configurations all_B_non_rel and all_B_contrastive, together with the Team Promptly baseline and Split 'n Simulate, consistently ranked among the top-performing systems. The remaining JOINorDIE configurations showed greater variability in their rankings.

When comparing the different context configurations of the JOINorDIE system, no clear pattern

Table 9
Simulator effectiveness across three snapshots, measured by average cosine similarity and the Top 1–Best gap as an indicator of ranking quality

Measures	**Avg. Cosine Similarity**			Δ(**Best**, **Top1**)		
Run / Snapshot	1	2	3	1	2	3
JOINorDIE_openai_all_B_non_rel	**0.5654**[1]	**0.4613**[3]	0.3891	**0.08**[2]	**0.07**[4]	0.10
JOINorDIE_openai_all_B_contrastive	**0.5550**[2]	**0.4474**[5]	0.3848	0.10	0.10	0.10
JOINorDIE_openai_last_A_scenario_2	**0.5539**[3]	0.3722	**0.3943**[2]	0.09	0.08	0.09
JOINorDIE_openai_first_A_scenario_2	**0.5537**[4]	0.3673	**0.3893**[5]	0.09	0.07	**0.07**[1]
JOINorDIE_openai_all_A_scenario_2	**0.5508**[5]	0.3702	**0.3909**[4]	0.09	0.08	**0.08**[4]
JOINorDIE_openai_last_B_non_rel	0.5483	**0.4535**[4]	0.3855	**0.08**[4]	0.08	0.10
JOINorDIE_openai_all_A_scenario_3	0.5468	0.3645	0.3840	0.10	**0.07**[3]	0.09
JOINorDIE_openai_first_A_scenario_3	0.5423	0.3664	0.3852	**0.08**[3]	**0.06**[1]	0.09
JOINorDIE_openai_last_A_scenario_3	0.5422	0.3608	0.3885	0.09	**0.07**[5]	0.08
JOINorDIE_openai_all_A_scenario_1	0.5408	0.3626	0.3887	0.09	0.08	**0.08**[5]
JOINorDIE_openai_all_B_rel	0.5397	0.4335	0.3735	0.12	0.11	0.11
JOINorDIE_openai_last_A_scenario_1	0.5393	0.3634	0.3876	0.09	**0.06**[2]	0.08
JOINorDIE_openai_first_A_scenario_1	0.5384	0.3673	0.3846	**0.08**[5]	0.07	**0.07**[3]
Promptly_simple-baseline-history-only	0.5342	**0.4829**[2]	0.3477	0.09	0.11	0.10
Split n' Simulate_Split-n-Simulate-run	0.5303	**0.5028**[1]	**0.4021**[1]	**0.04**[1]	0.08	**0.07**[2]
JOINorDIE_openai_last_B_contrastive	0.5295	0.4421	0.3825	0.10	0.09	0.09
JOINorDIE_openai_first_B_non_rel	0.5169	0.4408	**0.3912**[3]	0.09	0.08	0.10
Promptly_baseline-cot-refine	0.5150	0.3748	0.3305	0.10	0.12	0.10
Promptly_baseline-cot	0.5097	0.3762	0.3328	0.09	0.11	0.15
JOINorDIE_openai_last_B_rel	0.5095	0.4236	0.3690	0.11	0.10	0.10
JOINorDIE_openai_first_B_contrastive	0.5091	0.4281	0.3800	0.09	0.09	0.10
JOINorDIE_openai_first_B_rel	0.4905	0.4144	0.3686	0.10	0.09	0.10
Promptly_nlp-persona-cot-refinement	0.4727	0.3695	0.3227	0.11	0.10	0.10
Promptly_nlp-persona-history-only	0.4628	0.3689	0.3263	0.12	0.11	0.09
Promptly_nlp-persona-cot	0.4453	0.3654	0.3183	0.12	0.10	0.11

emerges regarding the optimal amount of session context. In several cases, even using only the first query of a session produced results that were competitive with configurations using the full session history or only the second-last query.

Measuring both the top-1 performance and the best-of-five similarities allows us to assess how effectively the approaches rank their best candidates at the top position. For most systems, the difference between the Average Best and Average Top-1 scores ranged between approximately 0.08 and 0.12. The only notable exception was the Split 'n Simulate approach for snapshot 1, where the difference was only 0.04. This indicates that its reranking strategy, which ranked candidates based on their cosine similarity to the second-last query of the session, successfully promoted stronger candidates to the top of the ranking. For snapshots 2 and 3, however, this advantage largely disappeared, with the differences becoming comparable to those of the other approaches. Overall, the gap between the Average Best and Average Top-1 scores remained relatively consistent across snapshots for each individual system.

