Can Generative LLMs Create Query Variants for Test Collections?

An Exploratory Study

Marwah Alaofi
RMIT University
Melbourne, Australia
marwah.alaofi@student.rmit.edu.au

Luke Gallagher
RMIT University
Melbourne, Australia
luke.gallagher@rmit.edu.au

Mark Sanderson
RMIT University
Melbourne, Australia
mark.sanderson@rmit.edu.au

Falk Scholer
RMIT University
Melbourne, Australia
falk.scholer@rmit.edu.au

Paul Thomas
Microsoft
Adelaide, Australia
pathom@microsoft.com

ABSTRACT

This paper explores the utility of a Large Language Model (LLM) to automatically generate queries and query variants from a description of an information need. Given a set of information needs described as backstories, we explore how similar the queries generated by the LLM are to those generated by humans. We quantify the similarity using different metrics and examine how the use of each set would contribute to document pooling when building test collections. Our results show potential in using LLMs to generate query variants. While they may not fully capture the wide variety of human-generated variants, they generate similar sets of relevant documents, reaching up to 71.1% overlap at a pool depth of 100.

CCS CONCEPTS

• **Information systems** → **Test collections**; **Query representation**.

KEYWORDS

Information retrieval; test collections; query variants; LLMs

ACM Reference Format:
Marwah Alaofi, Luke Gallagher, Mark Sanderson, Falk Scholer, and Paul Thomas. 2023. Can Generative LLMs Create Query Variants for Test Collections?: An Exploratory Study. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3539618.3591960

1 INTRODUCTION AND BACKGROUND

Information Retrieval (IR) has been dedicated to delivering relevant information in response to user queries. The realization of this objective has been facilitated by the use of offline test collections, which often provide a single representation (query) for each information need. The single query assumption is convenient for numerous reasons. It helps to make the judging process economically viable (along with the system pooling approach in the Cranfield paradigm of test collection construction [19]), and it has provided a consistent, reusable environment for the development of retrieval systems and evaluation measures. The importance of query variations for enumerating relevant documents in a test collection dates back several decades [18] and previous tracks at TREC have explored the significance of such variance [6]. More recently, there has been a line of research providing further insights from the user perspective with the advent of crowd-sourcing technologies [2, 13].

Query variants are alternative formulations of the same information need. For example, *"what hiking options are there in summer in sangre de cristo"* and *"sangre de Cristo, new mexico hiking"* are both query variants generated in response to the same information need, i.e., finding information for a hiking trip in the Sangre de Cristo mountain region during summer. Bailey et al. [2] showed that given the same backstory, users generate about 57 query variants on average - which is anticipated to increase as the number of participants grows. Similar findings are reported by Mackenzie et al. [12] for queries generated to find additional information in response to document summaries.

The impact of query variants on retrieval has been empirically demonstrated in prior research. Culpepper et al. [7] showed that variants impact effectiveness substantially more than that due to topic or ranking models. Penha et al. [17] tested the impact of variants using neural and transformer-based answer retrieval models. Their experimental results demonstrated a 20% effectiveness drop on average.

Alaofi et al. [1] empirically demonstrated the impact of query variants on a commercial search engine and different inverted indexes. Their results point to a substantial retrieval inconsistency and a concerning impact of variants on document retrievability.

Query variants have also been demonstrated to have a comparable impact on the pool size as that of systems, calling to consider incorporating them when building test collections [14]. The current abstraction of one query per topic in the majority of test collections raises two concerns about how realistic system evaluations are: (1) is limiting system evaluation to a single representation of the information need appropriate, and; (2) how the test collection is constructed in the first place. Can we offer a solution to (1) through the use of the LLMs to generate human-like queries? Or perhaps have

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591960

them at least act in place to generate similar pools for constructing test collections (2)?

