

# Leveraging Temporal Dynamics of Document Content in Relevance Ranking

Jonathan L. Elsas  
Language Technologies Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213  
jelsas@cs.cmu.edu

Susan T. Dumais  
Microsoft Research  
Redmond, WA 98052  
sdumais@microsoft.com

## ABSTRACT

Many web documents are dynamic, with content changing in varying amounts at varying frequencies. However, current document search algorithms have a static view of the document content, with only a single version of the document in the index at any point in time. In this paper, we present the first published analysis of using the temporal dynamics of document content to improve relevance ranking. We show that there is a strong relationship between the amount and frequency of content change and relevance. We develop a novel probabilistic document ranking algorithm that allows differential weighting of terms based on their temporal characteristics. By leveraging such content dynamics we show significant performance improvements for navigational queries.

## Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Miscellaneous

## General Terms

Algorithms, Experimentation

## Keywords

Web search, versioned documents, temporal change

## 1. INTRODUCTION

Web documents are dynamic. Newspaper homepages such as the New York Times<sup>1</sup> change several times a day, marketplace sites such as Craigslist<sup>2</sup> can change many times an hour and blogs are updated with varying frequencies when new posts and comments are added. Some of these changes are substantial and significant for information seekers – new

<sup>1</sup><http://nytimes.com>

<sup>2</sup><http://craigslist.org>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSDM'10, February 4–6, 2010, New York City, New York, USA.  
Copyright 2010 ACM 978-1-60558-889-6/10/02 ...\$10.00.

stories appearing on a homepage or new comments to a blog post. Others hold less interest for those looking for information – visitation counters, advertisement content, or formatting changes have little impact on the page content.

Currently, document ranking algorithms only have a static view of the page content. In this work we explore the interaction between the dynamics of web documents and relevance ranking, using document representations that view a document as a dynamic entity. We focus specifically on *navigational* searches, where there is very little variation across users on the clicked results, and there tend to be a small number of highly relevant documents that are consistently relevant across time. We find that, for these queries, there are significant relationships between the likelihood of change and the relevance level of the page. We develop a novel probabilistic retrieval model which takes into account dynamic content, and show significant performance improvements over a model that only views a document at a single point in time. To our knowledge, this is the first published study looking at content change within documents from a *relevance ranking* perspective.

Our contributions in this work include: the first evaluation of the relationship between document dynamics and relevance ranking, the introduction of a novel document ranking algorithm for use with dynamic documents, and a query-independent document prior based on document dynamics. We show that these two approaches to ranking dynamic documents are complementary and both yield significant performance gains.

## 2. RELATED WORK

Several studies have described characteristics of dynamic web content. Fetterly et al. [10] conducted a large scale exploration of the frequency and amount of change of approximately 150 million web pages over a ten week period. To measure the amount of page change, the authors use Broder et al.'s *shingleprinting* [7], described in more detail below. They find that roughly 65% of the pages studied do not change at all over the time period sampled. This analysis also shows correlations between change frequencies and top-level domains, for example .com domains are more likely to change than .org and .gov domains, and spam pages are more likely to change than others.

Ntoulas et al. [17] perform a similar study on a smaller set of web pages, 150 sites comprised of roughly 4.4 million pages downloaded every week for a year. The authors additionally investigate changes in link structure over time and “new” content created over the course of their collection. A

TF.IDF weighted cosine similarity is used to measure the amount of content change across samples, and the authors find similar change frequency and amount as the Fetterly et al. study above. They also find that link structure changes more rapidly than page content, suggesting that ranking algorithms which rely on link information may need more frequent crawls to accurately reflect the web graph. Using this same data set, Cho et al. [8] investigated how changes in link patterns could be used to identify high quality pages. In the research reported in this paper, we use changes in page content rather than link structure to set non-uniform document priors. The relationship of content change and page revisitation patterns is explored by Adar et al. [2, 3]. In these studies, the authors sample 55,000 pages with diverse revisitation characteristics at an hourly interval. A variety of ways to characterize change are introduced in this work, including measurements of document structural change and term-level content change. The authors show that the popularity of the page (number of unique visitors) is positively correlated with the frequency of change, but not the amount of change.

Summarization and visualization of dynamic documents and versioned collections has been explored in several studies [13, 4, 1]. Jatowt et al. [13] look at temporal characteristics of term frequencies over time and these temporal features are used to identify vocabulary for use in summarization. In that work, two classes of interesting terms are identified for inclusion in a summary: *prevalent* terms which occur in most snapshots of a page and *active* terms which appear and disappear in the document over time.

Implications of content change for web crawler policies have been investigated in several studies, and of particular pertinence to this work is that of Olston and Pandey [19]. In that work, the authors define the notion of *information longevity*, or the length of time a fragment of text remains on a page. A model of content generation is developed, which is designed to account for differing lifetimes of text. The authors refer to their different content generation models as *static*, *churn* and *scroll*. This model is then used as motivation for setting crawling policies.

Several temporal aspects of document collections have also been investigated, typically focusing on either document publication time-stamps or temporal mentions in the document text itself. The distributions of publication dates in result sets have been used for identifying query types or enhancing the presentation of those results. Jones and Diaz [9, 14] look at the temporal distribution of document time stamps returned for a query, and identify different query types based on those distributions. Alonso et al. [4] present a method of clustering and exploring search results based on temporal expressions within the text. Li and Croft explore retrieval models that leverage document timestamps, finding that for some classes of queries, favoring more recent documents improves performance [16]. Recently, Zhang et al. explored identifying and re-ranking search results for *time-sensitive queries* that implicitly refer to a year. They found that for this subset of queries, favoring recent documents can improve retrieval performance [21].

Work on versioned collections [5, 6, 11], such as source control systems or Wikis, generally explores the efficacy of indexing methods in providing access to previous versions of documents. This line of research focuses on indexing structures and efficiency, whereas our work is concerned with the

relationship between the changing document content and relevance ranking.

The work presented here is distinguished from that previous work by focusing specifically on the implications of content change to relevance ranking. Similarly to some of this previous work [13, 3], we identify interesting or important elements of a document’s vocabulary based on terms’ temporal characteristics. Previous studies have favored those terms for summarization or visualization, whereas here we focus the utility of those terms to improve relevance ranking. In addition, we develop a query-independent document prior using the overall temporal dynamics of the document content.