SERP Overlap In addition to RDS and cosine similarity, the organizers provided SERP Overlap as a system-centric evaluation metric that measures the similarity of the search results returned for the generated and ground-truth queries. Overall, the SERP Overlap scores were substantially lower than the semantic similarity scores, particularly for the Average and Top-1 evaluations. Nevertheless, the averaging across the best candidates achieved overlaps exceeding 20% in several cases. This indicates that search engine rankings are considerably more sensitive to lexical differences between queries than semantic similarity measures. The findings regarding the best-performing approaches and their performance across snapshots closely mirror those observed for RDS and cosine similarity, and do not provide many additional insights. Although the absolute SERP Overlap scores were lower, the differences between the Top-1 and Best evaluations were also considerably smaller. Consequently, none

Table 10
Simulator effectiveness across three snapshots, measured by average SERP overlap and the Top 1–Best gap as an indicator of ranking quality

Measures	**Avg. SERP Overlap**			Δ(**Best**, **Top1**)		
Run / Snapshot	1	2	3	1	2	3
JOINorDIE_openai_all_B_non_rel	**0.1149**[1]	**0.0770**[2]	**0.0486**[4]	0.08	0.09	0.04
JOINorDIE_openai_all_B_contrastive	**0.1132**[2]	**0.0770**[3]	**0.0476**[5]	0.09	0.09	0.05
JOINorDIE_openai_first_A_scenario_2	**0.1117**[3]	0.0571	0.0448	0.08	**0.05**[4]	0.04
JOINorDIE_openai_last_A_scenario_2	**0.1114**[4]	0.0573	0.0461	0.08	0.06	0.04
JOINorDIE_openai_all_A_scenario_3	**0.1077**[5]	0.0588	**0.0494**[2]	0.07	0.06	0.04
JOINorDIE_openai_all_B_rel	0.1062	0.0709	0.0461	0.10	0.09	0.05
Split n' Simulate_Split-n-Simulate-run	0.1051	**0.0951**[1]	**0.0489**[3]	**0.04**[1]	0.10	0.04
JOINorDIE_openai_first_A_scenario_3	0.1045	0.0541	0.0473	0.07	0.05	0.04
JOINorDIE_openai_all_A_scenario_2	0.1045	0.0575	0.0463	0.07	0.06	0.04
JOINorDIE_openai_last_A_scenario_3	0.1035	0.0527	0.0476	**0.06**[4]	**0.05**[5]	**0.03**[2]
JOINorDIE_openai_last_A_scenario_1	0.1024	0.0549	0.0460	**0.06**[3]	**0.04**[2]	**0.02**[1]
JOINorDIE_openai_all_A_scenario_1	0.1022	0.0576	0.0463	**0.06**[2]	0.06	0.04
JOINorDIE_openai_first_A_scenario_1	0.1003	0.0577	0.0438	**0.06**[5]	**0.04**[1]	**0.03**[3]
JOINorDIE_openai_first_B_non_rel	0.0959	0.0683	0.0472	0.07	0.07	0.04
Promptly_simple-baseline-history-only	0.0945	**0.0737**[4]	0.0298	0.07	0.10	0.04
JOINorDIE_openai_last_B_non_rel	0.0906	**0.0737**[5]	**0.0509**[1]	0.07	0.07	0.05
Promptly_baseline-cot-refine	0.0875	0.0399	0.0238	0.09	0.05	**0.03**[4]
JOINorDIE_openai_first_B_contrastive	0.0871	0.0710	0.0456	0.06	0.07	0.04
JOINorDIE_openai_first_B_rel	0.0859	0.0602	0.0426	0.08	0.05	0.05
JOINorDIE_openai_last_B_contrastive	0.0837	0.0715	0.0467	0.08	0.07	0.04
Promptly_baseline-cot	0.0836	0.0319	0.0256	0.06	0.05	0.04
JOINorDIE_openai_last_B_rel	0.0824	0.0680	0.0456	0.07	0.08	0.05
Promptly_nlp-persona-cot-refinement	0.0660	0.0351	0.0245	0.07	0.05	0.03
Promptly_nlp-persona-history-only	0.0556	0.0365	0.0218	0.07	0.05	0.03
Promptly_nlp-persona-cot	0.0469	0.0355	0.0184	0.08	**0.04**[3]	**0.03**[5]

of the approaches demonstrated a particularly effective reranking strategy with respect to this metric. Detailed quantitative results for all approaches and snapshots are provided in Table 10.

While Split 'n Simulate explicitly optimized its reranking based on cosine similarity between the generated candidate and the second-last session query, this optimization did not translate into improved SERP Overlap rankings.

