Smooth Operators for Effective Systematic Review Queries

Harrisen Scells
Leipzig University

Ferdinand Schlatt
Friedrich-Schiller-Universität Jena

Martin Potthast
Leipzig University and ScaDS.AI

ABSTRACT

Effective queries are crucial to minimising the time and cost of medical systematic reviews, as all retrieved documents must be judged for relevance. Boolean queries, developed by expert librarians, are the standard for systematic reviews. They guarantee reproducible and verifiable retrieval and more control than free-text queries. However, the result sets of Boolean queries are unranked and difficult to control due to the strict Boolean operators. We address these problems in a single unified retrieval model by formulating a class of smooth operators that are compatible with and extend existing Boolean operators. Our smooth operators overcome several shortcomings of previous extensions of the Boolean retrieval model. In particular, our operators are independent of the underlying ranking function, so that exact-match and large language model rankers can be combined in the same query. We found that replacing Boolean operators with equivalent or similar smooth operators often improves the effectiveness of queries. Their properties make tuning a query to precision or recall intuitive and allow greater control over how documents are retrieved. This additional control leads to more effective queries and reduces the cost of systematic reviews.

CCS CONCEPTS

• **Information systems** → **Specialized information retrieval**; **Query reformulation**.

KEYWORDS

systematic reviews, Boolean queries, retrieval models

ACM Reference Format:
Harrisen Scells, Ferdinand Schlatt, and Martin Potthast. 2023. Smooth Operators for Effective Systematic Review Queries. 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, 11 pages. https://doi.org/10.1145/3539618.3591768

1 INTRODUCTION

Developing effective queries to search the medical literature for systematic reviews takes a lot of time and effort. Systematic reviews in medicine aim to comprehensively synthesise all relevant literature on clearly defined research questions. Therefore, the query used to search the literature must also be comprehensive. This requirement has its price, because systematic reviews cost around 140,000 USD and take one to two years to complete [27]. Especially the screening phase is time-consuming [14]. It involves manually checking documents (in this case, abstracts of medical articles) retrieved by a Boolean query for relevance to the review. Most research in this area focuses on methods that directly accelerate the screening phase; for example, active learning [8, 29, 55], text mining [3, 10, 31, 50, 52], and document ranking [20, 26, 54]. We take a new approach and develop a method to directly improve the effectiveness of queries used to search for the literature of systematic reviews. This approach complements previous approaches in that a more effective query retrieves fewer documents to review.

The Boolean retrieval model is a powerful tool for formulating complex queries to retrieve relevant documents, but it has some long-known shortcomings [39, 56]: (1) A lack of control over the number of retrieved documents. Hence, finding a precise Boolean query that retrieves a specific number of documents within a screening budget comes down to an expensive iterative trial and error process [7]. (2) Retrieved documents are not ranked. This means that all documents are equally important, and there is no way to reliably and automatically determine a cut-off for non-relevance [35]. (3) Boolean operator strictness leads to unexpected results. For example, a document retrieved with one of four terms in an OR subquery is as important as a document retrieved with all four terms, while a document retrieved with three of four terms in an AND subquery is not retrieved. Extensions addressing the problems of the Boolean retrieval model [4, 40] have their own shortcomings and do not take recent advances in information retrieval, such as the use of large pre-trained language models, into account.

This paper proposes an extension to the Boolean retrieval model called the smooth operator model, which addresses the three outlined shortcomings (Section 3): the ability to control to what extent a document should be retrieved by a set of terms, the ability to rank documents according to how likely they should be retrieved, and the ability to express lenient or smooth Boolean operators. The main intuition for the smooth operator model is that a document's relevance for one part of a query should depend on its other parts. Our investigation into the model (Section 4) includes determining the effectiveness of smooth operators in improving the recall and precision of Boolean queries for systematic review literature search; ascertaining that the integration of pre-trained language models enhances effectiveness; and examining the strengths and limitations of smooth operators in different types of systematic reviews. The main findings (Section 5) of the study are that the smooth operator model has intuitive properties that enable optimisation of recall or precision, but determining the appropriate degree of smoothness is challenging. Despite this challenge, more effective queries are possible when incorporating basic ranking models, which turn out to be more effective and efficient than neural models based on pre-trained language models.[1,2]

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

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591768

[1] Our runs are available to download at https://dx.doi.org/10.5281/zenodo.7870122
[2] Our code is available at https://github.com/webis-de/SIGIR-23

2 BACKGROUND & RELATED WORK

The smooth operators we introduce are based on principles from two well-researched areas of information retrieval: the Boolean retrieval model and its extensions, and rank fusion. Beyond that, few attempts have been made to improve Boolean queries.

2.1 Basics of the Boolean Retrieval Model

Let $T = \{t_1, \ldots, t_m\}$ denote a set of m index terms, called terminology, and $D = \{d_1, \ldots, d_n\}$ denote a set of n documents, called document collection. Under the Boolean retrieval model, a real document $d \in D$ is represented as a subset of T, i.e. $d \subseteq T$.

A Boolean query q is a well-formed propositional formula in which the terms T are atoms and in which the operators AND, OR and NOT are usually used. The query language, like the propositional calculus, is inductively defined by the following rules:

(1) Every index term $t \in T$ is a query.
(2) If q is a query, so is NOT q.
(3) If q_1 and q_2 are queries, so are (q_1 AND q_2) and (q_1 OR q_2).
(4) Only expressions formed by the Rules (1)-(3) are queries.

Let Q be the set of all queries that can be formulated in the query language. Due to Rule (1), the terminology is a subset of all queries, i.e. $T \subset Q$. A well-formed Boolean query q can be split into an abstract syntax tree, where the nodes are operators and the leaves are atoms (see Figure 1). Each subtree of the syntax tree of q is also a well-formed boolean query in Q, which we call subquery.

To determine whether a document $d \in D$ is retrieved by a query $q \in Q$ under the Boolean retrieval model, its retrieval status value (RSV) is calculated [56]. This corresponds to determining whether q is satisfied by d. For this purpose, the semantics of q are defined using an interpretation function $\mathcal{I}_d : Q \to \{0, 1\}$, where 0 indicates that q is false for d and 1 that a query q is true for d. Given the query language above, $\mathcal{I}$ is recursively defined as follows:

$$\mathcal{I}_d(t) = \begin{cases} 1 & \text{if } t \in d, \\ 0 & \text{otherwise;} \end{cases}$$

$$\mathcal{I}_d(\text{NOT } q) = \begin{cases} 1 & \text{if } \mathcal{I}_d(q) = 0, \\ 0 & \text{otherwise;} \end{cases}$$

$$\mathcal{I}_d(q_1 \text{ AND } q_2) = \begin{cases} 1 & \text{if } \mathcal{I}_d(q_1) = \mathcal{I}_d(q_2) = 1, \\ 0 & \text{otherwise;} \end{cases}$$

$$\mathcal{I}_d(q_1 \text{ OR } q_2) = \begin{cases} 1 & \text{if } \mathcal{I}_d(q_1) = 1 \text{ or } \mathcal{I}_d(q_2) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

To retrieve the set of documents $D_q \subseteq D$ for which a query q is true, the RSV is calculated for each document in D, which is efficiently implemented using an inverted index of D.

2.2 Extensions of the Boolean Retrieval Model

The Boolean retrieval model is strict in how it assigns an RSV to a document. In particular, the AND operator has a strong influence. For example, if all but one subquery q_i of a query $q = q_1$ AND q_2 AND ... AND q_l are true for a document d, its RSV is 0. Conversely, the OR operator is often too broad and retrieves many more documents than necessary.

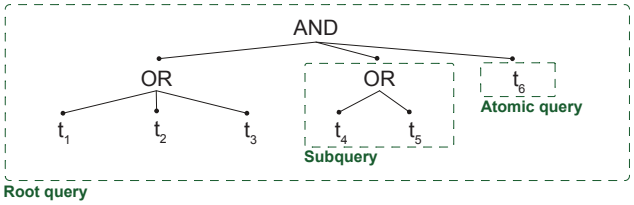


Figure 1: Boolean query syntax with annotated terminology.