Query variants have primarily been obtained through crowd-sourcing [2, 12], a process that is expensive to scale and may not be an accurate representation of query variants as the information needs are not naturally derived. A study by Zhang et al. [22], uses click graphs to collect query variants, based on the assumption that queries leading to the same click originate from the same information need. It is not clear if this assumption always holds true as a shared click may not necessarily indicate a shared intent and many shared intents may not lead to a shared click. This requires external labeling which is difficult to achieve objectively, and as in the case of crowd-sourcing is expensive to scale.

In-Context Learning (ICL) [5] emerges as a promising Natural Language Processing paradigm where no large domain-specific datasets are required to fine-tune LLMs on a specific downstream task. Instead, the LLMs are conditioned using a 'context' which is simply a textual description of the task with a few or even zero examples - often referred to as a few-, one-, or zero-shot learning. This approach achieves promising results and has surpassed some of the state-of-the-art models in some tasks. This holds the promise of addressing the challenge posed by the scarcity of large query variant datasets.

ICL has been recently used in IR, mainly to generate synthetic queries given documents [3, 8, 10]. The synthetic query-document pairs are then used to train a retrieval model. This approach builds on earlier efforts which, prior to indexing, used fine-tuned LLMs to extend documents by generating relevant queries, an approach that was simple yet effective to surpass state-of-the-art models on retrieval benchmarks [15]. Though the advances this research direction has made, it specifically aims to harness the power of LLMs to boost effectiveness scores by using some 'representative' queries, which may undergo some automated quality/consistency filtering [8, 9], resulting in a query set that improves performance but may not necessarily represent users.

The aim of this study is to explore using an LLM (GPT-3.5) as an alternative method to generate query variants given backstories (i.e., information need statements). We aim to compare the LLM-generated queries to human-generated ones. We approach this by quantifying the direct similarity between the two sets of queries and examining how they behave when used for pool construction. In particular, we pose the following questions:

RQ1 Can an LLM, with one-shot learning, generate queries that are similar or perhaps identical to the ones generated by humans?

RQ2 How do both sets compare when used for document pooling when constructing test collections?

2 EXPERIMENT DESIGN

We detail the query sets and metrics used in the experiment.

2.1 Query Sets, Model Prompting, and Runs

We use two sets of query variants: human and GPT-3.5 generated.

The human-generated query variants – referred to as the *human set* – were collected via crowd-sourcing as part of the UQV100 test collection [2], which has one hundred backstories to describe one hundred search topics derived from the TREC 2013 and 2014 web track. Crowd workers were asked to read a backstory and formulate an initial query for the search task.

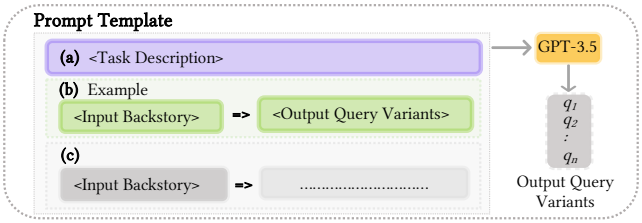

Figure 1: The prompt used to feed the GPT-3.5 model.

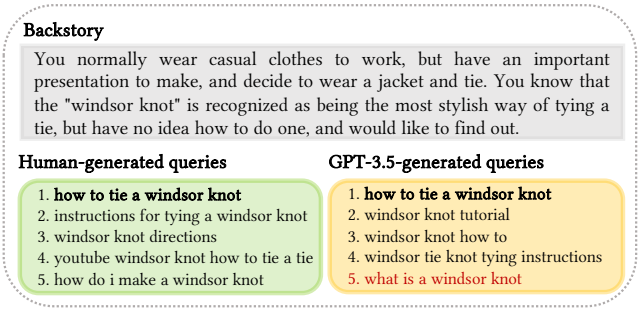

Figure 2: Randomly selected example variants generated by humans and GPT-3.5 ($temp = 0.5$). The variant in bold is reproduced by GPT-3.5 and the one in red appears less appropriate, although similar ones are generated by humans, e.g., "windsor knot wiki".

