Pseudo Test Collections for Learning Web Search Ranking Functions [*]

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

Abstract

Test collections are the primary drivers of progress in information retrieval. They provide a yardstick for assessing the effectiveness of ranking functions in an automatic, rapid, and repeatable fashion and serve as training data for learning to rank approaches. 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 the 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 evaluation or training learning to rank models. Experiments carried out on TREC Web track data show that learning to rank models trained using pseudo-test collections are capable of significantly outperforming unsupervised ranking functions and are statistically indistinguishable from models trained using manual judgments, thereby demonstrating the usefulness of the proposed approach.

1 Introduction

Reusable test collections play a central role in information retrieval 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, 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 approaches [17, 21, 31, 4, 28]. It would not be an exaggeration to say that test collections are the primary drivers of progress in IR today.

Academic researchers have access only to a small handful of test collections because they are very expensive to create. Traditionally, they are created as the byproduct of community-wide evaluations such as the NIST-organized Text Retrieval Conferences (TRECs). Using a process known as pooling [19, 35], 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

[1] University of Maryland, College Park. Email: nima@cs.umd.edu, jimmylin@umd.edu

[2] Information Sciences Institute, University of Southern California. Email: metzler@isi.edu

[3] King Abdullah University of Science and Technology (KAUST). Email: tamer.elsayedaly@kaust.edu.sa

[*] This paper is submitted to Computer Science Department of University of Maryland at College Park as a scholarly paper to partially satisfy requirements for Masters degree in the Spring 2011 academic semester. The paper has been accepted and will be presented at the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011)

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 [1, 18, 22, 33]. Both of these avenues are difficult to take for academic researchers.

This paper explores the feasibility of constructing Web search test collections automatically given only a large Web 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 provides a 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 a learning to rank model trained using the (unsupervised) pseudo test collection is capable of achieving significantly better effectiveness than other unsupervised models. It is further shown that our model is statistically indistinguishable from a learning to rank 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. Section 5 provides details of our experimental evaluation. Finally, Section 6 concludes the paper and describes several possible directions of future work.

2 Related Work

There are two primary 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 directions related to these two steps. However, there has not been any previous work that has combined these two steps within a single framework for the purpose of automatically constructing test collections. We now briefly describe previous work related to these two steps.

Sampling Pseudo Queries. Previous research has investigated methods for extracting implicit queries for contextual advertising [3, 25, 37] 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, ideas from these previous studies can be leveraged when sampling pseudo queries.

Other related work has shown anchor text to be a reasonable surrogate to query logs for the purpose of query reformulation [15]. We leverage this finding when constructing pseudo test collections for Web search. Our proposed approach samples pseudo queries from anchor text mined from a large Web crawl.

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 done *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 computing relevance scores for a large set of query-document pairs offline using a wide variety of (potentially expensive) relevance sources.

One related line of research deals with methods for inferring the relevance of unjudged documents when computing retrieval metrics [34, 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 far easier than pseudo test collection construction, because it is assumed that documents, a query, and some judgments are provided as evidence, whereas our framework will only rely on a minimal set of resources (e.g., a corpus).

Finally, it is important to note that our work differs from semi-supervised learning to rank approaches [16, 26, 27]. Although such approaches are designed to learn highly effective learning to rank models, it is typically not straightforward to adapt them to explicitly construct a test collection. Our goal is to construct test collections that can be used for a variety of tasks. Furthermore, our aim is to distill test collections in a completely unsupervised manner, whereas semi-supervised approach assume there is at least some labeled data 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 top of 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 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 data to train. 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 crowd-sourcing [2]. Both of these strategies are useful for low-cost *one-time* evaluations. However, they both 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 information. Indeed, as we will show in our experiments, even simple learning to rank approaches can be used with (noisy) pseudo test collections to learn highly effective ranking functions.

We propose constructing pseudo test collections by mimicking the process used to build manual test collections; that process 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 [19, 35], the minimal test collection paradigm [8, 7, 6], or crowd-sourcing [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 the current paper. Instead, it is assumed 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 document corpus, the framework will then automatically generate a set of queries. Since such queries are automatically generated, they are referred to as *pseudo queries*. It is important that the set of pseudo queries is 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 answered 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 have to be the output of a given retrieval system. Instead, it can be *any* subset of documents that can accurately and reliably be labeled in an unsupervised manner. It is also desirable, but not mandatory, 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.

