CIR at LongEval 2025: Exploring Temporal Sensitivity in Web Retrieval

Notebook for the LongEval Lab at CLEF 2025

Florian Braun[1], Timo Busch[1], Muhammed Sirac Coban[1], Maryam El Ghadioui[1], Davit Hovhannisyan[1], Kristine Jonina[1], Andreas Large[1], Felix Zhi Yong Lin[1], Eric Loewenstein[1], Lars Maaßen[1], Nadine Maron[1], Mark Henri Mörsheim[1], Joshua Azimoh Nduka Ofunim[1], Vadims Romanovskis[1], Alexander Simon[1], Jan Witalla[1], Max Wollenberg[1], Jüri Keller[1,*] and Philipp Schaer[1,*]

[1]*TH Köln (University of Applied Sciences), Claudiusstr. 1, Cologne, 50678, Germany*

Abstract

Temporal dynamics in retrieval settings have shown to carry helpful information for retrieval processes. In this submission to the CLEF LongEval lab we propose five different approaches: (1) Finding time-dependent queries with the help of LLMs and to treat these queries differently by boosting their retrieval scores based on the categorization; (2) Finding time-dependent queries and scoring them on a scale from 0 to 1 and to use that score to influence the final ranking; (3) Using relevance information from older sub-collections and to use relevance feedback on the current sub-collection by using query expansion using tf-idf; (4) Boosting known relevant documents-query pairs from older sub-collections but comparing the similarity of old and recent documents; finally, (5) a neural relevance re-ranking based on a topcial semantic clustering. In total we submitted seven runs to the WebRetrieval task of the lab. The results indicate that only four of them could outperform BM25.

Keywords

time-dependent queries, clustering, relevance feedback, similarity, LLM

1. Introduction

The LongEval lab at CLEF is focused on the evaluation of retrieval systems on changing test collections over time. In this lab notebook we summarize our submissions to the CLEF LongEval lab in 2025 that extend on previous work on the lab from 2023 [1] and 2024 [2]. The submissions are the result of a students' project course with the Cologne Information Retrieval group (CIR) at TH Köln - University Applied Sciences in Cologne, Germany. Five groups participated in the course (see Table 1).

2. Approaches and Implementations

The approaches tested in this submission range from (1) finding time-dependent queries with the help of Large Language Models (LLMs) and to treat these queries differently by boosting their retrieval scores based on the categorization (Section 2.1); (2) finding time-dependent queries and scoring them on a scale from 0 to 1 and to use that score to influence the final ranking (Section 2.2); (3) using relevance information from older sub-collections and to use relevance feedback on the current sub-collection by using query expansion using tf-idf (Section 2.3); (4) boosting known relevant documents-query pairs from older sub-collections but comparing the similarity of old and recent documents (Section 2.4); and finally (5) a neural relevance reranking based on a topcial semantic clustering (Section 2.5).

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

*Corresponding author.

jueri.keller@th-koeln.de (J. Keller); philipp.schaer@th-koeln.de (P. Schaer)

0000-0002-9392-8646 (J. Keller); 0000-0002-8817-4632 (P. Schaer)

Table 1
All five team submission included in this lab notebook.

Team name	Submissions and Implementations
CIR_SchaeredRetrieval	Categories of Time-dependent Queries (Section 2.1)
CIR_SuperTeam123	Scoring Time-dependent Queries (Section 2.2)
CIR_Sauerkraut	Time-dependent Relevance Feedback (Section 2.3)
CIR_JMFT	Filtering and Boosting of Document Pairs Across sub-collections (Section 2.4)
CIR_fair_schaer	Neural Relevance Re-ranking supported by Semantic Clustering (Section 2.5)

2.1. Categories of Time-dependent Queries

Queries are not all the same. We know that there are different query types in web search, like transactional, navigational, or informational queries [3]. Additionally, we know from studies on temporal retrieval [4] that users make a difference and most often prefer recent vs. old information, and that there are different temporal entities like events that should be treated differently in the retrieval process. We picked up these ideas and tried to first distinguish different types of temporal queries and use this classification information to apply a boost to recent documents that were ranked for these queries.

We define the following four query types: time-independent, explicit-time, event, and timeliness. We used GPT-4o mini to categorize each LongEval query into one of these categories. We instructed the LLM to categorize the queries using the following definitions:

- time-independent (timeless information not tied to a specific time or event, e.g., definitions, recipes, general rules),
- explicit-time (requests with explicit time references, e.g., years, dates, specific periods),
- event (requests about specific events, e.g., Named public events, Scheduled institutional events, Historical events), or
- timeliness (time-sensitive or current information where up-to-date info or availability matters e.g., weather, stock prices, live updates, buying intent, tax rates).