To generate the *GPT sets*, we use the same backstories from UQV100 to prompt the model. We use the *text-davinci-003* model.[1] This model is trained similarly to InstructGPT [16] using reinforcement learning with reward models fine-tuned on human preferences.[2] We experiment with different temperature settings $temp = \{0.0, 0.5, 1.0\}$, a parameter that controls how deterministic the model is in generating the text.

We prompt the model using the template in Figure 1. The prompt has (a) a *task description*, (b) an *example*, and (c) an *input backstory* for the model to generate the corresponding query variants. The task description is a natural language specification of the task, which provides some context and details to guide the model toward the expected distribution of the queries per backstory and the average number of words per query, with specific values of these settings based on prior research in query variant analysis [2, 12].

We follow a one-shot learning approach, in which the prompt contains an example input backstory with its associated human-generated output queries, randomly selected from UQV100. The same random example (i.e., topic 275) is used to prompt the model to generate query variants for the remaining 99 backstories. In order to avoid the influence of observed data, topic 275 has been excluded from our analysis.

We investigated zero-shot learning with no examples provided to the model. However, this approach resulted in the model producing

[1] Last accessed on 2 February 2023

[2] https://platform.openai.com/docs/model-index-for-researchers

Table 1: Query (Q) statistics of the human set and the three GPT sets under different temperature *temp* settings.

Query Variant Set	Number of Variants					Avg. Words/Q
	Total	Unique	Min.	Max.	Avg.	
Human	10726	5681	19	101	57.38	5.34
GPT ($temp = 0.0$)	4803	3638	11	172	36.75	5.95
GPT ($temp = 0.5$)	3061	2999	12	88	30.29	4.86
GPT ($temp = 1.0$)	2725	2719	12	48	27.46	4.65

long variants that closely resembled natural language questions. Few-shot learning, where multiple examples are provided to the model, might have produced better results [5], but as we were limited by the number of available backstories we opted to use one-shot learning. The distribution of all query sets are presented in Table 1. Some example query variants from the human set and one of the GPT sets for a given backstory are shown in Figure 2.

We generate runs for the human set and the GPT sets using Anserini [21] BM25 ($b = 0.4, k1 = 0.9$) on the ClueWeb12-B corpus[3] - which is also the corpus that was used to create relevance judgments for the UQV100 test collection. The prompt template we use and the generated GPT sets are publicly available to aid reproducibility.[4]

2.2 Metrics for Query Similarity

In addressing **RQ1**, we assume that the human set is the ideal set of query variants and measure how similar the GPT sets are to that set. We quantify that by measuring the average *Jaccard Index* between the human set and the three GPT sets in which the overlapping queries are an exact match between the two sets.

As keyword-based ranking models treat queries with slight variations equally, we incrementally relax the matching condition using text transformations over the two sets and report the average *Jaccard Index* score using the unique queries generated after each transformation. Specifically, the overlap between the two sets is quantified by initially determining the exact match of raw queries in both sets. This matching condition is then relaxed to cumulatively allow for variations in punctuation (T1), word forms (T2), stop words (T3), and word order (T4). We also show the *Coverage Ratio*, which quantifies the proportion of queries from the human set that are successfully reproduced in the GPT sets using the aforementioned matching conditions.

2.3 Metrics for Retrieval Similarity

To address **RQ2**, we find the overlap between the union of documents returned from the variants of the human set and the GPT sets. This measure quantifies the similarity of the two sets in their utility for constructing document pools. The overlap is quantified using the *Jaccard Index*, measured at different depths. While we are interested in measuring the overall overlap in documents (regardless of their relevance judgments), we believe that the impact of the overlap of relevant documents in particular holds greater importance in regards to system evaluation when constructing test collections.

