How Fresh Do You Want Your Search Results?

Shiwen Cheng, Anastasios Arvanitis, Vagelis Hristidis
Department of Computer Science & Engineering
University of California, Riverside, California, USA
{schen064, tasos, vagelis}@cs.ucr.edu

ABSTRACT

Researchers have recognized the importance of utilizing temporal features for improving the performance of information retrieval systems. Specifically, the timeliness of a web document can be a significant factor for determining whether it is relevant for a search query. Previous works have proposed time-aware retrieval models with particular focus on news queries, where recent web documents related with a real-world event are generally preferable. These queries typically exhibit bursts in the volume of published documents or submitted queries. However, no work has studied the role of time in queries such as “credit card overdraft fees” that have no major spikes in either document or query volumes over time, yet they still favor more recently published documents. In this work, we focus on this class of queries that we refer to as “timely queries”. We show that the change in the terms distribution of results of timely queries over time is strongly correlated with the users’ perception of time sensitivity. Based on this observation, we propose a method to estimate the query timeliness requirements and we propose principled ways to incorporate document freshness into the ranking model. Our study shows that our method yields a more accurate estimation of timeliness compared to volume-based approaches. We experimentally compare our ranking strategy with other time-sensitive and non time-sensitive ranking algorithms and we show that it improves the results’ retrieval quality for timely queries.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*

Keywords

Recency Query; Results Freshness; Web Search

1. INTRODUCTION

Previous works have shown that timeliness is a key aspect for determining the relevance of a web document for a search query [1, 2, 3, 4]. For instance, a user who issues a query for “US elections” in November 2012, is more likely to be interested for web pages related to the 2012 elections. Although there exist several web pages about previous elections that are highly topically relevant to the user’s query, it is expected that boosting the more recent documents will improve the retrieval quality. Under this assumption, several techniques have been proposed to incorporate the time dimension in the ranking model. Previous approaches can be broadly categorized into: *(i)* those that focus on news queries related with real-world events, favoring recent documents [5, 6, 7, 3], and *(ii)* those that target general time-sensitive queries, where the results are preferably published during a specific time range [2, 4].

For both query types, it has been found that the most relevant time range is usually associated with spikes in the number of related user queries, or in the volume of related documents published during that time frame. Based on this observation, current approaches analyze the time series of related queries or documents, in order to *(i)* classify queries as news queries or not [6] and *(ii)* to identify important time periods [2, 4]. Thereby, they boost the ranking of documents published around that time frame. For instance, a user searching for “Boston Marathon” might be looking for information about the terrorist attack that took place during the marathon on April 15, 2013; then documents published around that date are promoted.

For a large number of search queries however, recent studies [8, 9] have shown that, while freshness of query results is still very important, their query or document time series are either unclear or misleading. In particular, several queries exhibit no, multiple, random or periodic spikes in query popularity over a time period [8]. For instance consider a query like “credit card overdraft fees”. For such a query, the number of published documents and/or search queries remains more or less constant over time, normalized with the total query volume. However, more recent documents are clearly more relevant because, due to policy changes on behalf of banking institutions, information contained in older documents might no longer be valid. Other queries, exhibit a seasonal or unpredictable rise in popularity. A query such as “tax preparation tips” usually has a burst in popularity around mid April every year; on the other hand a query for “Rihanna new single” follows a more irregular trend pattern.

In general, for this type of queries, all other things being equal, a fresher document is probably much more relevant for a user’s search query. However, existing volume-based techniques cannot be applied for such queries, since the number of published documents or issued queries cannot be leveraged as an indication of users’ interests. In fact, the number of published documents or search queries might as well be negatively correlated with the users’ demands. For instance consider a query such as “long distance phone calls prices” or “fashionable haircuts”. Figure 1 shows the number of queries in a commercial search engine over years, normalized by the total query traffic during each time period, for the query “fashionable haircuts”.

CIKM’13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.
Copyright 2013 ACM 978-1-4503-2263-8/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505515.2505696.

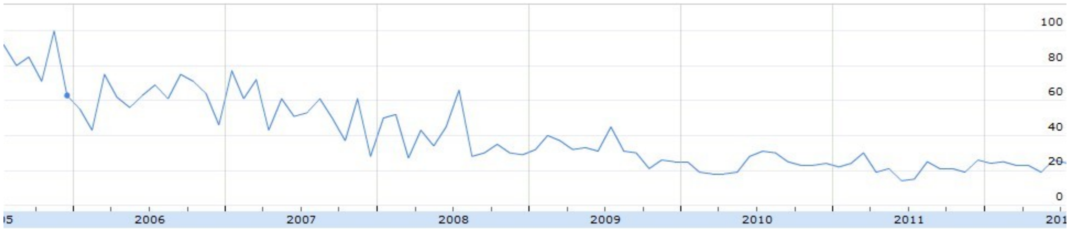


Figure 1: Number of search queries for “fashionable haircuts” over time

As depicted, user searches for this query have dropped during the recent years. At the same time, the keyword “fashionable” indicates that users are interested on new hairstyles and that web pages related to older ones are no longer relevant, which indicates that it is a time-sensitive query. For this query, an approach that focuses only on the query or document volume might instead give higher ranking to older documents.

Moreover, different queries can have very different degree of freshness requirements. For example for a query like “smartphone reviews”, users might consider as relevant a document that is up to 6 months old, whereas for a query like “new movies in theaters” the time range of interest is much shorter, typically 1-2 weeks. For the reasons mentioned above, volume-based approaches will not be able to capture different freshness requirements. Further, in addition to identifying the timely queries, we must also quantify how timely a query is.

In this work we will focus on this important type of queries, which we refer to as *timely queries*. Timely queries have the following properties: *(i)* the interest on the query from document publishers or consumers does not show significant variance over time (normalized by the total query traffic or document volume respectively), and *(ii)* more recently published documents are strongly preferred over older ones. Since, the popularity of such queries can be steady, previous approaches are not effective.

