Agent-based Document Expansion for Information Retrieval based on Topic Modeling of local Information

Oliver Strauß
Fraunhofer Institute for Industrial Engineering IAO Fraunhofer Society
Nobelstraße 12, Stuttgart, Germany
oliver.strauss@iao.fraunhofer.de
ORCID 0000-0003-1421-2744

Damian Kutzias
Fraunhofer Institute for Industrial Engineering IAO Fraunhofer Society
Nobelstraße 12, Stuttgart, Germany
damian.kutzias@iao.fraunhofer.de
ORCID 0000-0002-9114-3132

Holger Kett
Fraunhofer Institute for Industrial Engineering IAO Fraunhofer Society
Nobelstraße 12, Stuttgart, Germany
holger.kett@iao.fraunhofer.de
ORCID 0000-0002-2361-9733

***Abstract*—With the advent of data ecosystems finding information in distributed and federated catalogs and marketplaces becomes more and more important. One of the problems in data search and search in general is the mismatch between the terminology of users and of the searched items, be it dataset metadata or web pages. The paper proposes an agent-based approach to document expansion (ADE). The idea is to represent documents with agents that exploit local information collected from user searches and relevant signals to improve the representation of the document in a search index and subsequently to improve the search performance of the system. The agents collect terms from relevant queries and perform topic modeling on these terms and publish different variants expanded with the topic terms to the search index. We find that the approach achieves good improvement in search performance and is a valuable tool because is places no burden on the information retrieval pipeline and is complementary to other document expansion and information retrieval approaches.**

***Index Terms*—document expansion, agent-based system, information retrieval, topic modeling**

I. Introduction

The proliferation of data from a variety of sources in research and industry enables more and more data-centric services and applications that are increasingly organized in data ecosystems [1]. The International Data Spaces [2] and Gaia-X[1] initiatives are manifestations of this trend. They establish frameworks that allow data producers to share data while preserving control over their assets. Bringing together needs of data users with the offerings of data providers is an important aspect in these data ecosystems. This matching process is performed by data catalogues, data marketplaces and data brokers.

One problem in data search and information retrieval in general is the mismatch between the terminology of used by users in search queries and the language used in the documents (and their metadata) that the user is looking for [3]. While this issue is often addressed by expanding the user query with additional terms and re-ranking documents in post-retrieval, expanding the document representation itself before retrieval has gained attention with the advent of pre-trained transformers like Bidirectional Encoder Representations from Transformers (BERT) [4].

In this paper we propose an approach to iterative document expansion based on local information received from (implicit or exlicit) user feedback that indicates the relevance of a search result for a given query. The idea is to represent each document in a potentially distributed system by an agent, that tries to promote its document in a way that improves its discoverability. Agents that were considered relevant for a given query act locally by adjusting their own presentation to the terminology of the user query.

II. Related Work

The mismatch between the user intent which is expressed in the form of search queries in the users' terms and the documents in a corpus that are represented by their content and additional metadata is a well-known problem in information retrieval [3]. This problem has been addressed in various

This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the Incentives and Economics of Data Sharing Funding Action (IEDS0001). The author is responsible for the content of this publication.

Strauß, Oliver; Kutzias, Damian; Kett, Holger (2022): Agent-based Document Expansion for Information Retrieval based on Topic Modeling of local Information. In: Proceedings of 9th Intl. Conference on Soft Computing & Machine Intelligence (ISCMI 2022), November 26-27, 2022. Toronto, Canada. ISSN: 2640-0146, ISBN: 979-8-3503-2087-9