[3]https://www.lemurproject.org/clueweb12.php/
[4]https://github.com/MarwahAlaofi/SIGIR-23-SRP-UQV100-GPT-Query-Variants

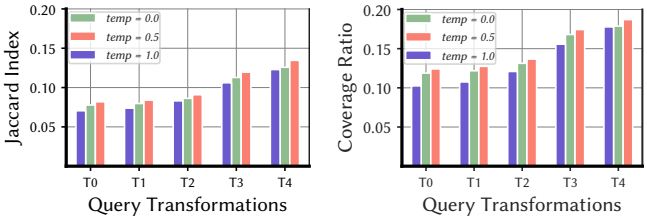

Figure 3: Jaccard index (left) and coverage ratio (right) between the human and the GPT sets under different temperature *temp* settings. *T0 denotes the original query set.*

That is, if a GPT set fails to retrieve irrelevant documents that are retrieved by the human set, those documents will remain unjudged and thus treated as irrelevant in most effectiveness metrics, unless using metrics that account for uncertainty (e.g., RBP).

We use the relevance judgments provided with the UQV100 collection and measure the overlap by considering relevant documents alone, i.e., those rated as *Essential*; *Very Useful*; *Mostly Useful* or *Slightly Useful*.

Different properties of the document pool are computed, mainly the *pool size growth* following [14], to measure the diversity of the GPT sets in comparison to the human set. We hypothesize that a diverse set of query variants given a topic is likely to retrieve different documents leading to a higher growth rate than a set of similar queries. This diversity is also examined using *Rank-Biased Overlap (RBO)* [20] to quantify the consistency between the retrieved documents of the query variants given a topic. A topic RBO score is the average score over all topic-variant pairs. Different query effectiveness metrics are also computed for comparison.

3 RESULTS AND DISCUSSION

We examine query and retrieval similarity.

3.1 Query Sets Similarity

Figure 3 shows the overlap between the human set and the GPT sets as measured by the Jaccard index and coverage ratio under four matching conditions. Results indicate a minimum of 7.1% Jaccard index between the GPT sets and the human set, with GPT sets demonstrating exact coverage of at least 10.3% of the human-generated queries. As expected, the queries demonstrate a greater degree of overlap as they undergo successive transformations, ultimately reaching a maximum Jaccard index of 13.5% and coverage ratio of 18.7% when using a temperature of 0.5.

While the observed overlap does not seem to indicate a high similarity, they should be interpreted within the limitation of the UQV100 human set - which is still somewhat artificial. That is, it cannot be conclusively stated that the unique query variants generated by GPT cannot be written by humans should we have more participants. The use of the UQV100 human-generated queries as a reference point to an ideal set of variants is not realistic, and while this comparison helps understand the capability of GPT, it may limit our interpretation as an exhaustive set of query variants given a topic would never exist. Incorporating human evaluation to assess the extent to which the GPT sets approximate human queries

Table 2: Average effectiveness metrics, RBO and pool properties at depth 10 for the human set and the GPT sets given all variants across all topics. RBP and RBO are measured using $p = 0.9$. Entries annotated with † and ‡ respectively indicate statistical significance for a Bonferroni pairwise t-test at $p < 0.05$ and $p < 0.01$ compared to the human query set baseline. Topic 275 (the example used to prompt the model) was removed from the computation and results are replicated independently.

Variant Set	P@10	NDCG@10	RBP	RBO	Pool Properties		
					Size	Relevant	Unjudged
Human set	0.443	0.274	0.406 +0.111	0.201	190.69	0.30	0.13
GPT ($temp = 0.0$)	0.386‡	0.246†	0.358 +0.254‡	0.235†	94.42	0.29	0.33
GPT ($temp = 0.5$)	0.393‡	0.249†	0.360 +0.238‡	0.220	93.55	0.29	0.31
GPT ($temp = 1.0$)	0.384‡	0.240‡	0.355 +0.263‡	0.235†	105.21	0.27	0.37

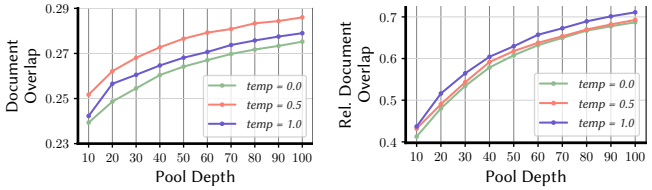

Figure 4: The average Jaccard index between the documents retrieved by the human set and the GPT sets at different depths given all documents (left) and relevant documents only (right).