Hence, to instantiate an instance of 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 information that are available. We hypothesize that 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 Automatically Constructing Web Search Test Collections

This section describes a specific instantiation of our general pseudo test collection framework that can be used to learn effective unsupervised learning to rank models for Web search. Web search is a particularly interesting domain to apply our proposed framework since the Web graph implicitly 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 [32], HITS [23], and SALSA [24]) 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 relevance judgments.

4.1 Implicit Relevance Signal

Given nothing but a collection of Web pages, anchor text provides a potentially high quality description of its target document. 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 result in poor pseudo queries and pseudo judgments. Examples of noisy 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 1961" which points to a biography as a reference). Thus, additional factors must be considered when using anchor text for construcitng pseudo test collections.

In order to reduce the effect of this noise and to sample higher-quality pseudo queries and pseudo judgments, a combination of strategies can be used. 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 are a number of techniques that can help achieve this goal.

In this paper, we only consider external anchor text (inter-domain links) as our source for sampling pseudo queries. In addition, we assign weights to each ¡anchor text, target document¿ pair based on a number of factors. One can think of these weights as a comparable measure of quality for each pair. The quality of a unique line of anchor text can then be computed by aggregating the weights assigned to all of the pairs the pair is associated with.

Here, we introduce a number of different weighting schemes to measure the quality of each ¡anchor text, target document¿ pair. An aggregation function is then defined in order to calculate the quality for each unique line of anchor text based on the weights assigned to those pairs.

4.1.1 Anchor Text-Document Weighting Schemes

Figure 1 illustrates the general structure of anchor text within the Web graph. From this struture, 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. Givent his information, our goal is to design a suitable weighting scheme for each of the anchor text-document pairs (i.e., ¡AT_k, $d_{k,i}$¿) that can later be used for extracting high quality pseudo queries and pseudo relevance judgments.

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

Spam. Spam documents are not reliable in terms of the information they contain. As a result, anchor text that describes a spam document should not be trusted. This weighting scheme measures the quality of an anchor-document pair solely on the basis of the spam score of the target document:

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

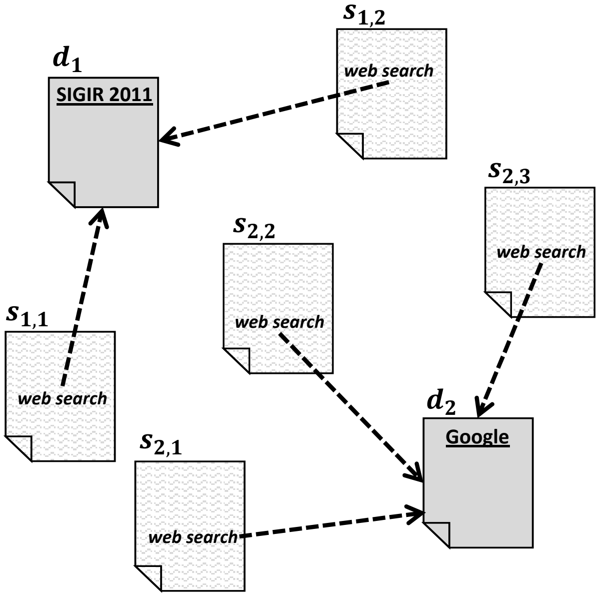

Figure 1: A sample structure of anchor text. In this figure, “web search” is a line of anchor text that points to two target documents d_1 and d_2. For each target document, anchor text originates from a number of source pages. For instance, $s_{1,2}$ and $s_{1,1}$ point to document d_1 with anchor text “web search”. In general, anchor 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 $1 \leq j \leq n_{k,i}$.

