# Pseudo Test Collections for Learning Web Search Ranking Functions

Nima Asadi[1], Donald Metzler[2], Tamer Elsayed[3], Jimmy Lin[1]

[1]University of Maryland, College Park
[2]Information Sciences Institute, University of Southern California
[3]King Abdullah University of Science and Technology (KAUST)

nima@cs.umd.edu, metzler@isi.edu, tamer.elsayedaly@kaust.edu.sa, jimmylin@umd.edu

## ABSTRACT

Test collections are the primary drivers of progress in information retrieval. They provide yardsticks for assessing the effectiveness of ranking functions in an automatic, rapid, and repeatable fashion and serve as training data for learning to rank models. However, manual construction of test collections tends to be slow, labor-intensive, and expensive. This paper examines the feasibility of constructing web search test collections in a completely unsupervised manner given only a large web corpus as input. Within our proposed framework, anchor text extracted from the web graph is treated as a pseudo query log from which pseudo queries are sampled. For each pseudo query, a set of relevant and non-relevant documents are selected using a variety of web-specific features, including spam and aggregated anchor text weights. The automatically mined queries and judgments form a pseudo test collection that can be used for training ranking functions. Experiments carried out on TREC web track data show that learning to rank models trained using pseudo test collections outperform an unsupervised ranking function and are statistically indistinguishable from a model trained using manual judgments, demonstrating the usefulness of our approach in extracting reasonable quality training data "for free".

**Categories and Subject Descriptors**: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

**General Terms**: Algorithms, Performance

**Keywords**: web, anchor text, hyperlinks, evaluation

## 1. INTRODUCTION

Reusable test collections play a central role in information retrieval (IR) research. A test collection consists of three components: a *corpus* of documents; a set of *queries* representing users' information needs (i.e., *topics*); and *relevance judgments*, which enumerate documents that are relevant (and not relevant) to a particular information need. These resources are critical to the development of ranking functions, one of the central problems in information retrieval research. Given a query and a corpus,[1] the task is to develop a ranking function that returns a ranked list of documents that maximizes the relevance of the retrieved documents with respect to the query. In this research framework, test collections serve two purposes: First, they provide a yardstick for assessing the effectiveness of ranking functions in an automatic, rapid, and repeatable fashion. Second, they provide training data for learning to rank models [19, 23, 33, 4, 30]. It would not be an exaggeration to say that test collections are the primary drivers of progress in IR today.

Academic researchers have access to only a small handful of test collections because they are very expensive to create. Traditionally, test collections are created as the byproduct of community-wide evaluations such as the NIST-organized Text REtrieval Conferences (TRECs). Using a process known as pooling [21, 37], NIST samples results from participating systems and coordinates a manual assessment process—this is a slow, labor-intensive, and expensive proposition. As a result, typical test collections contain perhaps a few dozen queries. Over time, yearly community-wide evaluations accumulate sufficient queries and relevance judgments to be useful for evaluation and learning to rank. However, if the underlying document corpus changes (for example, as when the field moved from newswire articles to web pages in the last decade), existing relevance judgments become mostly useless since they are corpus-specific. Thus, to some extent, the academic IR community suffers from the phenomenon of searching in the dark only under the lamp post, since that is where the test collections are.

In contrast, researchers in industry (i.e., at search engine companies) are able to circumvent these challenges primarily in two ways. First, such companies typically possess the financial resources to gather a large amount of human editorial judgments. Second, researchers at search engine companies have access to a variety of data resources, including query logs, click logs, and toolbar data. Such resources provide a rich source of implicit relevance judgments [24, 1, 35, 20]. Both of these avenues are mostly closed off to academic researchers.

This paper explores the feasibility of constructing web search test collections automatically given only a large web

*SIGIR'11,* July 24–28, 2011, Beijing, China.
Copyright 2011 ACM 978-1-4503-0757-4/11/07 ...$10.00.

[1] To reduce confusion in terminology (i.e., document collection vs. test collection), we use corpus to refer to the documents over which search is performed.

corpus. In particular, we propose novel approaches for extracting queries and relevance judgments using the web graph in an unsupervised manner. Since the queries and judgments extracted have not been vetted by humans, we call these *pseudo test collections*. If successful, automatic methods for distilling test collections would have several key benefits. From the academic perspective, such methods would provide the means for gathering a large amount of relevance information using minimal resources. The methods would also likely be useful within industrial research settings, providing a way for search engine companies to augment their human judgments and implicit behavioral-based judgments with a novel source of relevance information.

This paper has three primary contributions. First, we describe a general framework for constructing pseudo test collections. The framework includes components for sampling pseudo queries and distilling pseudo relevance judgments for the queries. As far as we know, this is the first attempt to develop a general-purpose methodology for automatically constructing test collections that can be used for evaluation and learning to rank. Second, we describe a specific instantiation of the framework for constructing web search pseudo test collections. The approach exploits the fact that anchor text serves as a strong implicit relevance signal. Various schemes that aggregate anchor text weights are introduced and used for sampling web search queries and generating relevance judgments. Finally, we evaluate the quality of a web search pseudo test collection in the context of learning to rank. We show that learning to rank models trained using the unsupervised pseudo test collection are more effective than a standard unsupervised model and are statistically indistinguishable from a model trained using manual judgments.

The remainder of this paper is laid out as follows. First, Section 2 describes related research. Section 3 outlines our proposed framework for generating pseudo test collections, and Section 4 describes a specific instantiation for web search. Section 5 provides details of our experimental evaluation. Finally, Section 6 concludes the paper and describes several possible directions for future work.

## 2. RELATED WORK

There are two steps involved in constructing pseudo test collections—sampling pseudo queries and inferring pseudo relevance judgments for the queries. A number of previous studies have explored work along these lines, described below. However, we are not aware of previous work that combines these two threads within a single framework for automatically constructing test collections.