To overcome such and similar limitations, many extensions to the Boolean retrieval model have been proposed that allow better control over the set of retrieved documents D_q. All extensions have in common that they replace the binary membership of a document d to the result set D_q for a query q by a gradual membership measure. This is usually achieved with a relevance function $\rho : D \times Q \to [0, 1]$, where 0 indicates no relevance and 1 maximum relevance.

Fuzzy set theory [62] has been extensively studied to express relationships between continuous relevance functions, most notably the fuzzy set retrieval model [5, 6]. The relationships between fuzzy sets are expressed with operators that correspond to those of the Boolean retrieval model, e.g. MIN and MAX instead of OR and AND, respectively. Since the fuzzy retrieval model assigns a continuous RSV to the documents, the total number of retrieved documents can be controlled with the help of a threshold. However, the fuzzy retrieval model has a major weakness that makes it less practical than the Boolean retrieval model. Under the original fuzzy set retrieval model, documents retrieved from all but one subquery in an AND expression are equivalent to documents retrieved from none, and documents retrieved from one subquery in an OR expression are equivalent to those retrieved from all subqueries. This weakness was addressed in later research by term weighting [32, 56]; however, the resulting operators have undesirable properties [36].

The extended Boolean model [39] is a generalisation of these concepts. Its basis of the relevance function is the L_p vector norm. Queries and documents are represented as vectors in an n-dimensional vector space, where each dimension typically corresponds to an index term, i.e. a controlled vocabulary of terms—an example in systematic review literature search is the MeSH vocabulary [23, 30]. A parameter between 1 and infinity controls the retrieval of documents. As it decreases, retrieval becomes an inner product between the query and document. As it approaches infinity, retrieval becomes equivalent to either the fuzzy-set or strict Boolean models, depending on how terms are represented (i.e., weighted or binary). However, while non-binary term weights can lead to more effective retrieval, choosing term weights is a complex problem, and manually assigning term weights can have undesirable or unintuitive effects on the result set of queries [51].

Many issues with extensions to the Boolean retrieval model arise from properties relating to term weighting. Rather than investigating more effective term weighting schemes, we take a different approach to relaxing Boolean operators. We estimate how likely a document is to a query indirectly using the output of a ranking function such as BM25 or a neural ranking function. This separation allows intermixing multiple ranking functions in a Boolean query.

2.3 Rank Fusion

The goal of rank fusion is to combine the result lists from multiple systems or queries into a single ranked list [49]. Rank fusion is a well-studied area of information retrieval and is well-known to improve ranking effectiveness. There are two main categories of rank fusion algorithms [16]: score-based [49] and rank-based [25].

Score-based rank fusion algorithms depend on the relevance scores ascribed to documents by the retrieval model or system. Naturally, scores must be normalised before applying a score-based rank fusion algorithm. The fused list is also heavily dependent on the score distributions of documents. Rank-based rank fusion algorithms instead directly depend on the rank positions of documents within each ranking, ignoring the score. These algorithms do not require any normalisation. Given the dependency of score-based rank fusion algorithms on document scores, we focus our attention on rank-based algorithms in this work. The use of rank-based fusion algorithms makes the document ranking of the smooth operator model independent of the underlying ranking model to score documents retrieved by each atomic query.

2.4 Improving Boolean Queries

Improving the effectiveness of systematic review literature search queries has been researched previously in several directions. Scells et al. [44, 48] investigated an automatic method that made modifications to queries such as the addition or removal of subqueries, changes in Boolean operators, or changes to which fields queries are restricted to searching. This method was shown to automatically improve the effectiveness of queries for systematic review literature search. However, the extent to which queries could be modified was not controllable, nor was it clear how many modifications should be made to a query before stopping. In other words, in practice, users would have no control over how much their queries could change. Pohl et al. [34] translated a small number of Boolean queries for systematic review literature into the extended Boolean model. However, they found that binary weights were more effective than more complicated weighting schemes, suggesting that choosing appropriate term weights is a challenging task, especially for the complex queries used for systematic review literature search. Wang et al. [57, 59] developed several methods for automatically suggesting MeSH terms for queries. These methods can suggest more effective MeSH terms for a query; however, the choice of the suggestion method is critical to the effectiveness of the query. In practice, there is no way to tell which MeSH suggestion method is the most effective a priori. There has also been some prior research investigating the automatic formulation of Boolean queries for different contexts, such as for systematic review literature search [47] and professional search [21]. Similar to the MeSH suggestion problem, where a modification's effect on a query is unknown, predicting which terms to add or remove to a query is difficult and often requires human expertise. The choice of terms can have a significant impact on the effectiveness of a query. Our smooth operator model has the advantage that queries can be broadened or restricted independently of the terms chosen, meaning that the choice of terms is less important for effective queries.

3 SMOOTH OPERATOR MODEL

There are two main components to the smooth operator model: the first is a probabilistic model that predicts the likelihood that a document is relevant to the children of a subquery; the second is a rank fusion model to rank documents and prevent ties.

3.1 Calculating the RSV for a Document

The intuition for the smooth operator model is that, given a root query q or a subquery q_i thereof, the RSV for a document d depends on the first-level children of q or q_i (see Figure 1). In other words, the more children that retrieve a document and the higher a document is ranked by each child, the more likely that document is relevant to the (sub)query q. We model the RSV as a probability that the children of a (sub)query retrieve a document. The RSV computed by $\rho(d, q)$ is the extent to which document d should belong to a (sub)query q:

$$\rho(d, q) = P(d|q) = \frac{P(d)P(q|d)}{P(q)},$$

The final value of $\rho(d, q)$ requires a recursive calculation for each subquery in a query in a bottom-up fashion. As a subquery comprises child queries (i.e., atomic queries or further subqueries), the probability of a document retrieved by a subquery necessarily depends on its children $q_i \in q$:

$$P(d|q_1, ..., q_k) = \frac{P(d)\prod P(q_i|d)}{P(d)\prod P(q_i|d) + P(\bar{d})\prod P(q_i|\bar{d})},$$

where $P(\bar{d}) = 1 - P(d)$, and likewise for $P(q|\bar{d})$.[3] This leaves two probability estimations: the prior probability $P(d)$ and the conditional probabilities $P(q_i|d)$ for each child. Intuitively, $P(d)$ can be reasoned as the collective contribution of all children to the document being retrieved. Meanwhile, $P(q_i|d)$ can be reasoned as the individual contribution by each child to the document being retrieved. We estimate $P(d)$ as the ratio of children that retrieve d:

$$P(d) = \frac{|\{\forall q_i \in q \ : \ d \in D_{q_i}\}|}{|q|},$$

where D_{q_i} are the documents retrieved by q_i and $|q|$ is the number of children contained in q. This probability models the intuition of coordination level matching [24], which assumes that the more children retrieve a document, the more likely that document is relevant. With this prior probability calculated, one now must calculate the contribution from each child query for a document.

$P(q_i|d)$ is estimated as the relevance between q_i and d, i.e., using the intuition from the probability ranking theory [38], which assumes that the higher ranked a document is in response to a query, the more relevant it is. In other words, $P(q_i|d)$ is the inverse position that d appears in a ranking for a child query q_i:

$$P(q_i|d) = 1 - \frac{pos(q_i, d)}{|D_{q_i}|},$$

where $pos(q_i, d)$ is the rank position of d in response to issuing q_i to a retrieval system. If the child is an atomic query, documents can be ranked by any retrieval model, e.g., BM25. If the child is another subquery, the documents will already be ranked, as assigning RSVs is bottom-up recursive.

[3] The Naive Bayes assumption shows that $P(\bar{d})$ and $P(q|\bar{d})$ are appropriately defined.

Table 1: Equivalents to the Boolean operators as expressed in the smooth operator model. Note that the NOT operator is nested: the inner function is an exclusive OR, and the outer function ANDs the result with the left-most query.

