Generating Natural Language Queries for More Effective Systematic Review Screening Prioritisation

Shuai Wang
The University of Queensland

Harrisen Scells
Leipzig University

Martin Potthast
Leipzig University & ScaDS.AI

Bevan Koopman
CSIRO & The University of Queensland

Guido Zuccon
The University of Queensland

ABSTRACT

Screening prioritisation in medical systematic reviews aims to rank the set of documents retrieved by complex Boolean queries. Prioritising the most important documents ensures that subsequent review steps can be carried out more efficiently and effectively. The current state of the art uses the final title of the review as a query to rank the documents using BERT-based neural rankers. However, the final title is only formulated at the end of the review process, which makes this approach impractical as it relies on ex post facto information. At the time of screening, only a rough working title is available, with which the BERT-based ranker performs significantly worse than with the final title. In this paper, we explore alternative sources of queries for prioritising screening, such as the Boolean query used to retrieve the documents to be screened and queries generated by instruction-based generative large-scale language models such as ChatGPT and Alpaca. Our best approach is not only viable based on the information available at the time of screening, but also has similar effectiveness to the final title.

CCS CONCEPTS

• **Information systems** → **Query suggestion**; • **Computing methodologies** → **Natural language generation**.

KEYWORDS

Systematic review, Screening prioritisation, Query variations, LLM

1 INTRODUCTION

Systematic reviews are a widely used type of literature review in evidence-based medicine to comprehensively identify, analyse and summarise all available research on a particular topic or question

Table 1: Our contribution at a glance: The post hoc effectiveness of Wang et al.'s [65] original approach can be achieved by generating queries from sources available in practice; * indicates statistical significant differences.

Source	Query	MAP	LastRel	WSS95	WSS100	Ref.
post hoc	final review title	0.295	634.975	**0.609**	**0.597**	[65]
	best generated query	**0.310**	**620.025**	0.589	0.569	**ours**
practice	working title	0.059*	976.275*	0.266*	0.271*	[65]
	generated queries	0.249	714.500	0.541*	0.521*	**ours**

in an unbiased manner [29]. They provide a rigorous and transparent pathway to medical decision-making tasks and minimise bias and errors that might otherwise result from an ad hoc literature search [59]. Systematic reviews are usually conducted according to a protocol of established steps [10, 20, 29]. As part of this process, complex Boolean queries are developed by specialists (e.g. trained experts in search) to obtain a large initial set of (candidate) documents. These candidates are manually screened to obtain a subset for in-depth analysis. One approach to increase the efficiency of relevance assessment in systematic reviews is know as *screening prioritisation* [28, 40]. The aim of this method is to rank the candidates so that the subsequent review tasks, such as full text screening, can start earlier and in parallel with the screening, resulting in a more timely and predictable completion. For rapid reviews, where screening is constrained by a limited budget, prioritising screening can help to identify more relevant documents and thus enable a review of higher quality [60].

Currently, the most advanced screening prioritisation methods use BERT-based neural rankers [65]. They are based on a fundamental but ultimately unjustified assumption that the title of the systematic review has already been conceived and is available as a query for ranking before screening. However, according to the systematic review protocol, the title of the review does not have to be formulated before the search [10, 20]: as a rule, the title is only determined at the time of writing. Instead, most systematic reviews have only a rough working title, usually just a few keywords of the study [62]. Our experiments show that using a working title for screening prioritisation is not competitive to using the final title [18]: When applied to the Seed Collection dataset [62], which contains both working and final titles of systematic reviews, the effectiveness of a neural ranker (Section 4.3.2) depends strongly on which of the two title versions is used (Table 1, top rows). Using the final title largely overestimates the achievable effectiveness.

Shuai Wang, Harrisen Scells, Martin Potthast, Bevan Koopman, and Guido Zuccon

In this paper, we investigate how screening prioritisation can be done based on the information available at the time of screening. Since it can take up to several weeks to work out the Boolean queries for the prior retrieval of candidate documents [45], we propose effective ways to exploit this valuable source of information that has so far been neglected by the state of the art [7, 27, 30, 31, 65, 66]. However, using a Boolean query to produce a ranked list is not straightforward. A Boolean query is complex, structured, and detailed; it is very different from the queries that are common in ad hoc retrieval [6]. BERT-based methods for ranking may perform poorly on these queries. Therefore, we investigate the use of two instruction-based models, namely OpenAI's ChatGPT [17] and Stanford's Alpaca [57], to generate natural language queries from Boolean queries. These generated natural language queries are in turn used as input for our neural-ranker-based screening prioritisation methods. The bottom rows of Table 1 show that the most powerful variants of our method are able to generate queries that compete with the use of the final title.[1] To guide our investigation, we have developed five research questions:

RQ1 How effective is screening prioritisation with Boolean queries compared to natural language queries generated from them?
RQ2 How do different generation models affect the effectiveness of natural language queries generated from Boolean queries?
RQ3 What impact do ranking methods have on the effectiveness of natural language queries derived from Boolean queries?
RQ4 Does generating multiple natural language queries from a single Boolean query improve effectiveness?
RQ5 How effective is screening prioritisation with natural language queries derived from Boolean queries compared to using the working titles of systematic reviews?

2 RELATED WORK

In this section, we review the literature on screening prioritisation for systematic reviews and instruction-based large language models.

2.1 Systematic Review Screening Prioritisation

Screening prioritisation has received considerable attention in technology-assisted systematic review generation. Various aspects were investigated, including the use of different input data sources [31, 50, 66], different algorithms or models for ranking purposes [1, 2, 8, 26, 30, 35, 38, 48, 62, 65], and active learning techniques that improve the efficiency of screening prioritisation through a human-in-the-loop approach [4, 7, 12, 13, 21, 35, 37, 54, 69, 71].

Boolean-driven screening prioritisation uses a Boolean query to rank candidate documents directly. While few studies have examined using Boolean queries alone, most used them in conjunction with the final review title [1–3]. Typically, keywords are extracted from the Boolean query and the review title to formulate a (bag of words) query, and then a lexical scoring function determines the relevance of a document to the query. However, these methods are impractical since the final title is not available at the time of screening. The coordination level fusion (CLF) approach proposed by Scells et al. [50] is the only existing method that examines screening prioritisation using Boolean queries only. It uses rank fusion to rank the documents retrieved by each clause of a Boolean query.

---
[1]Code: https://github.com/ielab/SIGIR-AP-2023-Bolean2Natural4SR

Neural ranker-based screening prioritisation methods rely on pre-trained models such as BERT and have achieved much higher effectiveness than traditional lexical rankers, on par with active learning methods that use relevance signals from the screened documents [65]. Despite the improvements they have brought to screening prioritisation, there are still challenges in using them: For instance, the input token length limitation imposed by most BERT-based models [11] is a critical limitation. It does not allow the model to process longer text inputs, such as the full text of candidate documents, extensive Boolean queries, or seed studies as a source of information. Previous approaches using neural rankers for screening prioritisation has focused only on using the review title as a query. We show that their effectiveness does not generalise when working titles are used instead (see Table 1).

2.2 Instruction-based Large Language Models

Recent advances in instruction-based large language models (LLMs), such as ChatGPT, have shown that they are able to accurately follow user instructions to complete tasks [17, 19, 46, 64]. These models typically contain tens of billions of parameters and are trained on diverse and extensive textual data so that they are able to generate relevant and coherent answers for a wide range of topics [17]. Several studies have evaluated the effectiveness of ChatGPT on various tasks, often observing an increase in effectiveness compared to previous approaches, e.g., in question answering [39, 56] or ranking [22, 55]. As part of a systematic review literature search, the use of ChatGPT to generate Boolean queries for systematic reviews has been investigated Wang et al. [64]. The results of this study showed that ChatGPT generates effective queries with appropriate prompting. In this paper we use ChatGPT and Alpaca [57]. Alpaca was fine-tuned based on an LLM developed by Meta with seven billion parameters, known as LLaMa [57, 58]. Alpaca has been fine-tuned using 52K instruction–output pairs generated from ChatGPT via a self-instruction approach Wang et al. [68], showing similar capabilities to ChatGPT in preliminary human evaluations [57].