**Sampling Pseudo Queries.** Previous research has investigated methods for extracting implicit queries for contextual advertising [3, 27, 39] and the automatic generation of titles and quick-links for web pages [12, 13]. The goal of both tasks is to extract short phrases that are relevant to a given web page. Such approaches extract important phrases from various sources, including high $tf.idf$ terms within a page, titles, anchor text, and query logs. Although these tasks are closely related to sampling pseudo queries, our work focuses on using implicit queries to automatically construct pseudo test collections, rather than solving advertising or user interface-related tasks. Of course, our work builds on these ideas.

Other related work has shown anchor text to be a reasonable surrogate to query logs for query reformulation [16]. We leverage this finding when constructing pseudo test collections for web search.

**Generating Pseudo Judgments.** The other step of the pseudo test collection construction process identifies a set of relevant (and non-relevant) documents for the sampled pseudo queries. A great deal of effort within the information retrieval community has been devoted to solving this problem. A key factor that differentiates our proposed approach from previous work is that all of our analysis and computation is performed *offline* using *global* information. Learning to rank systems often use a large number of features that are computed at runtime, typically based on evidence from a single document or a small set of top ranked documents. Our framework provides the ability to extract considerably more complex features that would likely be too costly (either in terms of space or in terms of computation) to use at runtime within a practical search engine. In this way, our approach can be thought of as computing relevance scores for a large set of query-document pairs offline using a wide variety of (potentially expensive) relevance signals. Similar approaches have been explored in the past, but almost exclusively in the context of leveraging implicit behavioral signals as pseudo judgments [23, 24, 17]. Our work instead focuses on using implicit corpus-based relevance signals to distill pseudo judgments.

One related line of research deals with methods for inferring the relevance of unjudged documents when computing retrieval metrics [36, 5, 7, 10, 9]. These approaches take as input a ranked list of documents retrieved in response to a query. Some of the documents have been judged, while the rest have not. The goal is then to estimate the relevance of the unjudged documents. This research has been shown to be useful for obtaining better estimates of retrieval system effectiveness in the presence of incomplete judgments. However, it should be clear that this task is easier than pseudo test collection construction, because it is assumed that documents, one or more queries, and some judgments are provided as evidence, whereas our framework relies only on having a corpus.

Finally, it is important to note that our work differs from semi-supervised learning to rank approaches [18, 28, 29]. Although such approaches are designed to learn highly effective ranking models, it is not straightforward to adapt them to constructing test collections. Our goal is to build test collections that can be used for a variety of tasks, and to do so in a completely unsupervised manner. In contrast, semi-supervised approaches assume that at least *some* labeled data are available.

## 3. PSEUDO TEST COLLECTIONS

As previously mentioned, information retrieval test collections consist of a document corpus, queries, and relevance judgments. It is often the case that researchers, when developing search technologies for a new task or domain, do not have access to all three components: in most cases, only a document corpus is available. Unfortunately, a corpus in isolation has limited utility; without queries and relevance judgments it would be very difficult to learn effective ranking functions (e.g., by learning to rank) or to evaluate the quality of a retrieval system built on the corpus. Even for tasks for which test collections already exist, there is never enough queries nor relevance judgments, since as with many machine learning tasks, algorithmic effectiveness increases with the amount of available training data.

Obtaining queries and relevance judgments is a labor-intensive and costly task. Commercial search engines are able to address this problem with data, in the form of query and click logs, and money, which can be used to hire large teams of humans to manually assess the relevance of documents. However, in resource-constrained academic settings, these options are not available, making it difficult to undertake research on new document corpora, new tasks, or work with machine learning algorithms that require lots of training data. This has led researchers to pursue low cost strategies for constructing *manual test collections.* Two emerging evaluation paradigms are minimal test collections [8, 7, 6] and crowdsourcing [2]. Both of these strategies are useful for low-cost *one-time* evaluations. However, they suffer from issues related to reusability [11, 9].

To overcome these issues, we propose to automatically construct test collections with minimal human effort. Given nothing but a document corpus $\mathcal{D}$, our goal is to automatically construct a high quality, reusable test collection that can be used to evaluate and train ranking functions over $\mathcal{D}$. Since the mined queries and relevance judgments are automatically *inferred*, we refer to the resulting test collections as pseudo test collections.

It is important to note that the goal of pseudo test collections is not to produce manual-quality test collections, but rather to minimize costs by automatically distilling *surrogate* test collections. It is assumed (and expected) that pseudo test collections will be noisy (i.e., have incorrect relevance assessments). However, this is not overly problematic, since recently-developed evaluation methodologies and learning to rank models are robust enough to handle certain amounts of missing or incorrect data. Indeed, as we will show in our experiments, even simple learning to rank approaches can be used with (noisy) pseudo test collections to learn effective ranking functions.

We propose constructing pseudo test collections by mimicking the process used to build manual test collections, which typically begins with a corpus. After a corpus has been obtained, a set of queries is chosen. The queries are either sampled from query logs or manually generated. Each query is then issued to one or more retrieval systems, which returns candidate documents that are then judged, either via pooling [21, 37], the minimal test collection paradigm [8, 7, 6], or crowdsourcing [2].

Our general pseudo test collection framework follows a similar process. The three primary steps are as follows:

1. **Corpus acquisition.** The only input to our proposed framework is a corpus of documents (e.g., a large web crawl, an archive of news articles, a collection of books, etc.). Automatic corpus acquisition and corpus expansion are beyond the scope of this paper. We assume that a corpus, generated in some way, is available. Since the corpus is the only input to the framework, it should be chosen with some care. Corpora that contain a large amount of potentially noisy implicit relevance information (e.g., anchor text, click information, user ratings, tags, metadata, etc.) are the most amenable to pseudo test collection construction.
2. **Pseudo query generation**. Given the corpus, we will then automatically generate a set of pseudo queries in a completely unsupervised manner. It is important that the pseudo queries are diverse, in terms of difficulty, topical coverage, user intent, etc. It is also important that the queries be "well-formed" and represent a realistic sample of information needs that can be addressed by documents in the corpus.
3. **Pseudo judgment generation**. For every sampled pseudo query, the final task is to assign automatically-generated relevance labels (e.g., "relevant" and "non-relevant") to some set of documents found in the corpus. Unlike the pooling method, this set of documents does not necessarily need to be the output of a retrieval system. Instead, it can be *any* subset of documents that can be accurately and reliably labeled in an unsupervised manner. It is also desirable, but not necessary, for each label to have a confidence score assigned to it. Such scores reflect the uncertainty in the automatically-generated label and may be informative for the purpose of evaluation or learning to rank.