We instructed the LLM to use the following categorization process: (1) Look for explicit time references (e.g., years, dates). Assign to explicit-time if present. (2) Check for event-related terms. Assign to the event if applicable. (3) If the request requires real-time or current information, assign to timeliness. (4) If the request is timeless and not tied to time or events, assign it to time-independent. The LLM was instructed to only respond with the category name.

Some examples of labels that this categorization process would assign are listed in Table 2. We see that some of the labels are debatable, like the assignment of World War II to the "explicit time" category instead of the "event" category. In a manual test with 100 random queries, we achieved a Cohen's Kappa between 0.33 and 0.42 (two separate annotation runs with GPT-4o mini). The LLM struggled the most with the event and explicit-time categories, where it was hard to find any matching queries. Instead of the default English prompt, we also tried a French prompt, which lowered the agreement rate to 0.24.

Based on the categories assigned by the LLM, we applied a boost to the original BM25 scores. As documents in LongEval don't have a specific timestamp, we applied a simple heuristic: If a document was detected in more than three LongEval sub-collections, it was considered old, and all other documents were considered recent. We then applied the boosting only on the recent document with the following boosting factors:

- time-independent: 1.20
- explicit-time: 1.15
- event: 1.15
- timeliness: 1.20

Table 2
Example labels from the categorization process to annotate time-dependent queries.

categorization	example queries	# queries
time-independent	“definition of gravity”, “chess rules”	23849
explicit-time	“World War II”, “US president 1990”	544
event	“Cannes festival 2025”, “French Revolution”	1146
timeliness	“Apple stock price”, “weather Lyon”	4601

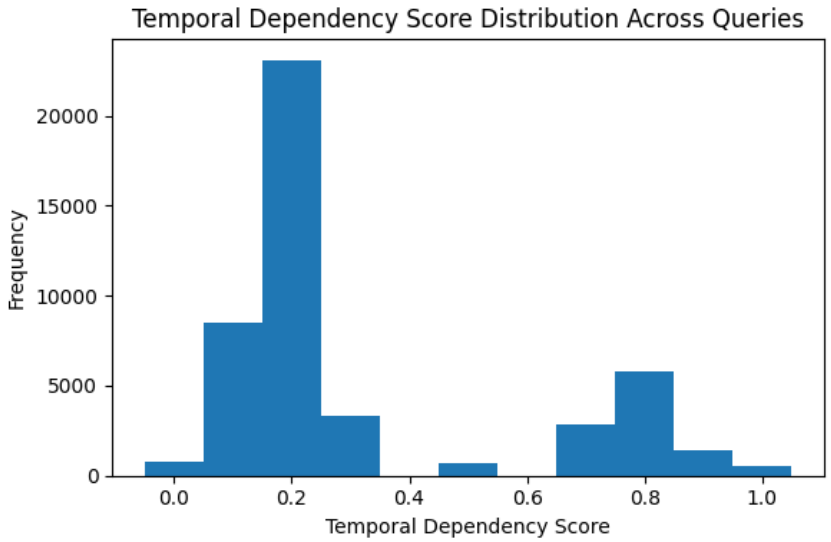

Figure 1: Distribution of the temporal dependency score across all queries.

2.2. Scoring Time-dependent Queries

In contrast to the previous approach, we are only interested in how time-dependent a query is. We prompt an LLM to assign a value between 0 and 1 to every query, encoding the time-dependency of a query.

We prompted GPT-4o with the following instruction: “You have this Query. Give a score on how time-dependent this Query: query_text. The Score is between 0 and 1. Don’t answer with anything more than the Score.” and received a score for 47.053 queries. In Figure 1, we see a plot of the distribution of temporal dependency scores. The average score was 0.325, indicating a clear majority of less time-dependent queries, with only a few time-dependent ones in comparison. Using a threshold of 0.6, only 22% of the queries were marked as time-dependent.