SrcSpam. While the level of “spamminess” of a target document is of high importance, the quality of sources that anchor text originates from tells how trustworthy that anchor text is as a descriptive phrase for a 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. Considering the fact that anchor text can originate from mutiple source pages, we need to define an aggregation function. In this weighting scheme, we aggregate the reliability of the sources using the harmonic mean:

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

PageRank. PageRank scores indicate the importance of a particular page based on 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,

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

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 points 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. [30]:

$$w(AT_k, d_{k,i}) = \textsc{Ham}(d_{k,i}) \times \sum_{s \in S(d_{k,i})} \frac{\delta(AT_k, d_{k,i}, s)}{|\textsc{anchors}(d_{k,i}, s)|} \tag{4}$$

where $S(d_{k,i})$ is the set of sites that link to $d_{k,i}$, $\delta(AT_k, d_{k,i}, s)$ is 1 if and only if anchor 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 link to $d_{k,i}$.

4.1.2 Anchor Text-Document Weight Aggregation

In order to measure how well a line of anchor text describes its target document, we propose to assign weights to each anchor-document pair according to different weighting schemes, as explained in the previous section. However, a line of anchor text can potentially 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 for a particular anchor text as follows:

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

where $w(AT_k, d_{k,i})$ is the weight associated with anchor text AT_k and its target document $d_{k,i}$, and n_k is the number of target documents 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 a set of pseudo queries and pseudo relevance judgments. We begin by describing how anchor text can be used for constructing a high quality set of pseudo queries.

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. The following sampling strategies are meant to address these factors to extract a high quality set of pseudo queries.

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 number of target documents often contain misspellings or otherwise are very specific topics. On the other hand, anchor text that have a large number of target documents are often broader topics and, in essense, very ambiguous; Examples of this is phrases such as "privacy policy" and "homepage". We consider these two types of anchor text as low quality anchor text, since they are either very specific and/or erroneous or very broad and ambiguous. When extracting pseudo queries, we would like to favor anchor text that have an average number of target documents.

Thus, we propose to 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 would like to favor lines of anchor text that have an average number of target documents, we are interested in partitions with an index that is not too small, but also not too large. Thus, we can define a probability distribution over indices n to satisfy this criteria. We chose to use a normal distribution $N(\mu, \sigma^2)$, since it satisfies the desired criteria. To sample pseudo queries, we first sample a partition according to the normal distribution, and then extract the highest weighted lines of anchor text, as weighted according to $w(AT_k)$ 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 will rely on unsupervised methods for estimating the relevance of query-document pairs.

4.3.1 Positive Judgments

Automatically identifying relevant documents is one of the core challenges of information retrieval. At the same time, as we mentioned earlier, anchor text serves as implicit relevance signal which can be used to extract limited high quality 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 relevance score 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 extracted, we sort its target documents according to their anchor text-document weight and select the top d_p documents as positive judgments for the query.

4.3.2 Negative Judgments

For a pseudo test collection to be complete, a set of negative judgments for a given pseudo query is as essential as a set of positive judgments for that query. Negative judgments must not only be non-relevant with respect to a query, but also must contain a diverse set of documents. Despite the utility of anchor text in extracting pseudo queries and positive judgments, anchor text is not a trusted source for extracting negative judgments.

Classical approaches in 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 given query. Documents that appear deep in the ranked list are likely to be "nearly relevant" or "non-relevant" with respect to the query. As a result, we retrieve a ranked list of R documents for each pseudo query based using a simple ranking function (e.g., language modeling) and sample d_n documents from the bottom of that list 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 with unsupervised and supervised learning to rank approaches.

5.1 Data and Methodology

We performed our experiments on the ClueWeb09 collection, a best-first Web crawl completed by CMU in early 2009. The collection contains one billion pages in ten languages totaling around 25 terabytes. 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). For evaluation purposes, we used a set of 50 queries and their corresponding judgments that were developed at TREC 2009 Web track.

The first segment of the English portion of the ClueWeb09 dataset consists of about 7.5 million unique lines of external (i.e. inter-domain) anchor text. We extracted the anchor text and used our proposed model to generate pseudo queries and pseudo judgments. Extracted anchor text, as depicted in Figure 1, provides us with information such as length and number of target documents. This information is necessary for computing the weighting functions and for the sampling process.