Therefore, to instantiate this framework, methods for generating pseudo queries and pseudo judgments must be defined. These methods will vary depending on a number of factors, including the corpus, the search task, and the sources of implicit relevance signals that are available. Naturally, generating pseudo test collections for certain tasks will be substantially easier than others. In the following section, we propose a methodology for constructing pseudo test collections for web search. This serves as the first instantiation of our proposed framework to illustrate its utility.

# 4. AUTOMATIC CONSTRUCTION OF WEB SEARCH TEST COLLECTIONS

This section describes a specific instantiation of our general pseudo test collection framework whose output can be used to train learning to rank models for web search. Web search is a particularly interesting domain for our framework since the web graph encodes a great deal of implicit relevance information, in the form of links and anchor text. This is evidenced by the fact that anchor text and link analysis features (e.g., PageRank [34], HITS [25], and SALSA [26]) are known to be important for web search. As we will show, the web graph can be leveraged to extract high quality pseudo queries and pseudo judgments.

## 4.1 Implicit Relevance Signals

Starting from only a collection of web pages, anchor text provides a potentially high quality description of target documents. This implicit signal serves as a strong source from which relatively high quality pseudo queries can be sampled. In addition, given a line of anchor text, its target documents (i.e., the set of documents the anchor text points to) are reasonable candidates for relevant pseudo judgments. However, there is always a level of noise in this relevance signal which, if sampled naïvely, could yield poor results. Examples of poor anchor text-target document pairs include common web terminology (e.g. "privacy policy" and "homepage"), ambiguous anchor text, links from spam pages, and links that serve as citations for a piece of information (e.g. "born in X" which points to a biography as a reference).

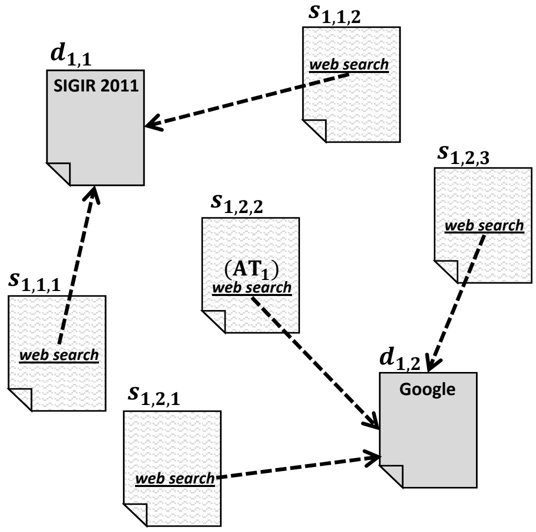

**Figure 1: A sample structure of anchor text. In this figure, "web search" is a line of anchor text ($\mathbf{AT}_1$) that points to two target documents $d_{1,1}$ and $d_{1,2}$. For each target document, anchor text originates from a number of source pages. For instance, $s_{1,1,1}$ and $s_{1,1,2}$ point to document $d_{1,1}$ with anchor text "web search".**

Therefore, additional factors must be considered when using anchor text for constructing pseudo test collections.

In order to reduce the effect of noise and to sample higher-quality pseudo queries and pseudo judgments, a combination of strategies can be used, for example: excluding intra-domain links (also called internal anchor text), measuring the quality of each unique line of anchor text as well as its target documents, and designing effective sampling techniques. We adopt all three strategies. In this paper, we only consider external anchor text (inter-domain links) as our source for pseudo queries. In addition, we assign weights to every anchor text-target document (henceforth, anchor-document) pair based on a number of factors. These weights can be thought of as confidence scores for relevance pairs. Finally, a score is computed for each unique line of anchor text to capture its quality.

In what follows, we introduce a number of different weighting schemes to compute weights for anchor-document pairs. An aggregation function is then defined to estimate the quality of lines of anchor text.

### 4.1.1 Anchor-Document Weighting Schemes

Figure 1 illustrates the general structure of anchor text within the web graph. In general, anchor text $\text{AT}_k$ points to $d_{k,i}$ documents (for $1 \leq i \leq n_k$). For each of the target documents $d_{k,i}$, the anchor text originates from a set of source pages $s_{k,i,j}$ for $1 \leq j \leq m_{k,i}$. From this structure, we extract a great deal of information about the anchor text, such as the set of all target documents, the set of sources that point at those documents, the URI of each of the documents, etc. Given this information, our goal is to design a suitable weighting scheme $w(\text{Anchor Text}, \text{Target Document}) \rightarrow \mathbb{R}$, that takes an anchor-document pair (e.g., <$\text{AT}_1$, $d_{1,2}$> in Figure 1) as input and returns a real number as a confidence score. This score can later be used for extracting high quality pseudo queries and pseudo relevance judgments.

We propose the following four weighting schemes; some are novel, others have been used by researchers in the past. In the following, $0 \leq \text{PR}(d) \leq 1$ is the PageRank score of document $d$, while $1 \leq \textsc{Ham}(d) \leq 100$ is the "ham" score (i.e., $100 - \textsc{Spam}(d)$, where $0 \leq \textsc{Spam}(d) \leq 99$).