Motivated by the above challenges, in this paper we propose a different approach to measure the freshness requirements of a user’s query, i.e., the *query timeliness*. In particular, we argue that the users’ freshness requirements from search results are strongly correlated with the degree of content change in the relevant documents. In other words, if the terms distribution inside the most relevant documents changes significantly over time, this indicates that older documents become stale shortly, and hence, they should be penalized with lower relevance scores.

In this paper we experimentally confirm this correlation. That is, if we identify significant content change in the relevant documents across time, then time becomes a major factor in our proposed ranking. We incorporate the query timeliness in our retrieval model in a principled manner by extending previous works on time-based language models [1, 3].

Note that, since our ranking model always favors more recent documents, our approach is not meant to handle news queries related with events that took place in the past. For example, for a query like “Supreme Court healthcare act”, results around June 28th 2012 are more relevant since this is when the Supreme Court ruled.

Contributions. The contributions of this paper can be summarized as follows:

- We show that the change in the terms distribution of results of timely queries over time is strongly correlated with the users’ perception of time-sensitivity.
- We propose principled ways to incorporate document freshness into the ranking model.
- We experimentally show that our proposed model improves the quality of the results for timely queries.

Outline. Section 2 presents related work on temporal ranking of search results. In Section 3 we discuss query timeliness and we provide a method to measure timeliness based on the terms distribution. In Section 4 we present our time-aware ranking model by incorporating the concept of query timeliness. Section 5 contains an experimental evaluation of our methods. In Section 6 we discuss how our proposed model can be applied in practice. Finally, in Section 7 we draw conclusions and sketch our future work.

2. RELATED WORK

Li and Croft [1] were the first that proposed a time-based language model to boost the ranking of more recent results. In particular, they introduced an exponential decay prior to the query likelihood language model, such that more recent documents are assigned a higher probability. Based on the proposed ranking model, experiments show significant improvements in retrieval quality for TREC queries that are related with recent events.

However, their proposed ranking model treats all recency queries as having the same freshness requirements from the search results. That is, they boost the ranking of recent results uniformly for all queries. Efron and Golovchinsky [3] improve upon this model [1] by proposing a query-specific exponential reranking method. In particular, they calculate a maximum likelihood estimator per query based on the time distribution of the most relevant results, as returned by a non time-aware ranking. A significant limitation with such an approach is that it will boost recent documents, only as long as the document volume increases over time, i.e., it can be applied mainly on news queries and not generally on time-sensitive ones. In contrast, in this paper, we provide a ranking model that can be used for different query types, which is not dependent on the distribution of documents over time or the underline (non time-sensitive) ranking. We experimentally compare to both methods [3, 1] in Section 5 and we show that our method yields better retrieval quality for timely queries.

Some works [2, 4] focus on more general time-sensitive queries, where the results are preferably published during a specific time range. Jones and Diaz [2] propose building a time series on the number of top ranked documents of the query. According to the detected number of spikes in the time series, they classify queries into three classes: *atemporal*, *temporally ambiguous* and *temporally unambiguous*, which represent queries that exhibit no spike, only one spike and more than one spikes in their document volumes, correspondingly. The class of queries that we study in this paper would be classified as atemporal by [2], since these queries do not exhibit major volume spikes (see Section 5.2.2). Therefore if we followed this approach, the freshness of the results would not be considered as important. Dakka et al. [4] propose alternative methods to learn the most relevant time period for a query, and present solutions to incorporate temporal relevance into several popular ranking algorithms. Both [2, 4] rely on the spikes in the distribution of relevant documents.

It has been empirically shown [1, 3] that applying a time-aware ranking model can generally harm the retrieval quality of non time-sensitive queries. In order to address this problem, Dai et al. [10] introduce a machine learning framework for simultaneously optimize both relevance and freshness of results, by utilizing both temporal and non-temporal document features.

Another approach is to automatically identify whether a search query is time sensitive or not. If a query is not time-sensitive, standard relevance methods can be used instead. Dong et al. [6] use machine learning techniques to classify a query as breaking news query or not. For this purpose, they measure the difference in the query probabilities in various time slots in the past, such as the last day, last week and last month. The probabilities are calculated based on the language model of both the query log and the document collection. If the query is classified as a breaking news query, the freshness of a document becomes important in ranking. Similarly to some of the above methods [2, 4], the underlying intuition is that in a specific time range, the results are much different from a regular search, for instance due to a burst in the number of relevant documents or due to the query being popular in the query stream. Similarly, using features such as the probabilities of the query generated from recent content, Styskin et al. [11] train a linear regression model to predict a probability for the query's freshness preference and they combine fresh documents with regular ones in order to enhance the temporal diversity of the results.

Assuming a news query, several works deal with how to improve the temporal relevance of returned results. Dong et al. [7, 12] extend former work [6] by enriching the results with documents discovered in the Twitter stream. For this purpose, they extract a set of features from both a regular documents' corpus and a tweets collection, and they learn a ranking model in order to merge recent tweets with regular results. Diaz [5] and König et al. [13] study the utility of showing news results among regular ones by using click-through data.

Elsas et al. [14] study the temporal factors of *navigational* queries, where there is usually a small number of highly relevant documents that are consistently relevant across time. For this type of queries, they experimentally show that there is a strong positive correlation between the relevance of a document and the frequency of the document's content change. However, within the same document, terms that are present across different time ranges are more important in estimating the overall document's relevance. Thereby, they propose the use of a document-specific prior in order to favor more dynamic documents. Our work has a similar motivation, i.e., to leverage the amount of content change in recency ranking. However, we focus on more general informational or transactional time-sensitive queries where more recent (and not necessarily highly dynamic) documents are preferable. Further, we measure the content change in the query rather than in the document level.

