Axiomatic Result Re-Ranking

Matthias Hagen Michael Völske Steve Göring Benno Stein

Bauhaus-Universität Weimar
99421 Weimar, Germany
<first name>.<last name>@uni-weimar.de

ABSTRACT

We consider the problem of re-ranking the top-k documents returned by a retrieval system given some search query. This setting is common to learning-to-rank scenarios, and it is often solved with machine learning and feature weighting based on user preferences such as clicks, dwell times, etc. In this paper, we combine the learning-to-rank paradigm with the recent developments on axioms for information retrieval. In particular, we suggest to re-rank the top-k documents of a retrieval system using carefully chosen axiom combinations.

In recent years, research on axioms for information retrieval has focused on identifying reasonable constraints that retrieval systems should fulfill. Researchers have analyzed a wide range of standard retrieval models for conformance to the proposed axioms and, at times, suggested certain adjustments to the models. We take up this axiomatic view—but, instead of adjusting the retrieval models themselves, we suggest the following innovation: to adopt the learning-to-rank idea and to re-rank the top-k results *directly* using promising axiom combinations. This way, we can turn every reasonable basic retrieval model into an axiom-based retrieval model. In large-scale experiments on the ClueWeb corpora, we identify promising axiom combinations for a variety of retrieval models. Our experiments show that for most of these models our axiom-based re-ranking significantly improves the original retrieval performance.

1. INTRODUCTION

Information retrieval research that deals with axioms for ranking quality plays a rather "theoretical" role in the community today. Most of the axiomatic research focuses on the question of whether the result rankings of retrieval models (e.g., BM25 or language models) are in accordance with specific reasonable axioms that formalize ranking preferences. E.g., from two documents of the same length, the document that contains the query terms more often should be favored. Some of these axiomatic studies also suggest subtle changes to the original retrieval models to better conform with specific axioms and then demonstrate retrieval performance improvements based on these changes. However, up to now, no "operationalized" axiomatic retrieval model has been proposed that *by construction* conforms with as many axioms as possible and that hence could lead to substantial retrieval performance gains.

This observation leads to the main research question of our paper: Is it possible—and how—to seamlessly integrate axioms for ranking preferences in order to improve the results of a given basis retrieval model? Our proposed solution is inspired by the learning-to-rank framework: Given some basis retrieval model, a carefully chosen axiom combination re-ranks the top-k results and produces an axioms-compliant output. In this regard we consider as many of the published axioms as possible and also suggest several newly developed term proximity axioms.

Most axioms in the information retrieval literature have a similar basic structure: for a pair or a triple of documents, ranking preferences are deduced from standard features such as document length, term frequency, or semantic similarity. When such an axiom is applied to all pairs or triples of documents in a retrieval model's result list, the matrix of the inferred preferences may induce a result re-ranking. For example, consider a situation with an axiom A and three initially retrieved documents d_1, d_2, and d_3. After applying axiom A to all document pairs, one might end up with the preferences $d_2 >_A d_1$, $d_2 >_A d_3$, and $d_1 >_A d_3$, where $d_i >_A d_j$ means that document d_i should be ranked above d_j according to axiom A. Only the ranking $[d_2, d_1, d_3]$ matches these preferences and will thus become the re-ranked document list. However, in the general case there are many axioms (typically of different importance) and contradictory rank preferences will become likely. As a solution and a way of combining the weighted axioms' matrices of rank preferences, we apply fusion algorithms that were developed in the field of computational social choice.

The effectiveness of our axiom-based retrieval system is studied in a large-scale evaluation with 17 basis retrieval models in the setting of the TREC Web tracks 2009–2014. As a result, the performance of almost all basis retrieval models is improved via axiomatic result re-ranking. It is thus possible to improve existing retrieval models in an "ex-post manner", considering the latest insights from the research on retrieval axioms. The main contributions of our paper are: (1) We show how combinations of known axioms can be incorporated into a learning-to-rank inspired result re-ranking for any given basis retrieval model. The resulting axiom-based retrieval systems are shown to significantly

CIKM '16, October 24–28, 2016, Indianapolis, IN, USA.

ISBN 978-1-4503-4073-1/16/10...$15.00

DOI: http://dx.doi.org/10.1145/2983323.2983704

increase retrieval performance for many standard retrieval models. (2) We propose axioms to model term proximity preferences and show their effect in the axiomatic re-ranking retrieval model.

2. RELATED WORK

We first briefly review the recent developments on axiomatic ideas for information retrieval. More detailed descriptions of the used axioms follow in Section 3. Furthermore, we give some background on the general learning-to-rank setting since it inspired our approach. The rank aggregation method borrowed from computational social choice is discussed in the description of our approach in Section 3.

2.1 Axioms for Information Retrieval

The earliest studies of axioms in the context of information retrieval systems date back more than 20 years now [30, 31, 4]. One of the first published ideas that possibly could be considered as "axiomatic" is a retrieval system based on production rules from artificial intelligence by McCune et al. [30], which led to some improvements over a simple Boolean model. Another approach using more formal rules (again, these could be viewed as axioms) is presented by Meghini et al. [31], who use terminological logics for building a retrieval model. The first real reference to a notion of axioms for information retrieval is contained in the aboutness study of Bruza and Huibers [4]. Actually, the authors do not propose a retrieval model but rather a way of expressing what should be expected from a good result ranking. It took a while but, especially in the last decade, the interest in this direction of using axioms to describe what a good ranking looks like increased substantially. Hui Fang's web page gives a good overview of the existing literature and axioms.[1] The goal of most of the recent studies is to propose new reasonable axioms and to evaluate how well existing retrieval models match the respective assumptions. While they also propose improvements for retrieval models, typically only a handful of specific axioms are considered. We propose a way of incorporating all of the known axioms in one retrieval system with the possibility of adding new axioms in the future.

We give a brief overview of the existing axiomatic literature divided by the goal of the axioms: term frequency and lower bounds on it, document length, query aspects, semantic similarity, term proximity, and other axiomatic ideas that do not fit any of these categories. The axioms that are part of our re-ranking scheme will be explained in more detail in Section 3.

Term frequency.

Term frequency axioms follow the idea that documents containing query terms more often should be ranked higher. Fang et al. define several such axioms (TFC1–TFC3 and TDC) [17, 19, 18] and experimentally show that these axioms should be satisfied in order to produce better rankings. We employ all of these axioms in our re-ranking scheme. Na et al. [32] propose some specific axiomatic term frequency constraints tailored to language modeling (LM) retrieval approaches. Since their axioms cannot be easily rephrased to be generally applicable to non-LM retrieval, we decided not to include these axioms.

Document length.