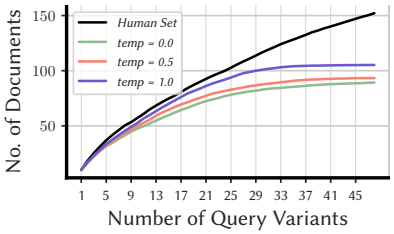

Figure 5: The average pool size at depth 10 as more variants are added. The growth lines are cut at 48, which is the maximum number of variants generated by GPT at $temp = 1$.

may yield more precise conclusions. This is, however, a question to be explored in future research.

3.2 Retrieval Similarity

Figure 4 shows the overlap between the document pools generated in response to the human set and the GPT sets at different depths measured by the Jaccard index. A relatively high average overlap of document pools is observed between the GPT sets and the human set, which increases as we increase the depth. When considering relevant documents only, the overlap is considerably high. With a temperature of 1.0, for example, the pools overlap at 43.7% at depth 10. This increases to 71.1% when examining the pools at depth 100.

Effectiveness metrics, RBO scores and pool properties are given in Table 2. It is evident that the human-generated variants yield a larger pool size, almost double that generated by any GPT sets, and which also grows faster (see Figure 5). This indicates a possible higher diversity (e.g., more distinct query terms) in the human set. This observation is supported by a lower consistency of the variants from the human set, as measured by RBO (significant with the temperature set to 0.0 or 1.0).

Human-generated variants are significantly more effective across all metrics. GPT queries, on the other hand, have higher residuals which indicate that, given more judgments, they may achieve higher effectiveness scores. This is also reflected in the higher proportion of the unjudged documents returned in the GPT generated pools. It would be interesting to further investigate the unjudged portion of the GPT sets to understand whether they retrieve relevant documents that were not found through the human set.

4 CONCLUSIONS AND FUTURE WORK

In this paper, we posed the following questions:

RQ1 Can an LLM, with one-shot learning, generate queries that are similar or perhaps identical to the ones generated by humans?

RQ2 How do both sets compare when used for document pooling when constructing test collections?

We found that for **RQ1**, GPT reproduced a reasonable portion of the human-generated queries. The similarity to human queries is yet to be fully understood given the limitation of the human set.

For **RQ2**, we found that GPT queries seem to have a substantial overlap in the pool of documents, particularly when we consider the relevant set alone. At 71.1% overlap at depth 100, GPT shows potential for replacing human query variants with synthetically generated ones during document pool construction.

This work presents a new opportunity to conveniently expand existing test collections, particularly those resembling TREC, which have information need statements that can be employed to condition LLMs. Further research could explore advanced prompting techniques and compare our approach of using an LLM to generate query variants from backstories with previous query simulation methods (e.g., [4, 11]), which were used to generate query variants from source documents.

ACKNOWLEDGMENTS

Marwah Alaofi is supported by a scholarship from Taibah University, Saudi Arabia. This work was also supported by the Australian Research Council (DP180102687, CE200100005). We thank the anonymous reviewers for their helpful feedback.

REFERENCES

[1] Marwah Alaofi, Luke Gallagher, Dana Mckay, Lauren L. Saling, Mark Sanderson, Falk Scholer, Damiano Spina, and Ryen W. White. 2022. Where Do Queries Come From?. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2850–2862. https://doi.org/10.1145/3477495.3531711

[2] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2016. UQV100: A Test Collection with Query Variability. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 725–728. https://doi.org/10.1145/2911451.2914671

[3] Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. InPars: Unsupervised Dataset Generation for Information Retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2387–2392. https://doi.org/10.1145/3477495.3531863

[4] Timo Breuer, Norbert Fuhr, and Philipp Schaer. 2022. Validating Simulations of User Query Variants. In *Advances in Information Retrieval - 44th European Conference on IR Research*. Springer, 80–94. https://doi.org/10.1007/978-3-030-99736-6_6

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 1877–1901.

