A Reproducibility Study of Information Retrieval Models

Peilin Yang
University of Delaware
Newark, DE 19716
United States
franklyn@udel.edu

Hui Fang
University of Delaware
Newark, DE 19716
United States
hfang@udel.edu

ABSTRACT

Developing effective information retrieval models has been a long standing challenge in Information Retrieval (IR), and significant progresses have been made over the years. With the increasiqng number of developed retrieval functions and the release of new data collections, it becomes more difficult, if not impossible, to compare a new retrieval function with all existing retrieval functions over all available data collections. To tackle this problem, this paper describes our efforts on constructing a platform that aims to improve the reproducibility of IR research and facilitate the evaluation and comparison of retrieval functions. With the developed platform, more than 20 state of the art retrieval functions have been implemented and systematically evaluated over 16 standard TREC collections (including the newly released ClueWeb datasets). Our reproducibility study leads to several interesting observations. First, the performance difference between the reproduced results and those reported in the original papers is small for most retrieval functions. Second, the optimal performance of a few representative retrieval functions is still comparable over the new TREC ClueWeb collections. Finally, the developed platform (i.e., *RISE*) is made publicly available so that any IR researchers would be able to utilize it to evaluate other retrieval functions.

1. INTRODUCTION

One of the key challenges in Information Retrieval (IR) is to develop effective retrieval models. Since the beginning of the field, many retrieval models have been proposed and studied [5, 10, 26, 19, 23, 6, 20, 9, 7, 32, 25, 13, 1, 21, 27, 30, 31] and many data collections have been released [29]. The large number of developed retrieval functions and the increasing number of collections make it more challenging to conduct a comprehensive comparison in terms of the retrieval performance. The commonly used practice when evaluating a new retrieval function is to hand-pick a few existing retrieval functions as baseline methods and then compare the performance of the new retrieval function with the baselines over several data collections. The choice of baseline methods and data collections varies based on the resources available for each researcher. As a result, there are many important IR basic research questions that remain unanswered. For example, given a standard TREC collection, which retrieval function is the most effective? How is the performance of function A compared with that of function B over collection C? We might be able to find the answers for some specific functions and collections from existing publications, but we are unable to answer the questions for any retrieval functions and collections. For example, it is difficult to find publications that report the performance comparison of traditional retrieval functions (e.g., Pivoted normalization method [27] and two stage language modeling method [30]) over newly released TREC ClueWeb collections. Moreover, it is also difficult to know which of two newly developed retrieval functions is more effective if such a comparison has not been reported in the paper. Thus, it is critical to conduct a comprehensive reproducibility study on information retrieval models to gain a better understanding on the performance of existing retrieval functions over a wide range of data collections.

In fact, there are quite a few recent studies that emphasized the importance of reproducibility in IR [2, 22, 28, 4, 3]. Armstrong et al. [3] evaluated the performances of five publicly available search systems over nine TREC collections and found no evidence that the retrieval models were improved from 1994 to 2005. Their follow-up study [4] further analyzed the retrieval results published at SIGIR and CIKM from 1998-2008, pointed out the baselines used in these publications were generally weak, and concluded that the ad hoc retrieval is not measurably improving. Both studies indicated the need of setting up a platform that can facilitate the reproducibility of existing retrieval functions and evaluation of new retrieval functions.

In this paper, we describe our efforts on conducting a reproducibility study for Information Retrieval models. First, we develop a web-based platform called *Reproducible Information retrieval System Evaluation* (***RISE***) for reproducing results for retrieval models. The RISE platform is designed to provide an easy yet controlled environment to facilitate the reproduce and fair comparison of different retrieval models. Second, we conduct a systematical comparison for a large number of representative retrieval functions over multiple data collections to see whether we can reproduce the reported performances and also generate benchmark results

ICTIR '16, September 12–16, 2016, Newark, Delaware, USA.
© 2016 ACM. ISBN 978-1-4503-4497-5/16/09...$15.00
DOI: http://dx.doi.org/10.1145/2970398.2970415

over the collections that have not been evaluated in the original papers.

Specifically, *RISE* can be regarded as an instantiation of Privacy Preserving Evaluation (PPE)[8] and Evaluation as a Service (EaaS)[17, 24]. When evaluating the retrieval performance of multiple retrieval functions over one collection, it provides the same underlying indexes of the collection and eliminates the impact of document pre-processing methods. Moreover, the format of queries is standardized and the evaluation measure can be comprehensive but yet flexible based on users need, e.g. choose title as the query of TREC query topic together with MAP reported as opposed to choose title and description as the query with P@10 reported. Thus, the *RISE* platform enables users to focus on the implementation of retrieval models themselves by automating other steps that are necessary for processing the document collections and evaluating retrieval models. Another important advantage of the *RISE* platform lies in its ability to evaluate retrieval models on the server side, which avoids the need of disseminating data collections.

With the developed *RISE* platform, we are able to conduct a more comprehensive reproducibility study for information retrieval models. In particular, we implement and evaluate more than 20 basic retrieval functions over 16 standard TREC collections. Experimental results allow us to make a few interesting observations. We first compare the evaluation results with those reported in the original papers, and find that the performance differences between the reproduced results and the original ones are small for majority of the retrieval functions. Among all the implemented functions, only one of them consistently generates worse performance than the one reported in the original paper. Moreover, we report the retrieval performance of all the implemented retrieval functions over all the 16 TREC collections including recently released ClueWeb sets. To the best of our knowledge, this is the first time of reporting such a large scale comparison of IR retrieval models. Such a comparison can be used as the performance references of the selected models.

The *RISE* platform is available at http://rires.info:8080/. Both source codes and evaluation results of the implemented retrieval functions can also be found on the website.

2. RELATED WORK

There have been significant efforts on developing various web services for IR evaluation. Lin et al. [16] proposed an open-source IR reproducibility challenge where they split the IR system into pieces of components such as two kinds of tokenization methods and four different IR toolkits. By easily configuring different combinations of these components, we can have a partially filled matrix indicating the performances of specific combinations of the components. Such transparent experiment set up makes it possible to have a better understanding about the impact of different components. Gollub et al. [11] described a reference implementation of their proposed IR evaluation web service which bears the important properties like web dissemination and peer-to-peer collaboration. Hanbury et al. [12] reviewed some of the existing automated IR evaluation approaches and proposed a framework for web service based component-level IR system evaluation. Lagun and Agichtein proposed a web service, which enables large scale studies of remote users[15]. Their system focused on providing a platform that reproduces and extends the previous findings on how users interact with the search engine especially the search results.