The yardstick that we will use to determine if our proposed approach is successful or not is whether a learning to rank model trained using an automatically constructed Web pseudo test collection can achieve better performance than a highly effective unsupervised baseline model. If this is the case, then we have shown that our pseudo test collections can be used to effectively combine evidence from an arbitrary set of features without any human judgments.

Hence, 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. We use BM25 as our baseline, since there are very few unsupervised retrieval models that are as effective. 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 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 as our proposed model. For the purpose of evaluation, all models are tested on the TREC 2009 Web track data.

Unless otherwise specified, we set the total number of sampled pseudo queries to 400 (Q), and 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. Also, 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 approaches, we report NDCG@20 [20] and ERR@20 [14], which are commonly used to evaluate the effectiveness of Web search tasks. To determine statistical significance between the various models, a one-side paired t-test is utilized.

The pseudo query sampling strategy used requires a mean and a variance for the underlying normal distribution $N(\mu, \sigma^2)$. Values for μ and σ can be chosen arbitrarily. However, based on our observation of the extracted anchor text, we found that anchor text that points to $\theta \leq 5$ target documents are not in general of high quality . Such anchor text make up a large portion of the extracted anchor text from our corpus, totaling about 7 million unique lines of anchor text. In order to ensure quality, we eliminate those anchor text that have fewer than 5 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, μ, based upon the statistics gathered from the remaining collection of anchor text. The distribution of number of targets per anchor text, after trimming those with $\theta \leq 5$ target documents, now has a median of 10. We choose this median to be the value for parameter μ used in the sampler. The value for σ^2 is set (arbitrarily) to 5.

5.2 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 that is learned from an automatically constructed pseudo test collection and another from manual judgments. We utilize the same features and learning algorithm to learn both models.

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 used in the model. The parameters of the linear model are learned using a greedy feature selection approach [29]. The model is iteratively learned by adding features to the model, one at a time, according to a greedy selection criteria. During each iteration, the feature that gives 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 then added to the model. 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 correla-

Weighting Scheme	Pseudo Query	Page Title	Relevance
Anchor	tax deductible	...Gifts, and Car Expenses	R
		Tax Deduction - Wikipedia	R
		Providence Hospitals...of Charity	NR
		Mortgage Tax Calculator	NR
	google labs	Google Labs	R
		Google Labs - Wikipedia	R
		Corporate Information ... Management	NR
		Blogger (service) - Wikipedia	NR
PageRank	download realplayer	RealPlayer - the best audio...	R
		RealPlayer	R
		Extreme sports games Free Download	NR
		Apple iPod, and format news...	NR
	yahoo privacy policy	Yahoo! Privacy Policy	R
		Yahoo! Store Privacy Policy	R
		free download yahoo messenger...	NR
		AltaVista - Privacy Policy...	NR

Table 1: Examples of pseudo queries and pseudo judgments extracted using the Anchor and PageRank weighting schemes. In the last column, R and NR show whether a document is extracted as a relevant or non-relevant page.

tion/redundancy between features. We also explored the use of AdaRank [36], an effective learning to rank approach, but found the results to be consistently worse than the simple greedy feature selection algorithm for this particular task.

5.3 Illustrative Examples

Table 1 shows selected examples of pseudo queries and their corresponding pseudo judgments extracted using the Anchor and PageRank weighting schemes. In this table, we provide the document ids for the relevant and non-relevant pages along with their titles.

The PageRank weighting scheme favors the anchor text that point to popular pages within the link structure. At the same time, popular pages with high PageRank values are highly likely to be website entry pages. Consequently, the anchor text associated with those pages are often navigational phrases. As a result, a large portion of the extracted pseudo queries are navigational 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 using the PageRank weighting scheme.

On the other hand, pseudo queries extracted using the Anchor weighting scheme consist of navigational phrases 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”.

Both pseudo queries and pseudo judgments can contain some noise as the result of imperfections in our weighting schemes. For instance, a query like “help forum” that was extracted using the Anchor weighting scheme is indeed a very ambiguous topic and covers a broad range of documents. In addition, as illustrated in Table 1, an extracted positive judgment for query “tax deductible” is in fact a document that contains specific examples for the requested concept that has weak relevance with respect to the original query.