3. ESTIMATING THE QUERY TIMELINESS

As we already noted, different queries can have very different timeliness degrees, i.e. requirements on the freshness of the search results. For instance, for a query such as "top graduate schools", a user might find results up to two years old as relevant, whereas for a query such as "Universal Studios coupon", she would be interested only on search results of the last few weeks, since older coupon offers will probably have already expired. Thus, the challenge is how to identify the appropriate timeliness degree for a given search query.

Previous works have used the query volume and the number of published documents as an indicator of the timeliness of a query. The most relevant work to our problem [3] computes a query-specific freshness parameter that is calculated based on the distribution of the publication times of the top k results that would be returned by a non time-sensitive ranking function. However, as we will demonstrate in our experiments in Section 5, the proposed approach [3] fails for the class of queries that we study in this work, i.e., those having a relatively steady document volume, such as the one shown in Figure 1.

In order to overcome this problem for the class of queries (timely query) we study in this paper, we introduce a new method to estimate query timeliness. In particular, we propose to use the degree of change in the content of the most relevant documents of a query, as a measure of the timeliness of the query. Kullback-Leibler (KL) divergence [15] is a popularly used measure to compute the divergence of text documents [16, 17]. In this paper, we apply KL divergence to measure the changes of text documents, i.e. we calculate the difference in term probability distributions of text documents using the KL divergence:

$$KL(P,R) = \sum_{t} P(t) \log \frac{P(t)}{R(t)} \tag{1}$$

where P and R are two probability distributions and $P(t)$, $R(t)$ denote the probability of term t in distributions P and R, respectively.

For a query Q, we define the degree of content change between two time slots as the difference in the probability distributions between the sets of documents relevant to Q in these time slots. For simplicity, hereafter we assume discrete time slots, which might denote weeks, months etc. Further, let T_i represent the set of documents that are relevant for query Q, and were published during time slot t_i (e.g., during July 2012). We will also symbolize as $LM(T_i)$ the language model produced by T_i. Then, we define the degree of content change between two time slots t_i, t_j as $KL(LM(T_i), LM(T_j))$.

Assuming n consecutive time slots $t_1, \cdots, t_n$, we define the terms distribution change for a query Q, denoted as $TDC(Q)$ as:

$$TDC(Q) = \frac{1}{n-1} \sum_{i=1}^{n-1} KL(LM(T_i),\ LM(T_{i+1})) \tag{2}$$

i.e., we take the average KL-divergence acquired from consecutive pairs of time slots. For calculating $LM(T_i)$ we will apply a unigram language model approach.

Several other measures have been proposed in order to quantify the amount of content change (the opposite of similarity) between documents, such as the cosine similarity, Dice similarity, and Jaccard distance.

Previous work [17] evaluated the effectiveness of these measures for computing the similarity of text documents for document clustering. The results have shown that all measures deliver similar results with KL divergence, and that differences depend on the particular characteristics of the document collection. Potentially, any of this measures could be used in our model to replace KL divergence.

4. INCORPORATING FRESHNESS INTO THE RANKING MODEL

In developing our time-aware ranking, we follow the language ranking model [18]. Li and Croft [1] were the first that proposed a ranking that incorporates a temporal dimension into the language model. According to the query likelihood approach, the probability that a document d is relevant to a query Q, $P(d|Q)$, is proportional to *(i)* the probability of deriving Q based on the language model of

d, termed as $P(Q|d)$ and *(ii)* an apriori probability of document d that depends on the publication date T_d, termed as $P(d|T_d)$:

$$P(d|Q) \propto P(Q|d) \cdot P(d|T_d) \tag{3}$$

In particular, in order to compute $P(d|T_d)$, they assume an exponential decay calculated as:

$$P(d|T_d) = \lambda e^{-\lambda \cdot \Delta t_d} \tag{4}$$

where Δt_d is the normalized age of document d, measured as the time distance between T_d and the date of the most recent document in the document collection. Note that Li and Croft [1] use the same freshness parameter λ for a set of queries that is manually identified as time-sensitive, regardless of the different degrees of timeliness requirements each query has. Thus, Efron et al. [3] improve [1] by proposing to use a query-specific λ_Q that is calculated based on the time distribution of relevant documents as returned by a non time-sensitive ranking model.

In this work we use Equations 3 and 4 as a starting point, and we modify them in order to consider the right amount of freshness for each query using the timeliness requirement estimation method proposed in Section 3. Specifically our ranking model assigns scores based on the following function:

$$Score(d,Q) = BM25(d,Q) \cdot \lambda_Q e^{-\lambda_Q \cdot \Delta t_d} \tag{5}$$

where $BM25(d,Q)$ denotes the ranking score of document d for a query Q by the popular ranking function Okapi BM25 [19], which is based on the probabilistic model. Based on our ranking model, λ_Q depends on the timeliness requirements of each query as measured by the amount of content change (Section 3). Intuitively, we would assign larger values of λ_Q for queries having higher timeliness degrees, as predicted by $TDC(Q)$. Larger values for λ_Q will result in penalizing the scores for older documents, thus favoring the most recent ones. We calculate λ_Q as:

$$\lambda_Q = \alpha \cdot (1 - e^{-TDC(Q)}) \tag{6}$$

where $\alpha > 0$ is a parameter of the ranking model and $1 - e^{-TDC(Q)}$ is the KL divergence score normalized in $(0,1]$[1]. We will assume a constant value of α for all queries. In Section 5.3 we provide more details on how we set α for experiments; in Section 6 we explain how α can be set up in practice.

Since the calculation of λ_Q is based on $TDC(Q)$, we will refer to the ranking model in Equation 5 as the *Timeliness-Aware Ranking* (*TAR*). In Section 5.3 we experimentally evaluate the retrieval quality of the proposed ranking function with the previously proposed time-aware ranking models [1, 3].

5. EXPERIMENTAL EVALUATION

In this section, we provide experimental results on the ranking quality of our methods. We first present the datasets that we used for our experiments in Section 5.1. Sections 5.2 and 5.3 contain the experimental evaluation of the proposed timeliness estimation method and timeliness-aware ranking respectively.