Boolean operator	smooth Boolean equivalent
OR	$f(\rho(d,q) \geq 0)$
AND	$f(\rho(d,q) = 1)$
NOT	$f(f(\rho(d,q) < 1) = 1)$

3.2 Defining Smooth Operators as RSV Cut-offs

Once all documents retrieved by the children of a subquery have been assigned an RSV, the documents are ranked by their RSV. Documents with a high RSV are more likely to be retrieved by all children of a subquery than documents with a low RSV. A smooth operator is a function that applies a cut-off, or threshold, to the ranked list of documents. Formally, a smooth operator takes the form $f()$, which binarises the RSV from a continuous value into a 1 or a 0; e.g., $f(\rho(d,q) > \theta)$, where θ is the desired RSV cut-off. The result set can be manipulated depending on the comparator.

One main advantage of the way result sets are cut-off is that when no cut-offs are applied, the smooth operator model is equivalent in retrieval to the Boolean retrieval model. Table 1 shows how the three Boolean operators OR, AND, and NOT can all be expressed as equivalent smooth functions. Figure 2 provides a visual intuition for how the smooth Boolean equivalent operators are smoothened to control the result set size.

One limitation of the smooth operator model is that $P(q|d)$ is dependent on the rank position of d. When $P(q_i|d) = 1$, the document is in the first rank position. If all other children also retrieve the document regardless of rank, the final RSV of this document will be 1. As each child query is likely to have a different document in the first-rank position, several documents can obtain the same RSV. In other words, multiple documents will likely have a tied RSV. Ties are not a problem for smoothing the result set size: intuitively, two independent documents ranked in the first position by two separate child queries are equally important. However, problems arise when inferring the rank positions for subqueries. A retrieval system cannot directly rank documents for subqueries and ties arise from the RSVs. We overcome these problems by independently ranking documents using rank fusion.

3.3 Breaking Ties with Rank Fusion

Rather than relying on the RSV for ranking document, which is highly likely to contain ties, we independently score documents using rank fusion in a way that ensures ties are less likely. Scells et al. [45] devised a ranking mechanism for Boolean queries, which fused document rankings bottom-up through the query. Rather than set-based retrieval, a rank-based retrieval model was applied to each atomic query. During document scoring, document rankings from each sub-query were fused to produce a single ranked list for a Boolean subquery. In that work, different rank fusion algorithms were applied depending on the Boolean operator, i.e., CombMNZ for OR and CombSUM for AND. In our work, we use a single rank fusion algorithm for ranking documents.

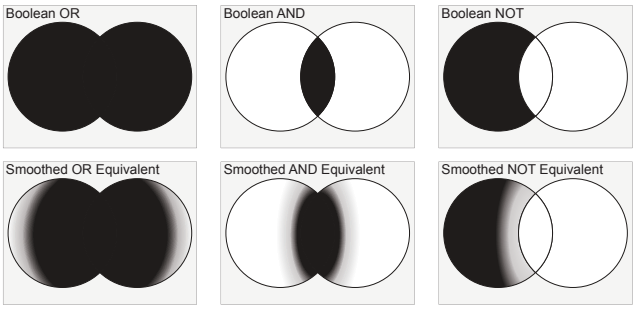


Figure 2: Boolean operators (above) and instantiations of smoothed operators in the smooth operator model (bottom). Note that the smooth NOT operator is a combination of two operators: NOT+AND. This is also how the NOT operator is expressed in Boolean queries (described in Table 1).

We propose a modification on the reciprocal rank fusion (RRF) [9] algorithm, which we denote RRFMNZ. The original RRF algorithm sums the reciprocal rank of a document among rankings, with some additional smoothing factor. We additionally multiply this sum by the number of rankings the document appears in:

$$RRFMNZ(d \in D) = |d \in R| \cdot \sum_{r \in R} \frac{1}{r(d) + k},$$

where D is the set of documents to be ranked, R are the rankings to be fused, $r(d)$ is the rank position of d in r, and k is a smoothing factor that mitigates high-ranking outlier documents. Intuitively, RRFMNZ lessens the dark horse effect [53] (i.e., outlier documents ranked highly by a single child query) and boosts the chorus effect [53] (i.e., documents retrieved by multiple children should be ranked higher). Initial experiments found that higher values of k are the most effective, so we set this parameter to 10,000.

4 EXPERIMENTAL SETUP

Test Collections. The primary objective of this research is to improve the retrieval effectiveness of systematic review literature search queries. Therefore, we use three information retrieval test collections of systematic review topics. We use two test collections from the CLEF Technology Assisted Reviews (TAR) track [17, 19]. These are referred to as the CLEF TAR 2017 and 2018 collections. We only use the 'testing' portions of these collections for our evaluation. We also use the seed study collection from Wang et al. [58]. We have chosen these test collections as they contain systematic reviews of different types and investigate the differences between them.

For our experiments, the queries must be executable in a retrieval system. We do not simply re-rank sets of already retrieved documents. To obtain comparable results, all the queries must be in the same format and executed within the same retrieval system. Since certain documents only exist in certain literature databases, several search engines are used in systematic review literature searches. For example, queries in the CLEF TAR collections are either executable in PubMed or Ovid. Both are proprietary search systems; however, PubMed makes much of the data freely available. Therefore, we convert all queries in the Ovid format into the PubMed format. While this conversion may not be perfect (i.e., Ovid has some fields

and operators that are not present in PubMed), this is what has been done in the literature [46, 57]. The Wang et al. [58] collection already contains queries specifically formulated for PubMed.

Training Collection. Our investigation includes training models for predicting suitable θ values. As we wanted to use the same classification model across all collections, we decided to use queries from a dataset of query logs as training data [42]. This query log contains approximately 6,000 queries, with a portion of the queries also containing references to potentially relevant documents that users self-defined to aid in the searching process. We filter these queries to only those containing these self-defined documents in PubMed format. The queries are also grouped into sessions, so to ensure training diversity while utilising as much data as possible, we use the first and last query in a session (so long as they are different). This leaves 362 queries for training, far more than if we were to perform, for example, leave one out cross-validation.

Retrieval Methods. The smooth operator model requires that documents retrieved by an atomic query are ranked. While the model imposes no restrictions on the method used to rank documents, we focus our efforts on two methods: BM25 and neural rankers derived from large pre-trained language models. BM25 is also used to retrieve an initial document ranking to be re-ranked by the neural rankers. The initial ranking for neural rankers uses the smooth Boolean equivalents. We use `pybool_ir` [43] to index and search the 2022 baseline PubMed collection. We use the default BM25 implementation in Lucene.

For neural re-ranking, we use a BERT-based cross-encoder architecture. The model receives a tokenised query and a document as input and outputs a relevance score. It is trained to output higher relevance scores for more relevant documents. This architecture has achieved state-of-the-art effectiveness [15] on TripClick [37], a medical document retrieval benchmark. The benchmark's license restricts the release of fine-tuned models. We, therefore, fine-tune three models based on pre-trained PubMedBERT [13], BERT [12], and DistilBERT [41] using the training triples released by Hofstätter et al. [15]. We train the models for 150,000 steps using the AdamW optimiser with a learning rate of $7 \cdot 10^{-6}$ and batch size 16. We chose PubMedBERT as it was pre-trained using medical terminology and should have a good representation of documents in PubMed. BERT and DistilBERT are used to compare model size and effectiveness.

During re-ranking, we opted to use the review title as the query to the re-ranker for each atomic query rather than the atomic query itself. These neural models may not be able to rank documents effectively when most atomic queries consist of only a single term. Additionally, only the top 100,000 documents for each atomic query are re-ranked to save computing resources. The total number of atomic queries that required re-ranking was 5,910.

Evaluation. Evaluation is performed using the documents included in the systematic reviews. We measure the effectiveness of our smooth operator model using recall, precision, and f-measure. As the smooth operator model also ranks documents, we employ recall@k and nDCG@k where $k \in \{100, 1000\}$.