**Spam.** Spam pages by definition contain low quality content. As a result, a line of anchor text that describes a spam page should not be trusted. This weighting scheme measures the confidence of an anchor-document pair solely on the basis of the ham score of the target document:

$$w(\text{AT}_k, d_{k,i}) = \log\left[\textsc{Ham}(d_{k,i})\right] \tag{1}$$

**SrcSpam.** While the level of "spamminess" of a target document is important, the quality of sources from which a line of anchor text originates tells us how trustworthy that anchor text is as a descriptive phrase for the target document. This weighting scheme follows the intuition that if an anchor text originates from a reliable set of sources, then that anchor text is likely to accurately describe the target document. Since anchor text can originate from multiple source pages, we need to define an aggregation function. In this weighting scheme, we aggregate the reliability of the sources using a harmonic mean:

$$w(\text{AT}_k, d_{k,i}) = \log\left[\frac{\textsc{Ham}(d_{k,i}) \cdot m_{k,i}}{\sum_{j=1}^{m_{k,i}} \left[\textsc{Ham}(s_{k,i,j})\right]^{-1}}\right] \tag{2}$$

**PageRank.** PageRank scores indicate the importance of a particular page based on global link structure: A page is important if other important pages point to it. This weighting scheme sets the quality score of a document based on its PageRank score and ham score:

$$w(\text{AT}_k, d_{k,i}) = \log\left[\text{PR}(d_{k,i}) \times \textsc{Ham}(d_{k,i})\right] \tag{3}$$

**Anchor.** The validity of anchor text can be measured based on the number of unique domains it originates from and the number of times it is used to point to a particular target document. This is the basis for the following weighting scheme that is a variant of the weighted scheme originally described by Metzler et al. [32]:

$$w(\text{AT}_k, d_{k,i}) = \textsc{Ham}(d_{k,i}) \sum_{s \in S(d_{k,i})} \frac{\delta(\text{AT}_k, d_{k,i}, s)}{|\textsc{Anchors}(d_{k,i}, s)|} \tag{4}$$

where $S(d_{k,i})$ denotes sites that link to $d_{k,i}$, $\delta(\text{AT}_k, d_{k,i}, s)$ is 1 if and only if anchor text $\text{AT}_k$ links to $d_{k,i}$ from some page within site $s$, and $|\textsc{Anchors}(d_{k,i}, s)|$ is the total number of unique anchors originating from site $s$ that links to $d_{k,i}$.

### 4.1.2 Anchor Text Quality Score

In order to measure how well a line of anchor text describes its target document, we assign weights to each anchor-document pair according to different weighting schemes as explained in the previous section. Next, we must define a function $w(\text{Anchor Text}) \rightarrow \mathbb{R}$ that computes a quality score for a line of anchor text.

A simple approach is to make use of anchor-document weights in order to estimate the quality of a line of anchor text. However, a line of anchor text can have multiple target documents and therefore can be present in multiple anchor-document pairs. Hence, the quality of a line of anchor text can be estimated by an aggregation function over the weights of individual anchor-document pairs for that particular anchor text.

In this paper, we use a simple arithmetic mean to combine the weights of individual anchor-document pairs as follows:

$$w(\mathrm{AT}_k) = \frac{1}{n_k}\sum_{i=1}^{n_k} w(\mathrm{AT}_k, d_{k,i})$$

where $w(\mathrm{AT}_k, d_{k,i})$ is the weight associated with anchor text $\mathrm{AT}_k$ and its target document $d_{k,i}$, and $n_k$ is the number of target documents $\mathrm{AT}_k$ points to. This weight can be interpreted as the quality of a unique line of anchor text.

## 4.2 Pseudo Queries

As we described previously, pseudo test collections consist of pseudo queries and associated pseudo relevance judgments. We begin by describing how anchor text can be used for constructing a high quality set of pseudo queries. Note that a naïve approach to extracting pseudo queries would be to rank all anchor text according to their weights and take the top $Q$ as pseudo queries. However, there are other factors that need to be considered.

Based on our observations, the number of target documents a unique line of anchor text points to, can, to some extent, verify whether an anchor text is of high quality or not. Anchor text that have very few target documents often contain misspellings or otherwise are overly-specific topics. On the other hand, anchor text that have a large number of target documents are often broader topics and, in essence, very ambiguous. Examples of this are phrases such as "privacy policy" and "homepage". We would like to avoid both extremes and thus favor anchor text that have an average number of target documents.

To accomplish this goal, we partition lines of anchor text based on the number of target documents they point to: a line of anchor text with $n$ target documents falls into partition $P_n$. Since we want to favor lines of anchor text that have an average number of target documents, we are interested in partitions with an index that is neither too small nor too large. We can define a probability distribution over indices $n$ to satisfy this criterion—in our case, we selected a normal distribution $N(\mu, \sigma^2)$. To sample pseudo queries, we first sample a partition according to the normal distribution, and then extract the highest weighted line of anchor text, weighted according to $w(\mathrm{AT}_k)$ from the partition.

## 4.3 Pseudo Judgments

The final step of our proposed pseudo test collection framework is the automatic generation of relevance judgments, which we refer to as pseudo judgments.

While there has been a great deal of effort devoted to accurately estimating the probability of relevance, many of them, especially learning to rank approaches, are only effective when trained using a large set of manual judgments. Therefore, many of these approaches are not directly applicable to our problem. Instead, we rely on unsupervised methods for estimating the relevance of query-document pairs.

**Positive Judgments.** As we mentioned earlier, anchor text provides an implicit relevance signal which can be used to extract positive judgments. Since pseudo queries are simply lines of anchor text, each pseudo query has a set of documents that it points to. These target documents serve as a reasonable set of potentially relevant documents. Assuming that the relevance of each document can be reliably estimated with respect to the anchor text, we can assert that the documents with the highest scores are relevant. We rely on the weighting schemes introduced earlier in this section to estimate the relevance of a document with respect to a pseudo query. For each pseudo query, we sort its target documents according to their anchor-document weight and select the top $d_p$ documents as positive judgments.