[6] Chris Buckley and Janet A Walz. 1999. The TREC-8 Query Track. In *Proceeding of Text Retrieval conference*. NIST Special Publication, 500–246.

[7] J. Shane Culpepper, Guglielmo Faggioli, Nicola Ferro, and Oren Kurland. 2021. Topic Difficulty: Collection and Query Formulation Effects. *ACM Transactions on Information Systems* 40, 1, Article 19 (2021), 36 pages. https://doi.org/10.1145/3470563

[8] Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. (2022). arXiv:2209.11755

[9] Mitko Gospodinov, Sean MacAvaney, and Craig Macdonald. 2023. Doc2Query–: When Less is More. In *Advances in Information Retrieval - 45th European Conference on IR Research*. Springer, 414–422. https://doi.org/10.1007/978-3-031-28238-6_31

[10] Vitor Jeronymo, Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, Jakub Zavrel, and Rodrigo Nogueira. 2023. InPars-v2: Large Language Models as Efficient Dataset Generators for Information Retrieval. (2023). arXiv:2301.01820

[11] Chris Jordan, Carolyn Watters, and Qigang Gao. 2006. Using Controlled Query Generation to Evaluate Blind Relevance Feedback Algorithms. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*. Association for Computing Machinery, 286–295. https://doi.org/10.1145/1141753.1141818

[12] Joel Mackenzie, Rodger Benham, Matthias Petri, Johanne R. Trippas, J. Shane Culpepper, and Alistair Moffat. 2020. CC-News-En: A Large English News Corpus. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, 3077–3084. https://doi.org/10.1145/3340531.3412762

[13] Alistair Moffat, Peter Bailey, Falk Scholer, and Paul Thomas. 2017. Incorporating User Expectations and Behavior into the Measurement of Search Effectiveness. *ACM Transactions on Information Systems* 35, 3, Article 24 (2017), 38 pages. https://doi.org/10.1145/3052768

[14] Alistair Moffat, Falk Scholer, Paul Thomas, and Peter Bailey. 2015. Pooled Evaluation Over Query Variations: Users Are as Diverse as Systems. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, 1759–1762. https://doi.org/10.1145/2806416.2806606

[15] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. (2019). arXiv:1904.08375

[16] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 27730–27744.

[17] Gustavo Penha, Arthur Câmara, and Claudia Hauff. 2022. Evaluating the Robustness of Retrieval Pipelines with Query Variation Generators. In *Advances in Information Retrieval - 44th European Conference on IR Research*. Springer, 397–412. https://doi.org/10.1007/978-3-030-99736-6_27

[18] Karen Spärck Jones and R. Graham Bates. 1977. *Report on a Design Study for the "Ideal" Information Retrieval Test Collection*. British Library Research and Development Report No. 5428. University of Cambridge.

[19] Ellen M. Voorhees. 2002. The Philosophy of Information Retrieval Evaluation. In *Workshop CLEF for European Languages*. 355–370.

[20] William Webber, Alistair Moffat, and Justin Zobel. 2010. A Similarity Measure for Indefinite Rankings. *ACM Transactions on Information Systems* 28, 4, Article 20 (2010), 38 pages. https://doi.org/10.1145/1852102.1852106

[21] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the Use of Lucene for Information Retrieval Research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 1253–1256. https://doi.org/10.1145/3077136.3080721

[22] Hongfei Zhang, Xia Song, Chenyan Xiong, Corby Rosset, Paul N. Bennett, Nick Craswell, and Saurabh Tiwary. 2019. Generic Intent Representation in Web Search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 65–74. https://doi.org/10.1145/3331184.3331198