Besides the term frequency axioms, Fang et al. also define document length axioms (LNC1, LNC2 and TF-LNC) [17] with the basic idea that in case of same term frequencies shorter documents should be ranked higher. We employ all of these axioms in our re-ranking scheme. A query-based document length constraint (QLNC) proposed by Cummins and O'Riordan [13] can not easily be reformulated to induce rank preferences so that we do not include it in our re-ranking scheme.

Lower bounds on term frequency.

Combining term frequency and document length, the idea of the lower bound axioms is that long documents should not be penalized too much. Lv and Zhai propose two respective axioms (LB1 and LB2) [28, 29]. We use adapted versions in our re-ranking scheme.

Query aspects.

Zheng and Fang [42], and Wu and Fang [39] propose axioms (REG and AND) that aim at ranking documents higher that match more query terms or aspects. Gollapudi and Sharma [22] developed axioms (DIV) with a similar purpose, modeling the diversity of a result set as a whole. Interestingly, they show that no diversification function can satisfy all the axioms simultaneously. However, these query aspect related axioms in their original formalizations do not induce rank preferences such that we use adapted versions in our re-ranking scheme.

Semantic similarity.

Often it can be very important not to rely on exact term matching between queries and documents but to also take documents into account that contain semantically similar terms. Yang and Fang propose five axioms in this regard (STMC1–STMC3, TSSC1, TSSC2) [20], which were later shown beneficial also in a query expansion setting [16]. We only use STMC1 and STMC2 in our re-ranking scheme since STMC3, TSSC1, and TSSC2 can not be restated to induce rank preferences.

Term proximity.

Term proximity axioms aim at describing the importance of query terms appearing close to each other in result documents (e.g., phrases). Tao and Zhai [37] introduce several respective axioms (DIST1–DIST5)—but rather with the goal of improving a retrieval model's proximity feature than to induce rank preferences. Since their axioms do not induce rank preferences, we propose five new proximity axioms as one of our contributions (cf. Section 3).

Other axiom ideas.

There is a wide range of other axiomatic studies that do not fit the above groups. Many of these are not helpful in our setting since either the axioms have a completely unrelated purpose (e.g., axioms for evaluation [3, 5]) or the axioms do not induce rank preferences by nature. An exception is Altman and Tennenholtz' study of properties implied by graph-theoretic axioms for link graphs [2]. They show that their axioms are satisfied by the PageRank algorithm but the axioms do not induce any rank preferences. We include a modified PageRank-based axiom as one of our contributions.

[1] http://www.eecis.udel.edu/~hfang/AX.html

Cummins and O'Riordan [11, 12] analyze axioms for learned ranking functions, but since none of the basis retrieval models we will use is machine-learning-based, the respective axioms would not help. Clinchant et al. [9, 10] describe axioms for pseudo-relevance feedback models (PRF) that are also not applicable in our setting since we do not employ PRF methods. Gerani et al. [21] propose axioms for combining scores in a multi-criteria relevance approach [21] that also do not fit the basis retrieval models we will employ. Zhang et al. [41] present an axiomatic framework for user-rating based ranking of items in Web 2.0 applications, but since our ad-hoc retrieval task is different, their axioms could not be applied. Karimzadehgan and Zhai [24] and Rahimi et al. [35] perform axiomatic analysis of translation language models in order to gain insights about how to optimize the estimation of translation probabilities; again, the purpose is different to our setting such that we do not include these axioms. Ding and Wang [14] show how axioms covering term dependency can be integrated into language-model-based retrieval approaches, but since their axioms do not induce preference lists and are not applicable to the non-LM approaches among our basis retrieval models, we do not include these axioms in our re-ranking scheme.

2.2 Learning to Rank

Our axiomatic re-ranking framework follows ideas developed in the learning-to-rank domain. There, the goal is to rank documents based on machine learning algorithms [27]. In general, three different approaches can be distinguished: pointwise, pairwise, and listwise ranking. In the pointwise approach, machine learning methods are used for each document to predict the rank based on document-individual feature values. The pairwise approach instead uses pairs of documents to conclude rank preferences for each pair. The listwise approach does not learn a ranking function for individual documents or pairs but processes entire result lists. Independent of the employed learning approach, most learning-to-rank systems are built on top of a basis retrieval model: An initial document set typically consisting of the basis model's top-k results is retrieved and then re-ranked, using the learned ranking method. In our system, we will follow this paradigm and employ a mixture of the pairwise and the listwise approach since the used axioms yield pairwise rank preferences, but the optimization criterion measures the performance over a range of result lists of different queries used for training—an approach inspired by a study of Cao et al. [6].

There are many directions for improving rankings in a learning-to-rank style. For example search engine logs provide a lot of implicit information that can be used to inform the learning process. Radlinski and Joachims [34] describe a learning-to-rank system that exploits click-through data in such a way. Since we do not have huge logs available for training, we stick to explicit feedback from the TREC relevance judgments for training. At first sight this may appear related to an approach of Veloso et al. [38] who use data mining techniques to learn association rules based on relevance judgments. However, instead of learning association rules, we take the set of axioms as given and learn only their importance by inferring an aggregation function. Our idea of training different axiomatic rankers while optimizing the target performance measure of nDCG @10 is inspired by the AdaRank framework [40] that also directly optimizes the performance measure instead of classification errors.

3. AXIOMATIC RE-RANKING

We put axiomatic re-ranking to work within three steps. First, an initial search is done with some basis retrieval model; the returned top-k results are used as re-ranking candidates (in our experiments we set $k = 50$). Recall that our approach is not restricted to a certain retrieval model—a fact which is later demonstrated in the experimental evaluation. Second, each axiom is evaluated regarding the retrieved documents, and the resulting pairwise rank preferences are stored as a matrix. Using a machine learning algorithm on a training set of document pairs with known relevance judgments, we infer an aggregation function to combine multiple axiom preferences into a joint preference matrix. Third, on the resulting matrix a rank aggregation is applied that utilizes ideas from the field of computational social choice. In particular, we derive the final re-ranked results by employing the KwikSort algorithm [1] to solve the Kemeny rank aggregation problem [25] on the sum matrix. We argue that the training should yield different axiom aggregation functions for different basis retrieval models. Hence, when applying axiomatic re-ranking given some basis retrieval model's results, we consult the corresponding learned aggregation function. The general setup of our approach is illustrated in Figure 1.

In the remainder of this section we explain which axioms from the axiomatic IR literature we use and the sometimes necessary modifications. We also present our newly developed term proximity axioms, and detail the employed rank aggregation method and the axiom aggregation scheme.