### 3. DOCUMENT DYNAMICS & RELEVANCE

Documents change for many reasons. The New York Times pages change whenever new stories are added or old stories are updated, Craigslist when new classified ads are added, and academics’ home pages when new papers are published. All of these pages change at different frequencies and in different amounts. In this section we provide some examples and intuitions about how such change may be used to improve relevance ranking. We examine two change features: (1) a query-relevant feature reflecting how the terms on a page (in particular those that match the query) change over time, and (2) a query-independent feature reflecting how frequently or by how much the page changes over time.

Different terms in a page’s vocabulary may be more stable or dynamic, they may remain constant over the lifetime of the page, or they may appear or disappear as the document changes. These differences in temporal term characteristics may lend some insight into the terms’ importance on the page for various information needs.

For example, on the page <http://allrecipes.com>, a popular website for sharing and rating recipes, stable terms that appear consistently over time include: allrecipes, cook, cookbooks, copyright, desserts, easy, healthy, newsroom, quick, recipe, and recipes. These terms represent a mix of characteristic terms that are descriptive of the overall central topic of the page and navigational elements. In contrast, terms that come and go during the summer months include: independence, themed, flag, fourth, macaroni, cream, zucchini, and grilled. These terms represent specific content that may have been on the page for a period of time, in this case relating to current holidays or the most recent recipes. This dynamic group of terms, although pertinent to the content of the page at a particular time, are not central to the main topic of the page.

When considering whether a document is relevant for a particular query, we may wish to consider whether the information need is more likely to be addressed by consistent or changing terms. Is the searcher more likely to be seeking dynamic or static content? Queries reflecting current events or late-breaking news may be better served by content that is recent (thus dynamic over time). In the above example, a searcher looking for recipes to cook for the Fourth of July holiday might be satisfied with term matches in the more dynamic portion of the page. On the other hand, for navigational searches we may want to favor content that is stable over a longer period of time and characteristic of the page in general. In our example, a searcher looking for the [allrecipes.com](http://allrecipes.com) homepage would be better served by that portion of the document that does not change.

Characteristics of document-level change such as how frequent or how much the document changes may also tell us something about the relevance of the page. A page that changes regularly may indicate that the page is actively maintained or frequently communicates with readers. This is an indication that the page may be more popular and possibly more relevant for some types of queries. Previous studies have shown that there is a strong relationship between web page popularity, frequency of revisitation and the frequency of page change [2]. Based on this observation, just knowing whether or not a page changes may be a useful feature in relevance ranking, independent of the query.

### 3.1 On Evaluating Dynamics and Relevance

Evaluating document dynamics and associated relevance judgments poses some special challenges. Information needs that reflect searches for late-breaking news or newly created content are particularly difficult to study in a traditional information retrieval evaluation. Queries for dynamic content and relevance judgments must be collected contemporaneously with the document collection. In the above example, an ongoing document collection must be underway when an event such as the July 4th holiday occurred. Queries relating to that event must be collection and assessed immediately, before dynamic documents change. Due to the possibly fleeting nature of the information need and the equally dynamic document set likely to be relevant, collecting accurate and realistic relevance judgments is impractical on a reasonably large scale. Although this is an interesting research direction, these types of information needs are not the focus of this work.

Navigational searches represent another category of information needs that could benefit from knowledge of the dynamics of document content. As in the `allrecipes.com` homepage search described above, stable terms that are characteristic of the page content are likely to be more important than transient content. The relevant pages for navigational queries are also unlikely to change over time. For this reason, it is feasible to create a test collection to investigate the relationship between relevance and content dynamics. The relevance assessment does not necessarily need to occur at the same time that the query is issued, and *any* version of the document over time should be equally relevant. Because of these factors, we choose to focus on navigational queries in this work.

## 4. DOCUMENT COLLECTION

### 4.1 Collection Description

For the purposes of studying content change and its relation to relevance ranking, we created a collection of roughly two million HTML web documents crawled every week for a period of ten weeks, from June 27, 2008 to August 27, 2008. Each of the individual crawls we refer to as a *time slice* or just *slice* of our combined collection.

The documents chosen for crawling were obtained from a collection of queries and documents for which human relevance judgments were available. Queries were chosen randomly from the logs of a web search engine. 18564 queries, each of which had at least 25 judged documents, were selected and the corresponding documents were crawled for ten weeks. This dataset is divided into training queries (60%) and test queries (40%). The training set is used to set all

smoothing and mixing parameters, as described below in Section 6.4, and the test set is used for evaluation. See Table 1 for detailed collection statistics.

Documents	2482367
All Queries	18564
Navigational Queries	2056
Ave. Document Length	2886.8 words
Ave. Query Length	2.70 words
Ave. Judgements per Query	145.6

Table 1: Collection Statistics

Documents were judged for graded relevance by human assessors using a five-point scale for relevance: Bad(0), Fair(1), Good(2), Excellent(3) and Perfect(4). Navigational results for a query (if any) were assigned the Perfect rating. Relevance assessments were collected over a period of several months and completed prior to the document collection. Although this collection period does not match exactly our crawl, we make the assumption that, particularly for navigational queries, the relevance of a page remains unchanged between the time of judgment and our crawl.

This temporal document collection was created with the intent of studying the relationship between document content dynamics and relevance ranking. Thus the documents comprising our collection were chosen because they had been returned by web search engines in response to a query. This differs significantly from previous collections used to study document dynamics over time, which are built from random samples of documents [10], documents with differing popularity and revisitation characteristics [3], or documents from popular domains [17].

### 4.2 Document Analysis

Due to the difference in selection of documents to create this collection as compared to previous collections, we first explore the temporal dynamics of this collection and compare it to other collections used to measure change on the web.

#### 4.2.1 Document Change

Several measures have been used in the past to assess the frequency and amount of content change: shingleprints [10], cosine similarity [17], and Dice similarity [3]. In this paper, we will use the shingleprinting algorithm, described below.