Our developed *RISE* system is closely related to the ideas of Privacy Preserved Evaluation (PPE)[8] and Evaluation as a Service (EaaS)[24, 17, 18]. The system is designed as a web service to provide a unified interface for the users to evaluate their models/algorithms. This design enables the system to host the data collections instead of shipping the data collections to researchers, which can ensure the privacy of the collections. VIRLab [8] provides similar Web service for users to implement retrieval functions, but it is mainly designed to facilitate teaching IR models. Thus, it does not support as many collection statistics as those provided in the *RISE* system, and the users can not see the functions implemented by other users. The uniqueness of *RISE* system is that it is specifically designed to facilitate the implementation and evaluation of retrieval functions.

The SIGIR 2015 Workshop on Reproducibility, Inexplicability, and Generalizability of Results (RIGOR) [14] is one of the venues that encourage the study of reproducibility. Their reproducibility challenge invited developers of 7 open-source search engines to provide baselines for TREC GOV2 collection. Trotman et. al. [28] and Muhleisen el. al. [22] have also tried to reproduce retrieval results for IR models, but the number of retrieval functions and the number of collections used in these studies (1 function 1 collection for [28] and 9 functions 10 collections for [22]) are not as large as what we studied in this paper.

Compared with the previous studies, our work is different in the following two aspects. First, the *RISE* system is specifically designed for the reproducibility study of retrieval models. It hides details about collection processing and evaluation, and enables users to focus on only the implementation of retrieval models. Due to its flexibility, we are able to implement and compare a wide range of retrieval functions that were not implemented in any other open-source toolkits. Second, our reproducibility study includes more retrieval functions and more data collections. The ultimate goal of the *RISE* system is to provide a complete set of benchmark results of IR models.

3. RISE - A REPRODUCIBILITY PLATFORM FOR RETRIEVAL MODELS

To reproduce the results of retrieval models, we implement a web-based Reproducible Information retrieval System Evaluation (*RISE*) platform. The platform is designed to provide a well-controlled environment for the users to implement and evaluate retrieval functions. Figure 1 shows the architecture of RISE. *RISE* is basically a web service built on top of a modified version of the Indri[1] toolkit. *RISE* hosts data collections on the server side, processes documents, and builds the indexes. Users need to upload their own implementations of retrieval functions based on the provided templates. After the code is uploaded, *RISE* automatically compiles it and evaluates it over the selected data collections. The evaluation results of the retrieval function will then be added to the score boards and thus be available for comparison.

Any registered users can contribute the implementation of a retrieval model to the system. Users are expected to be familiar with C++ but not necessarily familiar with In-

[1] http://sourceforge.net/projects/lemur/

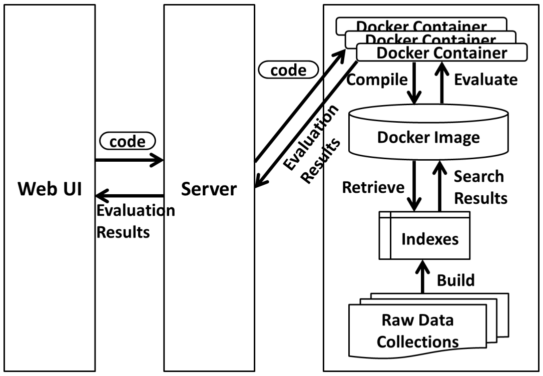


Figure 1: System Architecture

dri, as we provide detailed instructions with sample codes on how to access the statistics from the indexes and how to implement ranking models with the provided statistics. Moreover, *RISE* is an open system which allows any user to view any other users' implementations of the models. This functionality makes it possible for a user to easily try different variants of existing retrieval functions. The modified version of the Indri toolkit provides various statistics that are not available in the original version. These new statistics include query term frequency, average document term frequency, etc.

After the code is submitted to the server and successfully compiled, a Docker container is temporarily initiated on top of the static Docker image. (Docker container is like a sandbox which provides an isolated environment acting like an operating system. For more information please refer to https://www.docker.com/) The Docker image includes the indexes built from data collections and the modified Indri toolkit that will be used as the facility to run the model and generate the ranking list. A Docker image can be utilized by several Docker containers at the same time while keep the same underlying view of index and thus is the ideal choice for the system. Several Docker containers can be initiated in parallel so that multiple models can be compiled, run and evaluated at the same time. Moreover, by carefully setting the Docker container we can control the CPU and memory usage as well as security related settings (e.g. network, 3rd party libraries) so that the system is more robust against malicious/careless usage. The running Docker container compiles the codes, generate the ranking list, then evaluate the results. After that, the performance is rendered to the users and users can opt to choose other models to compare with.

There are several major benefits of the developed *RISE* platform. (1) The data collections are kept on the server side to preserve the data privacy. (2) The platform uses Docker to control the evaluation environment, which is more secure, faster and more reliable. (3) The platform provides a repository of the implementation of various retrieval functions. Registered users can upload their own codes, and these codes can be reused by other users. Such a repository could eliminate redundant efforts of implementing baseline methods among IR researchers. (4) The platform maintains the scores for each implemented retrieval function over all available data sets. The score boards could become a valuable reference when a new retrieval function needs to be evaluated and compared with the state of the art methods.

4. REPRODUCED RETRIEVAL FUNCTIONS

With the developed *RISE* platform, we conduct a reproducibility study of IR models with 21 representative retrieval functions. These retrieval functions include the representative ones from the vector space models [27, 23], the classic probabilistic models [25], the language modeling approaches [31, 30], the divergence from randomness models [1], the axiomatic models [9, 19], and the information theory based models [6].

Let us first explain the notations used in the paper.

- $|q|$: the number of terms in query q
- c_t^q: the number of occurrences of term t in query q
- c_t^d: the number of occurrences of term t in document d
- l_d: the length of document l
- c_d: the number of unique terms in document d
- F_t: the total number of term t in collection
- N_t: the number of documents containing term t
- $|C|$: the number of terms in collection
- N: the number of documents in collection
- L: the average document length in collection

We now provide more details about the retrieval functions that are included in the reproducibility study. All the implemented functions are summarized in Table 1 and Table 2.