5.1 Datasets

Query workload. In order to build our query workload we considered the Text REtrieval Conference (TREC) datasets (e.g. TREC Web Tracks [20]). Unfortunately most of the available datasets contain only a few timely queries. Thus, we manually created some additional queries that we considered as timely, following an approach similar to [3]. In particular, we asked 10 graduate students to suggest queries having diverse timeliness requirements. The complete query workload consists of 119 queries (taken from TREC or proposed by students) and is available at [21], in which we specified the queries from TREC.

Documents. For our experimental evaluation we also needed documents published during different time ranges and relevance judgements that take into account both the topical and the temporal relevance of results; however these data where not available in TREC dataset. Hence, we constructed our document collection by submitting each of the queries to a commercial search engine and we conducted a user relevance study as explained in Section 5.2.1.

In order to collect documents published at various time ranges, we specified different start and end dates in our search, such that the returned results contain only documents that were published during the specified time frame. Note that accurately identifying the publication date of each document is itself a challenging task [22, 6, 23]; moreover often the publication time and the time associated with the content contained in a document might differ. The search engine that we used for our experiments tries to estimate the publication date for each web page by using features such as the date when it was first crawled, or a byline date or an explicitly specified date of a news article or blog post if such information is available. For simplicity, hereafter we will assume that all documents returned by the search engine have been published during the specified time period. Following this method, we retrieved the top 400 results per year for years 2007-2011 and for the first half of 2012, i.e., we obtained for each query 2400 documents in total.

5.2 Estimation of the Query Timeliness

In the first set of experiments we compare the performance of our proposed method for estimating the query timeliness with the previous approaches that focused on the document [2, 6, 3, 4, 24] and query volume change [6] as discussed in Sections 2 and 3.

5.2.1 User Survey on Query Timeliness

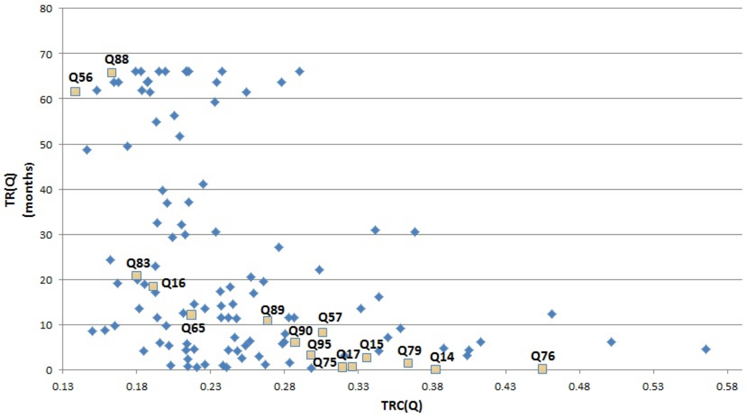
In order to calculate the degree of correlation between the document volume and the users' perception of query timeliness, we set up a user survey to collect judgments w.r.t. the timeliness demands for each query on behalf of the users. In addition to the 10 graduate students, our survey's subjects include users recruited through the Amazon Mechanical Turk [25]. In particular, we forwarded all queries to the graduate students, and 60 queries to each Amazon Mechanical Turk worker (110 workers in total). For each query, we asked the users to select among the following five options: *no time preference, up to 2 years old, up to 6 months old, up to 1 month old, up to 1 week old* the one that best describes their preferences in terms of freshness of the search results. In order to increase the quality of the user survey: *(i)* we disqualified low-quality workers from our experimental study as explained in the Appendix, and *(ii)* we filtered out the 5 highest and the 5 lowest outlier timeliness values for each query.

For each query we collected 10 timeliness judgments from students and 20 valid timeliness judgments from Amazon Mechanical Turk workers. Next, for each query we calculated the average Timeliness Requirement in months. For this purpose we mapped each label to a specific number in months that represents the respective timeliness class. Since the maximum age of any document in our collection is 5.5 years, we mapped each label for *no time preference* to 66 months. Similarly we mapped the other timeliness ranges to 24, 6, 1 and 0.25 months respectively. Then we took the average over all judgments, which we will refer to as $TR(Q)$. Figure 2a shows some representative queries, along with the respective aver-

[1] We use a normalized KL divergence score since the original KL divergence is unbounded.

#	Query	TR(Q)	TDC(Q)
Q88	public speaking tips	66.000	0.163
Q56	interview thank you letter	61.800	0.138
Q83	passport renewal	21.050	0.180
Q16	cancel a new car contract	17.158	0.193
Q65	low income housing	12.421	0.217
Q89	reality TV stars	11.158	0.267
Q57	keyboard reviews	8.474	0.306
Q90	retail sales index	6.316	0.287
Q95	smartphone reviews	3.500	0.298
Q15	California state parks jobs	2.842	0.335
Q79	newest tablet	1.670	0.363
Q17	celebrity gossips	0.868	0.326
Q75	NBA game schedule	0.838	0.319
Q76	NBA scores	0.408	0.454
Q14	California lottery results	0.288	0.382

(a)



(b)

Figure 2: TR(Q) vs. TDC(Q) based on timeliness judgments from students and Amazon Mechanical Turk workers

age timeliness requirements, as specified by the users. The query ids are taken from the complete list of the query workload [21].

As shown, users have very low freshness requirement for queries such as "public speaking tips" and "interview thank you letter". On the other hand, according to our user survey, users find the results published in the last 1-2 years for "passport renewal", "cancel a new car contract", or "low income housing" as relevant. Queries such as "retail sales index" have relevant results published in the last 6 months. Further, results published during the last month are considered as relevant for queries such as "newest tablet" or "celebrity gossips". Finally, for other queries such as "NBA scores" or "California lottery results" the relevant documents typically change per week, and users are looking for up-to-date information, as it is confirmed by the user survey.