3.1 Requirements on Axioms

We analyzed the literature on published retrieval model axioms and carefully selected those that can be restated to induce rank preferences for result lists. In this regard we decided to restrict to axioms that formalize rank preferences on pairs of documents—reflecting the pairwise approach to learning to rank. From its syntax, an axiom A in our framework is formulated as a triple:

$$A = (\textit{precondition}, \textit{filter}, \textit{conclusion}),$$

where *precondition* is any evaluable condition, *filter* is a more specific filter condition, and *conclusion* is a rank preference $d_i >_A d_j$ (semantics: document d_i should be ranked above d_j according to A). For each axiom A and for all pairs of documents these rank preferences are stored in a matrix M_A:

$$M_A[i,j] = \begin{cases} 1 & \text{if } d_i >_A d_j, \\ 0 & \text{otherwise.} \end{cases}$$

Note that, at least theoretically, the application of an axiom requires the iteration over all pairs of candidate documents to check *precondition* and *filter* to infer the rank preferences. In this paper, however, we do not focus on the practical efficiency of axiomatic re-ranking but demonstrate its effectiveness. Further tuning the efficiency of the axiomatic re-ranking approach will be an interesting task for future research given the promising experimental improvements of retrieval quality we achieve (cf. Section 4).

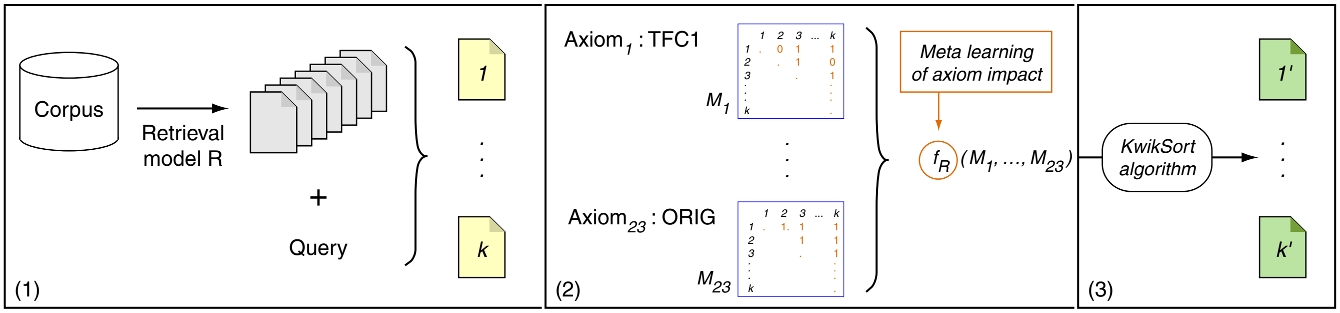

Figure 1: Illustration of our axiomatic result re-ranking approach. (1) Initial result set construction of size k using a basis retrieval model. (2) Deduction of axiom-specific partial orderings (matrices) for the result set documents, which are combined into a single matrix based on a previously learned axiom aggregation function. (3) Re-ranking of the original result list by solving the Kemeny rank aggregation problem with the KwikSort algorithm.

3.2 Existing Axioms And Modifications

We start with remarks on modifications that pertain to most axioms and then present the analyzed axioms with potential individual modifications.

General Modifications.

Some axioms from the literature rely on conditions or filters that require two to-be-compared documents having exactly the same length (e.g., TFC1 and TFC2). However, despite being theoretically sound, in real-world top-k search results there are not that many documents that fulfill this (or some other) value constraint perfectly. We hence relax such conditions and require only a fuzzy match, allowing a difference of at most 10%. A length condition $\mathit{length}(d_i) = \mathit{length}(d_j)$ requiring the exact same length of d_i and d_j would be adapted to $\mathit{length}(d_i) \approx_{10\%} \mathit{length}(d_j)$ with the following semantics:

$$\frac{|\mathit{length}(d_i) - \mathit{length}(d_j)|}{\max\{\mathit{length}(d_i), \mathit{length}(d_j)\}} \leq 0.10.$$

Consequently, a requested length difference then corresponds to a difference of more than 10%, denoted as $>_{10\%}$. Similarly, axioms with equality constraints on the term frequency are treated also with a 10%-relaxation. Some axioms' conditions require the same term discrimination value for two terms. We use *idf* in such cases but do not apply a 10%-rule since this would result in too many terms with the "same" *idf*-values. Instead, we round values to two decimal digits and then two terms have the "same" *idf*-value, when their rounded *idf*-values are the same. In some axioms, a semantic similarity measure $\mathrm{s}(w_1, w_2)$ for two terms is employed. We use WordNet[2] in such cases. Also note that some axioms conclude properties of some abstract query-document scoring function $\mathit{score}(d, q)$. However, we just use the induced rank preferences in these cases.

Term Frequency Axioms.

The basic idea of the term frequency axioms TFC1–TFC3 and TDC is to formulate reasonable assumptions on the correlation between term frequency and document ranks. For the example axiom TFC1, we give the original description and our employed restated version in full detail. For all the other axioms, we restrict the explanations to fewer details due to space constraints. Axiom TFC1 assigns higher scores to documents that contain a query term more often based on the following original definition [17]:

TFC1: Let $q = \{t\}$ be a query with only one term t. Assume $|d_1| = |d_2|$. If $\mathit{tf}(t, d_1) > \mathit{tf}(t, d_2)$, then $\mathit{score}(d_1, q) > \mathit{score}(d_2, q)$.

We transform TFC1 to our triple notation by setting

$$\begin{aligned} \mathit{precondition} &:= \mathit{length}(d_1) \approx_{10\%} \mathit{length}(d_2), \\ \mathit{filter} &:= \mathit{tf}(t, d_1) >_{10\%} \mathit{tf}(t, d_2), \text{ and} \\ \mathit{conclusion} &:= d_1 >_{\mathrm{TFC1}} d_2. \end{aligned}$$

In case of more than one query term, we use the sum of the individual term frequencies in the *filter* condition as a generalized version of TFC1. In a similar way, we generalize and transform TFC2, TFC3, and TDC. Axiom TFC2 compares three documents and checks the term frequency gaps between these documents. The problem is that it concludes *score* differences $\mathit{score}(d_2, q) - \mathit{score}(d_1, q) > \mathit{score}(d_3, q) - \mathit{score}(d_2, q)$ for the three documents that cannot directly be modeled in our framework. Based on the precondition of TFC2, d_3 has the highest term frequency and d_1 the lowest such that we change the *conclusion* to $d_3 >_{\mathrm{TFC2}} d_2$ and $d_2 >_{\mathrm{TFC2}} d_1$. This way, TFC2 could be seen as a transitive version of TFC1 such that it probably does not add much to an axiomatic re-ranking that also includes TFC1.