Unlike Split 'n Simulate, the remaining teams did not employ a dedicated reranking strategy. Consequently, their Top-1 rankings directly reflect the order in which the queries were generated, making the differences between the Top-1 and Best evaluations difficult to interpret in terms of reranking quality.

4.4. Conclusion for Task 3

The observations made during the preliminary analysis largely hold true. Nevertheless, the preliminary experiments remain valuable, as they provide insight into how the different teams iteratively optimized their approaches throughout the development phase and how individual design decisions influenced the final systems.

The official results show that simpler approaches often generalized more robustly than more complex reasoning-based methods. For Team Promptly, the history-based baseline (without personas) consistently outperformed the more sophisticated reasoning-based strategies, which generally resulted in a noticeable decrease in performance. For Team Split 'n Simulate, the persona-based simulation approach achieved the strongest overall results, suggesting that modeling user personas at a sufficiently fine-grained level can effectively improve query prediction. In contrast, Team Promptly's NLP-based persona tagging approach consistently underperformed compared to its simpler baseline, indicating that the automatically assigned persona tags were either not sufficiently discriminative or not well

aligned with the underlying prediction task.

The results of Team JOINorDIE further reinforce findings from previous work, particularly regarding the importance of prompt engineering. Across most configurations, providing the full query history consistently yielded the strongest performance compared to using only partial session context. However, the official results also reveal that these approaches generalized less effectively over time. While several JOINorDIE configurations achieved the best overall performance on the first snapshot, their performance decreased considerably for snapshots 2 and 3. In contrast, the comparatively simple baseline approach of Team Promptly and the persona-based Split 'n Simulate approach maintained a much more stable level of performance across all snapshots, resulting in stronger overall generalization.

Another interesting observation concerns the reranking of generated candidates. Team Split 'n Simulate was the only submission that incorporated an explicit reranking step, ranking generated candidates based on their cosine similarity to the second-last query in the session. While this strategy slightly improved the Top-1 cosine similarity for the first snapshot by promoting stronger candidates to the top of the ranking, its benefit largely disappeared for the later snapshots and did not translate into improved SERP Overlap scores. Consequently, this particular reranking component did not provide a substantial overall advantage. The remaining teams did not employ any dedicated reranking strategy, making direct comparisons of the gap between their Top-1 and Best candidate results difficult to interpret, as their candidate ordering simply reflects the order produced during generation rather than an explicitly optimized ranking.

Acknowledgments

This work is partially funded by the Deutsche Forschungsgemeinschaft (DFG) (grant numbers 407518790 and 509543643), the NFDIxCS grant numbers 501930651, and by the German Federal Ministry of Education and Research (BMBF) and the Joint Science Conference (GWK) through the FH-Personal funding program (PLan CV, reference number 03FHP109).

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly for Grammar and spell checking, ChatGPT for paraphrasing and rewording, and Gemini 3.1 pro for formatting assistance. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] S. Bouaraba, T. Breuer, M. Cancellieri, A. El-Ebshihy, M. Fröbe, P. Galuscáková, L. Goeuriot, G. Iturra-Bocaz, J. Keller, P. Knoth, A. K. Kruff, P. Mulhem, F. Piroi, D. Pride, P. Schaer, D. Schwab, Extended overview of the clef 2026 longeval lab on longitudinal evaluation of model performance, in: E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, 2026.

[2] S. Bouaraba, T. Breuer, M. Cancellieri, A. El-Ebshihy, M. Fröbe, P. Galuscáková, L. Goeuriot, G. Iturra-Bocaz, J. Keller, P. Knoth, A. K. Kruff, P. Mulhem, F. Piroi, D. Pride, P. Schaer, D. Schwab, Overview of the clef 2026 longeval lab on longitudinal evaluation of model performance, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, S. MacAvaney, J. M. Struß, E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Lecture Notes in Computer Science (LNCS), Springer, 2026.

[3] C. Macdonald, N. Tonellotto, Declarative experimentation in information retrieval using pyterrier, in: K. Balog, V. Setty, C. Lioma, Y. Liu, M. Zhang, K. Berberich (Eds.), ICTIR '20: The 2020 ACM SIGIR International Conference on the Theory of Information Retrieval, Virtual Event, Norway,

September 14-17, 2020, ACM, 2020, pp. 161–168. URL: https://doi.org/10.1145/3409256.3409829. doi:10.1145/3409256.3409829.

[4] J. Keller, M. Fröbe, G. Hendriksen, D. Alexander, M. Potthast, P. Schaer, Simplified longitudinal retrieval experiments: A case study on query expansion and document boosting, in: J. Carrillo-de-Albornoz, A. G. S. de Herrera, J. Gonzalo, L. Plaza, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction - 16th International Conference of the CLEF Association, CLEF 2025, Madrid, Spain, September 9-12, 2025, Proceedings, Lecture Notes in Computer Science, Springer, 2025, pp. 117–127. URL: https://doi.org/10.1007/978-3-032-04354-2_8. doi:10.1007/978-3-032-04354-2_8.