5.2.2 *Timeliness Estimation based on Volume-based Approaches*

Studying the Document Volume Change. We first examine to what extent previous approaches [2, 6, 3, 4, 24] can predict the timeliness of timely queries. For this experiment, for each query in our workload, we issued a web search where we also specified an one month time range. For each query, we retrieved the number of documents returned by the search engine for each month range, for each of the last 66 months (5 × 12 months for years 2007-2011 and 6 months for 2012).

Since the size of the web grows over time, we need to normalize the total number of documents per month, with the size of the web at the time. Since the total size of the (visible) web is unknown, we assume that its size can be approximated by a search query that returns as many relevant documents as possible. In particular, we issued a set of stopwords queries[2] and we calculated the average number of returned documents over all issued queries. Then, we used this number as an approximation of the size of the web for a specified time range. Finally, we normalized the document volumes for each query with the number of results of the stopwords query for this month, as shown in Figure 3. Figure 4 plots the normalized document volumes for some representative queries.

Anecdotal Examples. The queries "NBA lockout" and "Occupy Wall Street" are news queries that were not included in our query workload. The other two queries "MySQL cluster setup" and "Firefox updates" are from our query workload. By comparing the document volume time series of the news queries with our timely queries, we can observe that news queries have a peak during the specific time when the news event happened. For instance, the time series line for "Occupy Wall Street" suddenly hikes during the end of 2011, whereas the time series for "NBA lockout" exhibits an increase on document volume during the second half of 2011. In contrast, the time series of the document volumes for timely queries might not have any significant spikes. For instance, "MySQL cluster setup" query does not have any spikes and "Firefox updates" has spikes with insignificant variance.

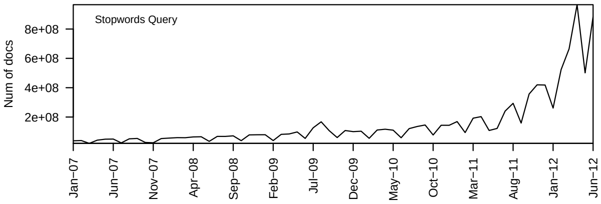


Figure 3: Time series of the number of documents returned for stopwords queries from Jan 1st, 2007 to Jun 30th, 2012

Correlation of Timeliness to Document Volume Change. In order to calculate the correlation between the document volume change and query timeliness we followed two different approaches based on how we measure the volume change.

First, we used the number of documents published during each of the 66 monthly slots that we considered in our experiments. Then, for each consecutive pair of months we calculated the absolute change in volumes. We also calculated the average number of documents per month, and we used it in order to normalize the differences between months. Finally, we calculated the Pearson correlation coefficient between the normalized document volume changes and TR across all queries in our query workload. The calculated Pearson correlation coefficient is -0.298. Note that the computed correlation value is negative because larger document volume changes result in more rapid change of the relevant information w.r.t. a query, which means that only the most recent documents should be considered as relevant. In that case the query would have a lower average timeliness value $TR(Q)$.

As a second measure, for each of the examined 119 queries we calculated the coefficient of the variation of its monthly time series, as:

$$CV = \frac{\sigma}{\mu} \tag{7}$$

where σ and μ represent the standard deviation and mean of each time series. Similarly, we calculated the Pearson correlation coefficient between CV and TR across all queries and we found a correlation equal to -0.281.

[2] Each query consists of one of the following stopwords: *a, the, and, to, of*

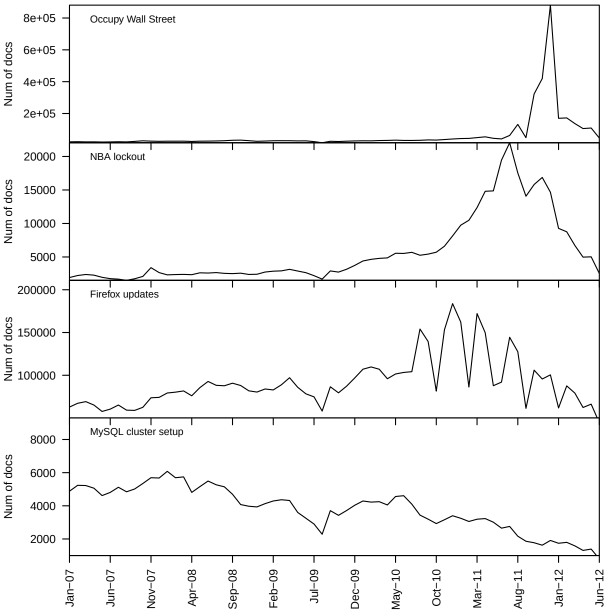


Figure 4: Time series of normalized documents number returned for several queries from Jan 1st, 2007 to Jun 30th, 2012

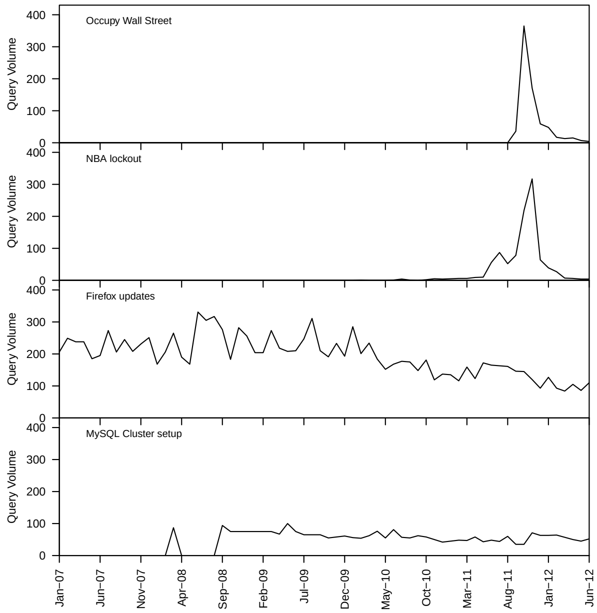


Figure 5: Time series of normalized query volume for several queries from Jan 1st, 2007 to Jun 30th, 2012, data from Google Insights [26]

Studying the Query Volume Change. Next, we examined the degree of correlation between the query volume change and the users' perception of timeliness. We built a time series of query volumes, by using the service provided by Google Insights [26]. We issued each query by specifying a date range of 5.5 years. Google Insights could provide monthly query volumes only for 87 out of the 119 queries issued when we collected these data. The results provided by Google Insights are already normalized with the size of the query traffic on Google. Figure 5 shows the time series constructed for several representative queries.