Axioms TFC3 and TDC conclude scoring properties for two-keyword queries based on term discrimination values (rounded *idf*-values in our setting). The document containing the terms more often or containing terms with higher *idf*-values is favored. To be applicable also to longer queries, we generalize TFC3 and TDC by applying them to every query term pair.

Document Length Axioms.

The axioms LNC1, LNC2, and TF-LNC are focusing on document length normalization [17]. In axiom LNC1 two documents are compared that have the same term frequency for all query terms (remember the 10%-softening in our setting). Then the shorter document is preferred.

Axiom LNC2 checks whether one document is an m-times copy of another document. In such a case, the "original" document (i.e., the shorter one) gets a better rank. Note that the m-times copy condition is a rather artificial case,

[2] http://wordnet.princeton.edu/

since hardly any real-world documents would contain one other document even only twice and nothing else. Hence, we modify this condition in the following way. We first calculate the Jaccard coefficient of the documents' vocabularies (i.e., overlapping terms). If it is at least 80%, we derive the value of m based on the ratio of the minimum and maximum term frequencies of the shared terms only.

Axiom TF-LNC combines term frequency and document length for single term queries. From a document pair the one with the higher term frequency is preferred when the documents without the term have the same length (10%-softening in our setting). We further generalize TF-LNC to multi-term queries using the sum of the term frequencies similar to the LNC1 generalization.

Lower Bound Axioms.

The axioms LB1 and LB2 capture a heuristic of lower-bounding term frequency such that long documents are not overly penalized [28]. In LB1, documents are examined that have the same retrieval score $\mathit{score}(q, d)$ (10%-softening in our case). If there is a query term t with $\mathit{tf}(t, d_1) = 0$ and $\mathit{tf}(t, d_2) > 0$, then $d_1 <_{\mathrm{LB1}} d_2$. The axiom LB2 concludes rank preferences for artificially generated documents not contained in the original result list such that we modify it to work on pairs of actual documents. If the *tf*-values of a query term pair t, t' are at most 20% different in two documents d_1 and d_2, the document with the higher frequency of the query term that comes earlier in the query is preferred.

Semantic Similarity Axioms.

Matching semantically similar terms instead of exact matches of the query terms might be helpful in vocabulary mismatch situations but also to enhance small result sets. We use WordNet to determine semantically similar terms and analyze the axioms STMC1–3 and TSSC1–2 [20, 16]. The conditions of STMC3, TSSC1, and TSSC2 are very specific and cannot be "softened" such that we do not include these axioms in our framework. The original formulation of STMC1 is for one-term queries and uses two single-term documents. We generalize this setting as follows. We calculate the semantic similarity for each word from a document d with each query term t of the query q to derive their average as the similarity $\sigma(d, q)$. From two documents d_1 and d_2 the one having the larger average value will be preferred. Also the formulation of STMC2 is generalized. For a pair of documents d_1 and d_2 we find the non-query term t from any of both documents that is maximally similar to any query term t'. We conclude $d_1 >_{\mathrm{STMC2}} d_2$ iff $|d_2|/|d_1| =_{approx_{20\%}} \mathit{tf}(t, d_2)/\mathit{tf}(t', d_1)$.

Query Aspect Axioms.

The axioms REG and AND [42, 39] focus on the individual query terms. We modify REG as follows. Let t be the query term most similar to the other query terms. If both d_1 and d_2 contain all the other query terms, the document is preferred that has a higher *tf*-value for t. In case of AND, from a document pair d_1 and d_2 where only d_1 contains all query terms, d_1 is preferred. To modify the original version of the diversity-inducing axiom DIV [22], let $J(d, q)$ be the Jaccard coefficient between the set of terms in document d and the set of query terms. If $J(d_1, q) < J(d_2, q)$, we conclude $d_1 >_{\mathrm{DIV}} d_2$. To penalize duplicate entries, we further propose a new axiom RSIM that computes the simhash-based similarity of all document pairs and from any formed similarity cluster only favors one particular document over all the others while not having preferences for documents from different clusters.

Table 1: Axioms included in our re-ranking scheme.

Purpose	Acronyms	Source	Incl.
Term frequency	TFC1–TFC3	[17]	Yes
	TDC	[17]	Yes
Document length	LNC1 + LNC2	[17]	Yes
	TF-LNC	[17]	Yes
	QLNC	[13]	No
Lower bound	LB1 + LB2	[28]	Yes
Query aspects	REG	[42, 39]	Yes
	AND	[42, 39]	Yes
	DIV	[22]	Yes
	RSIM	new	Yes
Semantic similarity	STMC1 + STMC2	[20]	Yes
	STMC3	[20]	No
	TSSC1 + TSSC2	[20]	No
Term proximity	QPHRA	new	Yes
	PROX1–5	new	Yes
	PHC + CCC	[37]	No
Other	ORIG	new	Yes
	P-RANK	[2]	Yes
	CPRF	[9]	No
	CTM	[24]	No
	CMR	[21]	No
	CEM	[3]	No

Other Axioms.

We include a straightforward new P-RANK axiom that simply prefers a document with higher PageRank. All the other published axioms discussed in Section 2 are either too specific, could not be formulated in our triple formulation, are already covered by other used axioms, or do not aim at retrieval (e.g., axiomatizing evaluation) such that we do not use them. Besides the above described axioms and our new proximity axioms (explained in the next section), we also include a simple axiom ORIG that represents the original top-k ranking and does not modify any rank decision. Its purpose is to give some "voting power" in the rank aggregation to the reasonable ideas underlying the employed basis retrieval model.

Summary.

Table 1 lists the known axioms and whether we could implement them for our re-ranking scheme; including our newly developed term proximity axioms explained in the next section and the ORIG axiom as a fallback option if no other axiom is effective.

3.3 Our New Term Proximity Axioms

As a first "proximity"-style axiom, we propose the new QPHRA axiom aimed at queries that contain highlighted phrases (e.g., via double quotes). A document containing all the query phrases is favored over a document not containing all phrases. Tao and Zhai [37] propose two axiomatic constraints on proximity importance for retrieval. However, these axioms are meant to describe properties a distance measure should have—not meaningful in our rank preference framework. Hence, we propose the new term proximity axioms PROX1–PROX5 inspired by Tao and Zhai's proposed proximity measures. Let $q = \{t_1, \ldots, t_n\}$ be a multi-term query and d_1, d_2 be two different documents. We also assume that all the query terms appear in both documents: $\forall t_j \in q : tf(t_j, d_i) > 0$, for $i \in \{1, 2\}$.

Our first proximity axiom captures proximity via the average position difference of all query term pairs.