Comparing our method to other methods in the literature is surprisingly difficult. Methods that automatically rewrite queries to make them more effective [2, 44, 48] use public test collections but have not shared their run files. In the CLEF TAR shared task, most submissions took part in the screening task [17–19]. This task involved re-ranking an already retrieved set of documents. It entirely disregards our task at hand, the retrieval step. Only four participants submitted entries to the retrieval subtask: AUTH, ECNU, SHEF, and UW. We include their runs as comparison baselines. We are unaware of other studies in the literature that perform tasks similar to what we perform that have publicly available results (i.e., run files) that we can fairly compare.

5 RESULTS

The following research questions guide our investigation.

RQ1 How effective is the smooth operator model at improving the effectiveness of existing systematic review literature search Boolean queries?

RQ2 Can the use of large pre-trained language models in conjunction with smooth Boolean operators enhance the effectiveness of systematic review literature search queries?

RQ3 How effective are smooth Boolean operators for systematic review literature searches across different types of systematic reviews, and what factors influence potential differences in effectiveness?

The first research question investigates instantiations of the smooth operators to determine if they can improve the retrieval effectiveness of queries. The second research question investigates use of neural rankers based on pre-trained language models to determine whether they can further improve the retrieval effectiveness of queries. The third research question investigates the strengths and limitations of smooth operators on different types of systematic reviews (i.e., intervention vs. diagnostic test accuracy).

5.1 Integration of Smooth Operators

We first investigate **RQ1**: *How effective is the smooth operator model at improving the effectiveness of existing systematic review literature search Boolean queries?* This research question is fundamentally concerned with whether the smooth operators can be integrated into existing query syntax for systematic review literature search queries. We split the investigation into this research question into three experiments: first, we perform an ablation study to understand how smoothing each Boolean operator impacts retrieval effectiveness; secondly we use an oracle to find appropriate θ parameters for each individual operator in the query; finally, we attempt to predict appropriate θ values. In the first experiment, we aim to understand if queries can be made more effective by globally modifying the smoothness of all operators of one type in a query. From a user perspective, this translates to a user that simply wants a more specific or broad query without tuning each operator in the query. In the second experiment, we aim to understand if queries can be made even more effective by tuning the smoothness of the operators for each subquery in the query. From a user perspective, this translates to a user that wants to carefully control the size of their result set without wanting to modify the syntactic or semantic properties of the query. In the third experiment, we aim to understand if it is possible to predict appropriate θ values. From a user perspective, this translates to a user that is uninterested in carefully tuning their query for effectiveness, but still desires a more effective query.

Table 2: Results of the ablation study, oracle, predictor, and neural methods. For each collection, a two-sided t-test; $p < 0.05$ with Bonferroni correction was performed between each method and the 'smooth Boolean equivalents' from Table 1. Rows in grey are runs from CLEF TAR 2018 that are not directly comparable to our results as they relied on explicit feedback mechanisms.

		Recall	R@100	R@1000	Precision	$F_{0.5}$	F_1	F_3	nDCG	nDCG@100	nDCG@1000
Wang et al.	Boolean operators	0.7149	-	-	0.0362	0.0509	0.0642	0.1081	-	-	-
	BM25 Title	0.7149	0.2082*	0.5325*	0.0362	0.0509	0.0642	0.1081	0.2605*	0.0972*	0.2071*
	smooth Boolean equivalents	0.7149	0.3787	0.6681	0.0362	0.0509	0.0642	0.1081	0.3675	0.2486	0.3503
	AND $\rightarrow f(\rho(d,q) \geq 0.99)$	0.7206	0.3472	0.6468	0.0019*	0.0029*	0.0038*	0.0075*	0.3595	0.2331	0.3357
	AND $\rightarrow f(\rho(d,q) \geq 0.9)$	0.7658*	0.3238	0.5753	0.0002*	0.0002*	0.0003*	0.0006*	0.3552	0.2180	0.3055*
	OR $\rightarrow f(\rho(d,q) \geq 0.01)$	0.3078*	0.2024*	0.2969*	0.0366	0.0484	0.0582	0.0861	0.1998*	0.1604*	0.1965*
	OR $\rightarrow f(\rho(d,q) \geq 0.1)$	0.0612*	0.0562*	0.0599*	0.0486	0.0419	0.0417	0.0438*	0.0637*	0.0620*	0.0632*
	Predictor	0.3876*	0.2315*	0.3764*	0.0364	0.0471	0.0566	0.0865	0.2189*	0.1651*	0.2150*
	Oracle	0.7437	0.4084	0.6989	0.0440	0.0613*	0.0769*	0.1275*	0.4062*	0.2885*	0.3897*
	PubmedBERT	0.7148	0.3577	0.6518	0.0363	0.0510	0.0643	0.1083	0.3524	0.2252	0.3310
	BERT	0.7149	0.3684	0.6472	0.0363	0.0510	0.0644	0.1083	0.3694	0.2447	0.3477
	DistilBERT	0.7118	0.3830	0.6628	0.0362	0.0508	0.0641	0.1078	0.3649	0.2449	0.3472
CLEF TAR 2017	Boolean operators	0.7521	-	-	0.0157	0.0214	0.0264	0.0429	-	-	-
	BM25 Title	0.7521	0.0239*	0.1268*	0.0157	0.0214	0.0264	0.0429	0.2261*	0.0224*	0.0629*
	smooth Boolean equivalents	0.7521	0.2033	0.4747	0.0157	0.0214	0.0264	0.0429	0.3300	0.1296	0.2421
	AND $\rightarrow f(\rho(d,q) \geq 0.99)$	0.7749	0.2069	0.4748	0.0020	0.0030	0.0040	0.0080	0.3366	0.1319	0.2440
	AND $\rightarrow f(\rho(d,q) \geq 0.9)$	0.8332	0.2102	0.4867	0.0006	0.0009	0.0011	0.0023	0.3553	0.1344	0.2513
	OR $\rightarrow f(\rho(d,q) \geq 0.01)$	0.4721*	0.1486	0.3284	0.0152	0.0205	0.0252	0.0403	0.2284*	0.1146	0.1877
	OR $\rightarrow f(\rho(d,q) \geq 0.1)$	0.2223*	0.1181	0.1868	0.0191	0.0239	0.0284	0.0422	0.1248	0.0870	0.1146
	Predictor	0.5106*	0.1461	0.3259	0.0107	0.0121	0.0137	0.0198	0.2410*	0.1146	0.1838
	Oracle	0.7695	0.2167	0.4812	0.0091	0.0122	0.0149	0.0245	0.3394	0.1395	0.2490
	PubmedBERT	0.7521	0.1834	0.4851	0.0157	0.0214	0.0264	0.0429	0.3243	0.1210	0.2369
	BERT	0.7521	0.2151	0.5045	0.0157	0.0214	0.0264	0.0429	0.3326	0.1348	0.2511
	DistilBERT	0.7514	0.1539	0.4617	0.0157	0.0214	0.0264	0.0429	0.3085	0.0980	0.2158
CLEF TAR 2018	Boolean operators	0.8344	-	-	0.0204	0.0297	0.0385	0.0699	-	-	-
	BM25 Title	0.8344	0.0245*	0.1960*	0.0204	0.0297	0.0385	0.0699	0.3266*	0.0232*	0.0951*
	smooth Boolean equivalents	0.8344	0.1807	0.5367	0.0204	0.0297	0.0385	0.0699	0.4567	0.1995	0.3410
	AND $\rightarrow f(\rho(d,q) \geq 0.99)$	0.8550	0.1623	0.5142	0.0050*	0.0075*	0.0099*	0.0194*	0.4576	0.1922	0.3306
	AND $\rightarrow f(\rho(d,q) \geq 0.9)$	0.8807	0.1655	0.5065	0.0007*	0.0010*	0.0013*	0.0026*	0.4632	0.1948	0.3263
	OR $\rightarrow f(\rho(d,q) \geq 0.01)$	0.5092*	0.1457	0.3611*	0.0193	0.0278	0.0356	0.0624	0.3069*	0.1794	0.2530*
	OR $\rightarrow f(\rho(d,q) \geq 0.1)$	0.2170*	0.0841	0.1572*	0.0271	0.0338	0.0397	0.0566	0.1454*	0.1143*	0.1272*
	Predictor	0.6205*	0.1464	0.3544*	0.0206	0.0293	0.0372	0.0637	0.3443*	0.1698	0.2418*
	Oracle	0.8487	0.1923	0.5375	0.0211	0.0307	0.0397	0.0718	0.4661	0.2125	0.3473
	PubmedBERT	0.8344	0.1867	0.5090	0.0204	0.0297	0.0385	0.0699	0.4593	0.2094	0.3399
	BERT	0.8344	0.1928	0.5168	0.0204	0.0297	0.0385	0.0699	0.4644	0.2191	0.3458
	DistilBERT	0.8344	0.1851	0.5004	0.0204	0.0297	0.0385	0.0699	0.4427	0.1798	0.3139
	auth_{run1,run2,run3} [28]	0.7705	0.2695	0.6386	0.0171	0.0249	0.0324	0.0593	0.4838	0.2804	0.4313
	ECNU_RUN1 [61]	0.5147*	0.2278	0.5147	0.0490	0.0661	0.0806	0.1248	0.3540	0.2440	0.3540
	ECNU_RUN2 [61]	0.3831*	0.1061	0.3831	0.0539	0.0695	0.0823	0.1190	0.2329*	0.1368	0.2329
	ECNU_RUN3 [61]	0.5147*	0.2318	0.5147	0.0490	0.0661	0.0806	0.1248	0.3487	0.2438	0.3487
	sheffield-bm25 [1]	0.4525*	0.1095	0.2875	0.0095	0.0138	0.0180	0.0328	0.2504*	0.1197	0.1852
	sheffield-boolean [1]	0.3048*	0.0555	0.1720*	0.0061	0.0089	0.0116	0.0212*	0.1519*	0.0562*	0.1018*
	sheffield-tfidf [1]	0.2572*	0.0169*	0.1052*	0.0059	0.0086	0.0112	0.0203	0.1123*	0.0154*	0.0523*
	UWX [11]	0.9749	0.3694	0.8677*	0.0254	0.0369	0.0478	0.0863	0.6484*	0.4084*	0.5988*