**Negative Judgments.** For a pseudo test collection to be complete, a set of negative judgments for each pseudo query is also essential. Negative judgments must not only be non-relevant with respect to the query, but also must contain a diverse set of documents. Despite the utility of anchor text for extracting pseudo queries and positive judgments, they do not appear to be a good source for negative judgments based on our experience.

Classic approaches to information retrieval such as BM25 or language modeling, on the other hand, define a content-based scoring function that measures the relevance of a given document with respect to a query. Documents that appear deep in the ranked list are likely to be "barely relevant" or "non-relevant". As a result, we retrieve a ranked list of $R$ documents for each pseudo query using a simple ranking function (e.g., language modeling) and sample $d_n$ documents from the bottom of that list to serve as negative judgments.

# 5. EXPERIMENTAL EVALUATION

This section describes the details of our experimental evaluation comparing the quality of learning to rank models trained using web search pseudo test collections and standard manually-constructed TREC test collections.

## 5.1 Data and Methodology

We performed our experiments on ClueWeb09, a best-first web crawl completed by Carnegie Mellon University in early 2009. The collection contains one billion pages in ten languages totaling around 25 TB. Of those, about 500 million pages are in English, divided into ten roughly equally-sized segments. Our experiments specifically focused on the first English segment, which contains 50 million documents (totaling 1.53 TB uncompressed, 247 GB compressed). The first segment of the English portion of the ClueWeb09 dataset contains approximately 7.5 million unique lines of external (i.e., inter-domain) anchor text, on which we applied our proposed methods to generate pseudo queries and pseudo judgments. For evaluation purposes, we used the set of 50 queries and their corresponding judgments from the TREC 2009 web track.

The criterion for determining if our proposed approach is successful or not is whether a learning to rank model trained using an automatically constructed pseudo web test collection can achieve higher effectiveness than alternative models. If this is the case, then we have shown that our pseudo test collections can be used to effectively extract implicit relevance signals without human judgments.

We compare our proposed approach against two models. The first is BM25, which is one of the most effective and widely used unsupervised retrieval models available. Most unsupervised models consist of a (semi-)heuristic combination of basic statistics, such as term frequency, inverse document frequency, and document length. We do not know of any existing retrieval models that can learn how to combine

| | Query | Page Title Keywords | |
|---|---|---|---|
| Anchor | tax deductible | Gifts and Car Expenses | R |
| | | Tax Deduct.-Wikipedia | R |
| | | Hospitals...of charity | NR |
| | | Mortgage Tax Calculator | NR |
| | google labs | Google Labs | R |
| | | Google Labs-Wikipedia | R |
| | | Info. Management | NR |
| | | Blogger - Wikipedia | NR |
| | ny public library | NYPL Print Collection | R |
| | | NYPL Photography Collection | R |
| | | New York Law | NR |
| | | Libraries and Schools | NR |
| | avoid plagiarism | Academic Integrity | R |
| | | Citing Sources and Avoiding Plagiarism | R |
| | | Research and Teaching Tools | NR |
| | | How to use Google Alerts | NR |
| | shipping tips | FedEx - Service Guide | R |
| | | Sportube Shipping Tips | R |
| | | eBay Tools | NR |
| | | Merchandise Coupon Codes | NR |
| | knee pain | Knee Pain - MayoClinic.com | R |
| | | Knee Pain Symptoms | R |
| | | Joint Replacement | NR |
| | | Discussing Knee Pain | NR |
| PageRank | download realplayer | RealPlayer ... audio | R |
| | | RealPlayer | R |
| | | Sports games Download | NR |
| | | Apple iPod... | NR |
| | yahoo privacy policy | Yahoo! Privacy Policy | R |
| | | Yahoo! Store Privacy Po. | R |
| | | download yahoo msngr | NR |
| | | AltaVista-Privacy Policy | NR |
| | united states copyright office | US Copyright Office | R |
| | | Copyright law of the US | R |
| | | Justia Regulation Trackers | NR |
| | | Sources and Notes | NR |
| | bbc weather | BBC Homepage | R |
| | | BBC Weather Center | R |
| | | Pippa Greenwood | NR |
| | | BBC News | NR |
| | free hit counters | Rapid Counter | R |
| | | Easy Counter | R |
| | | Web Site Design | NR |
| | | Jewelry and Watches | NR |
| | iphone sdk | Apple iPhone | R |
| | | iPhone Developer Connection | R |
| | | Google IO - Android | NR |
| | | Gps - iPhone Buzz | NR |

**Table 1: Examples of pseudo queries and pseudo judgments extracted using the Anchor and PageRank weighting schemes. In the last column, R indicates "relevant" and NR indicates "not relevant".**

an arbitrary set of features in a completely unsupervised manner. The second approach that we compare against is an upper bound, or "cheating" model, that is trained using the manual judgments from the TREC 2009 web track and then tested on the *same training set* (hence the "cheating" name). This is meant to approximate the effectiveness of a highly effective supervised learning to rank model trained over the same set of features using the same learning algorithm. For evaluation, all models are tested on TREC 2009 web track data.

Our pseudo query sampling strategy requires a mean and a variance for the underlying normal distribution $N(\mu, \sigma^2)$. Based on our observation, we found that anchor text pointing to $\leq 5$ target documents are generally of low quality. These make up a large portion of the extracted anchor text from our corpus, totaling about 7 million unique lines. To ensure quality, we eliminate those anchor text that have 5 or fewer target documents, thereby reducing the size of potential pseudo queries to only 6 percent of the original set. We set the mean of the normal distribution, $\mu$, based upon statistics gathered from the remaining collection. The filtered collection of anchor text has a median of 10 target documents, which we used for $\mu$ in the sampler. The value for $\sigma^2$ was set (arbitrarily) to 5. We used the Waterloo spam scores [15], computed for every document in the English portion of the ClueWeb09 dataset and freely available.[2]

Unless otherwise specified, we set the total number of sampled pseudo queries ($Q$) to 400, and the average number of pseudo positive ($d_p$) and negative judgments ($d_n$) for each query to 10 and 20, respectively, keeping the ratio of positive to negative judgments at 0.5. Pseudo negative judgments are sampled from the bottom of a ranked list of a thousand retrieved documents ($R$) using the language modeling query likelihood scoring function. To evaluate the effectiveness of our models, we report NDCG@20 [22] and ERR@20 [14], which are commonly used to evaluate the effectiveness of web search tasks. A one-side paired $t$-test (with $p = 0.05$) is used to determine statistical significance.