For our re-ranking, we take the BM25 scores based on our PyTerrier implementation with the default parameters for each pair of query q and document d and an additional boost:

$$\text{score\_combined}(q,d) = \text{score\_bm25}(q,d) + \lambda(\text{score\_time}(q) \times \text{score\_recency}(d)) \tag{1}$$

where *score_bm25* is the original BM25 score between q and d and λ is a weight factor for the boost based on the scores *score_time*, and *score_recency*. *score_time* is the temporal dependency factor of the query q as estimated by the LLM. *score_recency* is the recency of document d defined by the frequency of d in the previous snapshots P of the dataset S. It is calculated as follows:

$$\text{score\_recency}(d) = \frac{1}{1 + \log(1 + |\{\text{snapshot}|d \in \text{snapshot AND snapshot} \in P(S)\}|)} \tag{2}$$

Table 3
Time-dependent relevance feedback results based on one sub-collection from the training data (2023-02).

System	bpref	map	ndcg	ndcg@10	p@10
bm25 (2023-02)	0.529	0.223	0.324	0.258	0.057
time-dependent relevance feedback	0.578	0.262	0.366	0.299	0.061

2.3. Time-dependent Relevance Feedback

This submission builds upon our relevance_feedback approach as submitted in 2024 [2], where we used a query expansion method, making use of the relevance feedback provided by prior documents, i.e., those documents with a relevant label at earlier timestamps.

We reimplemented the original pipeline with the following modifications: (1) We removed French and English stopwords, (2) we removed terms with a length of less than 5 characters, (3) we only considered highly relevant (relevance score of 2) documents, and (4) we calculated the tf-idf weight using the whole PyTerrier index data, and not just the candidate documents. We calculated the tf-idf values for each term in the highly relevant documents for each query for all previous sub-collections, and we extracted the term with the highest tf-idf value per document. Up to 8 terms with the highest tf-idf values for each query were used to expand the original query. In most cases, only 2 to 4 terms were used, as for many queries, only a few highly relevant documents from previous sub-collections are available. On the training data, we see an improvement over a simple BM25 baseline (see Table 3).

2.4. Filtering and Boosting of Document Pairs Across sub-collections

The overall idea of this approach is to use relevance information from previous sub-collections by boosting known relevant documents. This approach was already proposed as qrel_boost in our submission 2024 and in 2025 we further refined this approach by including a filter step and a more fine-grained boosting mechanism. In the original approach we boosted all known relevant documents independent of any potential updates. This time we compare old and new documents not only based on the URL but also on the document content itself. It builds on observations from previous studies on pseudo relevance feedback [5]. Only if the old and new document content is the same or similar, we applied a boost on the original BM25 scores.

We developed and implemented three different methods to identify, filter, and boost document pairs that appear in two temporally distinct sub-collections of the LongEval dataset: A length matching approach, a document similarity comparison based on Sentence BERT, and a comparison by Jaccard index. The overarching goal was to recognize relevant document versions that remained stable over time, either structurally or semantically, and to integrate this information into a retrieval pipeline. All methods were grounded in a query-document relevance mapping (`query_doc_map`) derived from the official qrels file, which associates each query with its set of relevant documents. The result was a dictionary assigning each query ID to a standardized list of relevant document IDs, which served as the basis for our comparison strategies over periods. For the submitted approaches, only the qrels from the 2023-03 snapshot were used.

Length Matching The first method aimed to identify document pairs whose text content had the same length in both snapshots. We extracted and compared the text lengths for each document and retained only those pairs where the lengths matched perfectly. This strict filtering approach provided a fast and reliable pipeline for unchanged content, ensuring that only structurally identical document versions were boosted by taking the original BM25 score and multiplying it by two.

Length-Based Similarity with Tolerance Buckets Recognizing that minor formatting changes or metadata updates might alter document length slightly without affecting the core content, we introduced a more flexible filtering scheme based on length ratios. For each document pair, we computed the ratio between the shorter and longer version and classified them into predefined buckets and assigned a boosting factor (see Table 4). Separate filtered mappings were created for each category, allowing for

Table 4
Factors for time-dependent relevance boosting for a given document similarity (based on string length and SBERT similarity).

string length	SBERT similarity	boost factor
100%	100%	2
95-99%	95-99%	1.75
90-94%	90-94%	1.5
80-89%	80-89%	1.25

graded analysis or boosting strategies depending on the degree of length similarity.

Sentence-BERT Similarity (Hard Threshold) To move beyond structural comparison and capture true semantic stability, we employed Sentence-BERT embeddings using the `all-MiniLM-L6-v2` model. Each document version was encoded into a dense vector representation, and cosine similarity was calculated between the two versions. Only those document pairs with a similarity score above a strict threshold of 0.9 were retained and the original BM25 score was boosted with a factor of two. This method enabled us to preserve documents that may differ lexically but convey the same meaning, offering a more context-aware filtering strategy.