Anecdotal Examples. As shown, the query volume time series of "Occupy Wall Street" and "NBA lockout" have quite different behavior compared with the other two queries taken from the query workload. Also note that the hikes in query volumes of "Occupy Wall Street" and "NBA lockout" are consistent with the hikes of their document volumes in Figure 4.

Correlation of Timeliness to Query Volume Change. Similarly to how we calculated the timeliness estimation quality for the documents volume time series, we correlate (i) the normalized query change, and (ii) the coefficient variation of the query volumes time series of each query with the average timeliness requirement TR. The Pearson correlation coefficient calculated over the 87 queries was -0.132 using the normalized query volume change, and -0.130 using the coefficient of the variation.

Discussion. The relatively low correlation of both approaches shows that methods that leverage the document or query volume change are not suitable for timely queries, such as "MySQL cluster setup" and "Firefox updates", but can only be applied on news queries. Note that in addition to monthly, we also tried other range lengths with similar results.

5.2.3 Timeliness Estimation based on Terms Distribution Change

We now evaluate our method for estimating the timeliness of a query based on the terms distribution change in its relevant documents as presented in Section 3.

As discussed in Section 5.1, we collected 2400 documents per query over a period of 5.5 years. We created 6 document collections, where each collection contains all 400 documents retrieved for the corresponding time slot t_i. In order to build a language model for each document collection we concatenated all 400 documents into a single document T_i. Retrieving the relevant textual content from the HTML body of each document is a challenging and error-prone task [27]. In order to address this problem, mainly for performance reasons, we only considered the titles and snippets of each retrieved document.[3] Finally, when building each language model $LM(T_i)$, we ignored all common stopwords and all terms occurring fewer than 3 times over different time slots (we assumed that they are either typos or irrelevant terms). Then, based on the definition of $TDC(Q)$ in Equation 2, we calculated the terms distribution change for a query Q as:

$$TDC(Q) = \frac{1}{5}\sum_{i=2007}^{2011} KL(LM(T_i),\ LM(T_{i+1})) \tag{8}$$

Finally, we calculated the Pearson correlation coefficient between $TDC(Q)$ and $TR(Q)$ for all the queries of our workload and we got a correlation score -0.427. This score indicates that there is a strong correlation between the terms distribution change of the documents content and the users' perception of query timeliness. Further, it is much higher compared to the volume-based approaches that we presented in Section 5.2.2.

Note that instead of using one year as a unit to define the time slots, we also experimented with different time units. Because of the time range that we specified (5.5 years), there are not sufficient data in order to use a 2-year unit. Thus, we tried to use 1 week, 1 month, and 6 months units to build the language model $LM(T_i)$ for the most recent 10 time slots, and applied a similar method as above to calculate $TDC(Q)$. The Pearson correlation coefficients between $TDC(Q)$ and $TR(Q)$ based on 1 week, 1 month, and 6 months time slots are -0.298, -0.357 and -0.413 respectively, which

[3] Using the title and snippet instead of the textual content of a document can improve the efficiency and sometimes also the quality for some applications like search results clustering [28, 29] and query classification [30].

are all higher than volume-based approaches in Section 5.2.2. As the $TDC(Q)$ score from yearly time slots yields the highest prediction quality, we will use the $TDC(Q)$ results from yearly time slots in the following experiments.

Figure 2b plots the $TDC(Q)$ and $TR(Q)$ values for all queries in our workload. Some representative queries (those shown in Figure 2a) are labeled with different symbols and numbers. For instance, for query Q88: "public speaking tips" and Q56: "interview thank you letter", the relevant documents do not vary largely over time. Thus, the $TDC(Q)$ scores computed for these two queries are relatively small. According to our user survey, users roughly prefer the results published in the last two years for Q83: "passport renewal" and Q16: "cancel a new car contract", and last 1 or 2 months for queries such as Q79: "newest tablet". The $TDC(Q)$ scores linearly decrease according to their $TR(Q)$. For other queries such as Q76: "NBA scores" and Q14: "California lottery results", the relevant documents change very frequently, usually per week for the latter one or even everyday during the season time for the former one. This results in getting higher $TDC(Q)$ scores than other queries. At the same time, users are searching for up-to-date content, as it is confirmed by the $TR(Q)$ scores. As shown, the users' perception of timeliness for all of the above queries is captured sufficiently using our method.

5.3 Improving the Retrieval Performance Using Query Timeliness

In Section 4 we proposed a principled way to incorporate timeliness into our ranking algorithm TAR. In this section, we experimentally evaluate the retrieval performance of TAR, compared with other time-aware and non time-aware rankings. We first describe the experimental setup, which is in addition to the setup described in Section 5.1. Subsequently, we present our experimental results in Section 5.3.2.

5.3.1 Experimental Setup

Datasets. For our retrieval evaluation experiments we used the 2400 documents that we collected during the timeliness estimation survey. Note that for the performance evaluation experiments instead of using the results' titles and snippets, we built an index on the actual HTML content of each web page.