[1]https://gaia-x.eu/

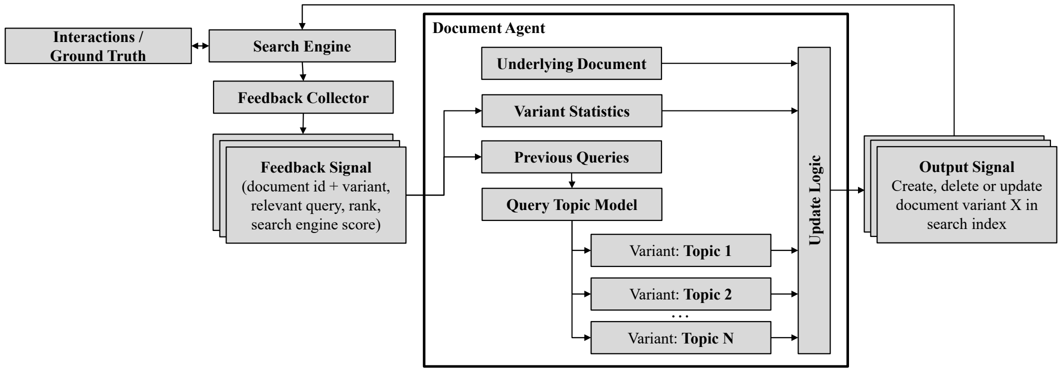

Fig. 1. Components of the agent-based system.

stages of the retrieval process: via document expansion in pre-precessing and query expansion during the retrieval phase.

Recent development of deep learning and transformer-based techniques lead to new methods to expand documents to better align with user queries. In doc2query [5] and its extension docTTTTTquery [6] a pre-trained BERT transformer, that was trained on query-passage pairs is used to derive plain text queries from given document passages. These queries are then added to the document before indexing which leads to an increase in search performance. In DeepCT [7] and its extension Context-aware Hierarchical Document Term weighting (HDCT) [8] a BERT transformer is trained to re-weight the terms of a passage or whole document according to their importance. The document is then altered so that the term frequencies match the calculated weights. The processed documents can be indexed with classic bag-of-words-based search engines.

Another approach that is performed during retrieval is query expansion. Here user queries are enriched with additional terms before the search is executed. The goal of query expansion is to adjust the query to better match the documents. [9] provides a extensive overview of this topic. Document expansion has the operational advantage, that it is decoupled from the query process and can involve computationally expensive computations while query expansion and re-ranking is performed on every query and has to be computed quickly to not degrade query execution times [9].

An intelligent agent is an autonomous component that possesses internal state, interacts with other agents and its environment, makes autonomous decisions based on its state and local environment and can learn and adapt to a changing environment [10]. Agents have been investigated as autonomous (distributed) mechanism in software development for a few decades. They can be seen as socially intelligent, autonomous problem solvers which achieve their objectives by interaction with other similar autonomous entities [11]. Agents have been used for self-organized, local optimization in a broad spectrum of applications such as autonomous network optimization [12] or distributed energy optimization [13].

The term agents is ambiguously used not only for agents optimizing local objectives, but also for software components with different functionalities. For example, [14] use nine modules (agents) for a system using intelligent evolution based on user queries to improve accuracy of the results. [15] survey various approaches that use cognitive agents to support information retrieval in the form of meta-search [16], semantic search or by taking users emotions into account [17]. In the context of this paper, we take the former view and understand agents as actors that operate on local information.

Topic modeling is an unsupervised machine learning method that uses statistics over the terms in documents and their co-occurrences to derive a set of topics that each consist of closely related words. The documents are assumed to be related to a combination of these topics. In this sense topic modeling reduces the dimension of a bag-of-word like document vector whose length corresponds to the size of the vocabulary to a vector on the topic space. This can be exploited for similarity search. The topics themselves are interesting because they distill different semantic aspects of the documents. Established topic modeling algorithms are Latent Semantic Indexing (LSI) [18] and Latent Dirichlet Allocation (LDA) [19]. Special algorithms have been developed for topic modeling on sort texts such as Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM) [20].

III. Agent-based Document Expansion

The paper proposes an agent-based approach to document expansion (ADE). The idea is to represent documents with agents that exploit local information collected from user searches and relevant signals to improve the representation of the document in a search index and subsequently to improve the search performance of the system. The agents collect terms from relevant queries, perform topic modeling on these terms, and publish different variants expanded with the topic terms to the search index.

A. Overview

The goal of the agents is to narrow the gap between the user and document terminology. They improve and adapt the presentation of the underlying document based on local information in the form of user feedback signals (Fig. 1). We assume a simple feedback model: every time a user indicates through her interaction with the search system that a document is relevant for the current query (e.g. by "purchasing" the document), the agent is notified with information about the query and its rank in the result list (cf. section III-B). In this paper we approximate the users' relevance indications by using the relevance judgments provided by the test collection.