4.1 Okapi BM25 and its variants

Okapi BM25 is one of the representative retrieval functions derived from the classical probabilistic retrieval model. It was first proposed at TREC-3 [25], and has become one of the most commonly used baseline retrieval functions. This function is denote as **BM25** in the paper.

Axiomatic approaches was first applied to Okapi BM25 to develop new retrieval functions [9] in 2005. The basic idea is to search for a retrieval function that can satisfy all reasonable retrieval constraints. Instead of blindly search for the function, one strategy is to start with an existing retrieval function, such as Okapi BM25, find its general form, and use the retrieval constraints to find different instantiations that can satisfy more retrieval constraints. The previous study derived two variants based on Okapi BM25, and they are referred to as **F2EXP** and **F2LOG** in the paper. Compared with the original BM25 function, these two variants have different implementations for both term frequency (TF) normalization part and the inverse document frequency (IDF) part.

Another variant of BM25 came from the study of Dirichlet Priors for term frequency normalization [13]. This variant, denoted as **BM3**, replaces the original TF normalization components in the BM25 function with the Dirichlet Priors TF normalization component.

Table 1: Retrieval functions that are reproduced in our study (Part 1)

Okapi BM25 and its variants	BM25	$\sum_{t\in q} \frac{(k_3+1)\cdot c_t^q}{k_3+c_t^q} \cdot \frac{(k_1+1)\cdot c_t^d}{c_t^d+k_1\cdot(1-b+b\cdot\frac{l_d}{L})} \cdot ln\left(\frac{N-N_t+0.5}{N_t+0.5}\right)$
	F2EXP	$\sum_{t\in q} \frac{c_t^d}{c_t^d+s+s\cdot\frac{l_d}{L}} \cdot \left(\frac{N+1}{N_t}\right)^k$
	F2LOG	$\sum_{t\in q} \frac{c_t^d}{c_t^d+s+s\cdot\frac{l_d}{L}} \cdot ln\left(\frac{N+1}{N_t}\right)$
	BM3	$\sum_{t\in q} \frac{(k_3+1)\cdot c_t^q}{k_3+c_t^q} \cdot \frac{(k_1+1)\cdot tfn}{k_1+tfn} \cdot ln\left(\frac{N-N_t+0.5}{N_t+0.5}\right)$ $tfn = \frac{c_t^d+\mu\cdot\frac{F_t}{\lvert C\rvert}}{l_d+\mu} \cdot \mu$
	BM25+	$\sum_{t\in q} \frac{(k_3+1)\cdot c_t^q}{k_3+c_t^q} \cdot \left[\frac{(k_1+1)\cdot c_t^d}{c_t^d+k_1\cdot(1-b+b\cdot\frac{l_d}{L})} + \delta\right] \cdot ln(\frac{N+1}{N_t})$
Pivoted and its variants	PIV	$\sum_{t\in q} \frac{1+ln(1+ln(c_t^d))}{(1-s)+s\cdot\frac{l_d}{L}} \cdot ln\left(\frac{N+1}{N_t}\right)$
	F1EXP	$\sum_{t\in q} (1+ln(1+ln(c_t^d))) \cdot \frac{L+s}{L+s\cdot l_d} \cdot \left(\frac{N+1}{N_t}\right)^k$
	F1LOG	$\sum_{t\in q} (1+ln(1+ln(c_t^d))) \cdot \frac{L+s}{L+s\cdot l_d} \cdot ln\left(\frac{N+1}{N_t}\right)$
	PIV+	$\sum_{t\in q} \left[\frac{1+ln(1+ln(c_t^d))}{(1-s)+s\cdot\frac{l_d}{L}} + \delta\right] \cdot ln\left(\frac{N+1}{N_t}\right)$
	NTFIDF	$\sum_{t\in q} \left[\left[\omega\cdot f\left(\frac{c_t^d}{l_d/c_d}\right) + (1-\omega)\cdot f\left(c_t^d\cdot log_2\left(1+\frac{L}{l_d}\right)\right)\right] \cdot \left[ln\left(\frac{N+1}{N_t}\right)\cdot f\left(\frac{F_t}{N_t}\right)\right]\right]$ $\omega = \frac{2}{1+log_2(1+\lvert q\rvert)}, f(x) = \frac{x}{1+x}$

Following the axiomatic methodology, Lv and Zhai [19] revealed a deficiency of the BM25 in its TF normalization component, i.e., the TF normalization component is not lower-bounded properly. To fix this problem, a variant of BM25, denoted as **BM25+**, was proposed. The main change is to add a lower bound to the TF normalization part.

4.2 Pivoted normalization function and its variants

Pivoted normalization method, denoted as **PIV**, is one of the most representative retrieval functions derived from the vector space model [27]. It can be regarded as one of the best-performing TF-IDF retrieval functions.

Axiomatic approaches were also applied to derive variants of the pivoted normalization method [9]. The two variants are denoted as **F1EXP** and **F1LOG**. Compared with the original function, the two variants are different in their implementations of IDF and TF normalization.

Similar to BM25, low-bounding term frequency normalization has also been applied to the pivoted function. The variant is denoted as **PIV+**, and it differs from the original function in having a lower bound added to the TF normalization component.

A novel TF-IDF term weighting scheme was proposed in 2013 to capture two different aspects of term saliency [23]. In particular, its TF component is a combination of two normalization strategies, in which one prefers short documents while the other prefers long documents. Its form is quite different from the pivoted normalization function. We include it as one of the variants for the pivoted normalization function because it uses a novel TF-IDF weighting strategy.

4.3 Language modeling approaches

Dirichlet prior method, denoted as **DIR**, is one of the representative retrieval functions derived using the language modeling approaches [31]. It uses the Dirichlet prior smoothing method to smooth a document language model and then ranks the documents based on the likelihood of the query is generated by the estimated document language models.

Two-stage language models were proposed to explicitly capture the difference influences of the query and document collection on the optimal parameter setting [30]. Compared with the Dirichlet prior method, the two-stage smoothing method (denoted as **TSL**) interpolates the smoothed document language model with a query background language model.