Sentence-BERT Similarity with Buckets) Expanding on the previous method, we categorized document pairs into semantic similarity buckets rather than applying a hard cutoff. This allowed us to group documents based on graded similarity levels similar to the string length approach (see Table 4).

Jaccard Similarity As an alternative to embedding-based methods, we also evaluated lexical overlap through Jaccard similarity. By tokenizing and lowercasing the document texts, we computed the ratio of intersecting to union word sets for each document pair. Document pairs exceeding a specified similarity threshold of 0.9 were retained and boosted by a factor of two. This approach was particularly useful for identifying minimal editorial changes.

2.5. Neural Re-ranking Supported by Semantic Clustering

This approach aims to develop a search engine that goes beyond keyword matching and also evaluates search results based on their content. Essentially, this means assigning new documents to thematic clusters. We used machine learning to identify hidden content-related connections between content clusters and their relevance.

First, we carried out a systematic clustering of all queries from the LongEval database. The queries were encoded with the help of OpenAI embeddings (`text-embedding-3-large`) into a 3072-dimensional semantic space. We used Uniform Manifold Approximation and Projection for Dimension (UMAP) to reduce the vectors to 50 dimensions, and then applied k-means clustering [6, 7]. The optimal number of clusters of 56 was determined by silhouette score analysis, resulting in thematically coherent groups (e.g. clusters with terms such as 4: “Job/Employment” or 32: “Food/Ingredients”) [8]. These clusters served as the basis for the subsequent modeling. In Figure 2, we see a visualization of ten high-level clusters from the original 56 topics discovered in the queries.

All relevant document were assigned to one or many different clusters. After pre-processing (lowercasing, normalization, stopword removal, SnowballStemmer for French, and punctuation removal), we extracted term frequencies for the top 10,000 terms and used a multi-hot encoding. For the model itself, we developed a dense neural network with TensorFlow/Keras. Our aim was to use the text content to predict which topic clusters a document fits into and how relevant it is. The model was built as a dual-output network with two separate prediction branches: (1) Cluster prediction: 56-neuron softmax layer for topic classification, and (2) Relevance estimation: sigmoid activation for continuous relevance assessment (0-1). The architecture consisted of three hidden layers ($512 \rightarrow 256 \rightarrow 128$ neurons) with LeakyReLU activation and dropout regularization ($p = 0.3$). The input features were the 10,000-dimensional multi-hot-encoded term vectors. The training was performed on documents until 2023-02 with class weighting to compensate for the relevance imbalance.

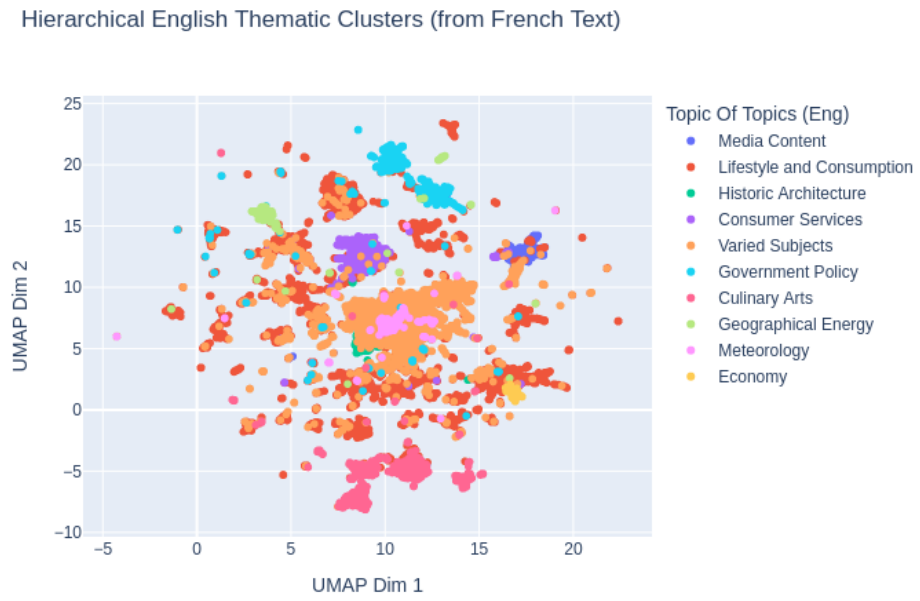

Figure 2: A visualization of ten high-level clusters from the original 56 topics discovered in the original queries.