The shingleprinting technique computes a hash signature for each term window in the document, deterministically samples those signatures and computes the signature overlap across subsequent versions of the document [7, 10]. In the limit as the number of samples increases, this measure approaches the Jaccard coefficient. This similarity computation is efficient, has a freely-available implementation<sup>3</sup>, and has proven to be effective in a variety of settings such as near duplicate detection. For the analysis here, we use shingleprints to measure the similarity of subsequent versions of a document over time, using the same parameters as previously published [10]. As our primary measure of content change, we average the shingleprint similarity values over all

<sup>3</sup><http://research.microsoft.com/en-us/downloads/4e0d0535-ff4c-4259-99fa-ab34f3f57d67/default.aspx>

time slices in our collection:

$$ShSim(D) = \frac{1}{T-1} \sum_{t=2}^T \frac{|Sh(D^{(t)}) \cap Sh(D^{(t-1)})|}{N} \quad (1)$$

where  $Sh(D^{(t)})$  are the sampled shingles in document  $D$  at time  $t$ ,  $N$  is the number of shingles sampled per document, and  $T$  is the number of time slices,  $T = 10$  in our collection. In our work, as in [10],  $N = 84$ . We define a measurement of the amount of change in a document over time as  $ShDiff(D) = 1.0 - ShSim(D)$ .

When looking at the change amount of pages over time, we see very similar trends as were observed in previous studies [10, 17]. We observed that 62.7% of pages remain virtually the same over all sampled timeslices, with on average greater than 95% of their shingle prints identical. There is a small percentage of pages (<2%) that change completely according to the  $ShDiff(D)$  measure on every crawl. The distribution of change amount over the course of our collection is also quite similar to previous studies, as can be seen in Figure 1. Figure 2 shows the distribution of the change frequency in the collection, with 6% of the documents having at least some change at each crawled version.

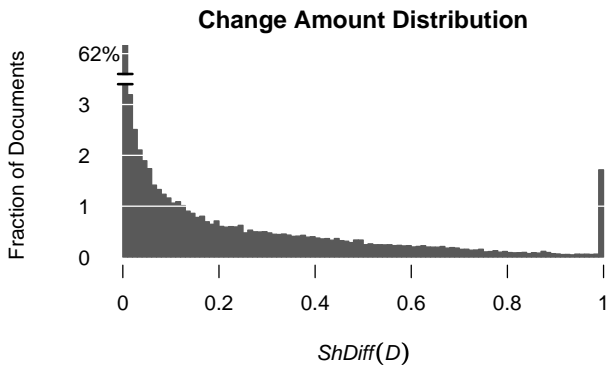


Figure 1: Change Amount ( $ShDiff(D)$ ) distribution. Vertical axis truncated to show low-frequency distribution.

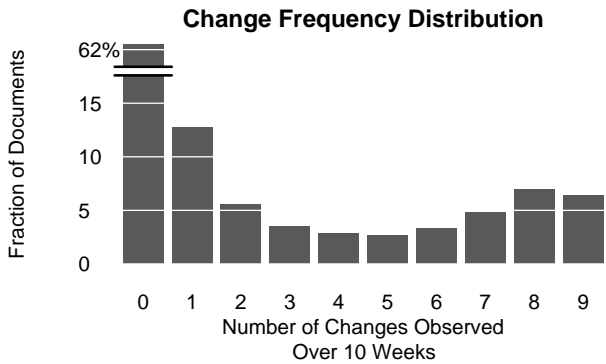


Figure 2: Change Frequency distribution. Vertical axis truncated to show low-frequency distribution.

#### 4.2.2 Document Change and Relevance

When looking at the relationship between change and relevance, several interesting trends emerge. Figure 3 shows the fraction of pages that change for each relevance level. We see that pages with higher relevance (judged 3 or 4) are more likely to change than others, regardless of the query: 62.9% of pages judged “4” change over the collection period, whereas only 37.3% change in general.

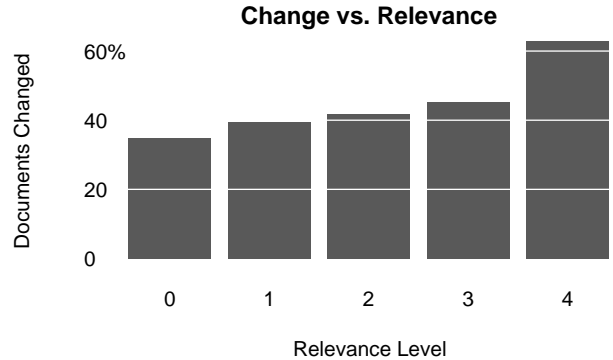


Figure 3: Percentage of documents at each relevance level that undergo any change.

Highly relevant documents are not only more likely to change than documents in general, they also tend to change to a greater degree than other documents. Figure 4 shows the average  $ShDiff(D)$  as a function of relevance level, for the documents that change. In this figure, we can see for those documents that change, the amount of change is greater for those documents judged more relevant.

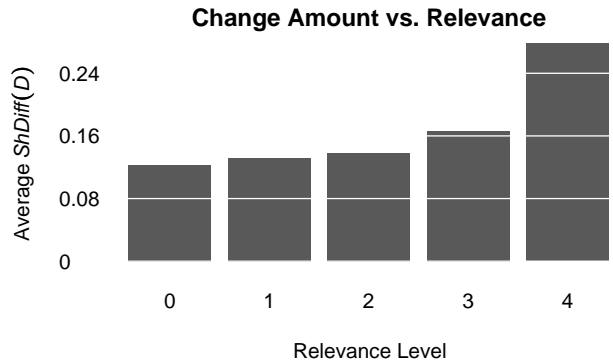


Figure 4: Amount of change at each relevance level, only documents that change.

### 4.3 Query Analysis

We now turn our attention to the query set used in our experiments. As stated above, we hypothesize that a retrieval algorithm that is able to differentially weight static and dynamic content may improve performance on navigational queries. Navigational queries have been defined in previous research as those queries with one or a small number of “right” results [7]. For collections that do not contain explicit relevance judgments, many measures have been proposed to identify navigational queries, including average click posi-

tion and the consistency of anchor text [15]. Since we have relevance judgments for the queries in our dataset, we define navigational queries using these explicit judgments. We consider a query as navigational if the query has a document judged as “perfect” (4). There are 2056 queries with at least one document judged “perfect” in our collection. Note that some queries have more than one document with a “perfect” judgement, but in most cases these documents are equivalent (e.g. redirect to the same page).