Instead of assuming document models take the form of a multinomial distribution over words, Multiple-Bernoulli language models assume that the document is a sample from a Multiple-Bernoulli distribution [21]. The retrieval function is denoted as **BLM**.

Similar to BM25 and Pivoted, Dirichlet prior method has also been studied using axiomatic approaches. Two variants derived using the axiomatic approaches [9] are denoted as **F3EXP** and **F3LOG**. The variant derived based on the lower bound term frequency normalization [19] is denoted as **DIR+**.

4.4 Divergence from Randomness Models

The **PL2** model is a representative retrieval function of the divergence from randomness framework [1]. It measures the randomness of terms using Poisson distribution with Laplacian smoothing.

The first variant of the PL2 is to replace the original TF normalization component with the Dirichlet prior TF normalization [13]. This variant is denoted as **PL3**.

The second variant of the PL2 considered in this paper is to apply the lower bound term frequency normalization [19]. It is denoted as **PL2+**.

4.5 Information-based Models

A family of information-based models was proposed for ad hoc IR [6]. These models focused on modeling relevance based on how a word deviates from its average behavior. Two power law distributions (e.g., a smoothed power-law

Table 2: Retrieval functions that are reproduced in our study (Part 2)

Language modeling approaches	DIR	$\sum_{t\in q} ln\left(\frac{c_t^d+\mu\cdot\frac{F_t}{\lvert C\rvert}}{l_d+\mu}\right)$
	TSL	$\sum_{t\in q}((1-\lambda)\cdot\frac{c_t^d+\mu\cdot\frac{F_t}{\lvert C\rvert}}{l_d+\mu}+\lambda\cdot\frac{F_t}{\lvert C\rvert})$
	BLM	$\sum_{t\in q}\frac{c_t^d+\mu\cdot\frac{F_t}{\lvert C\rvert}}{l_d+\frac{\lvert C\rvert}{F_t}+\mu-2}$
	F3EXP	$\sum_{t\in q}(1+ln(1+ln(c_t^d)))\cdot\left(\frac{N+1}{N_t}\right)^k-\frac{(l_d-\lvert q\rvert)\cdot\lvert q\rvert\cdot s}{L}$
	F3LOG	$\sum_{t\in q}(1+ln(1+ln(c_t^d)))\cdot ln\left(\frac{N+1}{N_t}\right)-\frac{(l_d-\lvert q\rvert)\cdot\lvert q\rvert\cdot s}{L}$
	DIR+	$\sum_{t\in q}\left[ln\left(1+\frac{c_t^d}{\mu\cdot\frac{F_t}{\lvert C\rvert}}\right)+ln\left(1+\frac{\delta}{\mu\cdot\frac{F_t}{\lvert C\rvert}}\right)\right]+\lvert q\rvert\cdot ln\frac{\mu}{l_d+\mu}$
Divergence from Randomness Models	PL2	$\sum_{t\in q}\frac{tfn\cdot log_2(tfn\cdot\lambda)+log_2e\cdot(\frac{1}{\lambda}-tfn)+0.5\cdot log_2(2\pi\cdot tfn)}{tfn+1}$ $tfn=c_t^d\cdot log_2\left(1+c\cdot\frac{L}{l_d}\right)$
	PL3	$\sum_{t\in q}\frac{tfn\cdot log_2(tfn\cdot\lambda)+log_2e\cdot(\frac{1}{\lambda}-tfn)+0.5\cdot log_2(2\pi\cdot tfn)}{tfn+1}$ $tfn=\frac{c_t^d+\mu\cdot\frac{F_t}{\lvert C\rvert}}{l_d+\mu}\cdot\mu$
	PL2+	$\sum_{t\in q,\lambda>1}\left[\frac{tfn\cdot log_2(tfn\cdot\lambda)+log_2e\cdot(\frac{1}{\lambda}-tfn)+0.5\cdot log_2(2\pi\cdot tfn)}{tfn+1}+\frac{\delta\cdot log_2(\delta\cdot\lambda)+log_2e\cdot(\frac{1}{\lambda}-\delta)+\frac{log_2(2\pi\delta)}{2}}{\delta+1}\right]$ $tfn=c_t^d\cdot log_2\left(1+c\cdot\frac{L}{l_d}\right)$
Information-based Models	SPL	$\sum_{t\in q}-ln\left(\frac{\lambda_t^{\frac{n_t^d}{n_t^d+1}}-\lambda_t}{1-\lambda_t}\right)$ $\lambda_t=\frac{F_t}{N}$ and $n_t^d=c_t^d+ln\left(1+c\cdot\frac{L}{l_d}\right)$
	LGD	$\sum_{t\in q}-ln\left(\frac{\lambda_t}{n_t^d+\lambda_t}\right)$ λ_t and n_t^d as shown above

distribution and log-logistic distribution) were used, and the corresponding functions are denoted as **SPL** and **LGD**.

5. EXPERIMENTS

We now describe the experiment design and results for our reproducibility study. The first set of experiments mainly focuses on whether we can reproduce the retrieval results that have been reported in the previous studies and whether the reproduced results are consistent with that have been reported. The second set of experiments aims to examine how well the retrieval functions perform on the newly released data sets and checks whether the conclusions are consistent with the previous findings. Finally, we also provide reference performance for all the reproduced retrieval functions over a wide range of TREC collections including the newly released ClueWeb collections.

5.1 Reproducibility study

5.1.1 Experiment Design

For the reproducibility experiments, we conduct experiments over 11 data sets that have been used in the ad hoc retrieval task at TREC-1, TREC-2, TREC-3, TREC-6, TREC-7, TREC-8; the small web track at TREC-8; the terabyte track at TREC 2004-2006; and the robust track at TREC 2004. The statistics of the data collections are summarized in Table 3.

All the collections are stemmed using Porter's stemmer. We mainly focus on the title part of the query topics. If the performance of title query is not reported by the original paper, then we use whatever query (e.g. description part or title+description+narrative) that was originally used. Please note that for some papers the authors reported the performances on the combination of multiple query topic sets, e.g. TREC678 as one query set. For this kind of query we treat the three years' topics as one query set like what the original authors did.

We evaluate the retrieval functions over these data collections and compare our results with what have been reported in the previous studies. The results are evaluated with MAP@1000, and the evaluation results are computed using `trec_eval`[2].