Once the agent has received enough feedback signals containing relevant queries, it uses topic modeling (cf. section III-D) to identify sets of mostly orthogonal terms in the collected queries (Fig. 1). The identified topics use terms of the user's terminology, and the intuition is that the identified topics approximate different information needs of users. The identified topic terms are added to the original document to produce a limited number of expanded document variants (cf. section III-C). These variants are then added to the search index and can be retrieved in subsequent queries. By representing the variants explicitly in the search index, the effect of the different expansions can be tracked and directly attributed to a variant. On this basis the agent can decide if unsuccessful variants should be removed or new variant should be created. This comes at the cost of an increase in index size.

The agents in the described system all use the same set of global parameters and compete for the high ranks in the result list. One possible extension is to allow each agent to have its own set of local parameters that are adjusted over time with a reinforcement learning approach. As this would require an extensive amount of user feedback it is not considered here. Another extension is to add cooperation to the model, e.g. by allowing agents to distribute feedback to similar agents and share the rewards. For sake of simplicity cooperation is also not considered in the current model.

B. User Interaction

Each agent receives signals based on user feedback. Users issue search queries and interact with the results. In our experiments we approximate user interaction with the given ground truth of the dataset and assume, that a relevant document has been "purchased" by the user, if it is in the first k elements of the result list $R_k(q)$ for user query q (see left side of Fig. 1). The set of relevant documents for a query q as given by user decisions or by some available ground truth is $F_k(q) \subseteq R_k(q)$.

For each relevant document $D_i \in F_k(q)$ found at rank r a signal $S(T_q, D_i, r)$ is produced that contains the local information collected from the search result: query terms T_q, relevant document D_i, and rank r.

$$q(T_q) \xrightarrow[Engine]{Search} \{S(T_q, D_i, r) \mid D_i \in F_k(q)\} \qquad (1)$$

After a batch of p_{batch} queries have been executed, the signals are collected and sent to the agents whereby a signal $S(T_q, D_x, r)$ is transmitted to the agent representing document D_x. The agents subsequently update their state and take action.

C. Agent Logic

An agent A_D represents a document D from the corpus. It can produce and publish variants $V_i(D, T_v^i)$ of this document to the search system that are copies of D expanded by terms T_v taken from the queries received via the feedback signals S. The expansion is performed by appending the terms T_v p_{boost} times to the document D. The parameter p_{boost} allows to boost the term weights of the expansion terms T_v relative to the terms of D.

Each agent maintains a variable t that is increased by one every time an agent is updated. t plays the role of a local time for the agent. Agents receive new signals $S(T_q, D, r)$ and are updated only when they were considered relevant for a query. The current model does not take negative feedback into account. No signals are created when agents are retrieved but not considered relevant.

After the agent receives a new signal $S(T_q, D, r)$ it checks if D corresponds to one of the document variants $V_i(D, T_v^i)$ and updates the statistics of the variant if this is the case. Each variant stores the number times N_{hit} that it was found as well as the time t_c when it was created. It also maintains a sum of the reciprocal ranks of previous hits. Given the current time t the fitness f of a variant is given by Eq. 2 as the product of the mean reciprocal rank $1/N_{hit} \sum_{i=0}^{N_{hit}} 1/r_i$ and the success rate $N_{hit}/(t - t_c)$ of the variant.

$$f_{Variant}(t) = \frac{1}{t - t_c} \sum_{i=0}^{N_{hit}} 1/r_i \qquad (2)$$

On each update the agent computes the fitness of all variants and keeps the best p_{var} variants while removing the others from the index.

D. Deriving Document Variants from Query Terms

After an agent has received more than p_{new} new query terms, it derives new expansion terms T_t using Latent Semantic Indexing (LSI) [18]. The intuition behind this approach is that users have different information needs that they express through their queries. We hypothesize that by using topic modeling different distinct topics can be distilled from the search terms that have a correspondence with the underlying information needs. Although there are topic modeling approaches tailored to small text corpuses [20], our experiments have shown that LSI provides acceptable results even when used on the small corpus of user query terms.