5.1.1 Ablation Study. We begin investigation into the first research question with an ablation study. In this experiment, we study the effect of the θ parameter on Boolean operators replaced by smooth operators. Table 2 contains the results of this experiment. As not all queries contain NOT operators, and they are far less common than OR and AND operators, we only study the replacement of OR and AND operators. The same θ value is set globally for all smooth operators within a query. For example, if the AND operator is replaced with the corresponding smooth operator $f(\rho(d,q) \geq \theta)$, then the θ value is the same for all replacements. All other Boolean operators in the query are replaced with equivalents from Table 1. Note that these smooth equivalent operators correspond to leaving the operators as is, as evidenced by the identical recall and precision in the first and third lines of each section in Table 2.

We use the same smooth operator $f(\rho(d,q) \geq \theta)$ for both OR and AND replacements. We test $\theta \in \{0.01, 0.1\}$ for the Boolean OR operator and $\theta \in \{0.99, 0.9\}$ for the Boolean AND operator. Overall, the results of the ablation study suggest that the replacement smooth operators have the desired effect: when replacing the Boolean AND operator, as θ decreases, the recall increases; when replacing the Boolean OR operator, as θ increases, the precision increases. The consequence of the increases in recall or precision is, naturally, a decrease in the respective other measure. This finding is captured most clearly by F-measure. Across the three collections, there was a decrease in F-measure compared to the Boolean operators except for the OR $\rightarrow f(\rho(d,q) \geq 0.1)$ replacement in the CLEF TAR 2017 and 2018 datasets. In these instances, the increase in F-measure compared to using the Boolean operators suggests that the increase in precision offsets the decrease in recall.

Next, we investigate how the smooth replacements affect the document ranking effectiveness. As smooth operators rank documents as a side-effect of the RSV calculation, we also study how different values of θ affect the document ranking effectiveness. For further comparison, we also re-rank the documents retrieved using the Boolean operators with BM25, using the title of the reviews as the query. Focusing first on recall@100 and recall@1000, the effectiveness is generally significantly worse for the OR replacements, except for CLEF TAR 2017. The AND replacements, on the other hand, often achieve a lower, but not significantly worse recall@k than the smooth Boolean equivalents. In rare cases, e.g. AND $\rightarrow f(\rho(d,q) \geq 0.9)$, R@1000 is insignificantly higher.

Moving to nDCG, AND operator replacement always achieves a higher effectiveness than OR operator replacements. The Wang et al. collection sees no improvement in nDCG over the Boolean equivalents. The CLEF TAR 2018 collection sees no improvement at shallow depths (nDCG@100 and nDCG@1000), but does improve at full depth (nDCG). The CLEF TAR 2017 collection sees an improvement across all three nDCG measures. In all cases where there was an improvement, the improvement was not statistically significant. Comparing the recall@k and nDCG measures using the BM25 title runs, the effectiveness is always significantly worse compared to the smooth Boolean equivalents. This is an important result because it demonstrates that simply replacing Boolean operators with the smooth equivalents, a more effective ranking can be achieved than retrieving and re-ranking using the systematic review title. This is a common baseline in many papers about screening prioritisation for systematic reviews [17–19, 22, 60].

Table 3: Computed features of each operator in queries. Children refer to the decedents, or operands, of an operator.

Feature	Description
Depth	Depth of the operator in the query.
Children	Number of children the operator has.
NumRet	Documents retrieved prior to smoothening.
Child Avg.	Average documents retrieved by children.
Child Std.	σ of documents retrieved by children.
Child Operators	Number of children that are operators.
Child Atoms	Number of children that are atomic queries.

5.1.2 Oracle Search. The results of the ablation study demonstrate that the smooth operators can be used to effectively manipulate the recall and precision of an existing Boolean query. However, the replacements often come with trade-offs: an increase in recall causes a decrease in precision, for example. Therefore, we next investigate whether it is possible to increase both precision and recall, using all three replacement operators. To accomplish this, we perform a parameter sweep of θ using the relevance assessments to find an acceptable value. When replacing the Boolean OR operator with a smooth replacement, we test values of θ in the range $\{0.0, 0.001, 0.01, 0.1, 0.15, 0.2\}$. For the Boolean AND and NOT smooth replacements, we test values of θ in the range $\{1.0, 0.999, 0.99, 0.95, 0.9, 0.8\}$. For each subquery in the query that contains an operator, we optimise that subquery by ensuring that recall increases or stays the same while F-measure increases. As the order of the θ values either increases or decreases the number of documents, when the optimisation criteria is no longer satisfied, the rest of the θ values are not tested.

The Oracle row in Table 2 reports the parameterisation which achieved the highest F-measure for each dataset. We find that across the three collections, it is possible to improve the effectiveness across all evaluation measures compared to using smooth Boolean equivalents, except for precision and F-measure on the 2017 CLEF TAR collection. In this instance, the bottom-up, local parameter search causes some topics to retain or have a higher recall, but a lower precision. Perhaps most interestingly, smooth operators are able to achieve a higher recall than the smooth Boolean equivalents.

5.1.3 Predicting θ. The results of the oracle θ parameter search demonstrate that more effective queries in terms of not only both recall and precision, but ranking quality are possible. This higher effectiveness requires one to have a thorough understanding of the effect that changing θ will have on the overall result set of the query. Instead, a method for predicting suitable θ values is desirable.

Before attempting supervised training to predict θ values, we first investigated whether there were any correlations between θ and features about the properties of an operator, such as the depth in the query or the number of children. Table 3 shows features used.