## 5. RETRIEVAL OVER SINGLE SLICES

Current search engines have a static view of the document collection, with only a single version of a document present in the index at any point in time. But, as documents change, the performance of our retrieval system may vary. In this section, we investigate this variance across time, and evaluate it in several ways: variance of the query-document scores, variance of the document *ranks*, and variance of the ranking algorithm’s performance (as measured using the explicit relevance judgments) over time. The stability of document scores (and ranks) over time, particularly for navigational queries, is a desirable feature of a retrieval system. Because these queries often function as a means to find a single known web page, the stability of those pages in the result set is important for a consistent user experience.

The following experiments explore the stability (or lack thereof) of the document scores, ranks and query performance over the different time slices. For these experiments we take a simple unigram language modeling retrieval approach, with Dirichlet-smoothed maximum likelihood estimates [20]:

$$\begin{aligned}
 P(D|Q)^{\text{rank}} &= P(D)P(Q|D) \\
 &= P(D) \prod_{q \in Q} P(q|D)^{n(q,Q)} \\
 &= P(D) \prod_{q \in Q} \left( \frac{n(q,D) + \mu P(q|C)}{|D| + \mu} \right)^{n(q,Q)} \quad (2)
 \end{aligned}$$

where  $D$  is the document in a single timeslice,  $q \in Q$  are the query terms,  $n(q,D)$  gives the term frequency of the term  $q$  in the document  $D$  (or query  $Q$ ),  $C$  is the collection and  $\mu$  is a smoothing parameter, set at  $\mu = 1500$  for these experiments. In the above formulation, we assume documents have a uniform prior  $P(D)$ . All experiments were conducted with the Indri search engine<sup>4</sup>.

We use Discounted Cumulative Gain (DCG) and Normalize Discounted Cumulative Gain (NDCG) at rank cutoff  $k$  as our primary evaluation metrics to measure retrieval performance [12]. The formulation used here is:

$$\text{DCG}@k = \sum_{i=1}^k \frac{2^{r(i)} - 1}{\log_2(1 + i)}$$

where  $r(i) \in \{0 \dots 4\}$  is the relevance level of the  $i$ th ranked document and  $\log_2$  is a base-2 logarithm. NDCG is the DCG value normalized by DCG of an optimal ranking,  $\text{DCG}^*@k$ :

$$\text{NDCG}@k = \frac{\text{DCG}@k}{\text{DCG}^*@k}$$

Figures 5 and 6 show the variance of document scores and ranks across the time slices. As we can see from the

<sup>4</sup><http://www.lemurproject.org/indri/>

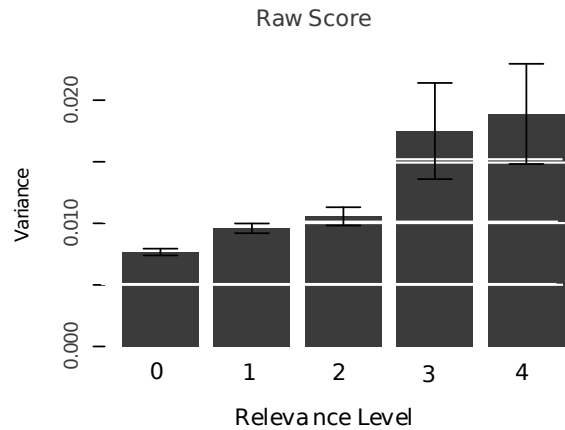


Figure 5: Variance of the document scores across time slices, with relevance level increasing from left to right. Error bars show one standard error.

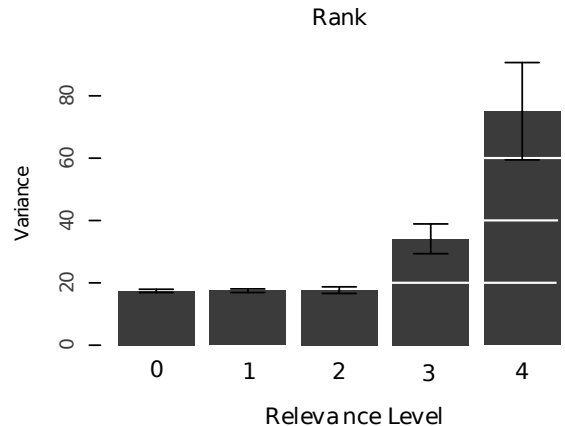


Figure 6: Variance of the document ranks across time slices, with relevance level increasing from left to right. Error bars show one standard error.

figure, document scores do vary across time, and this variance is higher for more relevant pages. The effect of this variance across time is also seen in measures of query performance, such as NDCG. Average NDCG@1 ranges from 0.215 to 0.227 across the different temporal slices.

In these experiments we have focused exclusively on scoring functions based on a document’s textual content to show that scores and performance vary across time slices. We do so to understand the variability over time of this important ranking feature. As stated above, particularly for navigational queries, this is an undesirable property of a retrieval model. Web search engines have other means to stabilize document scores through a richer feature set used in ranking. This data includes anchor text from the web graph, query logs, and click-through data, which all may be more stable than the document text itself.

## 6. RANKING DYNAMIC DOCUMENTS

In this section we explore ranking of dynamic documents. We present a retrieval model for dynamic documents, a document prior that leverages content dynamics, and present some insight into parameter fitting in these models.

## 6.1 A Language Model for Dynamic Document Retrieval

This section presents a novel document retrieval model that leverages changing document content. Based on the language modeling framework, this model allows differential weighting of content based on that content’s temporal characteristics. We will first present some notation used below, then the mathematical formulation of the general retrieval model, and finally provide the estimation details.

When we are dealing with documents over time, instead of a single term frequency for each word in the document, we have a vector of term frequencies. We will represent this vector of term frequencies with a term count function similar to Equation 2:

$$\mathbf{n}(q, D) = \langle n(q, D^{(1)}), n(q, D^{(2)}), \dots, n(q, D^{(T)}) \rangle$$

where the superscripts refer to time slices in our index, ranging from  $1 \dots T$ . We define the following functions as the sum of term frequencies across all time slices, and the number of non-zero entries in the term frequency vector respectively:

$$N(q, D) = \sum_{i=1}^T n(q, D^{(i)}) \text{ and}$$

$$c(q, D) = \sum_{i=1}^T \mathbb{I}(n(q, D^{(i)}) > 0)$$

where  $\mathbb{I}(\bullet)$  is an indicator function equal to 1 when  $\bullet$  is true and zero otherwise.