Effectiveness Metrics. For our retrieval evaluation, we applied two widely used relevance metrics: precision and Discounted Cumulative Gain (DCG) [31]. In particular, we measured the precision and DCG on the top-n results, denoted as $Prec@n$ and $DCG@n$ respectively. Instead of $DCG@n$, we adopted the Normalized Discounted Cumulative Gain (NDCG), which is a normalization of DCG in the range [0, 1] and is calculated as:

$$NDCG@n = \frac{DCG@n}{IDCG@n} \tag{9}$$

where IDCG@n is the ideal DCG@n, i.e., the maximum possible DCG value up to the ranking position n. DCG@n is calculated as [32]:

$$DCG@n = \sum_{i=1}^{n} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{10}$$

where rel_i denotes the binary relevance of the results at ranking position i, i.e., rel_i is equal to 1 if the result at position i is valid and 0 otherwise.

In our experiments we set $n = 5$, i.e., we measured Prec@5 and NDCG@5. In particular we calculated the average Prec@5 and NDCG@5 across the complete query workload [21].

Algorithms. In our evaluation we compared our TAR ranking with the following set of algorithms:

- BM25, the default (non time-aware) ranking provided by Lucene 3.5.0 [33], which uses BM25 ranking [19];
- BM25-T (Time-sorted BM25), which first retrieves the top-n documents based on BM25 and then sorts them by decreasing timestamp[4];
- EXP (Exponential time-based ranking) [1], which uses a constant exponential re-ranking rate for all queries, as defined in Equation 4;
- BEX (Bayesian EXponential ranking) [3], which calculates a query-specific exponential re-ranking rate based on the distribution of the top ranked documents obtained from a non time-aware ranking, as explained in Section 3.

Li and Croft [1] experimentally show that EXP achieves the best ranking quality by setting $\lambda = 0.01$. Therefore, we used this value for our experiments. For BEX, we used the recommended parameter settings as described in [3], i.e., we set $k = 500$, $\rho = 100$, and we calculated σ such that $(\rho - 1)/\sigma = 0.015$ (see [3] for more details).

Relevance User Survey. We set up a user study to collect the relevance judgments for our dataset. For this purpose we applied a pooling method that has been popularly used to build test collections in TREC [34, 35]. We randomly mixed the top results from each of the above ranking algorithms and asked a set of workers on the Amazon Mechanical Turk to label the results that are the most relevant to each query. The workers were asked to make their judgments considering both the topical and the temporal relevance of the presented results. The the whole dataset is available at [21].

We retrieved the top-5 results for each ranking algorithm BM25, EXP, BEX and TAR with different values of α[5]. After taking the union of the top-5 rankings of all algorithms (duplicate results from different algorithms will only show once), for each query we got 17 unique document results on average. Note that to compare with other algorithms, we will report the retrieval quality of TAR based on a single value of α. We split our query workload into 6 groups of 20 queries each, such that each worker would have to provide relevance judgments for 20 queries. Thus, each user had to evaluate around 340 (17 × 20) query-document pairs. A document is considered as relevant to a query if it is labeled by over 50% of the workers that provided judgments for this query. Again, we disqualified some low-quality workers from our study as detailed in the Appendix. We finally assumed as valid only the judgments provided from the 104 most high-quality workers (among the initial 126 workers). Thereby, each query has been evaluated by 17.3 (high-quality) workers on average. In total, for our experimental evaluation, we considered 35248 query-document pairs, out of which 11637 are labeled as relevant.

Setting of α: Additionally, we need to set the parameter α in Equation 6 for our TAR ranking. For this purpose we conducted a 5-fold cross-validation to train and test it. In particular, we split our query workload into 5 sets following a lexicographic order. Thereby, 4 out of the 5 test sets consist of 24 queries and one consists of 23 queries. We experimented with the following values for α = 0.01, 0.03, 0.05, 0.07, 0.09, 0.1, 0.3, 0.5, 0.7, 0.9, 1, 3, 5, 7, 9 and 11, which are all included in the above user survey. Each row in Table 1 shows the value of α that achieves the best averaged Prec@5 on each training set, and the averaged Prec@5 and NDCG@5 scores based

[4] When we study the effectiveness of top-n results for BM25-T, we retrieve and sort top-n results from BM25.

[5] We include various values for α as we will study the setting of α for TAR. In addition, this helps to retrieve results with diverse publication dates to increase the effectiveness of pooling.

Table 1: Cross-validation experiments for α using Prec@5

training set	α	Prec@5 (training set)	Prec@5 (testing set)	NDCG@5 (testing set)
1	0.5	0.383	0.325	0.410
2	0.3	0.371	0.383	0.472
3	0.3	0.375	0.367	0.393
4	0.5	0.371	0.375	0.425
5	0.3	0.371	0.383	0.477

on this α for the respective testing sets. Note that we also conducted experiments using NDCG@5 as our retrieval quality measurement for training; since it produced similar results with Prec@5, we do not show the results of this experiment.

As shown in Table 1, the values of α that achieve the best Prec@5 are quite stable on different training sets. Thus, in order to evaluate the overall performance of our TAR method against the baseline algorithms, we calculated the average Prec@5 and NDCG@5 for all testing queries in our query workload under the setting $\alpha = 0.3$.

5.3.2 *Experimental Results*

Measuring the ranking differences. First, we experimentally validate that the time-aware rankings yield quite different results compared to the BM25 ranking. For this purpose, we measured the normalized *Spearman footrule* distance [36] between each time-aware ranked list and the BM25 ranking. Figure 6 shows the Spearman footrule distance between BM25 and BM25-T, EXP, BEX and TAR for different values of α, when considering the top-*n* results for *n* = 5, 10, 15 and 20. Larger values for the Spearman footrule indicate more disagreement between two lists. As shown, the ranking of search results changes largely when applying a time-sensitive ranking algorithm.