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

Table 2: Effect of changing weighting schemes.

5.4 Results

This section describes the results of our experimental evaluation. We begin by describing our basic findings and then provide a deeper analysis of various aspects of our approach.

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. It is important to reiterate that our model is trained in a completely unsupervised manner using the pseudo test collections. All models are tested on the held-out TREC 2009 Web Track data. Table 2 shows the effectiveness for the different weighting schemes proposed in Section 4.1.1.

As the results values suggest, the PageRank weighting scheme leads to a lower quality learning to rank model. This finding is not unanticipated due to the fact that the PageRank weighting scheme estimates the quality of anchor text merely based upon the quality of its target documents regardless of the quality of its sources. Furthermore, the popularity of a document, as PageRank values indicate, does not necessarily guarantee the validity of the description its anchor text provides. Anchor text that points to a popular page might not contain a valid description of that document. Moreover, popular pages are more likely to be website entries and the anchor text that point to entry pages are often navigational phrases. Consequently, pseudo queries and pseudo judgments extracted using the PageRank weighting scheme are relatively of lower quality. The resulting pseudo collection, as the table suggests, is incapable of training an effective learning to rank model using the resulting pseudo test collection.

On the contrary, the Anchor weighting scheme makes use of the sources from which anchor text originates. The Anchor weighting scheme takes into consideration the number of unique domains that use the same line of anchor text to point to a target document. Moreover, it employs the number of unique lines of anchor text that point to a certain target document, approximately measuring the likelihood of broadness and ambiguity of topics within a certain document. These counts are intuitive metrics to gauge the quality of descriptiveness and relevance of a given anchor text regarding its target document. As a result, the pseudo test collection generated using this weighting scheme contains pseudo queries and pseudo judgments of higher quality, hence, produces a more effective learning to rank model.

This hypothesis is supported by the results, which show that a learning to rank model trained using the Anchor weighting scheme proves more effective, both in terms of NDCG@20 and ERR@20, compared to the BM25 baseline. Interestingly, the Anchor and SrcSpam weighting schemes 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 the learning to rank model compared to the “moderate” amount of manual training data used to train the cheating model. The results suggest that all of the weighting schemes, with the exception of PageRank, are effective for distilling pseudo test collections for the purpose of training learning to rank models.

We speculate that cheating model was only slightly better than BM25 in terms of NDCG due to

Feature	Value
lm-phrase	1.553
bm25-term	0.014
bm25-proximity	-0.567

Table 3: Learning to rank model resulting from the Anchor weighting scheme.

Feature	Value
lm-term	-0.0309
bm25-term	0.8911
bm25-phrase	0.1398

Table 4: Learning to rank model resulting from the cheating experiment.

the fact that all models were trained using ERR@20, even those evaluated using NDCG@20. This may simply be a case of so-called metric divergence. For this reason, the ERR@20 numbers provide a more valid 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 learning to rank model, then the ERR@20 would increase to 0.1610. However, when the spam feature was included in the model trained from the pseudo test collections, the effectiveness of the resulting model does not improve over the results presented in Table 2. This suggests that the strategy that we use for sampling non-relevant documents does not contain enough spam documents to allow the model to learn the importance of the spam feature.

This raises an important issue related to our proposed pseudo test collection framework. The strategies that are used for generating positive and negative judgments have the risk of significantly biasing the test set in unexpected ways. To minimize this risk, it is beneficial to select a very *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 more formally addressed and studied in the future. Although bias is a problem, we found that it does not ultimately affect the most important finding of our experiments, which is the fact that the learning to rank model learned (even without spam features) is capable of outperforming BM25 in a completely unsupervised manner.

5.4.2 Analysis of Learned Models

Table 3 shows the learning to rank model learned using a pseudo test collection with Anchor weighting, while Table 4 shows the cheating model that was learned using manual judgments.