In order to differentially weight terms, we will identify several classes of terms with different temporal characteristics. These classes are based on the term frequency distribution over time,  $\mathbf{n}(q, D)$ , and represent a variety of different criteria we may wish to use to favor (or avoid) during document ranking. For example, for a query seeking information on recent events, it may be advantageous to place a high weight on terms that recently entered the document’s vocabulary. Navigational queries, on the other hand, may be better served by content that is present in every time slice. This content is likely to be more reflective of the documents ongoing central topic, a critical aspect for these types of information needs.

In this work we define three classes of terms to weigh differentially. We will call these *long-term*, *mid-term* and *short-term*, reflecting the length of time the vocabulary is present on the page. These can be thought of as roughly equivalent to Olston and Pandey’s *static*, *scroll* and *churn* models [19], however our formulation is viewed at the term-level rather than the shingle-level. Three different language models can then be built from each of these term classes,  $P(q|D_L)$ ,  $P(q|D_M)$  and  $P(q|D_S)$ . Similar to work in fielded retrieval and combining representations [18], we model the document language model  $P(q|D)$  as a mixture of these models. As in Equation 2, we assume term independence and rank by  $P(D|Q) \approx P(D) \prod_{q \in Q} P(q|D)^{n(q, Q)}$ , using the following mixture model to estimate the query term likelihoods:

$$P(q|D) = \lambda_L P(q|D_L) + \lambda_M P(q|D_M) + \lambda_S P(q|D_S) \quad (3)$$

where  $\lambda_L, \lambda_M, \lambda_S \in [0, 1]$  and  $\lambda_L + \lambda_M + \lambda_S = 1$ .

It is convenient to think of these three models as being derived from three virtual documents. Terms only appearing

in a small number of time slices comprise the short-term virtual document, and terms in all of the time slices comprise the long-term document. Taking the three virtual documents together gives the union of the document across all time slices.

More formally, we derive the language models as follows. First, we will define term counting functions, where the subscript  $j \in \{L, M, S\}$  refers to the different mixture components:

$$\nu(q, D_j) = N(q, D) \phi_j(\mathbf{n}(q, D)). \quad (4)$$

The functions  $\phi_j \in [0, 1]$ ,  $\sum_j \phi_j(\mathbf{n}) = 1$  control the distribution of total term counts across the different mixture components.

With these term counting functions, shown in Equation 4, we can then estimate our mixture component language models as follows:

$$P(q|D_j) = \frac{\nu(q, D_j) + \mu_j P(q|C_j)}{|D_j| + \mu_j} \quad (5)$$

letting  $|D_j| = \sum_{w \in D_j} \nu(w, D_j)$  be the length of this virtual document. As in Equation 2,  $\mu_j$  is a component-specific smoothing parameter, and  $P(q|C_j)$  is a maximum-likelihood estimate of the collection probability, limited to the  $j$  mixture components:

$$P(q|C_j) = \frac{\sum_{D_j} \nu(q, D_j)}{\sum_w \sum_{D_j} \nu(w, D_j)}$$

The functions  $\phi_j$  enable the distribution of term counts across our different mixture component in varying proportions based on a term’s temporal distribution or term frequency vector,  $\mathbf{n}(q, D)$ . In this work, we define  $\phi$  over the number of slices this term occurs in,  $c(q, D)$  and apply a simple threshold as follows:

$$\phi_L(n) = \begin{cases} 1 & \text{if } c(q, D) \in [0.9 \times T, T] \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_M(n) = \begin{cases} 1 & \text{if } c(q, D) \in [0.5 \times T, 0.9 \times T] \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_S(n) = \begin{cases} 1 & \text{if } c(q, D) \in [0, 0.5 \times T] \\ 0 & \text{otherwise.} \end{cases}$$

where  $T = 10$  is the total number of slices in our collection.

There are many ways to distribute term counts (or, equivalently, probability mass) across the mixture components, and the method presented here is motivated by simplicity. This method allows efficient construction of the mixture components with term count statistics readily accessible in any search engine index that includes multiple versions of documents. Other techniques for indexing versioned collections, such as extensions to inverted file formats [6] or storage of term-count deltas [5], may provide more efficient querying. We leave exploration of these alternate indexing formats as potential future work.

This construction produces three language models with non-overlapping vocabulary so that terms occurring in almost all of the time slices will be present in the  $D_L$  language model, terms occurring in 50-90% of the time slices will be present in the  $D_M$  language model, and terms occurring in less than 50% of the time slices will be present in the  $D_S$  language model.

Different formulations of the  $\phi$  functions could allow a less partitioned set of language models with shared vocabulary across the models. For example, one could use a *staying power* statistic as in [3] to identify terms likely to persist in the document vocabulary. The above formulation, however, simplifies the problem considerably, allowing several independence assumptions to be made during training of the model parameters. We leave further explorations of the form of these functions to future work.

## 6.2 Observations on the retrieval model

This retrieval model has the effect of favoring more dynamic over more static documents. Take, for example, two documents  $A$  and  $B$  of the same length,  $|A| = |B|$ . Both documents contain the query term  $q$  in all time slices with the same frequency  $\nu(q, A_L) = \nu(q, B_L) > 0$ . Document  $A$  has a large fraction of long-term static content, and document  $B$  has a small fraction of long-term static content, so that  $|A_L| > |B_L|$  and  $|A_M| + |A_S| < |B_M| + |B_S|$ . Then, by Equation 5 we have:  $P(q|A_L) < P(q|B_L)$ . Given reasonable settings of the mixing parameters in Equation 3, so that the background models  $C_M$  and  $C_S$  don't dominate the scoring function, we have  $P(q|A) < P(q|B)$ .

By separating the transient, short-lived vocabulary from longer-lived vocabulary, we effectively shorten the length of the long-term virtual document. This, in turn, increases the influence of those terms that are stable across the lifetime of the document, and likely reflective of the central topic.