The LSI algorithm requires the number of topics p_{topics} to be specified in advance. Heuristically p_{topics} is chosen as $floor(\sqrt{N_Q}) + 1$ with N_Q being the total number of query terms collected by the agent so far. The result of the topic modeling run are p_{topics} lists of terms clustered by their co-occurrence in the queries the agent has received. From each list the first p_{terms} terms T_t^i that best define the topic are used to create new variants.

The T_t^i are not automatically turned into new variants since this would lead to new variants being constantly created and removed. For the variants to collect feedback on their fitness, a certain level of stability is required. A variant is only created for T_t, if its terms are sufficiently dissimilar to the existing

TABLE I
RESULTS OF THE EXPERIMENTAL RUNS (BEST RESULTS ARE IN BOLD).

Method	Metrics					
	P@10	*R@10*	*F1@10*	*MAP@10*	*MRR@10*	*nDCG@10*
BM25	0.188	0.117	0.010	0.084	0.406	0.244
BM25 + AQ	**0.252**	**0.163**	0.007	**0.118**	0.471	**0.320**
BM25 + ADE RS	0.228	0.126	0.015	0.093	0.496	0.290
BM25 + ADE TM	0.229	0.125	**0.016**	0.093	**0.499**	0.291

variants' terms T_v^i. The agent uses Jaccard similarity (Eq. 3) to compute the maximal similarity sim_{max} (Eq. 4).

$$sim_{Jaccard}(T_t, T_v) = \frac{|T_t \cap T_v|}{|T_t \cup T_v|} \tag{3}$$

$$sim_{max}(T_t) = max_i \ sim_{Jaccard}(T_t, T_v^i) \tag{4}$$

Only when $sim_{max}(T_t)$ is below some threshold p_{sim} a new variant $V(D, T_t)$ is created. Since new variants initially have low fitness, they are kept for at least three update cycles to have the chance to proof their value. The terms T_t of a newly created variant are finally copied p_{boost} times and appended to the base document D to form $V(D, T_t)$. p_{sim} and p_{boost} are parameters of the model. Finally the variants $V_i(D, T_t^i)$ are added to the search index.

IV. EVALUATION

We have tested the proposed system against data from the NFCorpus dataset [21]. This dataset consists of 9,964 documents from the medical domain that mostly come from PubMed and are written in expert terminology as well as 3,244 natural language queries that are written in non-technical English and have been gathered from the NutritionFacts.org site. The documents and queries are complemented by 169,756 automatically extracted relevance judgments. The divergence in terminology between queries and documents and the relatively high number of relevance judgments per query make it a suitable dataset to test the proposed system against. A split in training (80 %), development (10 %) and test sets (10 %) is provided.

For the documents, the title and abstract fields were concatenated while for queries only the title containing the actual query text was used. Pre-processing consisted of removing punctuation and symbols, tokenization, conversion to lower case, removal of one letter tokens, lematization and removal of stop words using the stop word list provided by NfCorpus. We compared four experiments to evaluate the effectiveness of ADE:

- BM25: We use BM25 [22] as baseline ranking model, which is a commonly used search and ranking method based on term based statistics. In this experiment, the full corpus was indexed using BM25 and the queries and relevance of judgments of the test set where used to calculate the performance metrics.
- BM25 + AQ: As an estimation of the potential upper bound of the effectiveness of ADE we used the queries and relevance of judgments of the training set to collect all relevant queries for each document in the full corpus and appended the query terms of these queries to the respective documents. Evaluation is then performed on the resulting modified corpus using BM25 and the queries and relevance of judgments of the test set.
- BM25 + ADE TM: Evaluation of the agent-based document expansion with topic modeling is performed in two phases: first an agent is initialized for every document in the full corpus and the initial BM25-index is created. Then the queries from the training set are split into batches of size p_{batch} and executed against the index. Using the relevance judgments of the training set, feedback signals are produced for the $k = 100$ top ranked relevant document variants in each query to to model a good willed user that always examines the first 100 hits. Here a document variant is considered relevant if the underlying document is relevant according to the relevance judgments. After all queries in the batch are executed, the agents update is triggered and the agents use topic modeling as described in sections III-C and III-D to derive new document variants that are added to the BM25-index. After all batches are processed the expanded index is evaluated against the queries and relevance of judgments of the test set. Here only best ranked variant of each document is counted, and all lower ranked variants of the same document were ignored.
- BM25 + ADE RS: To explore the effect of topic modeling this experiment is performed exactly as BM25 + ADE TM with the difference that the variants are created using uniform random samples without replacement drawn from the query terms available to the agent instead of topic modeling.