For ranking tasks, instruction-based language models are integrated with ranking models to achieve more effective results [16, 23, 61, 70]. Specifically, there are two common ways to combine these models: *retrieval-then-generation* and *generation-then-retrieval*. In the *retrieval-then-generation* approach, the ranking model first retrieves a set of relevant results based on the user's query. Then, the instruction-based language model generates a response based on the retrieved documents. This method relies primarily on the ranking model's ability to understand the user and extract corresponding information from their query, while the LLM is used to summarise the retrieved evidence to provide the user with a credible and comprehensive answer [23, 70]. In the *generate-then-retrieval* approach, the instruction-based LLM is first used to generate a response based on the user's query, and this response is then processed by a ranking model as a new query to retrieve documents that provide evidence for its statements [16, 61].

3 METHODOLOGY

Figure 1 gives an overview of our approach for screening prioritisation. One or more natural language queries are generated from a given Boolean query. Then the candidate documents to be screened are ranked based on the generated queries.
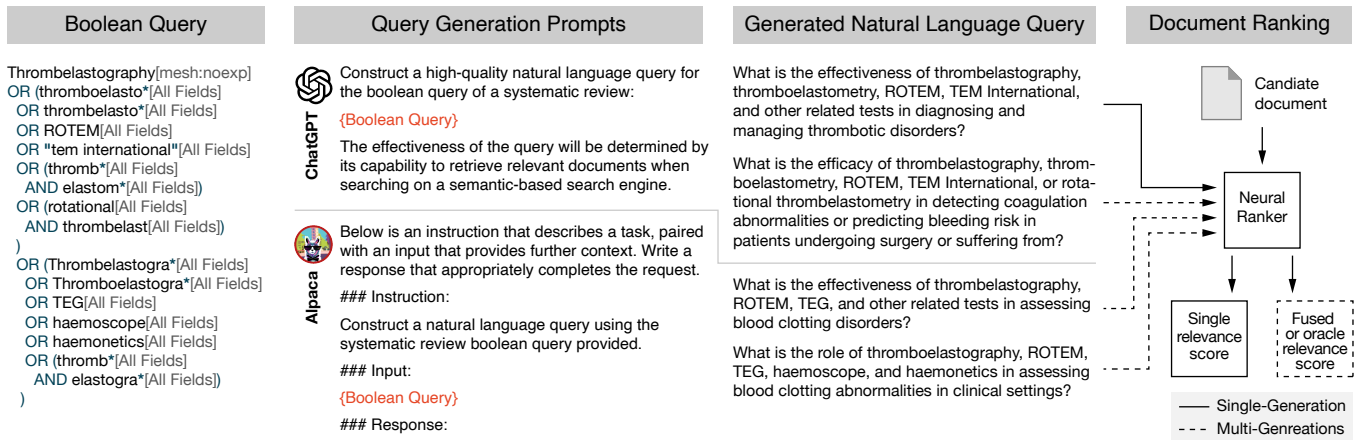
Figure 1: Illustration and examples of our screening prioritisation approach: Given a Boolean query, an instruction-based LLM is prompted to generate one or more natural language queries. Then, given a generated query and a candidate document, a neural ranker is used to predict one or more relevance scores for the document. In the latter case, the scores are fused by addition. As a baseline for our experiments, the score that maximises effectiveness is selected by an oracle.

3.1 Query Generation

Our task is to generate a natural language query from a Boolean query of a systematic review that best describes its information need. We evaluate two LLMs for this task: ChatGPT [17][2] and Alpaca [57].[3] Figure 1 shows an example consisting of a Boolean query, carefully optimized prompts for each of the two models, and four alternative generated natural language queries. Preliminary studies have shown that Alpaca has problems with zero-shot generation for virtually all Boolean queries, often returning the original Boolean query itself. We addressed this problem by fine-tuning Alpaca using pairs of Boolean queries and natural language queries generated by ChatGPT as training examples.

To adjust the "creativity" of an LLM, they often introduce a degree of stochasticity controlled by the so-called temperature parameter t, where $0 \leq t \leq 1$ [32]. Setting $t = 0$ causes the model to generate the same response over multiple inferences for a given prompt, whereas $t = 1$ causes the model's response to be randomly different each time. In other words, the lower the temperature value, the more deterministic a model's response. In our experiments, we investigate how the creativity of a model affects the effectiveness of screening prioritisation (RQ4). Therefore, we compare the generation of a single natural language query (*Single-Generation*) to generating multiple natural language queries (*Multi-Generations*) by adjusting the temperature accordingly.[4]

3.2 Document Ranking

To rank the documents, we follow the state-of-the-art screening prioritisation method developed by Wang et al. [65]. Here, a cross-encoder-based neural ranker is used to calculate the relevance score of a query–document pair. Specifically, the query and the document are first concatenated with a $[SEP]$ token, and then fed into a cross-encoder model that calculates the relevance score for the concatenated pair. The relevance score is then represented by the special classification token $[CLS]$ in the output of the model [9]. In the proposed pipeline, the query is obtained from a Boolean query as described above. However, in our experiments, we also explore the alternative of using the original Boolean query as input to the cross-encoder to address RQ1, and the alternative of using the working title of the review to address RQ5.

For fine-tuning, we first use a pre-trained BioBERT model, which has been shown to be effective in screening prioritisation when the title of the review is used as query [65]. Then, for each topic in the training set, we extract all relevant documents D^+ and a number of non-relevant documents D^-. For each pair of relevant document $(d^+, d^-) \in D^+ \times D^-$, we create training triples $\langle query, d^+, d^- \rangle$ and then fine-tune the model using localised contrastive loss as proposed by Gao et al. [15].

To investigate RQ4, we generate multiple queries per Boolean query in the Multi-Generation setup, calculate a relevance score for each pair of query and candidate document, and then apply two strategies to derive a single relevance score from them for a candidate document: *Fusion* and *Oracle* selection. In the *Fusion* strategy, the relevance scores of all natural language queries on the same topic that refer to a candidate document are summed to calculate the final relevance score of the document with respect to the topic of the systematic review. For the *Oracle* strategy, we first evaluate the ranked lists from different natural language queries, after which the best-performing ranked list, as measured by the mean average precision (MAP), is selected. This strategy serves as an upper bound baseline.

To investigate the effectiveness of combining the results of a generated natural language query with those of the original Boolean query, we also evaluate a setup that includes a fusion of their ranking results. For this purpose, we use the COMBSUM fusion technique to fuse the two ranked lists [14]: the relevance score of a document in the fused ranked list is the sum of the individual scores of the document in the two lists to be fused.

---

[2]We used the OpenAI's GPT-3.5-turbo API with a maximum of 4,097 tokens.
[3]The model was fine-tuned using the original setup of Stanford's Alpaca.
[4]As Hugging Face does not allow $t = 0$, we use $t = 0.0001$ instead.

Shuai Wang, Harrisen Scells, Martin Potthast, Bevan Koopman, and Guido Zuccon

4 EXPERIMENTAL SETUP

In this section, we outline the datasets we use, the methods we apply, and how we evaluate them.

4.1 Dataset

We use two collections in our experiments. The *CLEF TAR Collection* comprises three datasets from 2017, 2018, and 2019. In 2017, the dataset includes 50 systematic review topics divided into 20 for training and 30 for testing [24]. In 2018 the dataset is expanded including all 50 systematic review topics from 2017 as a training set and adding 30 new topics for testing [26]. The 2017 and 2018 datasets focus on Diagnostic Test Accuracy (DTA) systematic reviews. The 2019 dataset is divided into four categories of systematic reviews: the DTA category, which builds upon the 2018 dataset and uses it as the training set with eight new topics for testing; the Intervention category, containing 20 training topics and 20 testing topics; the Prognosis Review category and the Qualitative Review category, each featuring one topic [25]. In our experiments, we treat DTA and Intervention topics as two sub-collections, denoted as CLEF-2019-DTA, and CLEF-2019-Intervention. Each topic in the CLEF TAR Collection provides the review title, the Boolean query used for document retrieval, the documents retrieved as a result of the Boolean query, and the relevance labels for the documents at both abstract and full-text levels [24–26].