PROX1: Let $\pi(q, d)$ be the average difference of query term pair positions calculated as

$$\pi(q, d) = \frac{1}{|P|} \sum_{(i,j) \in P} \delta(d, i, j),$$

where $P = \{(i, j) \mid i, j \in q, i \neq j\}$ is the set of all possible query term pairs and $\delta(d, i, j)$ calculates the average number of words between the terms t_i and t_j in the document d based on all positions of t_i and t_j.

If $\pi(q, d_1) < \pi(q, d_2)$, we conclude $d_1 >_{\mathrm{PROX1}} d_2$ assuming that a document with a lower π-value (i.e., query term pairs are closer to each other) should get a better rank.

We note two caveats with PROX1: It uses all pairs of term occurrences in a document, but naturally many of these will be far apart even if some pairs are very close together. It might also be desirable for the first close co-occurrence of query terms to be near the document's beginning, such that the searcher will encounter them early while reading. The following axioms PROX2 and PROX3 address these issues.

PROX2: Let $\mathit{first}(t_i, d)$ be the position of the first occurrence of query term t_i in document d and let $\mu(d, q)$ be the sum of first positions over all query terms. If $\mu(d_1, q) < \mu(d_2, q)$, we conclude $d_1 >_{\mathrm{PROX2}} d_2$.

Axiom PROX2 considers every query term separately, disregarding whether documents contain phrases from the query.

PROX3: Let $\tau(d, q)$ be the first position of the whole query q as one phrase in the document d; if q is not a phrase in d, we set $\tau(d, q) = \infty$. If $\tau(d_1, q) < \tau(d_2, q)$, we conclude $d_1 >_{\mathrm{PROX3}} d_2$.

A problem of PROX3 is that important documents may not contain the whole query as one phrase but many subsets of the query terms as shorter phrases. The following axiom measures proximity using the closest tuples of query terms.

PROX4: Let $\omega(d, q)$ be a pair (a, b), where a is the number of non-query words within the closest grouping of all terms from query q in document d, and b is the frequency of this gap value. If $\omega(d_1, q) < \omega(d_2, q)$, we conclude $d_1 >_{\mathrm{PROX4}} d_2$.

We assume that the document with the lower ω-value better matches the query, since all query terms are closer together in the document. Further improving the proximity notion, we propose an axiom PROX5 focusing on the smallest window width that contains all query terms.

PROX5: Given a query term $t_i \in q$ and a document d, we consider the size of the smallest text span containing all query terms around each occurrence of t_i. Let $\bar{s}(d, q)$ be the average smallest text span across all occurrences of all query terms in d. If $\bar{s}(d_1, q) < \bar{s}(d_2, q)$, we conclude $d_1 >_{\mathrm{PROX5}} d_2$.

3.4 Rank-aggregation

As stated previously, each axiom's ranking preferences for a given top-k result set are expressed as a matrix M_A, whose elements (i, j) determine whether or not document d_i should be ranked before document d_j according to axiom A. In order to re-rank a top-k result set based on these preferences, we derive an aggregation function that yields a single, combined preference matrix M using a machine learning model described in detail in the next section.

However, after axiom preference aggregation, the resulting matrix M probably contains conflicts: if for instance $M[i, j] > M[j, i]$ and $M[j, k] > M[k, j]$ but $M[k, i] > M[i, k]$, it is not clear what document to rank the highest. This describes a typical rank-aggregation problem that can be translated to a social choice problem for which a variety of possible rank aggregation schemes exist [7]. We choose Kemeny rank aggregation since it has been shown beneficial in meta-search engines [15]. Kemeny rank aggregation merges m rankings into one global ranking while minimizing a distance function to the original m rankings (e.g., the number of pairs that are ranked in a different ordering) [25].

Solving Kemeny rank aggregation is a well known NP-complete problem [23]. From the different existing approximation schemes proposed in the literature, we choose the KwikSort approach [1]. KwikSort originally solves the minimum feedback arc set problem in weighted tournaments. It can be transferred to our setting, since the matrix M can be viewed as the incidence matrix of a directed weighted tournament graph with the vertex set $V = \{d_1, \ldots, d_n\}$.

3.5 Learning Axiom Preference Aggregation

We use the 23 axioms shown in Table 1 based on various different ideas (term frequency, proximity, etc.). Given a query q and a pair of documents (d_i, d_j) from the result set for q, each axiom A may express a preference for ranking d_i higher ($M_A[i, j] > M_A[j, i]$), lower ($M_A[i, j] < M_A[j, i]$) or the same ($M_A[i, j] = M_A[j, i]$) as d_j. In a set of documents with known relevance judgments, the optimal ordering for each document pair is known. Hence, we view the problem of axiom preference aggregation as a supervised classification problem at the level of document pairs, seeking to infer the aggregation function that best approximates the partial ordering induced by the relevance judgments.

We train a Random Forest classifier to predict the documents' relative ordering in an optimal ranking, using the individual axiom preferences as predictors, and relevance judgments as ground truth. For each document pair, we assign a class attribute from the set $\{\text{lower}, \text{higher}, \text{same}\}$. Since the relative ordering of documents with the same relevance has no influence on the measured quality of the final ranking, we employ an instance weighting scheme that halves the impact of the "same" class. And since not all axioms might be equally important for different retrieval models (e.g., *tf-idf* already has a term frequency component), we train separate preference aggregation functions for each retrieval model.

Due to the non-availability of large click logs or other large-scale implicit user feedback on our side, we use the nDCG @10 over relevance judgments for TREC queries as the performance measure in our experiments. We randomly split the queries into a training set to learn the retrieval-model-specific aggregation functions, and a test set to evaluate their retrieval performance before and after axiomatic re-ranking.

4. EVALUATION

Our experimental evaluation of the axiomatic re-ranking scheme is conducted as a large-scale study on the TREC Web tracks of 2009–2014 with a variety of basis retrieval models serving the initial top-k results. For the experiments on the 200 queries from the Web tracks 2009–2012, we employ 16 different basis retrieval models included in the Terrier framework [33], which we use to index the ClueWeb09 Category B. For the 100 queries from the Web tracks 2013 and 2014, we use the TREC-provided Indri[3] and Terrier baselines for the ClueWeb12 as our two basis retrieval models.

To speed up the experimental process, we perform the training and testing of the axiom aggregation schemes, each axiom's individual ranking, and the KwikSort Kemeny rank aggregation on a 135-node Hadoop cluster that also hosts the ClueWeb09 and ClueWeb12 documents and corpus statistics (e.g., *idf*-values) needed in some axioms.

4.1 Axiomatic Web Track Performance

We evaluate the axiomatic re-rankings on the queries from the TREC Web tracks of 2009–2014. From the ClueWeb09-based Web tracks of 2009–2012, there is a total of 198 queries with available relevance judgments. After discarding the 18 queries for which none of the basis retrieval models find any relevant results, we randomly select 120 of the remaining 180 queries as the training set, and use the other 60 as the test set. The 16 basis retrieval models shown in Table 2 are employed; more details on these models can be found in the extensive Terrier documentation.[4] We have set up Terrier to index the Category B part of the ClueWeb09 and train the axiom aggregation functions for each model separately on the training set topics as described in Section 3.