We use the default decision tree classifier from scikit-learn [33], and train it to predict the oracle search parameters from Section 5.1.2. We found that predicting appropriate θ values depending on different query contexts was challenging. Table 2 contains the results for predicting θ values ('Predictor' row for each collection). The results suggest that the classifier predicted values that smoothened

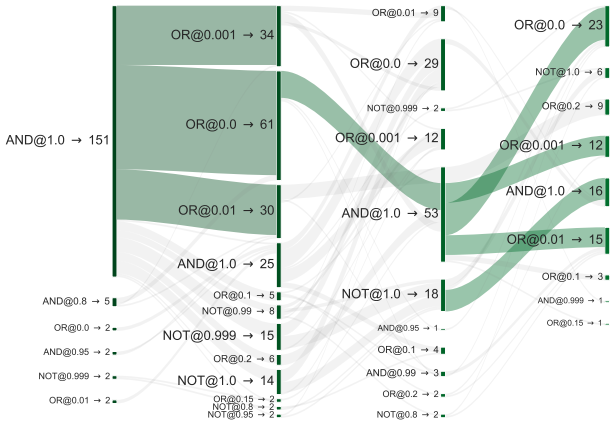
OR equivalents too much. The recall is dramatically lower than the Boolean operators, suggesting that relevant documents ranked poorly are being removed. In general, we found that the precision and recall of the 'Predictor' method lies somewhere in between the precision and recall of the replacement methods. That is, the recall of the 'Predictor' method is lower than AND replacement but higher than OR replacement; and likewise for precision but reversed. Developing an effective method for predicting appropriate values of θ is a challenge beyond the scope of this paper. We leave an investigation into more effective methods to future work.

5.1.4 Comparison with CLEF TAR 2018 participants. The CLEF TAR 2018 track introduced a new task from 2017 which involved participants retrieving documents rather than simply re-ranking the set of documents retrieved initially by the Boolean query. Table 2 contains the results of all participants for this task. Rows that have been marked in grey are not directly comparable to our method as they involve human intervention or explicit relevance feedback. For example, the Waterloo team used active learning and had annotators assess documents for their relevance [11]. Comparing the smooth Boolean equivalent runs, the runs from ECNU [61] were able to achieve a higher recall@k and nDCG@k. However, the smooth model was able to achieve a higher nDCG, likely because it achieves a higher recall than these methods. Although the oracle method is able to obtain a higher effectiveness across all measures compared to the smooth Boolean equivalents, the ranking effectiveness is still lower than the runs from ECNU. Despite this, the runs from ECNU are not statistically more effective than the smooth Boolean equivalents. Their use of pseudo-relevance feedback likely contributed the most to the ranking effectiveness. Additional ranking signals like those from pseudo-relevance feedback for improving ranking effectiveness are left to future work.

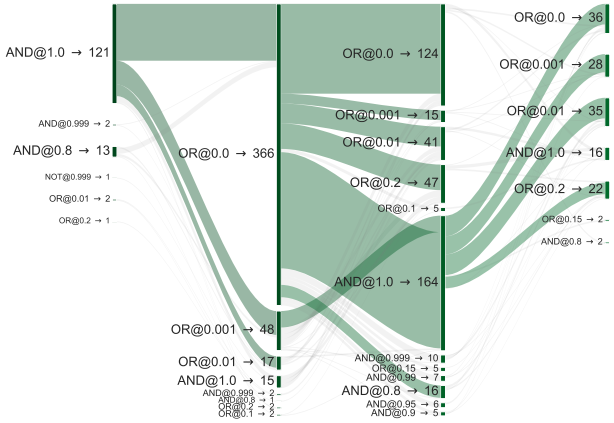
5.2 Effectiveness with Neural Rankers

Next, we investigate **RQ2**: *Can the use of large pre-trained language models in conjunction with smooth Boolean operators enhance the effectiveness of systematic review literature search queries?* We fine-tuned three BERT bi-encoder models on the TripClick collection [37]. Then, we used these rather than BM25 as the retrieval model for ranking documents using the smooth model. Table 2 contains the results of these experiments, under the names PubMedBERT, BERT, and DistilBERT. We found that across the three collections the BERT model was surprisingly more effective than PubMedBERT in all ranking metrics. Comparing BERT to the smooth Boolean equivalents, although BERT is able to achieve a higher nDCG@k than BM25, the recall@k is worse. This suggests that BERT is more effective than BM25 at shallow depths, but worse at deeper depths. More interestingly, the Oracle run has an even higher ranking effectiveness than the BERT ranker across the three collections. With appropriate θ values, an initially worse ranking can be refined to outperform an initially better ranking.

In summary, the use of complex neural rankers for this task provides only marginal gains in effectiveness, while being considerably less efficient. We note that a discrepancy between training and evaluation data most likely negatively affects the effectiveness of the neural rankers. In training, queries consisted of keyword-style queries from a click log, whereas, during inference, we used the



(a) Wang et al. [58] collection.



(b) Combined CLEF TAR 2017 [17] and 2018 [19] collections.

Figure 3: Flow diagrams that show the relationship between smooth operators as chosen by the oracle method. Operators at the leftmost side are at the highest depth in the query. The number following '→' indicates the total number of non-atomic subqueries beneath that subquery (i.e., children that are non-atomic). Green lines indicate subqueries that have more than ten kinds of operators of the same type leading into it. Only the first four depths are shown for space reasons.

systematic review titles as queries. Furthermore, we find that in the context of the smooth retrieval model, it is more important that rankers are effective at deeper depths than shallow depths, given that for systematic review literature search, recall is far more important than precision.

5.3 Systematic Review Types

Finally, we investigate **RQ3**: *How effective are smooth Boolean operators for systematic review literature searches across different types of systematic reviews, and what factors influence potential differences in effectiveness?* We investigate this research question by focusing

on the results obtained by the Oracle method across the three collections. One interesting finding is that while the Oracle results on the Wang et al. collection are statistically significantly more effective than the smooth Boolean equivalents, the same does not hold for the CLEF TAR collections. One potential reason for this could be the differences in the types of systematic reviews that form the basis of the topics of the three collections. The Wang et al. collection contains systematic reviews over a broad set of topics, while the CLEF TAR collections contains systematic reviews solely about diagnostic test accuracy. Different types of systematic reviews may focus on retrieving different documents, such as randomised controlled trials or meta-analyses, leading to differently designed queries.

Figures 3a and 3b show the differences in θ parameters chosen by the Oracle method between the CLEF TAR and Wang et al. collections. Interestingly, most OR subqueries at the first depth for CLEF TAR queries are not smoothed, whereas there is greater variability in how operators at the first depth are smoothed for the Wang et al. collection. Three other differences with the Wang et al. collection are (1) the higher number of AND and NOT operators at the first depth; (2) fewer operators that have more than ten parents; and (3) more operators where smoothing has been applied.

One possible reason many operators have no smoothing applied in the first three depths for the CLEF TAR collections is that these queries are already highly effective. That is, the reason that many operators are not smoothed, and as a result, do not obtain statistically higher effectiveness compared to the smooth Boolean equivalents, could be that many of these queries contain effective terms and are structured in such a way that further improving the effectiveness of these queries by any means would be difficult. Meanwhile, the fact that queries in the Wang et al. collection have the majority of their operators smoothed may indicate that these queries could be greatly improved by selecting more appropriate terms or better structuring. These differences may also be due to the kinds of literature these queries are developed for searching. For example, the topics in the Wang et al. collection may be more challenging to formulate queries for, given the nature of searching for randomised controlled trials, versus the kinds of documents that are relevant for systematic reviews of diagnostic test accuracy.

In short, this difference between the two collections regarding where and how soft operators are used has revealed interesting characteristics about the underlying topics being searched. There are considerable differences between the queries of the Wang et al. collection and the two CLEF TAR collections that should be considered when using smooth operators. We leave such investigation into how to exploit these differences to help predict appropriate values of θ to future work.