The *Seed Collection* contains 40 systematic review topics without training or testing portions [62]. The dataset also contains the review title, the Boolean query used during retrieval, documents retrieved, and relevance labels. However, unlike the CLEF TAR Collection, where abstract-level relevance and full-text level relevance are all included, the Seed Collection only contains full-text level relevance judgements directly extracted from published reviews. One major difference between the Seed Collection and the CLEF TAR Collection is that the dataset also includes more details of the review. For example, it includes a temporary working title for each review, named 'search name' in the collection, and a set of seed studies used for Boolean query creation [62].

Unlike previous studies that used only the training portions specified in each dataset, we re-split our training data to include distinct topics from all other datasets (CLEF TAR Collection and Seed Collection) that are not included in the test portion of the respective dataset. We chose this strategy due to the uneven allocation of training data across the datasets we use. For instance, the Seed Collection dataset contains no training topics, whereas CLEF-2017 comprises 20 training topics, and CLEF-2019-DTA holds 80 topics. By incorporating training data from a range of sources, we aim to establish a more balanced and comprehensive training environment for our fine-tuned models.

4.2 Baseline Methods

In our experiment, we employ BM25 and the Query Likelihood Model (QLM) as baseline ranking models [42, 44]. For query preprocessing, we begin by removing all field types in the query, leaving us with only the query terms. We then apply the matching algorithm to the candidate document, calculating a relevance score between the query and the document.

Similar to previous studies comparing neural rankers with traditional term-matching rankers, we utilise specific tools to implement our baseline models. For BM25, we employ the Gensim toolkit, an open-source library that offers robust implementations for a variety of information retrieval tasks [43]. For the QLM, we apply Jelinek-Mercer (JM) smoothing, a popular technique for query likelihood estimation [42].

In addition to the traditional ranking models, we also benchmark our models against the best-performing methods from participant runs in each CLEF-TAR dataset. It is important to note that certain participant runs have utilised relevance signals from relevance assessments to actively re-rank the remaining documents. We have excluded these runs from our baseline comparison, as they do not align with the scope of our screening prioritisation task, making the comparison unfair. The following participant runs have been selected as baselines for our study: CLEF-2017: *sheffield.run4* [2]; CLEF-2018: *shef-general* [1]; CLEF-2019-dta: *Sheffield/DTA/DTA_sheffield-Odds_Ratio* [3]; CLEF-2019-intervention: *Sheffield/DTA/DTA_sheffieldLog_Likelihood* [3].

Lastly, for the CLEF-2017 and 2018 datasets, we compare our method with the CLF approach proposed by Scells et al. [50]. The CLF approach stands out as the only existing methodology that has explored the application of Boolean queries for systematic review screening prioritisation.

4.3 Model Fine-tuning

In our experiments, we focus on fine-tuning two models: the Alpaca model for query generation and BioBERT for document ranking.

4.3.1 Fine-tuning the Alpaca Model. To fine-tune the Alpaca model, our first step involves using Single-Generation to convert the Boolean query into a natural language query for the training portion of each dataset. We use ChatGPT for this conversion task, and consider its output as the gold standard for the Alpaca model to learn from. Following this, we use the prompts shown in the second column of Figure 1 to further fine-tune the Alpaca model to generate a natural language query using the Boolean query of a topic. As Boolean queries for systematic reviews are complex and require many tokens, we opted to simplify the prompt used for Boolean query conversion in ChatGPT. To ensure minimal loss of information from the Boolean query, we increased the input token limit of the Alpaca model from 512 to 768.[5] Our fine-tuning process for each Alpaca model continues over three epochs, with batch size and gradient accumulation steps of one each, using three Nvidia 80GB A100 GPUs. For the remaining parameters, we adhered to those used in the original Alpaca work [57]. During inference, we use the same prompt as fine-tuned to convert the Boolean query to a natural language query in each test dataset.

4.3.2 Fine-tuning the neural ranker. In our experiments, we chose BioBERT as our pre-trained language model to fine-tune for ranking [34]. Previous research has demonstrated that BioBERT shows higher effectiveness in the task of title-driven screening prioritisation [65]. Same as the previous work, we utilise the Reranker toolkit [15] to fine-tune our model across 100 epochs. The key

---

[5]768 is the maximum token limit for three 80GB Nvidia A100 GPUs (batch size=1).

distinction in fine-tuning and inference pipeline lies in the maximum query length set for all models that utilise Boolean or natural language queries. Instead of the query limit of 64 that was set in previous work for the review title, we extend this to 256 to accommodate the naturally longer input derived from Boolean queries. This adjustment ensures that our models are capable of processing and learning from the full complexity of these queries, potentially enhancing their performance and the accuracy of their outputs.

4.4 Evaluation

For the CLEF-TAR Collection, we rely on abstract-level relevance to ensure a fair comparison with the submitted runs. However, for the Seed Collection where no abstract-level labels were provided, we utilise full-text level relevance signals.

To demonstrate the effectiveness of document ranking on screening prioritisation, we compute various evaluation metrics as established in at CLEF TAR. These metrics include Average Precision (AP), the rank of the last relevant document (Last_rel), Recall at several percentage cutoffs (1%, 5%, 10%, and 20%), and Work Saved over Sampling (WSS) at 95% and 100%. In accordance with the CLEF TAR tasks, we have used the same metrics for evaluating our work and used the tar-2018 evaluation script to evaluate our results [26].

5 MAIN RESULTS

In this section, we outline and interpret the results from our experiments. Specifically, we delve into the results derived from *Single-Generation* in Section 5.1, while Section 5.2 is devoted to examining *Multi-Generations*. Lastly, we perform an ablation study in Section 5.3 to further investigate the effectiveness of our method under various experimental configurations.

5.1 Effectiveness of Single-Generation

To understand the effectiveness of *Single-Generation*, Table 2 compares the ranking effectiveness of the generated query to the original Boolean query, our baseline methods, and title-driven methods (where the working title is used to rank candidate documents). We also evaluate the differences in effectiveness of screening prioritisation between queries generated by various generation models.

5.1.1 Boolean vs. Generated Query. First, we explore the overall effectiveness of neural-ranker-based screening prioritisation using the original Boolean queries versus generated natural language queries. The results suggest that transforming a Boolean query into a natural language query enhances the effectiveness of systematic review screening prioritisation. The only exception to this improvement is seen in CLEF-2019-DTA,[6] when using MAP. When evaluating using the recall and WSS measures, generating a natural language query for screening prioritisation achieves higher effectiveness on ranking non-relevant documents at the bottom of the ranking, as denoted by a higher value of Recall@5%, 10%, 20%, and WSS95, 100; but generally lower effectiveness of Recall@1%.

We also find that fusing the ranking results of the generated query with those from the Boolean query further improves effectiveness. Fusion leads to a significantly better ranking than when

using the Boolean query alone, particularly for CLEF-2017, 2018, and 2019-Intervention. This finding points to the potential benefits of using a fusion of converted natural language and Boolean queries to improve the ranking of systematic review screening.

5.1.2 Neural vs. Baselines. When comparing neural-based rankers with lexical methods, we observe that the results from neural-based rankers significantly outperform those from BM25 or QLM when the Boolean query alone is used to rank candidate documents. There are only two exceptions: one in CLEF-2018 when comparing the rank of the last relevant document (Last_rel), and the other in CLEF-2019-DTA when comparing Recall@1%. Even in these instances, although higher effectiveness was achieved, it did not reach statistical significance. These results highlight the substantial potential of neural rankers for boolean-driven screening prioritisation.