The evaluation results on the test set topics are depicted in Table 2. The models in the table are ordered according to their "Base" performance without axiomatic re-ranking. The average base performance over all test set topics is shown in the second column of the table. The two subsequent columns show the nDCG @10 after applying axiomatic re-ranking ("+AX"), and Terrier's Markov Random Field term dependency score modifier ("MRF") to the basis result set, respectively. We note that while MRF term dependency improves upon the average base performance in all cases, the magnitude of the improvement is larger for axiomatic re-ranking for nearly half of the studied retrieval models. The fifth column shows the average nDCG @10 when applying axiomatic re-ranking after MRF term dependency; the effect sizes reported in this column are computed with respect to the "MRF" values. The final column of Table 2 shows the maximum nDCG @10 achievable on the basis models top-50 result set—i.e., when ranking these documents in an "oracle"-stlye directly by their TREC relevance judgments.

Table 2: Retrieval performance (nDCG @10**) of the different retrieval models on the test set queries. The basis model's performance (Basis), with axiomatic re-ranking (+AX), and with MRF term dependence. Significant differences between Basis/+AX, Basis/MRF and MRF/MRF+AX (paired two-sided t-test, $p = 0.05$) are marked with a dagger†; the effect size (Cohen's d) is given in brackets below each value. The final column shows the** nDCG @10 **of the best possible re-ranking.**

Model	Basis	+AX	MRF	MRF+AX	max
DPH	0.273	**0.291** *(0.062)*	**0.307**† *(0.112)*	**0.314** *(0.025)*	0.642
DFRee	0.205	**0.236** *(0.121)*	**0.230** *(0.091)*	**0.245** *(0.057)*	0.599
In_expC2	0.205	**0.214** *(0.038)*	**0.229** *(0.091)*	**0.238** *(0.031)*	0.591
TF_IDF	0.202	**0.228** *(0.098)*	**0.239** *(0.134)*	0.200 *(-0.155)*	0.589
In_expB2	0.201	**0.202** *(0.006)*	**0.234** *(0.124)*	**0.237** *(0.011)*	0.592
DFReeKLIM	0.199	**0.213** *(0.057)*	**0.224** *(0.095)*	0.224 *(-0.001)*	0.591
BM25	0.198	0.188 *(-0.044)*	**0.229** *(0.116)*	0.216 *(-0.049)*	0.587
InL2	0.197	0.197 *(-0.001)*	**0.235** *(0.139)*	0.212 *(-0.091)*	0.593
BB2	0.195	**0.197** *(0.005)*	**0.236**† *(0.151)*	0.234 *(-0.006)*	0.587
DFR_BM25	0.194	**0.206** *(0.049)*	**0.236** *(0.156)*	0.220 *(-0.062)*	0.591
LemurTF_IDF	0.187	**0.224**† *(0.151)*	**0.221**† *(0.132)*	**0.237**† *(0.060)*	0.576
DLH13	0.164	**0.187** *(0.100)*	**0.184** *(0.080)*	**0.201** *(0.067)*	0.499
PL2	0.16	**0.213**† *(0.221)*	**0.190**† *(0.125)*	**0.211** *(0.084)*	0.550
DLH	0.153	**0.187** *(0.144)*	**0.181** *(0.113)*	**0.197** *(0.064)*	0.470
DirichletLM	0.139	**0.242**† *(0.456)*	**0.192**† *(0.276)*	**0.253**† *(0.249)*	0.564
Hiemstra_LM	0.107	**0.167**† *(0.277)*	**0.161**† *(0.245)*	**0.163** *(0.005)*	0.397

Except for two retrieval models of middling performance, our axiomatic re-ranking consistently improves the average basis retrieval performance. This improvement is statistically significant (paired two-sided t-test, p=0.05) for only four retrieval models at the lower end of the performance spectrum; however, we note that MRF term dependency achieves a significant improvement in only two further cases, while the magnitude of the effect tends to be smaller. Even on our fairly small test set, axiomatic retrieval yields a mid-sized effect on the performance of poorly-performing basis retrieval models. It should be noted that the performance improvements seen especially for the models with weaker base performance only come from the axiomatic re-ranking of the top-50 results of the weak model, not by incorporating knowledge from the better-performing models.

There are several interesting observations from these initial experiments. First, the retrieval model with the second-worst base performance (DirichletLM) achieves the second-best performance after axiomatic re-ranking, both with and without MRF term dependency scoring. Second, the differences between retrieval models after re-ranking are smaller than before. However, this leveling effect is not due to the re-ranked results being almost optimally ranked. As the final column of Table 2 shows, none of the studied re-ranking approaches achieve more than half of the nDCG @10 of the optimal re-ranking; there is a considerable potential for improvement in moving the retrieval performance closer to the optimum with stronger axioms. Future re-ranking ideas probably would need to include axioms capturing more sophisticated signals of relevance than the rather simplistic assumptions of the axioms used in our study. We will shed

[3] http://lemurproject.org/clueweb12/services.php

[4] http://terrier.org/docs/v4.0/configure_retrieval.html

Table 3: Retrieval performance (nDCG @10) on the Web track 2014 topics before and after applying the axiomatic re-ranking apporach. The axiom aggregation functions are trained on the topics of the Web track 2013. Significant differences between before and after (paired two-sided t-test, $p = 0.05$) are marked with a dagger (†) and effect size according to Cohen's d is given.

Model	Before	After	Effect size
Terrier DPH	0.471	0.446	-
Indri LM	0.346	0.502†	0.69

some more light on the influence of the different axioms in a second experimental study.

Before analyzing the individual axioms' impact, we conduct an experiment similar to the above for the TREC Web track baselines of the years 2013 and 2014. We did not index the ClueWeb12 ourselves for this experiment but relied on the rankings provided by the Web track organizers as the baselines. The preference aggregation schemes are trained for the topics of the Web track 2013 and tested on the topics of 2014, yielding 50 topics each for training and testing. The results are depicted in Table 3.

As can be seen, the performance of the Indri baseline is significantly improved with a medium effect size while the Terrier baseline's performance is decreased—although not significantly. One possible explanation for the decreased Terrier DPH performance is that for this ClueWeb12 experiment, we only used 50 topics for training, while for the ClueWeb09 experiments we used 120 topics. Applying the aggregation function trained for DPH on the Web track 2009–2012 topics to the Web track 2014 test set yields a slight, albeit non-significant, performance improvement to an nDCG @10 of 0.48 for DPH on the Web track 2014 topics. This indicates that the fifty Web track 2013 topics might not suffice to train a good aggregation function for axiomatic re-ranking of DPH results.