5.1.2 Results

We evaluate the retrieval performance for each of the 21 retrieval functions described in the previous section over all the data collections mentioned in Table 3. We then compare our reproduced results of a retrieval function with the original results reported in the paper that proposed the function. Due to the space limit, we can not report all the reproduced results, so we summarize a few main findings here.

WT2G and **disk4&5** are the two commonly used document collections in the previous study. We summarize the performance comparison between the reproduced results and the original results on these two data sets in Table 4 and Table 5 respectively. Note that **disk4&5** refers to all the data sets that use disk 4 and 5 as document collections, and it includes TREC6, TREC7 and TREC8. Let us first explain the notations in the two tables. The `orig.` column lists the originally reported results. The `repd.` column are the reproduced results. Either positive or negative difference between `orig.` and `repd.` is shown as percentage w.r.t the `orig.` in column `diff.`. The free parameter(s) used by the original

[2] http://trec.nist.gov/trec_eval/

Table 3: Data collections used for the reproducibility study

	Topics	Doc. collection	#documents	avdl
ad hoc task at TREC-1	51-100	**disk1&2**	741,856	412.89
ad hoc task at TREC-2	101-150			
ad hoc task at TREC-3	151-200			
ad hoc task at TREC-6	301-350	**disk4&5**	528,155	467.553
ad hoc task at TREC-7	351-400			
ad hoc task at TREC-8	401-450			
robust track at TREC 2004	601-700			
small web task at TREC-8	401-450	**WT2G**	247,491	1057.59
terabyte track at TREC 2004	701-750	**GOV2**	25,205,179	937.252
terabyte track at TREC 2005	751-800			
terabyte track at TREC 2006	801-850			

Table 4: Performance comparison of reproduced and original results on **WT2G**

Models	**orig.**	**repd.**	**diff.**	**para.**
BM25 and its variants				
BM25	0.310	0.315	+1.61%	$b = 0.2$
F2EXP	0.289	0.297	+2.77%	$s = 0.2^*$
F2LOG	0.295	0.301	+2.03%	$s = 0.3^*$
BM3	0.316	0.295	-6.65%	$\mu = 2700$
BM25+	0.318	0.318	+0.00%	$b = 0.2$ $\delta = 1.0$
PIV and its variants				
PIV	0.292	0.295	+1.03%	$s = 0.1$
F1EXP	0.288	0.278	-3.47%	$s = 0.0^*$
F1LOG	0.288	0.277	-3.82%	$s = 0.0^*$
PIV+	0.295	0.299	+1.36%	$s = 0.01$ $\delta = 0.4$
Language modeling approaches				
DIR	0.294	0.310	+5.44%	$\mu = 3000$
TSL	0.278	0.312	+12.23%	$\mu = 3500^*$ $\lambda = 0.0^*$
F3EXP	0.288	0.290	+0.69%	$s = 0.05^*$
F3LOG	0.290	0.293	+1.03%	$s = 0.05^*$
DIR+	0.312	0.312	+0.00%	$\mu = 3000^*$ $\delta = 0.01$
Divergence from Randomness Models				
PL3	0.293	0.288	-1.71%	$\mu = 9700$
PL2+	0.326	0.327	+0.31%	$c = 23$ $\delta = 0.8$

Table 5: Performance comparison of reproduced and original results on **disk4&5**

RM	**orig.**	**repd.**	**diff.**	**para.**
BM25 and its variants				
BM25	0.254	0.247	-2.76%	$b = 0.4$
BM3	0.251	0.238	-5.18%	$\mu = 950$
BM25+	0.255	0.249	-2.35%	$b = 0.4$ $\delta = 1.0$
PIV and its variants				
PIV	0.241	0.221	-8.30%	$s = 0.05$
PIV+	0.246	0.238	-3.25%	$s = 0.5$ $\delta = 0.01$
Language modeling approaches				
DIR+	0.253	0.252	-0.40%	$\mu = 1000^*$ $\delta = 0.01$
Divergence from Randomness Models				
PL3	0.230	0.239	+3.91%	$\mu = 1600$
PL2+	0.254	0.255	+0.39%	$c = 9$ $\delta = 0.8$
Information-based Models				
LGD	0.250	0.251	+0.40%	$c = 2.0$
SPL	0.254	0.251	-1.18%	$c = 9.0$

paper are reported in column `para.` where * means the parameter is not explicitly reported in the original paper and we just pick the optimal one by grid search. The original paper of BM25 and PIV did not report the performances on the collections that we select. Instead, we use what were reported in [13, 19] for these two models as their `orig.` results. Note that some retrieval functions are missing from the table because their original papers did not report the performance on the corresponding collection.

The results show that the performance differences with respect to the original performance, i.e, `diff.`, are small. Most of them are in the range of $[-5\%, +5\%]$. This indicates that we are able to successfully reproduce the retrieval performance for these functions.

To gain a better understanding of the reproduced results for all retrieval function, we summarize the performance difference (both mean and standard deviation) between the original and reproduced results for each of the retrieval function. The results are shown in Table 6. Although the reproduced results are not exactly the same as what were reported, the differences are generally small. We do not have the results for BLM because the authors of that paper did not report the performances on any collection that we have selected.

Among all the retrieval functions, PL2 has the largest standard deviation for the performance differences, and NTFIDF has the largest mean performance difference. We provide more detailed reproduced results for these two functions in Table 7. It is clear that the performance differences are consistent over almost all the collections. One possible explanation is that these two functions were originally implemented using the Terrier[3] retrieval system as opposed to Indri used in our paper. As pointed out in the previous study [22], using different toolkits could lead to different evaluation results.