## 6.3 Document Prior

Based on the previous observation that relevant documents tend to change, we may wish to bias our ranking function, independent of the query, in favor of dynamic documents. In the language modeling framework, this is done through the use of a non-uniform *document prior*. There are several ways to construct such a prior, such as the likelihood of generating a document at time  $t$  given the language model of documents at times  $t' < t$ . However, in this work we incorporate a document prior that is a simple transformation of the shingle overlap measure given in Equation 1. The change prior is given by the following, normalized to be a probability distribution across documents:

$$P_{ch}(D) \propto (ShDiff(D) + 1)^\gamma \quad (6)$$

where  $\gamma \geq 0$  is a parameter to be estimated. As  $\gamma$  approaches zero, documents are treated the same regardless of their change characteristics; when  $\gamma = 1$  this prior grows linearly; and when  $\gamma > 1$  this prior grows super-linearly with the volume of document change. This prior assigns maximal probability mass to documents that change completely at every slice in our collection. Although this case is somewhat counter-intuitive, only a small fraction of pages in our collection undergo this extreme change (see Figure 1) and the relative benefit of favoring more dynamic documents outweighs this risk, as we will show below.

## 6.4 Parameter Tuning

The models presented above require several parameters to be estimated from the training data:

For ease of fitting these parameters, we make the assumption that smoothing and mixing parameters are independent across the three mixture components. This assumption is not unreasonable, given that the three language models,  $D_L$ ,

$\mu_L, \mu_M, \mu_S$	Smoothing Parameters
$\lambda_L, \lambda_M, \lambda_S$	Mixing Weights
$\gamma$	Prior Parameter.

$D_M$  and  $D_S$  contain non-overlapping vocabulary (although the three background models  $C_L, C_M$  and  $C_S$  do overlap). Additionally, we also make the assumption that the prior parameter  $\gamma$  is independent of the others.

These assumptions allow us to fit each parameter separately, for which we use line-search. During the training phase we seek to identify parameters that maximize NDCG@1 for the single slice and dynamic models. We found that the models presented here are not sensitive to the smoothing and mixing parameters for the  $M$  and  $S$  language models. The parameters with the most influence over the model's performance are the long-term model smoothing and mixing weights,  $\lambda_L$  and  $\mu_L$ , and the prior weight,  $\gamma$ .

The learned parameters for this model are informative. First, the value of the prior parameter  $\gamma$  that maximizes performance on the training set is  $\gamma \approx 2.3$ . This indicates a very strong tendency to favor documents with a high level of content change.

Second, the smoothing parameters that maximize NDCG@1 are  $\mu_L = 5, \mu_M = \mu_S = 1500$ . This long-term model parameter is much lower than what is typically reported for Indri's smoothing on ad-hoc retrieval tasks. The value of this smoothing parameter influences how much of probability mass is assigned to "unobserved" terms in the document as compared to the terms present in the document's text [20]. A relatively lower value of  $\mu_L$  as compared to what is typical for the Dirichlet smoothing parameter indicates less of a need to smooth this model with the background model, and results in probability estimates closer to the maximum likelihood estimates for this model,  $\nu(q, D_L)/|D_L|$ .

Figure 7 shows the relationship between the long-term mixing weight  $\lambda_L$  and performance. This figure shows that  $\lambda_L \approx 0.1$  maximizes performance, and performance plateaus as  $\lambda_L$  increases. The slight decrease in performance beyond the maximum value indicates that the  $M$  and  $S$  models still have a role to play in ranking documents for navigational queries. For the experiments below, we use  $\lambda_L = 0.1$  and  $\lambda_M = \lambda_S = 0.45$ .

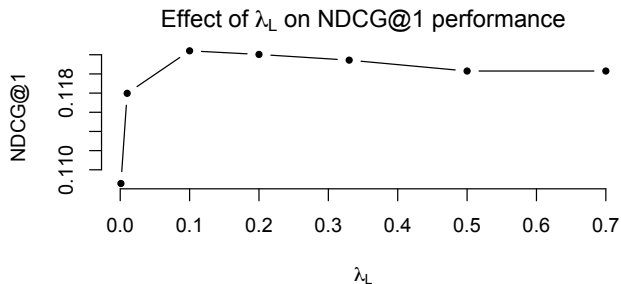


Figure 7: NDCG@1 as a function of the long-term mixing weight,  $\lambda_L$ . Remainder of mixing weight shared evenly:  $\lambda_M = \lambda_S = 0.5 \times (1.0 - \lambda_L)$

Although the value of the long-term model mixing weight,  $\lambda_L$ , that maximizes NDCG@1 is relatively smaller than the short- and mid-term model weights, the small value of the smoothing term  $\mu_L$  results in much higher probability esti-

mates for  $P(q|D_L)$ . For this reason, the long-term model typically dominates the final ranking formula.

## 7. EXPERIMENTS, RESULTS AND ANALYSIS

In this section we evaluate our retrieval model and our document change prior. As a baseline model, we use the performance of the unigram ranking model, Equation 2, averaged across all time slices. Because performance varies over the different versions of our documents, taking the expected performance over time gives a reasonable estimate of the performance at any single point in time. We refer to this baseline model as the *single slice model* below. We compare this model to the *dynamic model* presented in Section 6.1. We also study the effect of the change prior, presented in Section 6.3, on the both models. All experiments were conducted with the Indri search engine.

This baseline represents a strong content-only baseline, using the Indri search engine with parameters optimized for our dataset. In this paper we evaluate the effects of adding a temporal document prior (change prior) and/or temporal term information (dynamic model) compared with this baseline. In future work we will examine how our temporal models compare with other features commonly used in Web search engines (e.g., document priors based on link information such as PageRank or URL depth, or term weights based on click data or anchor text). Since temporal features can be used for other dynamic collections in addition to the Web which may have very different link structure or anchor text characteristics, we believe that it is important to start by understanding how temporal features contribute compared to content-only models.

As this is a high-precision web ranking task with graded relevance levels, we use NDCG at high ranks (1, 2, 3, 5 and 10) as our primary evaluation measure. All parameters in both the single slice and dynamic models are selected to maximize NDCG@1 on the training set. NDCG’s normalization serves to add a recall component to the evaluation measure, and for this reason, we also report DCG which focuses exclusively on graded precision. We are primarily interested in performance on the queries identified as navigational (Section 4.3). Complete performance results for the navigational queries are given in Table 2. Significance testing was performed with a two-sided paired t-test, using the Bonferroni correction to adjust for multiple testing of the four hypotheses. The four hypotheses involve comparing the single slice model with the dynamic model (SS vs. DY), and comparing the addition of the temporal prior,  $P_{ch}$ , (SS+ $P_{ch}$  vs. SS, DY+ $P_{ch}$  vs. DY, and DY+ $P_{ch}$  vs. SS). For completeness and as comparison, we report performance on the remainder of the query set in the text, as appropriate.