When we compare our approach to the CLF method, which also only uses Boolean queries for screening prioritisation, we find that our methods exhibit statistically higher effectiveness (except for WSS95 at CLEF-2017 and WSS100 at CLEF-2018). However, the margin is narrower than for methods like BM25 or QLM. Similarly, our methods consistently achieved higher effectiveness over the best participation runs from CLEF. However, similar to the previous comparison, the margin is narrower than when compared to BM25 or QLM, and the difference is not statistically significant in terms of MAP, except for the topics in the CLEF-2018 dataset. While our approach only used the Boolean query for screening prioritisation, the top CLEF entries typically utilised additional input sources, such as the final title of the review, which again shows that the neural method could be beneficial to screening prioritisation.

5.1.3 Comparison with using the Working Title. We are able to compare Boolean-driven screening prioritisation with working-title-driven screening prioritisation exclusively through the Seed Collection, as it is the only collection that provides systematic review working titles. To make this comparison, we trained an additional BioBERT ranker that uses the title from the CLEF dataset to prioritise relevant documents, with the same fine-tuning parameters as previous work [65]. Our findings suggest that although past studies have demonstrated substantial improvements in screening effectiveness when using the final review title as a query, this improvement does not extend to working titles. Remarkably, using working titles results in significantly lower effectiveness than Boolean-driven screening prioritisation methods and even underperforms when compared to basic term-matching methods.

5.1.4 ChatGPT vs. Alpaca. Comparing the effectiveness of natural language queries generated by ChatGPT and Alpaca, our results indicate that ChatGPT often outperforms Alpaca in terms of MAP, with the sole exception being the Seed Collection. A significant discrepancy is also noted in CLEF-2019-Intervention, where the natural language query generated by the Alpaca model considerably underperforms compared to both the original Boolean query and the query generated by ChatGPT. This effectiveness drop could be attributed to the difference in systematic review types in the dataset to other datasets (intervention versus DTA). Therefore, the Alpaca model, which has learned to generate queries based on DTA, may not be as effective for intervention topics.

---

[6]Note that the CLEF-2019-DTA dataset is notably smaller, containing only eight topics (30 topics for other datasets on average), which may make it vulnerable to outliers.

Table 2: Evaluation results for comparing methods for Boolean-driven screening prioritisation by generating natural language queries. We use natural language queries generated by ChatGPT and Alpaca, and the fusions of Boolean/ChatGPT and Boolean/Alpaca. Statistical significant differences (Student's two-tailed paired t-test with Bonferroni correction, $p < 0.05$) between using the Boolean query with the BioBERT ranker, and other approaches are indicated by ∗.

Dataset	Query	Ranker	MAP	Last_Rel	Recall@x				WSS95	WSS100
					$x = 1\%$	$x = 5\%$	$x = 10\%$	$x = 20\%$		
CLEF-2017	Boolean	BM25	0.114*	3242.733*	0.083*	0.215*	0.324*	0.491*	0.252*	0.188*
	Boolean	QLM	0.122*	3223.400*	0.073*	0.209*	0.325*	0.476*	0.243*	0.195*
	Boolean	CLF	0.217	3028.033*	0.149	0.341*	0.473*	0.671*	0.442*	0.327*
	Best Participation Run		0.218	2382.467*	0.131	0.332*	0.499*	0.688*	0.488*	0.395*
	Boolean	BioBERT	0.278	1790.867	0.166	0.488	0.656	0.812	0.600	0.536
	ChatGPT	BioBERT	0.293	1991.167	0.150	0.476	0.643	0.801	0.590	0.501
	Boolean/ChatGPT	BioBERT	**0.300***	1843.133	0.170	**0.499**	**0.664**	0.823	0.610	0.532
	Alpaca	BioBERT	0.284	1866.000	0.165	0.435	0.607	0.789	0.591	0.502
	Boolean/Alpaca	BioBERT	0.295	**1759.233**	**0.171**	0.483	0.663	**0.827***	**0.615**	**0.539**
CLEF-2018	Boolean	BM25	0.154*	6033.067*	0.082*	0.242*	0.391*	0.563*	0.361*	0.264*
	Boolean	QLM	0.157*	6097.133*	0.080*	0.252*	0.380*	0.557*	0.384*	0.251*
	Boolean	CLF	0.272*	5743.267*	0.152	0.393*	0.546*	0.729*	0.552*	0.411*
	Best Participation Run		0.258*	5519.200	0.129*	0.383*	0.545*	0.729*	0.552*	0.431
	Boolean	BioBERT	0.353	4830.933	0.202	0.517	0.681	0.845	0.656	0.503
	ChatGPT	BioBERT	0.381	**4508.933**	0.247*	**0.555***	**0.713***	0.865	**0.692***	0.528
	Boolean/ChatGPT	BioBERT	**0.386***	4603.767	**0.247***	0.551*	0.705*	0.859*	0.685*	**0.537***
	Alpaca	BioBERT	0.333	4957.233	0.191	0.493	0.662	0.827	0.640	0.485
	Boolean/Alpaca	BioBERT	0.365	4628.233	0.220	0.525	0.688	0.849	0.668	0.523
CLEF-2019-DTA	Boolean	BM25	0.125*	2766.875*	0.068	0.163*	0.303*	0.463*	0.299*	0.163*
	Boolean	QLM	0.121*	2614.750*	0.042	0.185*	0.278*	0.432*	0.271*	0.180*
	Best Participation Run		0.248	2183.500	0.168	0.439	0.594	0.742	0.490*	0.347*
	Boolean	BioBERT	**0.272**	1146.000	0.174	0.419	0.565	0.751	0.651	0.528
	ChatGPT	BioBERT	0.247	1173.250	0.183	**0.454**	**0.594**	**0.757**	0.660	0.528
	Boolean/ChatGPT	BioBERT	0.268	**1134.375**	**0.183**	0.446	0.584	0.755	**0.665**	**0.545**
	Alpaca	BioBERT	0.241	1217.875	0.170	0.483	0.622	0.784	0.666	0.520
	Boolean/Alpaca	BioBERT	0.251	1146.125	0.173	0.458	0.592	0.783	0.659	0.537
CLEF-2019-Intervention	Boolean	BM25	0.154*	1479.450*	0.070*	0.181*	0.264*	0.417*	0.289*	0.264*
	Boolean	QLM	0.148*	1473.850*	0.041*	0.201*	0.287*	0.450*	0.295*	0.252*
	Best Participation Run		0.293	1132.000	0.165	0.419	0.542	0.722	0.458	0.381*
	Boolean	BioBERT	0.389	1064.300	0.195	0.458	0.619	0.733	0.557	0.499
	ChatGPT	BioBERT	0.433*	993.300	0.217	0.487	**0.654**	0.788*	0.573	0.503
	Boolean/ChatGPT	BioBERT	**0.446***	**975.750**	**0.233**	**0.508***	0.651*	**0.789***	0.578	0.529
	Alpaca	BioBERT	0.317	1087.300	0.120	0.337*	0.510*	0.656	0.491	0.448
	Boolean/Alpaca	BioBERT	0.377	1024.700	0.169	0.460	0.616	0.730	0.551	0.504
Seed Collection	Boolean	BM25	0.087*	990.100*	0.034*	0.140*	0.249*	0.412*	0.252*	0.253*
	Boolean	QLM	0.085*	986.475*	0.018*	0.141*	0.212*	0.397*	0.254*	0.260*
	Working Title	BioBERT	0.059*	976.275*	0.008*	0.070*	0.134*	0.273*	0.266*	0.271*
	Boolean	BioBERT	0.199	785.900	0.085	0.248	0.412	0.600	0.481	0.467
	ChatGPT	BioBERT	0.217	**727.150**	0.082	**0.330***	**0.482***	0.670*	0.530*	0.505
	Boolean/ChatGPT	BioBERT	0.219	744.775*	0.078	0.279	0.468*	0.677*	0.525*	**0.506***
	Alpaca	BioBERT	0.221	780.600	0.083	0.314*	0.473	0.655	0.529*	0.500
	Boolean/Alpaca	BioBERT	**0.230***	765.925	**0.098**	0.268	0.466*	0.661*	**0.531***	0.505*