The following metrics were collected using the ranx library (version 0.2.12) [23]: precision, recall, f1, mean reciprocal rank (mrr), mean average precision (map) and normalized discounted cumulative gain. We chose a small value of $k = 10$ to measure the improvement, a user would see on the first page of rearch results. For BM25 + ADE TM the hyper parameters were tuned by hand. The following parameter values were used: p_{batch} = 500, p_{var} = 5, p_{new} = 5, p_{topics} = 2, p_{terms} = 7, p_{sim} = 0.4 and p_{boost} = 10. BM25 + ADE TM and BM25 + ADE RS use the same set of parameters. To eliminate the effect of the order of query execution, 10 runs with random sequences of all queries were performed and averaged over. Table I shows the results of these experiments.

Unsurprisingly both AQ und ADE (both TM and RS) perform better than the BM25 baseline, since the documents were expanded with additional terms in the user's terminology. Also expected is the better performance in most metrics of AQ compared to ADE, since ADE was restricted to signals contained in the top 100 hits in each search while AQ uses the complete set of query terms for expansion. Interestingly ADE performs better than AQ in the MRR@10 metric. This means that on average the first relevant document appears higher in the result list. ADE achieved the biggest improvements over the baseline in f1@10 (60 %), MRR@10 (23 %) and Precision@10 (22 %).

The use of topic modeling in ADE TM leads only to slightly better performance than random sampling of terms in ACE RS. At least when paired with a term based search strategy like BM25, selecting expansion terms with topic modeling only had a small effect.

For ADE, the number of documents in the index increase from initially 5371 to an average of 7316 after the runs, which is an increase of 36 %.

V. Discussion and Future Work

The experiments show that a strategy based on agents that act on local information to improve search performance can be beneficial. ADE was able to improve search performance against the baseline by more than 20 % in some metrics using less information than AQ. It would be interesting to combine the agent based approach with doc2query in pre-processing to enhance documents with queries generated from document descriptions.

The tests were performed using BM25 which is a search algorithm based on term statistics that provides exact search based on the actual terms in the document. It would be interesting, if ADE performs better when paired with a semantic search algorithm such as Latent Semantic Indexing [18]. Term-statistics-based algorithms like BM25 find a document that contains a set of terms in most cases similarly to the best of multiple documents that contain subsets of the document's terms. In similarity search however the document expansion based on topic modeling might have a bigger impact, since the documents point in space is spread out by the topic-enhanced variants which might make the utility of topic modeling in this context more tangible.

The agent strategy presented in this paper does not use all local information that the agents have access to. in addition to the query terms and rank, interactions with other agents that share the same result list can be explored. The agents could exchange their relevant search terms or analyze each other's content for overlapping terms and concepts. The non-relevant documents can provide negative feedback.

Agent-based document expansion has its place in the information retrieval pipeline between pre-processing and retrieval and ranking. It complements pre-processing-based document expansion methods like doc2query [5] and DeepCT [7] in that it expands documents with actual information gathered from user feedback. It is decoupled from the retrieval and ranking step and thus does not increase the computational cost of these tasks. ADE also benefits the later retrieval stages because it provides them with additional information that can be exploited in retrieval and (re-)ranking.

Acknowledgment

This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the Incentives and Economics of Data Sharing Funding Action (IEDS0001). The author is responsible for the content of this publication.

References

[1] B. Otto, D. Lis, J. Jürjens, J. Cirullies, S. Opriel, F. Howar, S. Meister, M. Spiekermann, H. Pettenpohl, and F. Möller, "DATA ECOSYSTEMS: Conceptual foundations, constituents and recommendations for action: ISST-Report," tech. rep., Fraunhofer-Institut für Software- und Systemtechnik ISST, Dortmund, 2019.

[2] L. Nagel and D. Lycklama, "Design Principles for Data Spaces: Position Paper," tech. rep., International Data Spaces Association, Dortmund, 2021.