The models are markedly different. For example, in the cheating model, bm25-term, which is the standard BM25 score is assigned a very high weight, while the same feature is assigned a low weight in the pseudo collection-based model. This is likely the result of biases found in both the pseudo test collection and the actual test collection. The pseudo test collection seems biased towards *phrase* features, which may stem from the fact that the page a piece of anchor text points to is likely to contain the anchor text itself, as a phrase, within the page. On the other hand, the judgments collected from TREC used pooling, and it is well known that such pools are heavily biased towards BM25, since many TREC participants utilize the ranking function. Therefore, regardless the source of judgments, it is difficult to avoid inherent biases.

At a higher level, both of the models include the same feature types (i.e., term, phrase, and proximity), but the relative weighting of each is different. It is interesting that the proximity score in the model trained using pseudo test collections has a large negative weight. It is likely that the phrase feature is actually too large and the proximity feature is simply offsetting it.

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: Effect of changing query size and number of negative judgments. Model are resulted from the Anchor weighting scheme and number of positive judgments is set to 10.

5.4.3 Pseudo Collection Size

The number of pseudo queries as well as the number of pseudo judgments indicate the size of a pseudo collection. More pseudo queries along with more pseudo judgments result in larger pseudo collections, which in effect provide a larger training set for a learning to rank model. However, sampling more pseudo queries or more pseudo judgments is not always good news, since more noisy data find their way to the final pseudo collection. Thus, the effect of altering these two parameters, i.e. number of pseudo queries and number of pseudo judgments, needs to be studied.

In the previous section, the number of generated pseudo queries and the number of positive and negative pseudo judgments were kept constant throughout the experiments. In this section, we build larger datasets by changing these numbers. However, to avoid an exhaustive seach over the space created by altering these two parameters, we pick the weighting scheme that result in more effective models according Table 2 (i.e., the Anchor weighting scheme) and present the behavior of that particular weighting scheme under different circumstances (i.e. different number of pseudo queries and pseudo judgments). To further simplify the presentation, we only change the number of pseudo negative judgments (d_n). This serves for two purposes. By altering the number of negative judgments we are changing the size of the resulting pseudo collection. Additionally, we are changing the ratio of positive to negative judgments. This shows the stability of sampled positive judgments.

Table 5 shows the effect of these changes for the Anchor weighting scheme for a small subspace of parameters Q and d_n. This table suggests that extracting more pseudo queries, slightly improves the resulting model in general. In addition, lowering the ratio of positive to negative judgments (or in particular, increasing d_n), adds more noise to the training data and eventually results in a slightly less effective model. The last row and last column in Table 5 show the average for each row and column respectively.

6 Conclusions and Future Work

In this paper, we proposed a general unsupervised framework for constructing high quality test collections given nothing but a corpus. The resulting pseudo test collections, which consists of a set of pseudo queries and pseudo relevance judgments, can be used to evaluate and train learning to rank models. We also described an instantiation of the proposed framework for Web search that leverages anchor text as a source of implicit relevance signals. This new problem was motivated by the fact that test collections are essential components in information retrieval research, but at the same time labor-intensive and expensive to manually construct. Our proposed model can be viewed as a way to mine implicit relevance signals from a corpus of documents and automatically generate test collections without any human intervention, thereby increasing the amount of data available for evaluation and training learning to rank models.

We evaluate automatically generated pseudo test collections by training learning to rank models over the extracted pseudo queries and pseudo judgments. Our experimental evaluation carried out on TREC Web track data showed that completely unsupervised learning to rank models trained using pseudo test collections can outperform existing unsupervised ranking funtions. This demonstrates the utility of the proposed approach.

There are several possible directions of future work. First, we would like to develop a better understanding of the weighting schemes within the current instantiation of the general framework. Second, we would like to create other methods of extracting relevance judgments, especially negative judgments. In addition, we would like to study other sources of implicit relevance signals such as page titles and high term frequencies. Finally, it would be interesting to use our pseudo test collections for evaluation. We are particularly interested in the reusability of the test collections and their ability to automatically order systems according to their effectiveness.

References

[1] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10, New York, NY, USA, 2006. ACM.

[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. In *SIGIR'07*, pages 559–566. ACM Press, 2007.

[4] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, pages 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. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 63–70, New York, NY, USA, 2007. ACM.