Similarly to the ClueWeb09 setting, this experiment again indicates that axiomatic re-ranking can even out performance differences between different basis retrieval models, such that the specific model used for the initial top-k retrieval has less of an impact. Still, there is room for further improvement, as can be seen from the possible performance given an optimally re-ranked top-k result set—for this experiment, the optimal nDCG @10 is close to 1.0 on average.

4.2 Impact of the Different Axioms

To gain further insights into the influence of the different axioms on the re-ranking, we analyze the ClueWeb09 experiment in more detail. In particular, we investigate the performance of different axiom subsets, and how often they are applied and actually change the ranking decisions compared to the ORIG axiom that does not change the basis model's ranking. Further, we analyze the overlap of the different retrieval models' top-k results to account for the more homogenous performance of the different retrieval models after axiomatic re-ranking.

Axiom subsets.

We study the individual axioms' influence in a follow-up on the ClueWeb09 experiment with interesting subsets of the axioms. We run the same experimental process (learning the aggregation function on the training set topics, testing on the topics of 2012) for individual groups of axioms and for the set of all axioms without each group (the ORIG axiom is always included). A summary of the results is shown in Table 4.

Table 4: Improvements in nDCG @10 on the Web track 2012 topics for different axiom sub-groups. The second column shows the number of retrieval models (out of 16) whose performance is improved on average across the test set topics. The last column shows the average difference in nDCG @10 across retrieval models.

Axiom Group	Improved	Avg. Diff.
Term frequency axioms only	5	-0.80%
Document length axioms only	1	-0.02%
Lower bound axioms only	2	-7.79%
Query aspects axioms only	0	-15.62%
Semantic similarity axioms only	0	-14.70%
Term proximity axioms only	6	+1.37%
All without term frequency	5	-1.78%
All without document length	10	+4.54%
All without lower bound	1	-6.55%
All without query aspects	1	-11.57%
All without semantic similarity	6	-1.24%
All without term proximity	2	-5.98%

Of the six axiom subsets—document length, lower bound, query aspects, semantic similarity, term frequency, and proximity—query aspects and semantic similarity don't improve any of the retrieval models by themselves. The other four groups improve at least one model on their own, with the term proximity axioms improving the largest number of basis retrieval models, albeit by a small percentage.

A further observation can be made about subsets containing all axioms except one of the groups. Without the lower bound axioms, without the query aspects axioms, and without the proximity axioms, the fewest improvements are possible. This hints at the relative importance of these axioms. Without document length, 10 improvements are still possible. For all of the axiom subsets, the relative improvements in nDCG @10 are much smaller than for the full set. This hints at a rather complex interplay between the different axioms in achieving a better top-10 ranking.

The subset experiments show large differences in the importance of individual axioms that we further examine by analyzing the impact of the different axioms in the preference aggregation function, and to how many document pairs the different axioms could be applied.

Axiom importance, usage and rank differences.

In order to examine the different axioms' importance, we study how much they contribute to the performance of the learned preference aggregation functions. Table 5 exemplifies the mean decrease in model accuracy for the axiom preference aggregation functions of the best- and worst-performing basis retrieval models. For each axiom, the corresponding value in the table shows by what percentage the aggregation model's accuracy would decrease without that variable. The contributions of the different axioms tend to be fairly similar across retrieval models, but there are some key differences: The contribution of the ORIG axiom de-

Table 5: Feature importance for a selection of axioms in the axiom preference aggregation function for the best- and worst-performing base retrieval model.

	Mean Decrease Acc.	
Axiom	DPH	Hiemstra_LM
TFC1	19.21	10.53
TFC2	9.70	1.97
TDC	0.15	1.99
LNC1	3.18	3.93
TF-LNC	1.44	0.00
LB1	33.22	26.04
LB2	5.54	3.73
REG	21.33	20.31
DIV	27.71	24.85
STMC1	31.18	27.32
STMC2	15.25	15.16
PROX1	19.41	18.64
PROX2	16.61	12.96
PROX3	25.08	18.59
PROX4	17.76	17.84
PROX5	17.60	15.66
P-RANK	18.77	12.79
ORIG	23.54	14.82

creases with the performance of the basis retrieval model. While certain axioms, such as TDC, TF-LNC and LB2 never have a large impact on the aggregation functions, there is at least one high-impact axiom in each of the axiom groups. The exception to this are the document length axioms, which never contribute more than five percent of the aggregation accuracy.

In a similar avenue, we examine how many ranking preferences of $d_1 >_A d_2$ each individual axiom A specifies in the ClueWeb09 experiment—i.e., how often its *preconditions* are met. The distributions are quite similar for the different retrieval models. Interestingly, STMC1 (semantic similarity) is applied most frequently by far, but as can be seen from the axiom subsets experiment, it probably draws non-useful conclusions often. The axioms PROX2 and LB1 are the second most commonly applied, followed by the other proximity axioms, then TFC1, TFC2 and LNC1; LNC2, TFC3 and TF-LNC are used very rarely.

To underpin this investigation, we study the difference in the top-10 results caused by individual axioms. Again, STMC1 alone would yield the highest difference, but it does not have a high impact in any of the learned aggregation functions. The term proximity axioms, as well as TFC1 and LB1, change about 50% of the top-10 result sets. Given the high impact of especially PROX3 and LB1 and the results of the axiom subsets, this indicates that their share of the improved re-ranked performance is the highest. The axioms LNC1, TF-LNC, TDC, and LB2 alone will hardly ever change a top-10 ranking. Along with their lower impact, this indicates that they are the least important among our selection.

Result overlap of the basis models.

An unexpected finding of our axiomatic re-ranking results is that the performances of different basis retrieval models are more similar after re-ranking. A large overlap between the top-k result sets of the different retrieval models would explain not just this effect, but also the rather similar axiom impact across retrieval models.

To analyze the overlap, we measure the Jaccard coefficient between any two basis models' top-50 results. The average Jaccard coefficient of the 7 200 possible pairs is 0.6, underpinning our hypothesis of a large overlap. Furthermore, limiting the analysis to the documents with a TREC judgment of 2 or more (i.e., at least highly relevant), the average overlap increases to 0.8. When such documents are re-ranked to the top of a ranking, the nDCG @10 is significantly improved. Since these highly relevant documents are treated the same way for the individual basis retrieval models by the similarly aggregated axiom combinations, the leveled performance effect is explained.