6 CONCLUSION

We have introduced the smooth operator model, which can broaden or restrict the document set size of Boolean queries. The smooth operator model has several properties that improve upon previous extensions to the Boolean retrieval model. Most notably, the syntax and semantics of queries are identical to Boolean queries, except for how smooth an operator should be. If no smoothing is applied to any operator, it is equivalent to the Boolean retrieval model.

One property of the smooth operator model is that it produces document rankings. Using the smooth Boolean equivalents, we found that the smooth operator model was significantly more effective than ranking the set of retrieved documents by the systematic review title (a common baseline in related research). Furthermore, neural retrieval models did not statistically improve the smooth operator model's ranking effectiveness, suggesting that the smooth operator model's rank fusion component produces effective rankings regardless of the ranking function used to rank atomic queries.

When used to modify the set of retrieved documents, i.e., broadening or narrowing the set of retrieved documents, queries could be directly optimised for precision or recall with the θ parameter alone. This is important for specialised search scenarios, as identifying specific terms to broaden or narrow the scope of a query is challenging even for expert searchers. We also found that improving the precision and recall of queries is possible using only smooth operators. However, we have not identified any heuristics for how smooth an operator should be in different search contexts. As we could not reliably predict the smoothness of operators using a feature-based supervised learning method, we leave the effective prediction of suitable θ values to future work.

The smooth operator model was demonstrated to improve the effectiveness of systematic review literature search queries. Improving the effectiveness of these queries is a challenging problem, given that expert human searchers develop the queries. Using our smooth operator model, the effectiveness of existing systematic review literature search queries can be improved without changing the syntactic or semantic structure of queries. In practice, this also means that when formulating new queries using the smooth operator model, the choice of terms that broaden or restrict the scope of a query may become less important, as the relationship between these terms and clauses can be relaxed. Another advantage of the smooth operator model is the ability to intermix exact match and neural retrieval models. Although we did not deeply investigate the combination of exact match and neural retrieval models, in practice, it would allow expert searchers to combine the restrictiveness of exact match (e.g., for filtering specific studies) with the semantic expressiveness of neural models (e.g., not needing to search synonyms or plurals of terms explicitly). We believe that queries developed from scratch with the smooth operator model in mind could be more effective than adapting them to Boolean queries as we have done in this paper. More effective queries for systematic review literature search lead to fewer documents that need to be screened for the review. As a result, systematic reviews can be completed in a more cost-effective and timely manner; overall leading to more positive healthcare outcomes.

ACKNOWLEDGMENTS

Dr Harrisen Scells is the recipient of an Alexander von Humboldt Stiftung Research Fellowship. This work was partially funded by the European Commission under GA 101070014 (OpenWebSearch.EU). The authors wish to thank Lukas Gienapp, Theresa Elstner, and Niklas Deckers for their feedback on early revisions of this paper. The authors also wish to thank the reviewers for their helpful and insightful feedback on this paper.

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: Working Notes of CLEF 2018: Conference and Labs of the Evaluation Forum*, Vol. 2125. CEUR Workshop Proceedings.

[2] Amal Alharbi and Mark Stevenson. 2020. Refining Boolean Queries to Identify Relevant Studies for Systematic Review Updates. *Journal of the American Medical Informatics Association* 27, 11 (Nov. 2020), 1658–1666. https://doi.org/10.1093/jamia/ocaa148

[3] Sophia Ananiadou, Brian Rea, Naoaki Okazaki, Rob Procter, and James Thomas. 2009. Supporting Systematic Reviews Using Text Mining. *Social Science Computer Review* 27, 4 (2009), 509–523.

[4] Abraham Bookstein. 1980. Fuzzy Requests: An Approach to Weighted Boolean Searches. *Journal of the american Society for Information Science* 31, 4 (1980), 240–247.

[5] Gloria Bordogna and Gabriella Pasi. 1993. A Fuzzy Linguistic Approach Generalizing Boolean Information Retrieval: A Model and Its Evaluation. *Journal of the American Society for Information Science* 44, 2 (March 1993), 70–82. https://doi.org/10.1002/(SICI)1097-4571(199303)44:2<70::AID-ASI2>3.0.CO;2-I

[6] Duncan A Buell. 1981. A General Model of Query Processing in Information Retrieval Systems. *Information Processing & Management* 17, 5 (1981).

[7] Justin Clark. 2013. Systematic Reviewing. In *Methods of Clinical Epidemiology*, Gail M. Williams Suhail A. R. Doi (Ed.).

[8] A.M. Cohen, W.R. Hersh, K. Peterson, and P.Y. Yen. 2006. Reducing Workload in Systematic Review Preparation Using Automated Citation Classification. *Journal of the American Medical Informatics Association* 13, 2 (2006), 206–219.

[9] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '09*. ACM Press, Boston, MA, USA, 758. https://doi.org/10.1145/1571941.1572114

[10] Gordon V Cormack and Maura R Grossman. 2015. Autonomy and Reliability of Continuous Active Learning for Technology-Assisted Review. *arXiv preprint arXiv:1504.06868* (2015). arXiv:1504.06868

[11] Gordon V Cormack and Maura R Grossman. 2018. Technology-Assisted Review in Empirical Medicine: Waterloo Participation in CLEF eHealth 2018. In *CEUR Workshop Proceedings: Working Notes of CLEF 2018: Conference and Labs of the Evaluation Forum*.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]* (May 2019). arXiv:1810.04805 [cs]

[13] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *ACM Transactions on Computing for Healthcare (HEALTH)* 3, 1 (2021), 1–23.

[14] Julian Higgins, James Thomas, Jacqueline Chandler, Miranda Cumpston, Tianjing Li, Matthew Page, and Vivian Welch. 2022. *Cochrane Handbook for Systematic Reviews of Interventions version 6.3 (updated February 2022)*. Cochrane. https://training.cochrane.org/handbook/current/chapter-i

[15] Sebastian Hofstätter, Sophia Althammer, Mete Sertkan, and Allan Hanbury. 2022. Establishing Strong Baselines For TripClick Health Retrieval. In *European Conference on Information Retrieval*. Springer, 144–152.

[16] D Frank Hsu and Isak Taksa. 2005. Comparing Rank and Score Combination Methods for Data Fusion in Information Retrieval. *Information retrieval Journal* 8, 3 (2005), 449–480.

[17] Evangelos Kanoulas, Dan Li, Leif Azzopardi, and Rene Spijker. 2017. CLEF 2017 Technologically Assisted Reviews in Empirical Medicine Overview. In *CEUR Workshop Proceedings: Working Notes of CLEF 2017: Conference and Labs of the Evaluation Forum*.

[18] Evangelos Kanoulas, Dan Li, Leif Azzopardi, and Rene Spijker. 2019. CLEF 2019 Technology Assisted Reviews in Empirical Medicine Overview. In *CEUR Workshop Proceedings: Working Notes of CLEF 2018: Conference and Labs of the Evaluation Forum*, Vol. 2380.

[19] Evangelos Kanoulas, Rene Spijker, Dan Li, and Leif Azzopardi. 2018. CLEF 2018 Technology Assisted Reviews in Empirical Medicine Overview. In *CEUR Workshop Proceedings: Working Notes of CLEF 2018: Conference and Labs of the Evaluation Forum*.

[20] Sarvnaz Karimi, Justin Zobel, Stefan Pohl, and Falk Scholer. 2009. The Challenge of High Recall in Biomedical Systematic Search. In *Proceedings of the 3rd International Workshop on Data and Text Mining in Bioinformatics*. 89–92.

[21] Youngho Kim, Jangwon Seo, and W Bruce Croft. 2011. Automatic Boolean Query Suggestion for Professional Search. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[22] Grace E. Lee and Aixin Sun. 2018. Seed-Driven Document Ranking for Systematic Reviews in Evidence-Based Medicine. In *Proceedings of the 41st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 455–464.

[23] Carolyn E. Lipscomb. 2000. Medical Subject Headings (MeSH). *Bulletin of the Medical Library Association* 88, 3 (July 2000), 265–266.

[24] Robert Losee. 1987. Probabilistic Retrieval and Coordination Level Matching. *Journal of the American Society for Information Science* 38, 4 (1987), 239–244.