Similar to the ChatGPT-generated queries, those from Alpaca also achieve higher effectiveness when fused with the results from the Boolean query. This approach leads to higher effectiveness across all datasets compared to using only the Boolean query. Notably, in the Seed Collection, the fusion of derived and Boolean queries achieves significantly higher effectiveness compared to the Boolean query alone. This provides further evidence that the Alpaca model can be trained to generate high-quality natural language queries from Boolean queries, equalling the effectiveness of Chat-GPT. While Alpaca provides a degree of transparency in its process, unlike ChatGPT, making this comparison even more compelling.
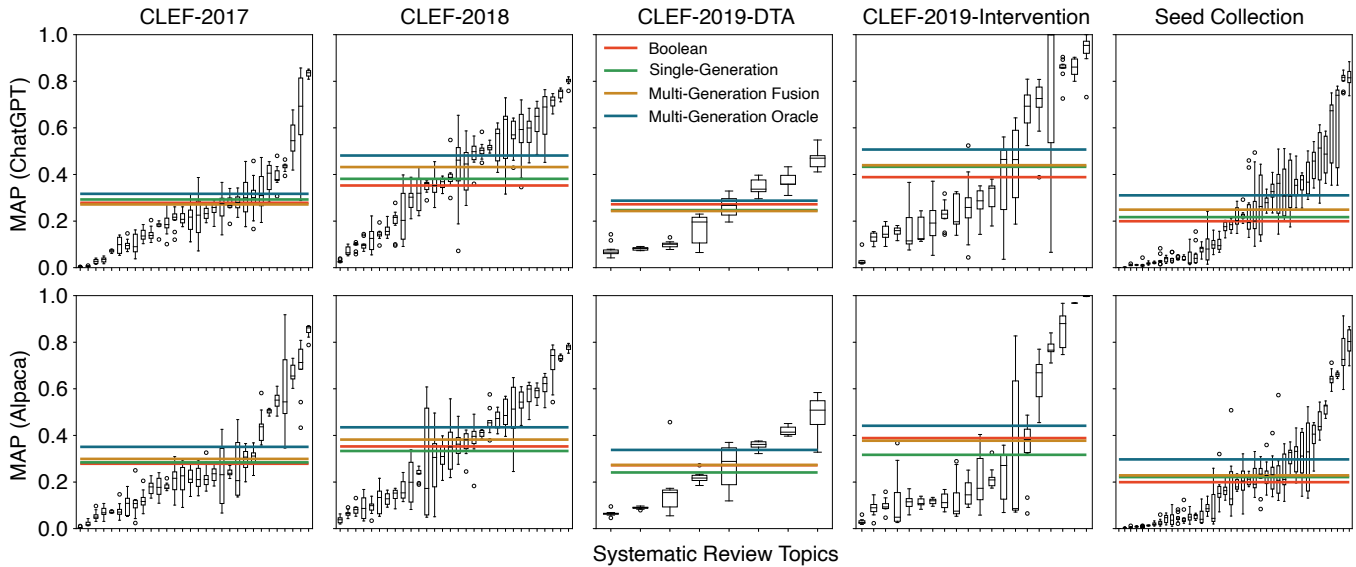
Figure 2: Topic-by-topic variability graph for the effectiveness of the Multi-Generations setup, using a single generated natural language query to rank documents. The coloured horizontal lines indicate the average effectiveness of different methods (Boolean, Single-Generation, Multi-Generation Fusion, and Multi-Generation Oracle).

5.2 Variability and Impact of Multi-Generations

Figure 2 shows the results of multiple generations from both the ChatGPT and Alpaca models. Here, both the average effectiveness and the per-topic effectiveness are measured by the MAP metrics. Our findings reveal high variation in effectiveness when using converted queries from both models, with intervention queries appearing as the most unstable topics. We note that extreme variability in effectiveness has also been observed in previous work for both user-edited and system-generated queries, across a range of search domains [33, 36, 41, 49, 51–53, 63, 73].

In evaluating the variability of effectiveness across different generation models, we observe a difference in terms of different topic types. In the case of DTA topics, the Alpaca model shows a higher degree of variability compared to the ChatGPT model. This is evidenced by a higher variance observed in the CLEF-2017, CLEF-2018, and CLEF-2019-DTA datasets, where the variance of the Alpaca model is 14.3%, 7.7%, and 166% greater than that of the ChatGPT model, respectively. On the other hand, for intervention topics, or topics in the Seed Collection that are not classified, the Alpaca model demonstrates more stability. Specifically, its variance is 39.1%, and 28.6% lower than that of ChatGPT.

Upon examining the average effectiveness, the fusion of multiple generations generally outperforms Single-Generation. Exceptions occur in the CLEF-2017 and CLEF-2019-DTA datasets with ChatGPT queries, and in the CLEF-2019-Intervention dataset with Alpaca queries. Moreover, the fusion of Multi-Generations from the Alpaca model consistently performs better than Boolean queries.

Without a doubt, Multi-Generation Oracle queries consistently achieve the highest effectiveness, marking a considerable margin over the other ranking methods. This tells that with a proper technique or investigation to know how to select the best query over Multi-Generations, it could potentially lead to significant improvements in the effectiveness of screening prioritisation tasks.

5.3 Ablation Studies

To gain deeper insights into why generating a natural language query could yield higher effectiveness, and to understand the role of fusion, and the training process in the effectiveness of screening prioritisation, we conduct a series of ablation studies that investigates these factors.

5.3.1 Generate query vs generate title. In our first ablation experiment, our underlying intuition for generating a natural language query instead of a systematic review title from the Boolean query is that we believe a title may only cover a narrow aspect of the Boolean query. Therefore, if vital information from the title is missed, it could result in lower effectiveness. To test this assumption, we compare generating a systematic review title and a natural language query from a Boolean query for the task of screening prioritisation.

To accomplish this, we first train a cross-encoder BioBERT model using the training portions of each dataset to rank documents using the final review title. For generating the review title, we employ ChatGPT in a zero-shot fashion, as fine-tuning the model is not yet available. For Alpaca, we fine-tune the model using the review titles in the training portion of our dataset using the same parameters as described in Section 3.1, and then test on the testing portion.

Our results, presented in Table 3, clearly demonstrate that generating titles almost always yields lower effectiveness than generating natural language queries, regardless of whether the generation is done using ChatGPT or Alpaca (with the only exceptions being WSS95 on CLEF-2017 and WSS100 on CLEF-2018 when comparing the Alpaca model). Nevertheless, generating titles using ChatGPT appears to be significantly lower than when generated through the Alpaca model, with most results showing statistical significance.

Shuai Wang, Harrisen Scells, Martin Potthast, Bevan Koopman, and Guido Zuccon

Table 3: Results comparing the effectiveness of generating a title (GT) versus generating a natural language query (GQ) from the Boolean query of a systematic review for screening prioritisation. Statistical significant differences ($p < 0.05$) between the effectiveness of a generated title versus a generated natural language query are indicated by $*$.

Dataset	Model	Query	AP	WSS95	WSS100
CLEF-2017	ChatGPT	GQ	**0.293**	**0.590**	**0.501**
	ChatGPT	GT	0.140*	0.486*	0.396*
	Alpaca	GQ	**0.284**	0.591	**0.502**
	Alpaca	GT	0.270	**0.595**	0.502
CLEF-2018	ChatGPT	GQ	**0.381**	**0.692**	**0.528**
	ChatGPT	GT	0.277*	0.626*	0.491
	Alpaca	GQ	**0.333**	**0.640**	0.485
	Alpaca	GT	0.307	0.637	**0.501**
CLEF-2019-DTA	ChatGPT	GQ	**0.247**	**0.660**	**0.528**
	ChatGPT	GT	0.175	0.565*	0.504
	Alpaca	GQ	**0.241**	**0.665**	**0.521**
	Alpaca	GT	0.164	0.544*	0.458
CLEF-2019-Intervention	ChatGPT	GQ	**0.433**	**0.573**	**0.503**
	ChatGPT	GT	0.164*	0.443*	0.404
	Alpaca	GQ	**0.317**	**0.491**	**0.448**
	Alpaca	GT	0.232	0.458	0.408
Seed Collection	ChatGPT	GQ	**0.217**	**0.530**	**0.505**
	ChatGPT	GT	0.127*	0.494	0.490
	Alpaca	GQ	**0.221**	**0.529**	**0.500**
	Alpaca	GT	0.164	0.432*	0.439

5.3.2 Impact of Fusion. We further explore how the fusion of results from both Boolean and generated queries impacts the effectiveness of screening prioritisation. In Figure 3, we compare the effectiveness of using Boolean queries and generated queries separately versus using their fused results for screening prioritisation.