[3] http://terrier.org/

Table 6: The mean and standard deviation of the performance difference between the reproduced and original results

Functions	Mean	Std.
BM25 and its variants		
BM25	-2.08%	4.11%
F2EXP	+0.68%	2.18%
F2LOG	+0.22%	1.63%
BM3	-5.92%	0.74%
BM25+	-0.67%	1.19%
PIV and its variants		
PIV	-3.64%	4.67%
F1EXP	-6.62%	2.23%
F1LOG	-7.76%	2.79%
PIV+	-0.94%	2.31%
NTFIDF	-17.08%	4.71%
Language modeling approaches		
DIR	+1.03%	3.26%
TSL	+4.09%	6.18%
F3EXP	-2.65%	2.72%
F3LOG	-4.11%	3.74%
DIR+	-0.20%	0.20%
Divergence from Randomness Models		
PL2	+5.54%	17.73%
PL3	+0.59%	2.41%
PL2+	+0.35%	0.04%
Information-based Models		
SPL	-4.60%	3.42%
LGD	-2.04%	2.45%

Table 7: Reproduced performance comparison for PL2 and NTFIDF

Functions	collections	orig.	repd.	diff.	para.
PL2	TREC1	0.207	0.257	+24.46%	$c = 1.0$
	TREC2	0.238	0.285	+19.60%	$c = 1.0$
	TREC3	0.271	0.327	+20.89%	$c = 1.0$
	TREC6	0.257	0.233	-9.30%	$c = 1.0$
	TREC7	0.221	0.196	-11.39%	$c = 1.0$
	TREC8	0.256	0.228	-11.01%	$c = 1.0$
NTFIDF	TREC678	0.234	0.209	-10.64%	
	ROBUST04	0.302	0.245	-18.84%	
	GOV2	0.317	0.248	-21.77%	

Table 9: Free Parameters used in Parameter Tuning

Model	Para. Range	Incr.
BM25	$b \in [0, 1]$	0.05
PIV, F1EXP, F1LOG, F2EXP, F2LOG, F3EXP, F3LOG	$s \in [0, 1]$	0.05
DIR, BLM,	$\mu \in [500, 5000]$	500
TSL	$\mu \in [500, 5000]$ $\lambda \in [0, 1]$	500 0.1
PL2	$c \in [0.5] \cup [1, 25]$	1
BM3, PL3	$c \in [0.5] \cup [0.75] \cup [1, 9]$ $\mu \in [500, 5000]$	1 500
BM25+	$b \in [0, 1]$	0.05
PIV+	$s \in [0, 1]$	0.05
DIR+	$\mu \in [500, 5000]$	500
PL2+	$c \in [0.5] \cup [1, 25]$	1
BM25+, PIV+, DIR+, PL2+	$\delta \in [0.0, 1.5]$	0.1

5.2 Performance Comparison on Web Search Collections

Not only can the *RISE* system provide a platform to reproduce the results of existing IR models, but also minimize the efforts when evaluating IR models over new collections. Whenever there is a new data collection available, the *RISE* system can easily run all the implemented retrieval functions on the new data collection and generate evaluate results for each function.

We conduct experiments to evaluate the performance of retrieval functions over 5 data sets used in the Web track from TREC 2010 to TREC 2014. The Web track at TREC 2010 to TREC 2012 used the ClueWeb09[4] as the document collection. Each year's Web track has 50 topics. Since the entire ClueWeb09 collection is too big to host on our server, we used the category B colleciton, which contains a subset of about 50 million English pages. The Web track at TREC 2013 to TREC 2014 used the ClueWeb12[5] as the document collection. Each data set has 50 topics developed by NIST. Again, due to the huge size of the original ClueWeb12 data set, we evaluate the retrieval functions over a subset of collection. The subset is generated by sampling documents from the raw collection. We use Indri default query likelihood baseline to retrieve top 10,000 documents for each query and make these documents as the sampled collection. Following the measured used at the TREC Web track, ERR@20 is used to evaluate the performance for these data sets. Due to the space limit, instead of reporting the performance over each Web track data set, we report the performance based on the document collection used. For example, **CW09** corresponds to the data set combining data used in the Web track at TREC 2010-2012. Similarly, **CW12** corresponds to the data set combining data used in the Web track at TREC 2013-2014.

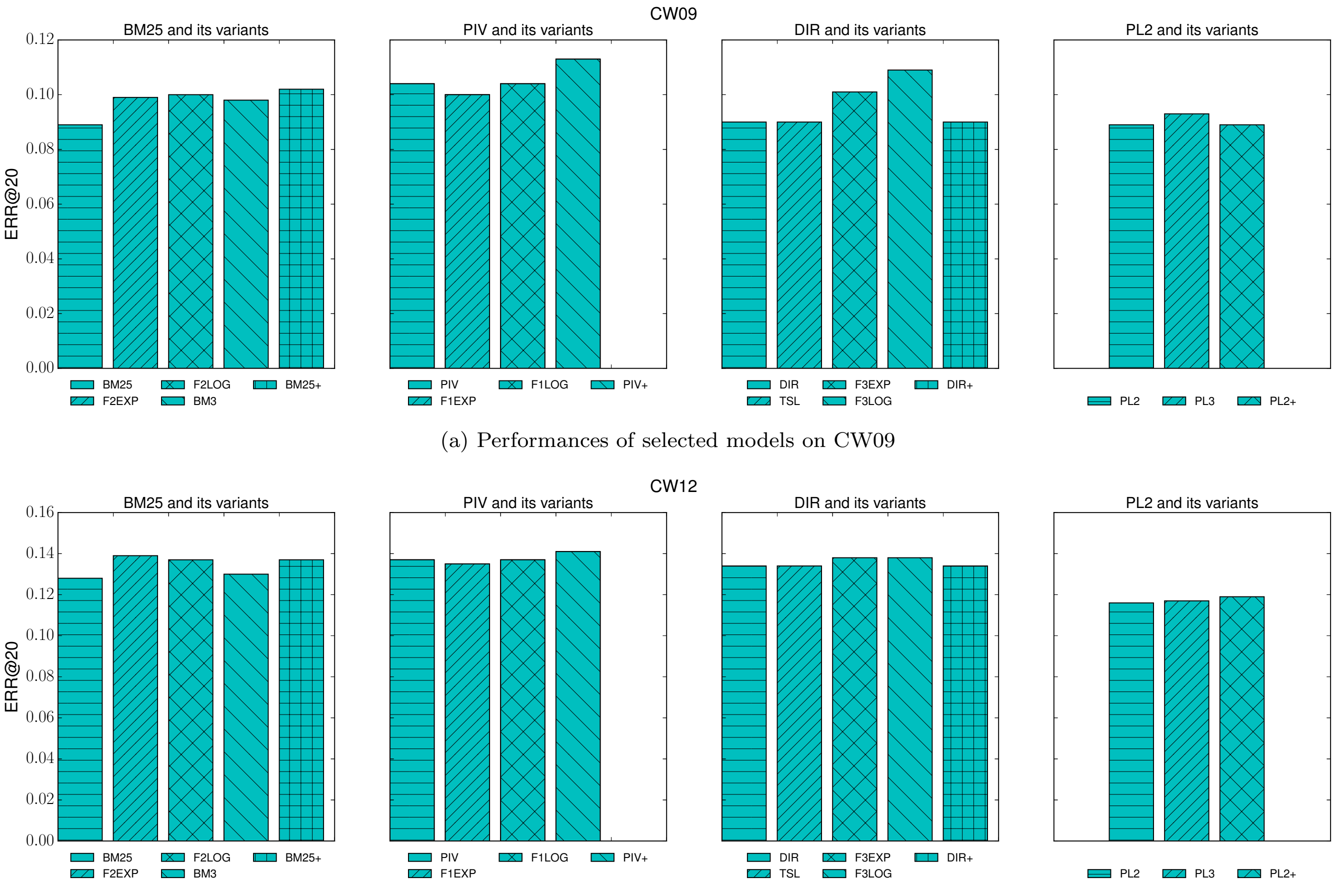
As discussed in the previous section, many variants have been proposed to improve the performance of representative retrieval functions such as BM25, PIV, DIR and PL2. All those studies were conducted over the traditional TREC collections. Thus, it would be interesting to see whether the improvement would still exist on the new Web collections.