[5] S. MacAvaney, A. Yates, S. Feldman, D. Downey, A. Cohan, N. Goharian, Simplified data wrangling with ir_datasets, in: F. Diaz, C. Shah, T. Suel, P. Castells, R. Jones, T. Sakai (Eds.), SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021, ACM, 2021, pp. 2429–2436. URL: https://doi.org/10.1145/3404835.3463254. doi:10.1145/3404835.3463254.

[6] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T. Liu, Lightgbm: A highly efficient gradient boosting decision tree, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 3146–3154. URL: https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

[7] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, Association for Computational Linguistics, 2019, pp. 3980–3990. URL: https://doi.org/10.18653/v1/D19-1410. doi:10.18653/V1/D19-1410.

[8] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, M. Zhou, Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[9] R. Alkhalifa, I. M. Bilal, H. Borkakoty, J. Camacho-Collados, R. Deveaud, A. El-Ebshihy, L. E. Anke, G. G. Sáez, P. Galuscáková, L. Goeuriot, E. Kochkina, M. Liakata, D. Loureiro, P. Mulhem, F. Piroi, M. Popel, C. Servan, H. T. Madabushi, A. Zubiaga, Overview of the CLEF-2023 longeval lab on longitudinal evaluation of model performance, in: CLEF, volume 14163 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 440–458.

[10] R. Alkhalifa, H. Borkakoty, R. Deveaud, A. El-Ebshihy, L. E. Anke, T. Fink, P. Galuscáková, G. G. Sáez, L. Goeuriot, D. Iommi, M. Liakata, H. T. Madabushi, P. Medina-Alias, P. Mulhem, F. Piroi, M. Popel, A. Zubiaga, Overview of the CLEF 2024 longeval lab on longitudinal evaluation of model performance, in: CLEF (2), volume 14959 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 208–230.

[11] M. Cancellieri, A. El-Ebshihy, T. Fink, M. Fröbe, P. Galuscáková, G. G. Sáez, L. Goeuriot, D. Iommi, J. Keller, P. Knoth, P. Mulhem, F. Piroi, D. Pride, P. Schaer, Longeval at CLEF 2025: Longitudinal evaluation of IR systems on web and scientific data, in: J. Carrillo-de-Albornoz, A. G. S. de Herrera, J. Gonzalo, L. Plaza, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction - 16th International Conference of the CLEF Association, CLEF 2025, Madrid, Spain, September 9-12, 2025, Proceedings, volume 16089 of *Lecture Notes in Computer Science*, Springer, 2025, pp. 363–387. URL: https://doi.org/10.1007/978-3-032-04354-2_20. doi:10.1007/978-3-032-04354-2_20.

[12] G. N. G. Sáez, P. Mulhem, L. Goeuriot, Towards the evaluation of information retrieval systems on evolving datasets with pivot systems, in: K. S. Candan, B. Ionescu, L. Goeuriot, B. Larsen, H. Müller,

A. Joly, M. Maistro, F. Piroi, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction - 12th International Conference of the CLEF Association, CLEF 2021, Virtual Event, September 21-24, 2021, Proceedings, volume 12880 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 91–102. doi:10.1007/978-3-030-85251-1_8.

[13] J. Keller, T. Breuer, P. Schaer, Evaluation of temporal change in IR test collections, in: ICTIR, ACM, 2024, pp. 3–13.

[14] J. Keller, M. Fröbe, B. Engelmann, F. Haak, T. Breuer, B. Larsen, P. Schaer, Formalized information needs improve large-language-model relevance judgments, CoRR abs/2604.04140 (2026). URL: https://doi.org/10.48550/arXiv.2604.04140. doi:10.48550/ARXIV.2604.04140. arXiv:2604.04140.

[15] L. Team, The llama 3 herd of models, CoRR abs/2407.21783 (2024). URL: https://doi.org/10.48550/arXiv.2407.21783. doi:10.48550/ARXIV.2407.21783. arXiv:2407.21783.

[16] A. K. Kruff, C. K. Kreutz, T. Breuer, P. Schaer, K. Balog, Sim4ia-bench: A user simulation benchmark suite for next query and utterance prediction, in: R. Campos, A. Jatowt, Y. Lan, M. Aliannejadi, C. Bauer, S. MacAvaney, A. Anand, Z. Ren, S. Verberne, N. Bai, M. Mansoury (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2026, pp. 594–609.

[17] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, Bertscore: Evaluating text generation with BERT, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020. URL: https://openreview.net/forum?id=SkeHuCVFDr.