[25] Craig Macdonald and Iadh Ounis. 2006. Voting for Candidates: Adapting Data Fusion Techniques for an Expert Search Task. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*. ACM, 387–396.

[26] D. Martinez, S. Karimi, L. Cavedon, and T. Baldwin. 2008. Facilitating Biomedical Systematic Reviews Using Ranked Text Retrieval and Classification. In *Proceedings of the 13th Australasian Document Computing Symposium*.

[27] Matthew Michelson and Katja Reuter. 2019. The Significant Cost of Systematic Reviews and Meta-Analyses: A Call for Greater Involvement of Machine Learning to Assess the Promise of Clinical Trials. *Contemporary Clinical Trials Communications* 16 (Dec. 2019), 100443. https://doi.org/10.1016/j.conctc.2019.100443

[28] 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 *CEUR Workshop Proceedings: Working Notes of CLEF 2018: Conference and Labs of the Evaluation Forum*.

[29] M. Miwa, J. Thomas, A. O'Mara-Eves, and S. Ananiadou. 2014. Reducing Systematic Review Workload through Certainty-Based Screening. *Journal of Biomedical Informatics* 51 (2014), 242–253.

[30] National Library of Medicine (US). 1963. *Medical Subject Headings: Main Headings, Sub-headings, and Cross References Used in the Index Medicus and the National Library of Medicine Catalog*. US Department of Health, Education, and Welfare. Public Health Service.

[31] 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.

[32] Chris D Paice. 1984. Soft Evaluation of Boolean Search Queries in Information Retrieval Systems. *Information Technology: Research and Development* 3, 1 (1984), 33–41.

[33] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 85 (2011), 2825–2830.

[34] Stefan Pohl, Justin Zobel, and Alistair Moffat. 2010. Extended Boolean Retrieval for Systematic Biomedical Reviews. *Computer Science* 102 (2010).

[35] Piotr Przybyła, Austin J. Brockmeier, Georgios Kontonatsios, Marie-Annick Le Pogam, John McNaught, Erik von Elm, Kay Nolan, and Sophia Ananiadou. 2018. Prioritising References for Systematic Reviews with RobotAnalyst: A User Study. *Research Synthesis Methods* 9, 3 (2018), 470–488. https://doi.org/10.1002/jrsm.1311

[36] Tadeusz Radecki. 1979. Fuzzy Set Theoretical Approach to Document Retrieval. *Information Processing & Management* 15, 5 (1979), 247–259.

[37] Navid Rekabsaz, Oleg Lesota, Markus Schedl, Jon Brassey, and Carsten Eickhoff. 2021. TripClick: The Log Files of a Large Health Web Search Engine. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2507–2513.

[38] Stephen E Robertson. 1977. The Probability Ranking Principle in IR. *Journal of documentation* 33, 4 (1977), 294–304.

[39] Gerard Salton, Edward A Fox, and Harry Wu. 1982. *Extended Boolean Information Retrieval*. Technical Report. Cornell University.

[40] Gerard Salton, Edward A Fox, and Harry Wu. 1983. Extended Boolean Information Retrieval. *Commun. ACM* 26, 11 (1983), 1022–1036.

[41] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *ArXiv* (Oct. 2019).

[42] Harrisen Scells, Connor Forbes, Justin Clark, Bevan Koopman, and Guido Zuccon. 2022. The Impact of Query Refinement on Systematic Review Literature Search: A Query Log Analysis. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*. ACM, Madrid Spain, 34–42. https://doi.org/10.1145/3539813.3545143

[43] Harrisen Scells and Martin Potthast. 2023. pybool_ir: A Toolkit for Domain-Specific Search Experiments. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Taipei Taiwan.

[44] Harrisen Scells and Guido Zuccon. 2018. Generating Better Queries for Systematic Reviews. In *Proceedings of the 41st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 475–484.

[45] Harrisen Scells and Guido Zuccon. 2020. You Can Teach an Old Dog New Tricks: Rank Fusion Applied to Coordination Level Matching for Ranking in Systematic Reviews. In *Proceedings of the 42nd European Conference on Information Retrieval*. 399–414.

[46] Harrisen Scells, Guido Zuccon, and Bevan Koopman. 2019. Automatic Boolean Query Refinement for Systematic Review Literature Search. In *Proceedings of the 28th World Wide Web Conference*. 1646–1656.

[47] Harrisen Scells, Guido Zuccon, and Bevan Koopman. 2020. A Comparison of Automatic Boolean Query Formulation for Systematic Reviews. *Information Retrieval Journal* (2020), 1–26.

[48] Harrisen Scells, Guido Zuccon, Bevan Koopman, and Justin Clark. 2019. Automatic Search Strategy Reformulation Interface for Systematic Reviews. In *Proceedings of the 2019 Cochrane Colloquium*.

[49] Joseph A Shaw and Edward A Fox. 1995. Combination of Multiple Searches. *NIST SPECIAL PUBLICATION SP* (1995), 105–105.

[50] I. Shemilt, A. Simon, G.J. Hollands, T.M. Marteau, D. Ogilvie, A. O'Mara-Eves, M.P. Kelly, and J. Thomas. 2014. Pinpointing Needles in Giant Haystacks: Use of Text Mining to Reduce Impractical Screening Workload in Extremely Large Scoping Reviews. *Research Synthesis Methods* 5, 1 (2014), 31–49.

[51] Maria Smith. 1990. Aspects of the P-Norm Model of Information Retrieval: Syntactic Query Generation, Efficiency, and Theoretical Properties. (May 1990).

[52] CM Stansfield, Alison O'Mara-Eves, and James Thomas. 2015. Reducing Systematic Review Workload Using Text Mining: Opportunities and Pitfalls. *Journal of the European Association for Health Information and Libraries* 11, 3 (2015), 8–10.

[53] Christopher C Vogt and Garrison W Cottrell. 1999. Fusion via a Linear Combination of Scores. *Information retrieval Journal* 1, 3 (1999), 151–173.

[54] Byron C Wallace, Kevin Small, Carla E Brodley, Joseph Lau, and Thomas A Trikalinos. 2012. Deploying an Interactive Machine Learning System in an Evidence-Based Practice Center: Abstrackr. In *Proceedings of the 2nd ACM International Health Informatics Symposium*. 819–824.

[55] Byron C Wallace, Thomas A Trikalinos, Joseph Lau, Carla Brodley, and Christopher H Schmid. 2010. Semi-Automated Screening of Biomedical Citations for Systematic Reviews. *BMC bioinformatics* 11, 1 (2010), 55.

[56] WG Waller and Donald H Kraft. 1979. A Mathematical Model of a Weighted Boolean Retrieval System. *Information Processing & Management* 15, 5 (1979), 235–245.

[57] Shuai Wang, Hang Li, Harrisen Scells, Daniel Locke, and Guido Zuccon. 2021. MeSH Term Suggestion for Systematic Review Literature Search. In *Australasian Document Computing Symposium*. ACM, Virtual Event Australia, 1–8. https://doi.org/10.1145/3503516.3503530

[58] 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*.

[59] 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* 16 (Nov. 2022), 200141. https://doi.org/10.1016/j.iswa.2022.200141

[60] Shuai Wang, Harrisen Scells, Bevan Koopman, and Guido Zuccon. 2022. Neural Rankers for Effective Screening Prioritisation in Medical Systematic Review Literature Search. https://doi.org/10.1145/3572960.3572980 arXiv:2212.09017 [cs]

[61] Huaying Wu, Tingting Wang, Jiayi Chen, Su Chen, Qinmin Hu, and Liang He. 2018. Ecnu at 2018 Ehealth Task 2: Technologically Assisted Reviews in Empirical Medicine. *Methods-a Companion to Methods in Enzymology* 4, 5 (2018), 7.

[62] Lotfi A. Zadeh. 1965. Fuzzy Sets. *Information and control* 8, 3 (1965), 338–353.