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

[7] B. Carterette and J. Allan. Semiautomatic evaluation of retrieval systems using document similarities. In *Proc. 16th Intl. Conf. on Information and Knowledge Management*, pages 873–876, New York, NY, USA, 2007. ACM.

[8] B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 268–275, New York, NY, USA, 2006. ACM.

[9] B. Carterette, E. Gabrilovich, V. Josifovski, and D. Metzler. Measuring the reusability of test collections. In *WSDM '10: Proceedings of the third ACM international conference on Web search and data mining*, pages 231–240, New York, NY, USA, 2010. ACM.

[10] B. Carterette and R. Jones. Evaluating search engines by modeling the relationship between relevance and clicks. In *Advances in Neural Information Processing Systems 20 (NIPS 2007)*, pages 217–224, 2008.

[11] B. Carterette, E. Kanoulas, V. Pavlu, and H. Fang. Reusable test collections through experimental design. In *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 547–554, New York, NY, USA, 2010. ACM.

[12] D. Chakrabarti, R. Kumar, and K. Punera. Generating succinct titles for web urls. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 79–87, New York, NY, USA, 2008. ACM.

[13] D. Chakrabarti, R. Kumar, and K. Punera. Quicklink selection for navigational query results. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 391–400, New York, NY, USA, 2009. ACM.

[14] O. Chapelle, D. Metzler, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Proc. 18th Intl. Conf. on Information and Knowledge Management*, 2009.

[15] V. Dang and W. B. Croft. Query reformulation using anchor text. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM 2010)*, pages 41–50, New York, New York, 2010.

[16] K. Duh and K. Kirchhoff. Learning to rank with partially-labeled data. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 251–258, New York, NY, USA, 2008. ACM.

[17] F. Gey. Inferring probability of relevance using the method of logistic regression. In *Proc. 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994.

[18] Q. Guo and E. Agichtein. Exploring mouse movements for inferring query intent. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 707–708, New York, NY, USA, 2008. ACM.

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

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

[21] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, New York, NY, USA, 2002. ACM.

[22] T. Joachims, L. Granka, B. Pang, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005)*, pages 154–161, Salvador, Brazil, 2005.

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

[24] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (salsa) and the tkc effect. *Comput. Netw.*, 33(1-6):387–401, 2000.

[25] H. Li, D. Zhang, J. Hu, H.-J. Zeng, and Z. Chen. Finding keyword from online broadcasting content for targeted advertising. In *ADKDD '07: Proceedings of the 1st international workshop on Data mining and audience intelligence for advertising*, pages 55–62, New York, NY, USA, 2007. ACM.

[26] M. Li, H. Li, and Z.-H. Zhou. Semi-supervised document retrieval. *Inf. Process. Manage.*, 45:341–355, May 2009.

[27] Y. Lin, H. Lin, Z. Yang, and S. Su. A boosting approach for learning to rank using svd with partially labeled data. In *Proceedings of the 5th Asia Information Retrieval Symposium on Information Retrieval Technology*, AIRS '09, pages 330–338, Berlin, Heidelberg, 2009. Springer-Verlag.

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

[29] D. Metzler. Automatic feature selection in the Markov random field model for information retrieval. In *CIKM 2007*, pages 253–262.

[30] D. Metzler, J. Novak, H. Cui, and S. Reddy. Building enriched document representations using aggregated anchor text. In *Proc. 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 219–226, New York, NY, USA, 2009. ACM.

[31] R. Nallapati. Discriminative models for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004)*, pages 64–71, Sheffield, United Kingdom, 2004.

[32] 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.

[33] F. Radlinski and T. Joachims. Query chains: Learning to rank from implicit feedback. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2005)*, pages 239–248, Chicago, Illinois, 2005.

[34] I. Soboroff, C. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 66–73, New York, NY, USA, 2001. ACM.

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

[36] J. Xu and H. Li. AdaRank: A boosting algorithm for information retrieval. In *SIGIR 2007*, pages 391–398. ACM.

[37] W.-t. Yih, J. Goodman, and V. R. Carvalho. Finding advertising keywords on web pages. In *Proceedings of the 15th International Conference on World Wide Web*, 2006.