[3] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais, "The vocabulary problem in human-system communication," *Communications of the ACM*, vol. 30, no. 11, pp. 964–971, 1987.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint*, 2018. arXiv:1810.04805.

[5] R. Nogueira, W. Yang, J. Lin, and K. Cho, "Document Expansion by Query Prediction," *arXiv preprint*, 2019. arXiv:1904.08375.

[6] R. Nogueira and J. Lin, "From doc2query to docTTTTTquery," *online preprint*, 2019. https://cs.uwaterloo.ca/ jimmylin/publications/.

[7] Z. Dai and J. Callan, "Context-Aware Sentence/Passage Term Importance Estimation For First Stage Retrieval," *arXiv preprint*, 2019. arXiv:1910.10687.

[8] Z. Dai and J. Callan, "Context-Aware Document Term Weighting for Ad-Hoc Search," in *Proceedings of The Web Conference 2020* (Y. Huang, ed.), ACM Digital Library, (New York,NY,United States), pp. 1897–1907, Association for Computing Machinery, 2020.

[9] A. Yates, R. Nogueira, and J. Lin, "Pretrained Transformers for Text Ranking: BERT and Beyond," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (L. Lewin-Eytan, ed.), ACM Digital Library, (New York,NY,United States), pp. 1154–1156, Association for Computing Machinery, 2021.

[10] J. C. Jackson, D. Rand, K. Lewis, M. I. Norton, and K. Gray, "Agent-Based Modeling," *Social Psychological and Personality Science*, vol. 8, no. 4, pp. 387–395, 2017.

[11] L. M. J. Hogg and N. R. Jennings, "Socially Intelligent Reasoning for Autonomous Agents," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 31, no. 5, pp. 381–393, 2001.

[12] A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, and S. Shenker, "On a Network Creation Game," *Proceedings of the Twenty-Second Annual Symposium on Principles of Distributed Computing*, pp. 347–351, 2003.

[13] C. Hinrichs and M. Sonnenschein, "A distributed combinatorial optimisation heuristic for the scheduling of energy resources represented by self-interested agents," *International Journal of Bio-Inspired Computation*, vol. 10, no. 2, pp. 69–78, 2017.

[14] B. Xue and G.-l. Yan, "Research on multi-agents information retrieval system based on intelligent evolution," in *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology (ICCSNT 2012)*, (Piscataway, NJ), pp. 1042–1045, IEEE, 2012.

[15] S. P. Meenakshi, G. Agarwal, S. Bakshi, S. Bhatter, and P. Sivakumar, "Cognitive Agents for Web Based Search Engines: A Review," in *2017 Second International Conference on Recent Trends and Challenges in Computational Models* (P. Vijayakumar and L. J. Deborah, eds.), (Piscataway, NJ), pp. 201–206, IEEE, 2017.

[16] S. M. H. Mahmud, M. F. Rabbi, and K. N. Guy-Fernand, "An Agent-based Meta-Search Engine Architecture for Open Government Datasets Search," *Communications on Applied Electronics*, vol. 4, no. 7, pp. 21–25, 2016.

[17] N. Gulati and A. Garg, "A Secured Cognitive Agent based Multi-strategic Intelligent Search System," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 2, pp. 206–222, 2018.

[18] S. T. Dumais *et al.*, "Latent semantic analysis," *Annu. Rev. Inf. Sci. Technol.*, vol. 38, no. 1, pp. 188–230, 2004.

[19] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[20] J. Qiang, Z. Qian, Y. Li, Y. Yuan, and X. Wu, "Short Text Topic Modeling Techniques, Applications, and Performance: A Survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 3, pp. 1427–1445, 2022.

[21] V. Boteva, D. Gholipour, A. Sokolov, and S. Riezler, "A full-text learning to rank dataset for medical information retrieval," in *Proceedings of the 38th European Conference on Information Retrieval*, 2016.

[22] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford, *et al.*, "Okapi at trec-3," *Nist Special Publication Sp*, vol. 109, p. 109, 1995.

[23] E. Bassani, "ranx: A blazing-fast python library for ranking evaluation and comparison," in *ECIR (2)*, vol. 13186 of *Lecture Notes in Computer Science*, pp. 259–264, Springer, 2022.