## 5.2 Illustrative Examples

Table 1 shows selected examples of pseudo queries and their corresponding pseudo judgments extracted using the Anchor and PageRank weighting schemes.

The PageRank weighting scheme favors anchor text that point to popular pages. Popular pages with high PageRank values are likely to be website entry pages. Consequently, associated anchor text are navigational in nature, which yields many navigational pseudo queries. As shown in the examples, queries such as "download realplayer" and "yahoo privacy policy" as well as other navigational queries like "wiki help", "yahoo myweb", and "metawiki" are extracted with the PageRank weighting scheme.

On the other hand, pseudo queries extracted using the Anchor weighting scheme consist of navigational as well as informational phrases, as illustrated in the examples. Other examples include "weather maps", "landscapes scenery", "free hit counter code", "national center health statistics nchs", and "passport services".

Of course, pseudo queries and pseudo judgments are noisy. For instance, a query such as "help forum" that was extracted using the Anchor weighting scheme seems like a poor query due to its ambiguity. Examining the results, we observe many judgments that humans are not likely to consider relevant. Despite the presence of noise, experimental evidence provided later illustrates that our pseudo test collections are nevertheless *useful*.

## 5.3 Learning to Rank Model

We make use of a relatively straightforward learning to rank model in our experiments. Recall that we need to learn two models—one from an automatically constructed pseudo test collection and another from manual judgments. To support a fair comparison, both models use the same features and the same learning algorithm.

[2] http://plg.uwaterloo.ca/~gvcormac/clueweb09spam/

| Weighting Scheme | NDCG@20 | ERR@20 |
|---|---|---|
| BM25 | 0.291 | 0.135 |
| Cheating | 0.292 | 0.141 |
| Anchor | 0.298 | 0.149 |
| SrcSpam | 0.298 | 0.149 |
| Spam | 0.298 | 0.146 |
| PageRank | 0.110 | 0.062 |

**Table 2: Evaluation of different ranking models on TREC 2009 web track data, comparing BM25, a "cheating" greedy feature selection model, and the same greedy feature selection model trained on pseudo test collections generated using different weighting schemes.**

We learn a simple linear ranking function using a standard suite of features consisting of basic information retrieval scores (e.g., language modeling and BM25 scores), term proximity features (exact phrases, ordered windows, unordered windows, etc.), and query-independent features (e.g., PageRank). There are a total of 45 features. The parameters of the linear model are learned using a greedy feature selection approach [31]. The model is iteratively constructed by adding features to the model, one at a time, according to a greedy selection criterion. During each iteration, the feature that provides the biggest gain in effectiveness (as measured by ERR@20 [14], which is also our primary evaluation metric) after being added to the existing model is selected. This yields a sequence of one-dimensional optimizations that can easily be solved using line search techniques. The algorithm stops when the difference in ERR between successive iterations drops below a given tolerance ($10^{-4}$). This training procedure is simple, fast, and yields a model with minimal correlation/redundancy between features. We also explored the use of AdaRank [38], an effective learning to rank approach, but found the results to be consistently worse than the simple greedy feature selection algorithm with these features for this task.

## 5.4 Results

This section describes the results of our experimental evaluation. We first describe our basic findings and then provide a more in-depth analysis.

### *5.4.1 Basic Results*

We begin by comparing the effectiveness of learning to rank models trained using pseudo test collections against BM25 and the cheating model. Table 2 shows the effectiveness for the different weighting schemes proposed in Section 4.1.1. All models are tested with TREC 2009 web track data. It is important to reiterate that the models trained with pseudo test collections do not require *any* manual intervention and are completely unsupervised.

As the results suggest, the PageRank weighting scheme leads to a less effective ranking model. This finding is not unexpected since the PageRank weighting scheme estimates the quality of anchor text based solely on the quality of its target documents, regardless of the quality of its sources. Furthermore, this weighting scheme suffers from the issues already discussed in Section 5.2. The extracted pseudo queries and pseudo judgments are low in quality, and hence it is not possible to learn an effective ranker.

On the other hand, the Anchor weighting scheme makes use of the sources from which anchor text originates. It takes into consideration the number of unique domains that use the same line of anchor text to point to a target document. In addition, it factors in the number of unique lines of anchor text that points to a target document, which approximates the breadth and ambiguity of topics within the document. These counts nicely capture the quality of a given anchor-document pair. As a result, the pseudo test collection generated using this weighting scheme contains higher quality pseudo queries and pseudo judgments, and therefore yields a more effective ranking model.

Results show that a learning to rank model trained using a pseudo test collection generated using the Anchor weighting scheme is more effective than baseline BM25, both in terms of NDCG@20 and ERR@20. Interestingly, all weighting schemes except for PageRank outperform the cheating experiment according to both metrics. Although the differences are not statistically significant, this suggests that our pseudo test collections, although noisy, provide valuable training data for learning to rank, compared to the "moderate" amount of TREC manual training data. We speculate that the cheating model is only slightly better than BM25 in NDCG due to the fact that all models were optimized for ERR@20. This may simply be a case of so-called metric divergence. For this reason, the ERR@20 numbers provide a more reliable comparison.

Finally, it is important to note that our learning to rank model does not include any spam features. If we were to include spam features in the "cheating" condition, it achieves an ERR@20 of 0.161. However, if the spam feature were also included in the models trained from the pseudo test collections, the results are no more effective than the results reported in Table 2. This suggests that the strategy we use for sampling non-relevant documents does not contain enough spam documents to allow the models to learn the importance of the spam feature.