Figure 2 shows the optimal performance comparison of the representative retrieval functions with their variants on the new Web collections. We can make a few interesting observations. First, it is interesting to see that most variants can outperform their original retrieval functions. For example, all the variants of BM25 performs better than BM25 on both collections. The only exception is the PIV function. PIV performs really well on the two new collections. Second, divergence from randomness models do not perform as well as other retrieval functions. Finally, the optimal performances of the BM25 variants, PIV variants and DIR variants are comparable.

[4] http://lemurproject.org/clueweb09/

[5] http://lemurproject.org/clueweb12.php/

Figure 2: Optimal Performances on ClueWeb Collections



(a) Performances of selected models on CW09

(b) Performances of selected models on CW12

5.3 Summary

To serve as a future reference, we summarize the optimal performance of all the retrieval functions over all the data sets in Table 8. Due to the space limit, the data sets are categorized based on the collections used, so data sets used in multiple tracks might be grouped into one because they used the same document collections. For each retrieval function, the free parameters are tuned via grid search and the parameter ranges are summarized in Table 9.

The optimal performances for the selected retrieval models on all collections are shown in Table 8. To the best of our knowledge this is the first time of reporting such large scale and comprehensive performances of retrieval models.

6. CONCLUSIONS AND FUTURE WORK

This paper describes our efforts on building the Reproducible Information retrieval System Evaluation (*RISE*) platform. *RISE* is a Web service that facilitates the implementation and evaluation of IR models. In particular, it can serve as an implementation repository of retrieval functions. Users can not only submit their own implementations but also view the implementations submitted by other users. With such an implementation repository, the *RISE* can also facilitate the evaluation of existing retrieval functions over new data collections. As demonstrated in the paper, we have implemented 21 retrieval functions and evaluate them over 16 TREC data sets. All the implementations and the evaluation results are available at the *RISE* platform [6].

For the future work, we plan to provide continuous support to the *RISE* system so that other researchers from the community can contribute and leverage the system for their own research. Regarding the system design, we plan to provide more functionalities, such as training/testing data splitting, to facilitate the evaluation process.

7. ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant Number IIS-1423002. We thank the reviewers for their useful comments.

8. REFERENCES

[1] G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, 2002.

[2] J. Arguello, F. Diaz, J. Lin, and A. Trotman. Sigir 2015 workshop on reproducibility, inexplicability, and generalizability of results (rigor). In *Proceedings of the*

[6] http://rires.info:8080/

Table 8: Optimal MAP/ERR@20 for all collections. * indicates the model is significant better than the base model in its category (always the first one). † indicates the model is the best performed in its category. ‡ indicates the model is significant better than all other models in its category. All significant tests are at p = 0.05 by a paired one-tailed t-test.

RM	**disk12**	**disk45**	**WT2G**	**GOV2**	**CW09**	**CW12**
BM25 and its variants						
BM25	0.204	0.248	0.315	0.297	0.089	0.128
F2EXP	0.228^{*}	0.251	0.297	0.284	0.099^{*}	$0.139^{\dagger}$
F2LOG	0.231^{*}	$0.252^{\dagger}$	0.302	0.297	0.100^{*}	0.137
BM3	0.234^{*}	0.241	0.296	0.283	0.098^{*}	0.130
BM25+	$0.235^{*\dagger}$	0.249	$0.318^{\dagger}$	$0.301^{\dagger}$	$0.102^{*\dagger}$	0.137
PIV and its variants						
PIV	0.201	0.221	0.294	0.254	0.104	0.137
F1EXP	0.198	0.221	0.278	0.240	0.100	0.135
F1LOG	0.200	0.217	0.277	0.255	0.104	0.137
PIV+	$0.207^{*\dagger}$	$0.239^{*\ddagger}$	0.299^{*}	0.265^{*}	$0.113^{\dagger}$	$0.141^{\dagger}$
NTFIDF	0.205^{*}	0.213	$0.307^{*\dagger}$	$0.296^{*\ddagger}$	0.097	0.129
Language Modeling Approaches						
DIR	0.227	$0.252^{\dagger}$	0.312	0.299	0.090	0.134
BLM	0.208	0.233	$0.314^{\dagger}$	0.222	0.072	0.113
TSL	$0.228^{\dagger}$	0.252	0.312	$0.300^{\dagger}$	0.090	0.134
F3EXP	0.205	0.234	0.290	0.250	0.101^{*}	0.138
F3LOG	0.203	0.232	0.293	0.263	$0.109^{*\dagger}$	$0.138^{\dagger}$
DIR+	0.227	0.252	0.312	0.299	0.090	0.134
Divergence from Randomness Models						
PL2	0.228	0.252	0.325	$0.303^{\dagger}$	0.089	0.116
PL3	$0.228^{\dagger}$	0.241	0.290	0.269	$0.093^{\dagger}$	0.117
PL2+	0.214	$0.255^{*\ddagger}$	$0.328^{*\ddagger}$	0.301	0.089^{*}	$0.119^{*\dagger}$
Information-based Models						
LGD	$0.215^{\dagger}$	$0.251^{\dagger}$	$0.320^{\dagger}$	$0.300^{\dagger}$	0.086	$0.131^{\dagger}$
SPL	0.213	0.251	0.313	0.299	$0.093^{\dagger}$	0.130

38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15, pages 1147–1148, New York, NY, USA, 2015. ACM.