The results indicate that the fusion, on average, consistently outperforms using the generated query alone, but it is not always more effective than using the Boolean queries alone. The effectiveness of Boolean queries should not be overlooked. When comparing results across the two generation models, we observe that the effectiveness gains over Boolean queries obtained tend to be more stable when ChatGPT is used. Using ChatGPT in query generation may thus contribute to more consistent improvements when the results are combined with those from Boolean queries.

5.3.3 Train Ranker using Single-Generation or Multi-Generations. In our experiments, we train our natural language query-based ranker using Single-Generation results from generation models for reproducibility purposes, as Multi-Generations setup do not yield deterministic results each time, even when given the same prompt. However, we are interested in understanding how using Single-Generation versus Multi-Generations impacts the final outcome of the trained ranking model. To explore this, we formulated four distinct training and inference strategies for our downstream ranking model, which we refer to as Single-Train, Multi-Train, Single-Inference, and Multi-Inference.
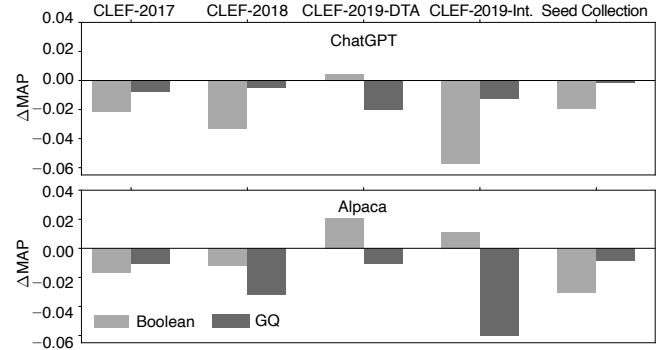
Figure 3: Differences in MAP from Boolean, Generated Query (GQ) to their fused effectiveness.
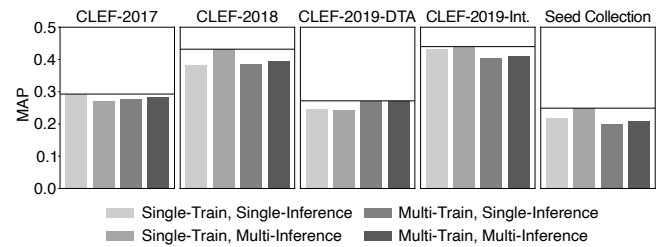
Figure 4: Effectiveness when different training and inference settings are used for ranking candidate documents using the generated natural language query from ChatGPT.

For Single-Train, we train our model using the Single-Generation result from each Boolean query. For Multi-Train, we incorporate all generations from the generation model for each Boolean query in our training data. For Single-Inference, we test our model using a Single-Generation result from each Boolean query. Lastly, for Multi-Inference, we test our model using all generated queries from each Boolean query, and fuse them together.

With the same training parameters applied, we present the resulting effectiveness of screening prioritisation from ChatGPT using a bar chart in Figure 4. From the results, it is apparent that training the neural ranker using multiple creative queries does not typically yield higher effectiveness compared to training on a single deterministic query. The sole exception to this observation is the CLEF-2019-DTA dataset. However, when it comes to Multi-Inference, models generally exhibit improved effectiveness. This implies that the diversity introduced in the inference stage can positively impact the effectiveness of the ranking model, allowing it to generalise better and handle different query formulations. On the other hand, training using a diverse number of generated queries for the same topic may not significantly improve the effectiveness of the ranking model. This is likely due to the model being trained to generalise over multiple query formulations, which could lead to an averaging effect on the learned query-document relevance patterns.

6 SUMMARY OF FINDINGS

Finally, we answer our research questions based on our results:

RQ1: *Comparison of original Boolean queries and generated natural language queries.* We find that generating natural language queries generally results in higher effectiveness than using Boolean queries. This is valid both when the Boolean query is used in the context of the SOTA neural rankers for screening prioritisation, and when used within the previously proposed CLF technique [50], the only published technique for screening prioritisation that explicitly uses the Boolean query for ranking.

We also find that large gains can be obtained when the rankings obtained when using the original Boolean query and the generated natural language query are fused together. This result was obtained when using a simple rank fusion method, CombSUM: further improvements might be possible if using more sophisticated fusion methods [67, 72]. This result suggests that these queries have complementary characteristics that can benefit screening prioritisation.

RQ2: *Impact of generation models.* When the effectiveness of two generation models, ChatGPT and Alpaca, are compared, we observe that ChatGPT consistently generates natural language queries that are more effective in screening prioritisation. The gap in effectiveness is more pronounced for the CLEF-2019-Intervention dataset. This may be attributed to the training of Alpaca models on primary DTA topics, with intervention topics only contained in the test portion. This could have affected Alpaca's ability to perform effectively on the intervention topics. Conversely, ChatGPT, used in a zero-shot fashion, is not specifically tailored towards any topic; thus, its effectiveness is not significantly influenced by different types of systematic reviews.

RQ3: *Impact of ranking methods.* We find that neural methods consistently outperform traditional term-matching methods; they also outperform runs submitted by the research teams that participated in the CLEF TAR shared tasks associated with the datasets we use. This finding highlights the robustness and effectiveness of neural ranking methodologies for the screening prioritisation task.

RQ4: *Effect of Multi-Generations.* We identify considerable variance in the effectiveness of multiple natural language queries generated from both ChatGPT and Alpaca when applied to screening prioritisation. Notably, the Alpaca model tends to generate more unstable queries. However, when the results derived from these diversified queries are integrated, they often outperform the strategy of generating and using just one deterministic query for ranking documents. This occurs in 52.3% of cases when using ChatGPT and in 61.7% of cases when using Alpaca. If we also consider instances where the effectiveness is tied, these percentages increase to 55.5% for ChatGPT and 70.3% for Alpaca.

This finding suggests that the creativity of generative LLMs can enhance the natural language query generation task. Importantly, our findings also indicate that if a method could be implemented to effectively select the highest-performing generated query, the effectiveness of the downstream screening prioritisation task can be significantly improved (Oracle results). This potential for query selection may open new avenues for improving systematic review processes, pointing to the value of research into query performance

predictors for systematic reviews; research on query performance predictors has been substantial in general information retrieval [5], but very scarce in the context of systematic reviews, where common predictors have been shown to be mostly ineffective [47].

RQ5: *Derived natural language queries vs. working titles.* We find that using a systematic review's working title as an input query for screening prioritisation generally results in lower effectiveness when compared to the use of our methods that uses the Boolean query for the same review to derive a natural language query to rank the candidate documents. This is different from when the final titles of the review are used: a practice that is common when experimenting with automation methods for systematic review, but only possible in retrospective evaluation, and not in practise. This discrepancy between working title effectiveness and final title effectiveness may be due to the evolving nature of the review title throughout the research process for the systematic review.

7 CONCLUSION

Our approach to screening prioritisation advances the state of the art by combining the power of large language models, neural rankers, and relying only on information available at the time of screening during the production of a systematic review. Previous work relied instead on the final review title as the query for ranking candidate documents for screening, which is only available at the end of producing a systematic review. This led to overestimated effectiveness scores, as our experiments show. Using instruction-based LLMs to generate queries from the Boolean queries available at the time of screening is competitive with the state of the art using the final title. We also show that improvements in effectiveness can be achieved when rankings based on Boolean queries and generated natural language queries are combined with rank fusion.