The top of Table 2 compares performance for the dynamic model and the single slice baseline model, for both the train and test sets. We see consistent and significant performance improvements for the dynamic model in both DCG and NDCG at all rank cutoffs. We see an increase of more than 5% for all positions on the training set, and more than 4% on the test set for both the NDCG and DCG. This shows significant and stable advantages for the dynamic model over the single slice model. This validates our hypothesis that navigational queries are better served by a model that can differentially weight static and dynamic content.

The bottom section of Table 2 shows the effects of adding the change prior to both the single slice and dynamic models for the navigational queries. For the training set, the addition of the prior improves performance of both the SS and DY models — we see an increase of more than 10% for all positions in both models, roughly doubling the performance improvements seen with the dynamic term model alone. On the test set, the performance improvement does not seem to generalize as well at the top rank, but is consistently significant at higher cutoffs. This small drop in performance at the first ranks may indicate the change prior too aggressively favors documents with a high volume of change, and possibly is over-fit to the training set. From the table, it is clear that the change prior is consistently effective at higher ranks (>2) on both the training and test sets.

We see significant improvements over the single slice model (SS) with both the change prior (SS+ $P_{ch}$ ) and the dynamic model (DY). These two enhancements on their own are statistically indistinguishable. But, using both together yields further significant improvements (DY+ $P_{ch}$ ). The advantages are more than 20% compared to SS on the training set and, except for position 1, also significant on the test set. Thus, these two methods for leveraging dynamic document content are complementary. The bolded entries in Table 2 indicate the best results for each level of DCG and NDCG. In all but two cases (DCG@1 and NDCG@1 in the test set), the combination of dynamic term model and change prior produces the best overall retrieval performance.

These experimental results show that significant performance gains over a strong content-only baseline on navigational queries. Using just the dynamic retrieval model results in more than a 4% improvement in performance, and this advantage is consistent for all ranks. Adding a temporal document prior to either the single slice (SS) or dynamic (DY) model provides additional retrieval benefits and these effects are most evident at higher ranks. These results are quite promising. They represent the first experiments attempting to model content dynamics in a document ranking algorithm. We describe several extensions below which we believe can further improve retrieval performance by taking into account the dynamics of document content over time.

Although our main focus is on navigational queries (and we tuned the retrieval model for such queries), we also investigated the performance of our models on the full query set. On the full query set, we see roughly a 1.3% decrease in NDCG@1 when using the dynamic model as compared to the single-slice model. The application of the change prior to the single-slice model results in approximately a 1.7% increase in performance. Neither of these differences are significant.

## 8. CONCLUSION

In this paper, we have presented the first published analysis of using document content change characteristics in relevance ranking. We demonstrated a strong relationship between content change and relevance, and have developed two methods for leveraging this in ranking algorithms. First, we developed a probabilistic retrieval model that can differentially weight terms based on their temporal characteristics. Second, we developed a query independent document prior that can be used to favor dynamic documents. Both of these independently led to significant performance improvements on navigational queries. Additionally, these two methods

	Train Set			Test Set						
	SS	DY	% over SS	SS	DY	% over SS				
nDCG@1	0.1112	0.1214	9.17**	0.1409	<b>0.1466</b>	4.04*				
2	0.1378	0.1453	5.44*	0.1611	0.1700	5.53*				
3	0.1591	0.1692	6.35***	0.1827	0.1907	4.36*				
5	0.1921	0.2045	6.45***	0.2126	0.2220	4.43**				
10	0.2524	0.2693	6.70***	0.2766	0.2903	4.94***				
DCG@1	1.6686	1.8216	9.17**	2.1137	<b>2.2004</b>	4.10*				
2	2.7475	2.9033	5.67*	3.2786	3.4454	5.09*				
3	3.5871	3.8210	6.52***	4.2046	4.3725	3.99*				
5	4.9106	5.2359	6.62***	5.5329	5.7768	4.41**				
10	7.4550	7.9617	6.80***	8.2670	8.6976	5.21***				
	SS+ $P_{ch}$	% over SS	DY+ $P_{ch}$	% over DY	% over SS	SS+ $P_{ch}$	% over SS	DY+ $P_{ch}$	% over DY	% over DY
nDCG@1	0.1275	14.70**	<b>0.1368</b>	12.68*	23.02**	0.1338	-5.08*	0.1397	-4.68*	-4.68*
2	0.1558	13.06***	<b>0.1692</b>	16.47**	22.81***	0.1670	3.66*	<b>0.1759</b>	3.45*	3.45*
3	0.1767	11.06***	<b>0.1929</b>	14.01***	21.25***	0.1955	6.99*	<b>0.2049</b>	7.44**	7.44**
5	0.2144	11.60***	<b>0.2354</b>	15.09***	22.52***	0.2404	13.10**	<b>0.2470</b>	11.27**	11.27**
10	0.2796	10.79***	<b>0.3076</b>	14.21***	21.85***	0.3143	13.60***	<b>0.3177</b>	9.43***	9.43***
DCG@1	1.9131	14.65**	<b>2.0519</b>	12.64*	22.97**	2.0065	-5.07*	2.0960	-4.75*	-4.75*
2	3.1140	13.34***	<b>3.3757</b>	16.27**	22.86***	3.3876	3.33*	<b>3.5334</b>	2.55*	2.55*
3	4.0078	11.73***	<b>4.3810</b>	14.65***	22.13***	4.4823	6.60*	<b>4.6786</b>	7.00**	7.00**
5	5.5340	12.69***	<b>6.1267</b>	17.01***	24.76***	6.3244	14.31**	<b>6.4584</b>	11.80**	11.8**
10	8.3868	12.50***	<b>9.2854</b>	16.63***	24.55***	9.6036	16.17***	<b>9.6536</b>	10.99***	10.99***

**Table 2: Full Performance Results for Navigational Queries.** Top shows Single Slice Model (SS) vs. Dynamic Model (DY). Bottom shows the effect of Prior ( $+P_{ch}$ ) on both models. \*, \*\*, \*\*\* indicate differences at the 0.05, 0.01 and 0.001 adjusted significance levels with a two-sided paired t-test. Bolded entries indicate the best result for each level of DCG and NDCG.

are complimentary, and when used together yield further performance improvements.