The actual retrieval was a two step approach: (1) 1000 candidate documents were retrieved using PyTerrier's BM25, (2) then we applied a re-ranking based on the overlap of query and document cluster:

$$\text{score\_combined} = \begin{cases} \text{score\_bm25}(q,d) \times 2 \times \text{sigmoid}(\text{score\_cluster}(d)), & \text{if cluster overlap} \\ \text{score\_bm25}(q,d), & \text{otherwise} \end{cases} \tag{3}$$

where score_cluster is the prediction of the cluster relevance predictor. So, for document that don't have a topical cluster overlap with the queries, we take the original bm25 score, but for overlapping documents we alter the score to enforce a re-reranking. We can implement this process to be trained only on few or only the preceding sub-collection of different sub-collection and therefore a longer time span to enhance the training process.

3. Results

The retrieval effectiveness of the presented approaches was evaluated using the nDCG@10 metric [9], which aligns with the web-search context of this task. We also report the effectiveness of the BM25 [10] baseline, as most systems, excluding the Sauerkraut approach, function as re-rankers applied to this initial retrieval stage. The comprehensive results are depicted in Table 5 and Figure 3.

Among the evaluated systems, the SchaeredRetrieval approach, which boosted known documents based on the temporal type of the query, demonstrated the weakest performance across nearly all snapshots, with nDCG@10 scores ranging from 0.20 to 0.25. Similarly, the timeliness-focused approach by SuperTeam123 performed on par with the BM25 baseline for most snapshots. Although the nDCG@10 scores typically differed by only a few thousandths, a substantial drop occurred at the 2023-04 snapshot, where it recorded the lowest score of 0.192 among all systems and snapshots.

In contrast, the relevance feedback approach was the first to outperform the BM25 baseline. It shows similar performance trends as the baseline but consistently achieved better results. It is the second best unique approach. The JMFT team submitted three variations of their approach: Jaccard, Bert, and length. All of which outperform the baseline and the other approaches. Notably, for the first two

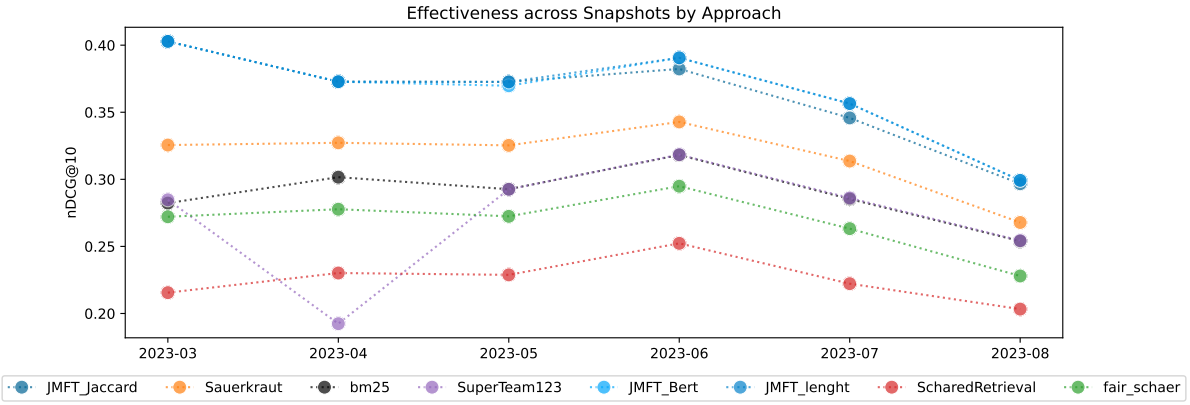

Figure 3: Line plot of the NDCG@10 scores for the different approaches at the different test snapshots. The three variations of the JMFT approach and the Sauerkraut approach outperform the bm25 baseline.

Table 5
NDCG@10 scores for the different approaches at the different test snapshots. The best results per snapshot are highlighted in **bold**.