Our results also show that while Alpaca, an open-source generation model, can match ChatGPT's effectiveness in some cases, ChatGPT generally produces better natural language queries, leading to more effective screening prioritisation. We also found that multiple generations of natural language queries, while leading to high variance in effectiveness, have the potential to yield a significant increase in effectiveness when effective query performance predictors are available to identify the best query variants, which leaves room for future work.

In summary, this paper has demonstrated the value of instruction-based models in generating and improving queries for screening prioritisation with neural rankers. Our future work involves investigating the potential of combining the query generation capability of instruction-based models with the highly effective ranking capability of neural rankers. In short, we believe that end-to-end training of instruction and ranking models can lead to even higher effectiveness in ranking documents.

ACKNOWLEDGMENTS

REFERENCES

[1] Amal Alharbi, William Briggs, and Mark Stevenson. 2018. Retrieving and ranking studies for systematic reviews: University of Sheffield's approach to CLEF eHealth 2018 Task 2. In *CEUR Workshop Proceedings*, Vol. 2125. CEUR Workshop Proceedings.

[2] Amal Alharbi and Mark Stevenson. 2017. Ranking Abstracts to Identify Relevant Evidence for Systematic Reviews: The University of Sheffield's Approach to CLEF eHealth 2017 Task 2.. In *CLEF (Working Notes)*.

[3] Amal Alharbi and Mark Stevenson. 2019. Ranking studies for systematic reviews using query adaptation: University of Sheffield's approach to CLEF eHealth 2019 task 2 working notes for CLEF 2019. In *Working Notes of CLEF 2019-Conference and Labs of the Evaluation Forum*, Vol. 2380. CEUR Workshop Proceedings.

[4] Antonios Anagnostou, Athanasios Lagopoulos, Grigorios Tsoumakas, and Ioannis P Vlahavas. 2017. Combining Inter-Review Learning-to-Rank and Intra-Review Incremental Training for Title and Abstract Screening in Systematic Reviews.. In *CLEF (Working Notes)*.

[5] Negar Arabzadeh, Maryam Khodabakhsh, and Ebrahim Bagheri. 2021. BERT-QPP: contextualized pre-trained transformers for query performance prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2857–2861.

[6] Wichor M Bramer, Gerdien B De Jonge, Melissa L Rethlefsen, Frans Mast, and Jos Kleijnen. 2018. A systematic approach to searching: an efficient and complete method to develop literature searches. *Journal of the Medical Library Association: JMLA* 106, 4 (2018), 531.

[7] Andres Carvallo, Denis Parra, Hans Lobel, and Alvaro Soto. 2020. Automatic document screening of medical literature using word and text embeddings in an active learning setting. *Scientometrics* 125 (2020), 3047–3084.

[8] Jiayi Chen, Su Chen, Yang Song, Hongyu Liu, Yueyao Wang, Qinmin Hu, Liang He, and Yan Yang. 2017. ECNU at 2017 eHealth Task 2: Technologically Assisted Reviews in Empirical Medicine.. In *CLEF (Working Notes)*.

[9] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341* (2019).

[10] Jonathan J Deeks, Patrick M Bossuyt, Mariska M Leeflang, and Yemisi Takwoingi. 2022. *Cochrane handbook for systematic reviews of diagnostic test accuracy*. John Wiley & Sons.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[12] Giorgio Maria Di Nunzio, Federica Beghini, Federica Vezzani, and Geneviève Henrot. 2017. An Interactive Two-Dimensional Approach to Query Aspects Rewriting in Systematic Reviews. IMS Unipd At CLEF eHealth Task 2.. In *CLEF (Working Notes)*.

[13] Giorgio Maria Di Nunzio, Giacomo Ciuffreda, and Federica Vezzani. 2018. Interactive Sampling for Systematic Reviews. IMS Unipd At CLEF 2018 eHealth Task 2.. In *CLEF (Working Notes)*.

[14] Edward Fox and Joseph Shaw. 1994. Combination of multiple searches. *NIST special publication SP* (1994), 243–243.

[15] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Rethink training of BERT rerankers in multi-stage retrieval pipeline. In *European Conference on Information Retrieval*. Springer, 280–286.

[16] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496* (2022).

[17] Roberto Gozalo-Brizuela and Eduardo C Garrido-Merchan. 2023. ChatGPT is not all you need. A State of the Art Review of large Generative AI models. *arXiv preprint arXiv:2301.04655* (2023).

[18] Ann T Gregory and A Robert Denniss. 2018. An introduction to writing narrative and systematic reviews—Tasks, tips and traps for aspiring authors. *Heart, Lung and Circulation* 27, 7 (2018), 893–898.

[19] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597* (2023).

[20] Julian PT Higgins, James Thomas, Jacqueline Chandler, Miranda Cumpston, Tianjing Li, Matthew J Page, and Vivian A Welch. 2019. *Cochrane handbook for systematic reviews of interventions*. John Wiley & Sons.

[21] Noah Hollmann and Carsten Eickhoff. 2017. Ranking and Feedback-based Stopping for Recall-Centric Document Retrieval.. In *CLEF (Working Notes)*.

[22] Yunjie Ji, Yan Gong, Yiping Peng, Chao Ni, Peiyan Sun, Dongyu Pan, Baochang Ma, and Xiangang Li. 2023. Exploring ChatGPT's Ability to Rank Content: A Preliminary Study on Consistency with Human Preferences. *arXiv preprint arXiv:2303.07610* (2023).

[23] Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983* (2023).

[24] Evangelos Kanoulas, Dan Li, Leif Azzopardi, and Rene Spijker. 2017. CLEF 2017 Technologically Assisted Reviews in Empirical Medicine Overview. In *CLEF'17*.

[25] Evangelos Kanoulas, Dan Li, Leif Azzopardi, and Rene Spijker. 2019. CLEF 2019 technology assisted reviews in empirical medicine overview. In *CEUR workshop proceedings*, Vol. 2380.

[26] Evangelos Kanoulas, Rene Spijker, Dan Li, and Leif Azzopardi. 2018. CLEF 2018 Technology Assisted Reviews in Empirical Medicine Overview. In *CLEF 2018 Evaluation Labs and Workshop: Online Working Notes, CEUR-WS*.

[27] Sarvnaz Karimi, Stefan Pohl, Falk Scholer, Lawrence Cavedon, and Justin Zobel. 2010. Boolean versus ranked querying for biomedical systematic reviews. *BMC medical informatics and decision making* 10, 1 (2010), 1–20.

[28] Jon Karnon, Elizabeth Goyder, Paul Tappenden, Seonaid McPhie, Isabel Towers, John Brazier, and Jason Madan. 2007. A review and critique of modelling in prioritising and designing screening programmes. *HEALTH TECHNOLOGY ASSESSMENT-SOUTHAMPTON-* 11, 52 (2007).

[29] Barbara Kitchenham. 2004. Procedures for performing systematic reviews. *Keele, UK, Keele University* 33, 2004 (2004), 1–26.

[30] Grace Eunkyung Lee. 2017. A study of convolutional neural networks for clinical document classification in systematic reviews: sysreview at CLEF eHealth 2017. (2017).

[31] Grace E. Lee and Aixin Sun. 2018. Seed-driven Document Ranking for Systematic Reviews in Evidence-Based Medicine. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. ACM, New York, NY, USA, 455–464. https://doi.org/10.1145/3209978.3209994

[32] Sharon Levy, Michael Saxon, and William Yang Wang. 2021. Investigating Memorization of Conspiracy Theories in Text Generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 4718–4729. https://doi.org/10.18653/v1/2021.findings-acl.416

[33] Binsheng Liu, Nick Craswell, Xiaolu Lu, Oren Kurland, and J Shane Culpepper. 2019. A comparative analysis of human and automatic query variants. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. 47–50.