These results show that for navigational queries, two aspects of document dynamics are significant for relevance ranking. The effect of the document change prior ( $P_{ch}$ ) indicates that favoring dynamic pages in relevance ranking can lead to performance improvements. The effect of the dynamic ranking model (DY) indicates that favoring static content within those pages also improves performance. These two effects are different and complementary, with their combination ( $DY + P_{ch}$ ) yielding the best retrieval performance.

These results demonstrate that document dynamics can play a significant role in relevance ranking. The models here are only a first step in understanding the complex interaction between document retrieval algorithms and content dynamics. Several areas of this research open the door to further experimentation with ranking algorithms that are sensitive to dynamic documents. Other methods of modeling stable and dynamic content in documents, for example, may lead to richer language models which can be tailored towards different types of information needs. Similarly, alternative methods for using temporal change patterns to develop document priors could be explored.

This work focused on navigational queries, but other information needs may also be appropriate to study in the context of document dynamics. Query volume often follows bursty or cyclical patterns, corresponding to public interest in news or recurring events. Understanding the relationship between query dynamics and document dynamics could lead to further insight into how document change can be leveraged in ranking algorithms. As noted earlier, there are some interesting challenges in developing test collections for this since relevance judgments may change over time.

There are also several interesting systems-related issues in

representing document change over time. In the research reported in this paper, we simply saved multiple versions of the collection, but this may not be feasible in all settings. Developing methods for identifying sufficient statistics, optimal sampling frequencies, etc. are important research directions.

Although we have focused on web documents in our experiments, these techniques are equally applicable to any document collection with similar temporal dynamics. Versioned collections such as Wikis are a prime example, where documents can undergo continuous editing and revision. Corporate intranet environments also frequently contain documents that undergo periodic changes, but often lack the rich link structure, descriptive anchor text, and large amount of user interaction data that are available on the web and important in web ranking algorithms. These environments may also benefit from retrieval models that are sensitive to term dynamics, like the ones presented here.

## 9. REFERENCES

- [1] E. Adar, M. Dontcheva, J. Fogarty, and D. S. Weld. Zoetrope: Interacting with the ephemeral web. In *UIST '08: Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, pages 239–248, New York, NY, USA, 2008. ACM.
- [2] E. Adar, J. Teevan, and S. T. Dumais. Resonance on the web: Web dynamics and revisitation patterns. In *CHI '09: Proceedings of the 27th International Conference on Human Factors in Computing Systems*, pages 1381–1390, New York, NY, USA, 2009. ACM.
- [3] E. Adar, J. Teevan, S. T. Dumais, and J. L. Elsas. The web changes everything: Understanding the dynamics of web content. In *WSDM '09: Proceedings of the Second ACM International Conference on Web*

- Search and Data Mining*, pages 282–291, New York, NY, USA, 2009. ACM.
- [4] O. Alonso and M. Gertz. Clustering of search results using temporal attributes. In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 597–598, New York, NY, USA, 2006. ACM.
- [5] P. Anick and R. Flynn. Versioning a full-text information retrieval system. In *SIGIR '92: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 98–111, New York, NY, USA, 1992. ACM.
- [6] K. Berberich, S. Bedathur, T. Neumann, and G. Weikum. A time machine for text search. In *SIGIR '07: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 519–526, New York, NY, USA, 2007. ACM.
- [7] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. *Computer Networks and ISDN Systems*, 29(8-13):1157–1166, 1997.
- [8] J. Cho, S. Roy, and R. E. Adams. Page quality: In search of an unbiased web ranking. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, pages 551–562, New York, NY, USA, 2005. ACM.
- [9] F. Diaz and R. Jones. Using temporal profiles of queries for precision prediction. In *SIGIR '04: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 18–24, New York, NY, USA, 2004. ACM.
- [10] D. Fetterly, M. Manasse, M. Najork, and J. Wiener. A large-scale study of the evolution of web pages. In *WWW '03: Proceedings of the 12th International Conference on World Wide Web*, pages 669–678, New York, NY, USA, 2003. ACM.
- [11] M. Herscovici, R. Lempel, and S. Yogev. Efficient indexing of versioned document sequences. *Lecture Notes in Computer Science*, 4425:76–87, 2007.
- [12] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.
- [13] A. Jatowt, Y. Kawai, and K. Tanaka. Visualizing historical content of web pages. In *WWW '08: Proceeding of the 17th International Conference on World Wide Web*, pages 1221–1222, New York, NY, USA, 2008. ACM.
- [14] R. Jones and F. Diaz. Temporal profiles of queries. *ACM Transactions on Information Systems (TOIS)*, 25(3):14, 2007.
- [15] U. Lee, Z. Liu, and J. Cho. Automatic identification of user goals in web search. In *WWW '05: Proceedings of the 14th International Conference on World Wide Web*, pages 391–400, New York, NY, USA, 2005. ACM.
- [16] X. Li and W. B. Croft. Time-based language models. In *CIKM '03: Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pages 469–475, New York, NY, USA, 2003. ACM.
- [17] A. Ntoulas, J. Cho, and C. Olston. What’s new on the web?: The evolution of the web from a search engine perspective. In *WWW '04: Proceedings of the 13th International Conference on World Wide Web*, pages 1–12, New York, NY, USA, 2004. ACM.
- [18] P. Ogilvie and J. Callan. Combining document representations for known-item search. In *SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 143–150, New York, NY, USA, 2003. ACM.
- [19] C. Olston and S. Pandey. Recrawl scheduling based on information longevity. In *WWW '08: Proceeding of the 17th International Conference on World Wide Web*, pages 437–446, New York, NY, USA, 2008. ACM.
- [20] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214, 2004.
- [21] R. Zhang, Y. Chang, Z. Zheng, D. Metzler, and J.-y. Nie. Search result re-ranking by feedback control adjustment for time-sensitive query. In *NAACL '09: Proceedings of the 2009 Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 165–168, Morristown, NJ, USA, 2009. Association for Computational Linguistics.