This raises an important issue related to our pseudo test collection framework. The strategies used for generating positive and negative judgments run the risk of significantly biasing results in unexpected ways. To minimize this risk, it is beneficial to select a *diverse* set of relevant and non-relevant documents based on a wide variety of features and selection strategies. This is an important issue that should be better studied in the future. Nevertheless, the conclusion appears clear: learning to rank models trained in a completely unsupervised fashion outperform BM25 and are just as good as a model trained with manual judgments. In other words, we are able to exploit implicit relevance signals to improve ranking effectiveness in web search, and this improvement comes basically "for free".

### *5.4.2 Analysis of Learned Models*

Analysis of the ranking models themselves shed some insight on feature selection. Table 3 shows the model learned using the pseudo test collection built with Anchor weighting, while Table 4 shows the cheating model learned using manual judgments.

The models are markedly different. For example, in the cheating model, bm25-term, the BM25 score on query unigrams, is assigned a high weight, while the same feature is assigned a low weight in the other model. This is likely the result of biases in both the pseudo test collection and the manual test collection. The pseudo test collection appears to be biased towards *phrase* features, which may stem from

| Feature | Weight |
|---|---|
| lm-phrase | 1.553 |
| bm25-term | 0.014 |
| bm25-proximity | -0.567 |

**Table 3: Ranking model trained on a pseudo test collection extracted using the Anchor weighting scheme.**

| Feature | Weight |
|---|---|
| lm-term | -0.0309 |
| bm25-term | 0.8911 |
| bm25-phrase | 0.1398 |

**Table 4: Ranking model trained using TREC 2009 web track data.**

the fact that the page a piece of anchor text points to is likely to contain the anchor text itself as a phrase. On the other hand, TREC judgments are collected via pooling, and it is known that such pools are biased towards BM25, since many TREC participants utilize that ranking function. Therefore, regardless of the source of judgments, it is difficult to avoid inherent biases.

We see that both models include the same feature types (i.e., term, phrase, and proximity), but their relative weights are different. It is interesting that the proximity score in the model trained using the pseudo test collection has a large negative weight. It is likely that the phrase feature is too large and the proximity feature is simply offsetting it.

### *5.4.3 Pseudo Collection Size*

The number of pseudo queries and pseudo judgments together determine the size of a pseudo collection. Larger collections provide more training data. However, deeper sampling will draw in pseudo queries and pseudo judgments with lower quality scores, yielding nosier data. The tradeoff between these two factors needs to be studied.

In previous experiments, the number of sampled pseudo queries and positive/negative pseudo judgments were kept constant. Here, we present results from altering these parameters. However, to avoid an exhaustive search over all possible parameter and weighting scheme combinations, we focused on the Anchor weighting scheme, based on the results in Table 2. To further simplify the presentation, we only change the average number of pseudo negative judgments ($d_n$). This serves two purposes: first, by altering the number of negative judgments we are changing the size of the resulting pseudo collection. Second, this alters the ratio of positive to negative judgments.

Table 5 shows the effect of these changes for the Anchor weighting scheme for a small subspace of parameters $Q$ and $d_n$. The table suggests that extracting more pseudo queries improves the resulting models slightly. In addition, lowering the ratio of positive to negative judgments (i.e., increasing $d_n$) adds more noise to the training data and results in slightly less effective models. The last row and last column in Table 5 show averages over the respective conditions.

## 6. CONCLUSIONS AND FUTURE WORK

Motivated by the fact that although test collections are essential for information retrieval research, they remain labor-intensive and expensive to manually construct, this paper proposes an unsupervised framework for constructing high quality test collections given nothing but a corpus. The resulting pseudo test collections, which consist of a set of pseudo queries and pseudo relevance judgments, can be used to evaluate and train learning to rank models. We describe an instantiation of the proposed framework for web search that leverages anchor text as a source of implicit relevance signals. Our pseudo test collections are used to train simple learning to rank models. Experiments on TREC 2009 data illustrate that the resulting models outperform BM25 and are statistically indistinguishable from equivalent models trained using manual judgments. This illustrates the utility of our approach in being able to extract reasonable training data "for free".

| $Q$ / $d_n$ | 5 | 10 | 20 | 30 | 40 | Avg. |
|---|---|---|---|---|---|---|
| 200 | 0.112 | 0.111 | 0.111 | 0.113 | 0.111 | 0.112 |
| 400 | 0.150 | 0.148 | 0.149 | 0.146 | 0.147 | 0.148 |
| 800 | 0.150 | 0.152 | 0.152 | 0.146 | 0.147 | 0.149 |
| 1600 | 0.152 | 0.152 | 0.147 | 0.147 | 0.147 | 0.149 |
| Avg. | 0.141 | 0.141 | 0.139 | 0.138 | 0.138 | |

**Table 5: Effects of varying the number of sampled queries and the number of negative judgments (Anchor weighting scheme, average number of positive judgments set to 10).**

There are several possible directions for future work. First, we would like to develop a better understanding of the weighting schemes within the current instantiation of the framework. Second, we would like to explore alternative methods for extracting relevance judgments, particularly negative judgments. In addition, we would like to study other sources of implicit relevance signals such as page titles and terms with high term frequencies. Finally, it would be interesting to use our pseudo test collections for evaluation purposes, comparing with alternatives such as pooling, minimal test collections, or crowdsourcing. We are particularly interested in the reusability of pseudo test collections and their ability to discriminate the effectiveness of ranking algorithms. Overall, we believe that this work opens up many future directions in information retrieval research.

## 7. ACKNOWLEDGMENTS

This work has been supported in part by NSF awards IIS-0836560, IIS-0916043, and CCF-1018625. Any opinions, findings, conclusions, or recommendations expressed are the authors' and do not necessarily reflect those of the sponsors. The first author's deepest gratitude goes to Katherine, for her invaluable encouragement and wholehearted support. The last author is grateful to Esther and Kiri for their loving support and dedicates this work to Joshua and Jacob.