[3] T. G. Armstrong, A. Moffat, W. Webber, and J. Zobel. Has adhoc retrieval improved since 1994? In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 692–693, New York, NY, USA, 2009. ACM.

[4] T. G. Armstrong, A. Moffat, W. Webber, and J. Zobel. Improvements that don't add up: Ad-hoc retrieval results since 1998. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 601–610, New York, NY, USA, 2009. ACM.

[5] C. Buckley, J. Allan, and G. Salton. Automatic routing and ad-hoc retrieval using smart : Trec 2. In *Proceedings of the Second Text REtrieval Conference (TREC-2), pages 45–56. NIST Special Publication*, pages 45–56, 1994.

[6] S. Clinchant and E. Gaussier. Information-based models for ad hoc ir. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 234–241, New York, NY, USA, 2010. ACM.

[7] A. P. de Vries and T. Roelleke. Relevance information: A loss of entropy but a gain for idf? In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 282–289, New York, NY, USA, 2005. ACM.

[8] H. Fang, H. Wu, P. Yang, and C. Zhai. Virlab: A web-based virtual lab for learning and studying information retrieval models. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 1249–1250, New York, NY, USA, 2014. ACM.

[9] H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 480–487, New York, NY, USA, 2005. ACM.

[10] M. Franz and J. S. McCarley. Word document density and relevance scoring (poster session). In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 345–347, New York, NY, USA, 2000. ACM.

[11] T. Gollub, B. Stein, and S. Burrows. Ousting ivory tower research: Towards a web framework for

providing experiments as a service. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 1125–1126, New York, NY, USA, 2012. ACM.

[12] A. Hanbury and H. Müller. Automated component-level evaluation: Present and future. In *Proceedings of the 2010 International Conference on Multilingual and Multimodal Information Access Evaluation: Cross-language Evaluation Forum*, CLEF'10, pages 124–135, Berlin, Heidelberg, 2010. Springer-Verlag.

[13] B. He and I. Ounis. A study of the dirichlet priors for term frequency normalisation. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 465–471, New York, NY, USA, 2005. ACM.

[14] F. Hopfgartner, A. Hanbury, H. Müller, N. Kando, S. Mercer, J. Kalpathy-Cramer, M. Potthast, T. Gollub, A. Krithara, J. Lin, K. Balog, and I. Eggel. Report on the evaluation-as-a-service (eaas) expert workshop. *SIGIR Forum*, 49(1):57–65, June 2015.

[15] D. Lagun and E. Agichtein. Viewser: Enabling large-scale remote user studies of web search examination and interaction. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 365–374, New York, NY, USA, 2011. ACM.

[16] J. Lin, M. Crane, A. Trotman, J. Callan, I. Chattopadhyaya, J. Foley, G. Ingersoll, C. MacDonald, and S. Vigna. Toward reproducible baselines: The open-source ir reproducibility challenge. In N. Ferro, F. Crestani, M.-F. Moens, J. Mothe, F. Silvestri, G. M. D. Nunzio, C. Hauff, and G. Silvello, editors, *ECIR*, volume 9626 of *Lecture Notes in Computer Science*, pages 408–420. Springer, 2016.

[17] J. Lin and M. Efron. Evaluation as a service for information retrieval. *SIGIR Forum*, 47(2):8–14, Jan. 2013.

[18] J. Lin and M. Efron. Infrastructure support for evaluation as a service. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, pages 79–82, New York, NY, USA, 2014. ACM.

[19] Y. Lv and C. Zhai. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 7–16, New York, NY, USA, 2011. ACM.

[20] D. Metzler. Generalized inverse document frequency. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 399–408, New York, NY, USA, 2008. ACM.

[21] D. Metzler, V. Lavrenko, and W. B. Croft. Formal multiple-bernoulli models for language modeling. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 540–541, New York, NY, USA, 2004. ACM.

[22] H. Mühleisen, T. Samar, J. Lin, and A. de Vries. Old dogs are great at new tricks: Column stores for ir prototyping. In *Proceedings of the 37th International ACM SIGIR Conference on Research &#38; Development in Information Retrieval*, SIGIR '14, pages 863–866, New York, NY, USA, 2014. ACM.

[23] J. H. Paik. A novel tf-idf weighting scheme for effective ranking. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 343–352, New York, NY, USA, 2013. ACM.

[24] J. Rao, J. Lin, and M. Efron. Reproducible experiments on lexical and temporal feedback for tweet search. In A. Hanbury, G. Kazai, A. Rauber, and N. Fuhr, editors, *Advances in Information Retrieval*, volume 9022 of *Lecture Notes in Computer Science*, pages 755–767. Springer International Publishing, 2015.

[25] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. pages 109–126, 1996.

[26] S. E. Robertson, S. Walker, and M. M. Hancock-Beaulieu. Large test collection experiments on an operational, interactive system: Okapi at trec. In *Proceedings of the Second Conference on Text Retrieval Conference*, TREC-2, pages 345–360, Elmsford, NY, USA, 1995. Pergamon Press, Inc.

[27] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, pages 21–29, New York, NY, USA, 1996. ACM.

[28] A. Trotman, A. Puurula, and B. Burgess. Improvements to bm25 and language models examined. In *Proceedings of the 19th Australasian Document Computing Symposium*, ADCS '14, Melbourne, VIC, Australia, 2014. ACM.

[29] E. M. Voorhees and D. K. Harman. *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press, 2005.

[30] C. Zhai and J. Lafferty. Two-stage language models for information retrieval. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 49–56, New York, NY, USA, 2002. ACM.

[31] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, Apr. 2004.

[32] J. Zhu, J. Wang, I. J. Cox, and M. J. Taylor. Risky business: Modeling and exploiting uncertainty in information retrieval. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 99–106, New York, NY, USA, 2009. ACM.