[34] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. BioGPT: Generative Pre-Trained Transformer for Biomedical Text Generation and Mining. *Briefings in Bioinformatics* 23, 6 (Nov. 2022), bbac409. https://doi.org/10.1093/bib/bbac409

[35] Adamantios Minas, Athanasios Lagopoulos, and Grigorios Tsoumakas. 2018. Aristotle University's Approach to the Technologically Assisted Reviews in Empirical Medicine Task of the 2018 CLEF eHealth Lab.. In *CLEF (Working Notes)*.

[36] 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 on conference on information and knowledge management*. 1759–1762.

[37] Christopher Norman, Mariska Leeflang, and Aurélie Névéol. 2018. LIMSI@ CLEF eHealth 2018 Task 2: Technology Assisted Reviews by Stacking Active and Static Learning.. In *CLEF (Working Notes)*.

[38] Christopher Norman12, Mariska Leeflang, and Aurélie Névéol. 2017. Limsi@ clef ehealth 2017 task 2: Logistic regression for automatic article ranking. (2017).

[39] Reham Omar, Omij Mangukiya, Panos Kalnis, and Essam Mansour. 2023. Chatgpt versus traditional question answering for knowledge graphs: Current status and future directions towards knowledge graph chatbots. *arXiv preprint arXiv:2302.06466* (2023).

[40] Alison O'Mara-Eves, James Thomas, John McNaught, Makoto Miwa, and Sophia Ananiadou. 2015. Using Text Mining for Study Identification in Systematic Reviews: A Systematic Review of Current Approaches. *Systematic reviews* 4, 1 (2015), 5.

[41] Joao Palotti, Guido Zuccon, Johannes Bernhardt, Allan Hanbury, and Lorraine Goeuriot. 2016. Assessors agreement: A case study across assessor type, payment levels, query variations and relevance dimensions. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 7th International Conference of the CLEF Association, CLEF 2016, Évora, Portugal, September 5-8, 2016, Proceedings 7*. Springer, 40–53.

[42] Jay M. Ponte and W. Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Melbourne, Australia) *(SIGIR '98)*. Association for Computing Machinery, New York, NY, USA, 275–281. https://doi.org/10.1145/290941.291008

[43] Radim Rehurek, Petr Sojka, et al. 2011. Gensim—statistical semantics in python. *Retrieved from genism. org* (2011).

[44] Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*. Springer, 232–241.

[45] Ahlam A Saleh, Melissa A Ratajeski, and Marnie Bertolet. 2014. Grey literature searching for health sciences systematic reviews: a prospective study of time spent and resources utilized. *Evidence based library and information practice* 9, 3 (2014), 28.

[46] Malik Sallam. 2023. ChatGPT utility in healthcare education, research, and practice: systematic review on the promising perspectives and valid concerns. In *Healthcare*, Vol. 11. MDPI, 887.

[47] Harrisen Scells, Leif Azzopardi, Guido Zuccon, and Bevan Koopman. 2018. Query variation performance prediction for systematic reviews. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1089–1092.

[48] Harrisen Scells, Guido Zuccon, Anthony Deacon, and Bevan Koopman. 2017. QUT ielab at CLEF eHealth 2017 technology assisted reviews track: Initial experiments with learning to rank. In *CEUR Workshop Proceedings: Working Notes of CLEF 2017: Conference and Labs of the Evaluation Forum*, Vol. 1866. CEUR Workshop Proceedings, Paper–98.

[49] Harrisen Scells, Guido Zuccon, and Bevan Koopman. 2019. Automatic Boolean query refinement for systematic review literature search. In *The world wide web conference*. 1646–1656.

[50] Harrisen Scells, Guido Zuccon, and Bevan Koopman. 2020. You Can Teach an Old Dog New Tricks: Rank Fusion applied to Coordination Level Matching for Ranking in Systematic Reviews. In *European Conference on Information Retrieval*. Springer, 399–414.

[51] Harrisen Scells, Guido Zuccon, and Bevan Koopman. 2021. A comparison of automatic Boolean query formulation for systematic reviews. *Information Retrieval Journal* 24 (2021), 3–28.

[52] Harrisen Scells, Guido Zuccon, Bevan Koopman, and Justin Clark. 2020. Automatic boolean query formulation for systematic review literature search. In *Proceedings of the web conference 2020*. 1071–1081.

[53] Harrisen Scells, Guido Zuccon, Mohamed A Sharaf, and Bevan Koopman. 2020. Sampling Query Variations for Learning to Rank to Improve Automatic Boolean Query Generation in Systematic Reviews. In *Proceedings of The Web Conference 2020*. 3041–3048.

[54] Jaspreet Singh and Lini Thomas. 2017. IIIT-H at CLEF eHealth 2017 Task 2: Technologically Assisted Reviews in Empirical Medicine.. In *CLEF (Working Notes)*.

[55] Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent. *arXiv preprint arXiv:2304.09542* (2023).

[56] Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. 2023. Evaluation of ChatGPT as a question answering system for answering complex questions. *arXiv preprint arXiv:2303.07992* (2023).

[57] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

[58] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[59] David Tranfield, David Denyer, and Palminder Smart. 2003. Towards a methodology for developing evidence-informed management knowledge by means of systematic review. *British journal of management* 14, 3 (2003), 207–222.

[60] Andrea C Tricco, Jesmin Antony, Wasifa Zarin, Lisa Strifler, Marco Ghassemi, John Ivory, Laure Perrier, Brian Hutton, David Moher, and Sharon E Straus. 2015. A scoping review of rapid review methods. *BMC medicine* 13, 1 (2015), 1–15.

[61] Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query Expansion with Large Language Models. *arXiv preprint arXiv:2303.07678* (2023).

[62] Shuai Wang, Harrisen Scells, Justin Clark, Bevan Koopman, and Guido Zuccon. 2022. From little things big things grow: A collection with seed studies for medical systematic review literature search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3176–3186.

[63] Shuai Wang, Harrisen Scells, Bevan Koopman, and Guido Zuccon. 2022. Automated MeSH Term Suggestion for Effective Query Formulation in Systematic Reviews Literature Search. *Intelligent Systems with Applications* (2022), 200141.

[64] Shuai Wang, Harrisen Scells, Bevan Koopman, and Guido Zuccon. 2023. Can ChatGPT write a good boolean query for systematic review literature search? *arXiv preprint arXiv:2302.03495* (2023).

[65] Shuai Wang, Harrisen Scells, Bevan Koopman, and Guido Zuccon. 2023. Neural Rankers for Effective Screening Prioritisation in Medical Systematic Review Literature Search. In *Proceedings of the 26th Australasian Document Computing Symposium* (Adelaide, SA, Australia) *(ADCS '22)*. Association for Computing Machinery, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.1145/3572960.3572980

[66] Shuai Wang, Harrisen Scells, Ahmed Mourad, and Guido Zuccon. 2022. Seed-driven document ranking for systematic reviews: A reproducibility study. In *European Conference on Information Retrieval*. Springer, 686–700.

[67] Shuai Wang, Shengyao Zhuang, and Guido Zuccon. 2021. Bert-based dense retrievers require interpolation with bm25 for effective passage retrieval. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 317–324.

[68] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560* (2022).

[69] Eugene Yang, Sean MacAvaney, David D Lewis, and Ophir Frieder. 2022. Goldilocks: Just-right tuning of bert for technology-assisted review. In *European Conference on Information Retrieval*. Springer, 502–517.

[70] Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Richard James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Retrieval-augmented multimodal language modeling. (2023).

[71] Zhe Yu and Tim Menzies. 2017. Data Balancing for Technologically Assisted Reviews: Undersampling or Reweighting.. In *CLEF (Working Notes)*.

[72] Xin Zhou, Adrien Depeursinge, and Henning Muller. 2010. Information fusion for combining visual and textual image retrieval. In *2010 20th International Conference on Pattern Recognition*. IEEE, 1590–1593.

[73] Guido Zuccon, Joao Palotti, and Allan Hanbury. 2016. Query variations and their effect on comparing information retrieval systems. In *Proceedings of the 25th ACM international on conference on information and knowledge management*. 691–700.