5. CONCLUSION

We introduce an axiom-based framework to re-rank a basis retrieval model's top-k results. This way, we exploit all the findings from the last decade on axiomatic IR in a unified setup. For the first time, we demonstrate how a variety of axioms can be used in a practical setting. Our experimental analyses show the axiom-based re-ranking to improve retrieval performance for almost all the basis models—often with a medium effect size. Still, our experiments also showed that there is room for further improvements since the possible optimal re-rankings of the top-k results could achieve much higher retrieval scores. The inclusion of more sophisticated axiomatic ideas as part of the re-ranking thus is a very promising direction for future research.

Potentially to-be-included axioms comprise the axioms on query aspects that we could not restate to fit our axiomatic scheme, but also axioms on document readability or near duplicates in the results. The formulation of such practically applicable axioms might increase the possible performance improvements since such facets of relevance are not yet covered by any of our current axioms.

Another branch of interesting future work is the efficiency of the re-ranking process. In this paper, our goal was to show the possible effectiveness by improvements of retrieval performance. However, the currently achieved performance of about 2 seconds for the re-ranking of the top-50 results of a single query are far from being acceptable in a live system. Still, our current experimental setup is not yet optimized for speed such that the necessary efficiency gains to reach practical applicability should be tractable.

6. REFERENCES

[1] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5), 2008.

[2] A. Altman and M. Tennenholtz. Ranking systems: the pagerank axioms. In *EC 2005*, pp. 1–8.

[3] E. Amigó, J. Gonzalo, and F. Verdejo. A general evaluation measure for document organization tasks. In *SIGIR 2013*, pp. 643–652.

[4] P. Bruza and T. W. C. Huibers. Investigating aboutness axioms using information fields. In *SIGIR 1994*, pp. 112–121.

[5] L. Busin and S. Mizzaro. Axiometrics: An axiomatic approach to information retrieval effectiveness metrics. In *ICTIR 2013*, paper 8.

[6] Z. Cao, T. Qin, T. Liu, M. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *ICML 2007*, pp. 129–136.

[7] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. A short introduction to computational social choice. In *SOFSEM 2007*, pp. 51–69.

[8] C. L. A. Clarke, K. Collins-Thompson, P. Bennett, F. Diaz, and E. M. Voorhees. Overview of the TREC 2013 web track. In *TREC 2013*.

[9] S. Clinchant and É. Gaussier. A document frequency constraint for pseudo-relevance feedback models. In *CORIA 2011*, pp. 73–88.

[10] S. Clinchant and É. Gaussier. A theoretical analysis of pseudo-relevance feedback models. In *ICTIR 2013*, paper 6.

[11] R. Cummins and C. O'Riordan. An axiomatic comparison of learned term-weighting schemes in information retrieval: clarifications and extensions. *Artif. Intell. Rev.*, 28(1):51–68, 2007.

[12] R. Cummins and C. O'Riordan. Analysing ranking functions in information retrieval using constraints. *Information Extraction from the Internet*, 2009.

[13] R. Cummins and C. O'Riordan. A constraint to automatically regulate document-length normalisation. In *CIKM 2012*, pp. 2443–2446.

[14] F. Ding and B. Wang. An axiomatic approach to exploit term dependencies in language model. In *AIRS 2008*, pp. 586–591.

[15] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW 2001*, pp. 613–622.

[16] H. Fang. A re-examination of query expansion using lexical resources. In *ACL 2008*, pp. 139–147.

[17] H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. In *SIGIR 2004*, pp. 49–56.

[18] H. Fang, T. Tao, and C. Zhai. Diagnostic evaluation of information retrieval models. *ACM Trans. Inf. Syst.*, 29(2):7, 2011.

[19] H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. In *SIGIR 2005*, pp. 480–487.

[20] H. Fang and C. Zhai. Semantic term matching in axiomatic approaches to information retrieval. In *SIGIR 2006*, pp. 115–122.

[21] S. Gerani, C. Zhai, and F. Crestani. Score transformation in linear combination for multi-criteria relevance ranking. In *ECIR 2012*, pp. 256–267.

[22] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *WWW 2009*, pp. 381–390.

[23] E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theor. Comput. Sci.*, 349(3):382–391, 2005.

[24] M. Karimzadehgan and C. Zhai. Axiomatic analysis of translation language model for information retrieval. In *ECIR 2012*, pp. 268–280.

[25] J. G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.

[26] J. Li and R. R. Rhinehart. Heuristic random optimization. *Computers & Chemical Engineering*, 22(3):427–444, 1998.

[27] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[28] Y. Lv and C. Zhai. Lower-bounding term frequency normalization. In *CIKM 2011*, pp. 7–16.

[29] Y. Lv and C. Zhai. A log-logistic model-based interpretation of *tf* normalization of BM25. In *ECIR 2012*, pp. 244–255.

[30] B. P. McCune, R. M. Tong, J. S. Dean, and D. G. Shapiro. RUBRIC: A system for rule-based information retrieval. *IEEE Trans. Software Eng.*, 11(9):939–945, 1985.

[31] C. Meghini, F. Sebastiani, U. Straccia, and C. Thanos. A model of information retrieval based on a terminological logic. In *SIGIR 1993*, pp. 298–307.

[32] S. Na, I. Kang, and J. Lee. Improving term frequency normalization for multi-topical documents and application to language modeling approaches. In *ECIR 2008*, pp. 382–393.

[33] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A high performance and scalable information retrieval platform. In *OSIR 2006 Workshop*, pp. 18–25.

[34] F. Radlinski and T. Joachims. Query chains: Learning to rank from implicit feedback. In *KDD 2005*, pp. 239–248.

[35] R. Rahimi, A. Shakery, and I. King. Axiomatic analysis of cross-language information retrieval. In *CIKM 2014*, pp. 1875–1878.

[36] F. J. Solis and R. J.-B. Wets. Minimization by random search techniques. *Mathematics of Operations Research*, 6(1):19–30, 1981.

[37] T. Tao and C. Zhai. An exploration of proximity measures in information retrieval. In *SIGIR 2007*, pp. 295–302.

[38] A. Veloso, H. M. de Almeida, M. A. Gonçalves, and W. M. Jr. Learning to rank at query-time using association rules. In *SIGIR 2008*, pp. 267–274.

[39] H. Wu and H. Fang. Relation based term weighting regularization. In *ECIR 2012*, pp. 109–120.

[40] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. In *SIGIR 2007*, pp. 391–398.

[41] D. Zhang, R. Mao, H. Li, and J. Mao. How to count thumb-ups and thumb-downs: User-rating based ranking of items from an axiomatic perspective. In *ICTIR 2011*, pp. 238–249.

[42] W. Zheng and H. Fang. Query aspect based term weighting regularization in information retrieval. In *ECIR 2010*, pp. 344–356.