Approach	2023-03	2023-04	2023-05	2023-06	2023-07	2023-08
bm25	0.282	0.302	0.293	0.318	0.285	0.254
ScharedRetrieval	0.216	0.230	0.229	0.252	0.222	0.203
fair_schaer	0.272	0.278	0.272	0.295	0.263	0.228
SuperTeam123	0.285	0.192	0.293	0.318	0.286	0.254
Sauerkraut	0.326	0.327	0.325	0.343	0.314	0.268
JMFT_Jaccard	**0.403**	**0.373**	**0.373**	0.382	0.346	0.297
JMFT_Bert	**0.403**	**0.373**	0.370	**0.391**	**0.356**	**0.299**
JMFT_lenght	**0.403**	**0.373**	**0.373**	**0.391**	**0.356**	**0.299**

snapshots, all three JMFT approaches yielded identical nDCG@10 scores, with only minor variances observed thereafter.

The final team, fair_schaer, proposed a neural relevance re-ranking model. This approach achieved an effectiveness that positioned it between the BM25 baseline it was designed to re-rank and the ScharedRetrieval system. This approach also did not outperform BM25. Over time, the performance gap between this system and the BM25 baseline narrowed slightly.

Overall, all submitted approaches exhibited broadly similar trends in retrieval effectiveness. A greater variance in performance among the systems was observed in the initial two snapshots when the training data was most recent. This variance diminished in the later snapshots, with the final snapshot showing the least variance between the systems.

4. Conclusion

We proposed five distinct approaches for leveraging temporal information within test collections. While some of these methods are further developments of recent submissions, others are novel and previously untested. Ultimately, only two approaches, relevance feedback and qrel_boosting, managed to outperform the BM25 baseline on the test data. These results confirm, once again, that both are effective strategies for improving retrieval effectiveness at a low computational cost. In contrast, our findings indicate that the timeliness of a query could not yet be successfully utilized as an effective relevance signal.

Acknowledgments

We gratefully acknowledge the support of the German Research Foundation (DFG) through project grant No. 407518790.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] J. Keller, T. Breuer, P. Schaer, Evaluating temporal persistence using replicability measures, in: M. Aliannejadi, G. Faggioli, N. Ferro, M. Vlachos (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023, volume 3497 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 2441–2457. URL: https://ceur-ws.org/Vol-3497/paper-196.pdf.

[2] J. Keller, T. Breuer, P. Schaer, Leveraging prior relevance signals in web search, in: G. Faggioli, N. Ferro, P. Galuscáková, A. G. S. de Herrera (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024), Grenoble, France, 9-12 September, 2024, volume 3740 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 2396–2406. URL: https://ceur-ws.org/Vol-3740/paper-220.pdf.

[3] A. Z. Broder, A taxonomy of web search, SIGIR Forum 36 (2002) 3–10. URL: https://doi.org/10.1145/792550.792552. doi:10.1145/792550.792552.

[4] H. Joho, A. Jatowt, R. Blanco, A survey of temporal web search experience, in: L. Carr, A. H. F. Laender, B. F. Lóscio, I. King, M. Fontoura, D. Vrandecic, L. Aroyo, J. P. M. de Oliveira, F. Lima, E. Wilde (Eds.), 22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume, International World Wide Web Conferences Steering Committee / ACM, 2013, pp. 1101–1108. URL: https://doi.org/10.1145/2487788.2488126. doi:10.1145/2487788.2488126.

[5] T. Breuer, M. Pest, P. Schaer, Evaluating elements of web-based data enrichment for pseudo-relevance feedback retrieval, 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. 53–64. URL: https://doi.org/10.1007/978-3-030-85251-1_5. doi:10.1007/978-3-030-85251-1_5.

[6] L. McInnes, J. Healy, UMAP: uniform manifold approximation and projection for dimension reduction, CoRR abs/1802.03426 (2018). URL: http://arxiv.org/abs/1802.03426. arXiv:1802.03426.

[7] J. B. MacQueen, Some methods for classification and analysis of multivariate observations, University of California Press, 1967, pp. 281–297.

[8] P. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65. URL: https://doi.org/10.1016/0377-0427(87)90125-7. doi:10.1016/0377-0427(87)90125-7.

[9] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, ACM Trans. Inf. Syst. 20 (2002) 422–446. URL: http://doi.acm.org/10.1145/582415.582418. doi:10.1145/582415.582418.

[10] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gatford, Okapi at TREC-3, in: D. K. Harman (Ed.), Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994, volume 500-225 of *NIST Special Publication*, National Institute

of Standards and Technology (NIST), 1994, pp. 109–126. URL: http://trec.nist.gov/pubs/trec3/papers/city.ps.gz.