## 8. REFERENCES

[1] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. *SIGIR*, pp. 3–10, Seattle, Washington, 2006.

[2] O. Alonso, D. E. Rose, and B. Stewart. Crowdsourcing for relevance evaluation. *SIGIR Forum*, 42(2):9–15, 2008.

[3] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel. A semantic approach to contextual advertising. *SIGIR*, pp. 559–566, Amsterdam, The Netherlands, 2007.

[4] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. *ICML*, pp. 89–96, Bonn, Germany, 2005.

[5] S. Büttcher, C. L. A. Clarke, P. C. K. Yeung, and I. Soboroff. Reliable information retrieval evaluation with incomplete and biased judgements. *SIGIR*, pp. 63–70, Amsterdam, The Netherlands, 2007.

[6] B. Carterette. *Low-Cost and Robust Evaluation of Information Retrieval Systems*. PhD thesis, University of Massachuetts, Amherst, Massachusetts, 2008.

[7] B. Carterette and J. Allan. Semiautomatic evaluation of retrieval systems using document similarities. *CIKM*, pp. 873–876, Lisbon, Portugal, 2007.

[8] B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. *SIGIR*, pp. 268–275, Seattle, Washington, 2006.

[9] B. Carterette, E. Gabrilovich, V. Josifovski, and D. Metzler. Measuring the reusability of test collections. *WSDM*, pp. 231–240, New York, 2010.

[10] B. Carterette and R. Jones. Evaluating search engines by modeling the relationship between relevance and clicks. *NIPS*, pp. 217–224, Vancouver, British Columbia, Canada, 2008.

[11] B. Carterette, E. Kanoulas, V. Pavlu, and H. Fang. Reusable test collections through experimental design. *SIGIR*, pp. 547–554, Geneva, Switzerland, 2010.

[12] D. Chakrabarti, R. Kumar, and K. Punera. Generating succinct titles for web URLs. *KDD*, pp. 79–87, Las Vegas, Nevada, 2008.

[13] D. Chakrabarti, R. Kumar, and K. Punera. Quicklink selection for navigational query results. *WWW*, pp. 391–400, Madrid, Spain, 2009.

[14] O. Chapelle, D. Metzler, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. *CIKM*, pp. 621–630, Hong Kong, China, 2009.

[15] G. V. Cormack, M. D. Smucker, and C. L. A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *CoRR*, abs/1004.5168, 2010.

[16] V. Dang and W. B. Croft. Query reformulation using anchor text. *WSDM*, pp. 41–50, New York, 2010.

[17] F. Diaz, D. Metzler, and S. Amer-Yahia. Relevance and ranking in online dating systems. *SIGIR*, SIGIR, pp. 66–73, Geneva, Switzerland, 2010.

[18] K. Duh and K. Kirchhoff. Learning to rank with partially-labeled data. *SIGIR*, pp. 251–258, Singapore, 2008.

[19] F. C. Gey. Inferring probability of relevance using the method of logistic regression. *SIGIR*, pp. 222–231, Dublin, Ireland, 1994.

[20] Q. Guo and E. Agichtein. Exploring mouse movements for inferring query intent. *SIGIR*, pp. 707–708, Singapore, 2008.

[21] D. K. Harman. The TREC test collections. In E. M. Voorhees and D. K. Harman, editors, *TREC: Experiment and Evaluation in Information Retrieval*, pp. 21–52. MIT Press, 2005.

[22] K. Järvelin and J. Kekäläinen. Cumulative gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[23] T. Joachims. Optimizing search engines using clickthrough data. *KDD*, pp. 133–142, Edmonton, Alberta, Canada, 2002.

[24] T. Joachims, L. Granka, B. Pang, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. *SIGIR*, pp. 154–161, Salvador, Brazil, 2005.

[25] J. Kleinberg. Authoritative sources in a hyperlinked environment. *J. of the ACM*, 46(5):604–632, 1999.

[26] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks*, 33(1-6):387–401, 2000.

[27] H. Li, D. Zhang, J. Hu, H.-J. Zeng, and Z. Chen. Finding keyword from online broadcasting content for targeted advertising. *The 1st International Workshop on Data Mining and Audience Intelligence for Advertising*, pp. 55–62, San Jose, California, 2007.

[28] M. Li, H. Li, and Z.-H. Zhou. Semi-supervised document retrieval. *Information Processing and Management*, 45:341–355, May 2009.

[29] Y. Lin, H. Lin, Z. Yang, and S. Su. A boosting approach for learning to rank using SVD with partially labeled data. *The 5th Asia Information Retrieval Symposium on Information Retrieval Technology*, pp. 330–338, Sapporo, Japan, 2009.

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

[31] D. Metzler. Automatic feature selection in the Markov random field model for information retrieval. *CIKM*, pp. 253–262, Lisbon, Portugal, 2007.

[32] D. Metzler, J. Novak, H. Cui, and S. Reddy. Building enriched document representations using aggregated anchor text. *SIGIR*, pp. 219–226, Boston, Massachusetts, 2009.

[33] R. Nallapati. Discriminative models for information retrieval. *SIGIR*, pp. 64–71, Sheffield, United Kingdom, 2004.

[34] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. Stanford Digital Library Working Paper SIDL-WP-1999-0120, Stanford University, 1999.

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

[36] I. Soboroff, C. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. *SIGIR*, pp. 66–73, New Orleans, Louisiana, 2001.

[37] K. Spärck Jones and C. J. van Rijsbergen. Information retrieval test collections. *J. of Documentation*, 32(1):59–75, 1976.

[38] J. Xu and H. Li. AdaRank: A boosting algorithm for information retrieval. *SIGIR*, pp. 391–398, Amsterdam, The Netherlands, 2007.

[39] W.-t. Yih, J. Goodman, and V. R. Carvalho. Finding advertising keywords on web pages. *WWW*, pp. 213–222, Edinburgh, Scotland, 2006.