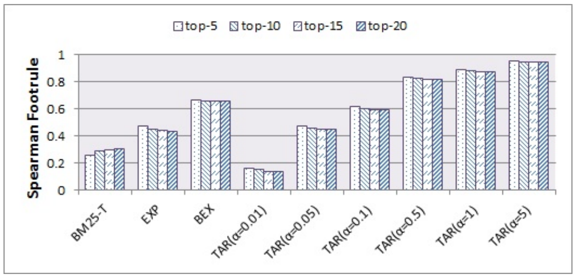


Figure 6: Spearman footrule between the BM25 and different time-based rankings for various top-*n* lists

Retrieval Quality. Table 2 compares the retrieval quality of TAR against the competitor methods for the complete query workload. As depicted, TAR achieves superior retrieval quality compared to all four competitor rankings. TAR performs 15% and 11% better than BEX in terms of average Prec@5 and NDCG@5, respectively. Further, BEX has better performance than EXP, which is consistent with the experimental findings in the original paper [3]. BM25-T delivers better NDCG@5 than BM25 because of the property of timely queries: more recently published relevant documents are preferred. Note that, as we described before, top-*n* results of BM25-T is the reranking of top-*n* results of BM25 based on time. Thus, for a query the value of Prec@5 will be the same for BM25-T and BM25 but NDCG@5 may be different. Further, the improvements of TAR are statistically significant with p-value < 0.01 using the paired Student's t-test over all competitor methods.

Sensitivity of Retrieval Quality wrt. Query Timeliness. Next, we studied the retrieval quality of all ranking algorithms for queries with different timeliness requirements. Specifically, we split the query workload into three timeliness groups according to the values of $TR(Q)$, as specified by the users in the survey described in Section 5.2.1. The three timeliness groups that we constructed have the following ranges: [0-6 months], [6-24 months] and [24-66 months] and contain 34, 45 and 40 queries respectively.

Table 2: Retrieval quality for BM25, BM25-T, EXP, BEX and TAR

ranking algorithm	AVG Prec@5	AVG NDCG@5
BM25	0.176	0.202
BM25-T	0.176	0.228
EXP	0.277	0.332
BEX	0.324	0.393
TAR	**0.373**	**0.438**

Figures 7 and 8 show the average Prec@5 and NDCG@5 results for each timeliness group for the different rankings that we examined. The "*" symbol over the TAR bar denotes that the respective improvements of TAR over all baselines are statistically significant with p-value < 0.05 using the paired Student's t-test.

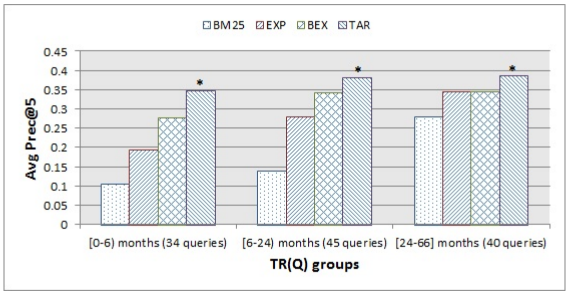


Figure 7: Average Prec@5 of examined algorithms on different timeliness groups

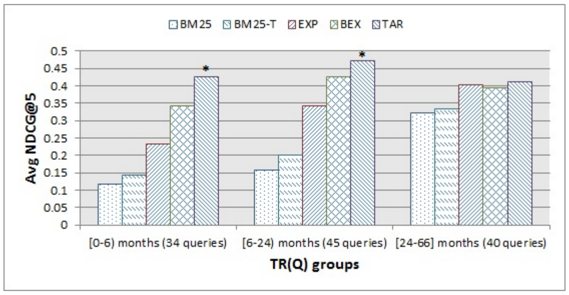


Figure 8: Average NDCG@5 of examined algorithms on different timeliness groups

These two figures show that for all timeliness groups, our proposed ranking achieves better retrieval quality than both BM25 and the other time-aware ranking algorithms. For the case of the [0-6 months] group, in which queries have the highest timeliness requirement, the improvements of TAR over all baselines (over 24% better than BEX on both Prec@5 and NDCG@5) are larger compared to the other two groups. This shows that our proposed model is especially useful for the queries with very intense timeliness requirements. For the case of the [6-24 months] group, TAR delivers over 10% improvement than BEX which is still the best among the baselines. Finally, for the queries in the [24-66 months] group, where the results freshness is a less important factor than the other two groups, TAR has a slight improvement over the baselines. Further, compared to the previous timeliness groups for the queries of this group we notice that BM25 achieves quite better retrieval quality because the content relevance is becoming the more important factor when the timeliness requirement drops.

Examples. We examine the retrieval quality for the query examples that we studied in Section 5.2.1. The Prec@5 and NDCG@5 scores for the example queries are shown in Tables 3 and 4 respectively.

Table 3: Prec@5 for a sample of queries using different rankings.

Queries		Algorithms			
		BM25 (BM25-T)	EXP	BEX	TAR
Q88	public speaking tips	0.4	0.4	0.4	0.2
Q56	interview thank you letter	0.4	0.4	0.4	0.4
Q83	passport renewal	0	0	0	0.2
Q16	cancel a new car contract	0.4	0.4	0.6	0.6
Q65	low income housing	0	0	0	0.4
Q89	reality TV stars	0	0.2	0.4	0.4
Q57	keyboard reviews	0.2	0.2	0.6	0.6
Q90	retail sales index	0	0.4	0.4	0.6
Q95	smartphone reviews	0.6	0.6	0.6	0.8
Q15	California state parks jobs	0.2	0.4	0.6	0.6
Q79	newest tablet	0	0.2	0.2	0.4
Q17	celebrity gossips	0	0	0.2	0.4
Q75	NBA game schedule	0.2	0.2	0.2	0.6
Q76	NBA scores	0	0.2	0.2	0.6
Q14	California lottery results	0	0	0	0

Table 4: NDCG@5 for a sample of queries using different rankings

Queries		Algorithms				
		BM25	BM25-T	EXP	BEX	TAR
Q88	public speaking tips	0.360	0.360	0.383	0.301	0.146
Q56	interview thank you letter	0.586	0.319	0.319	0.319	0.319
Q83	passport renewal	0	0	0	0	0.182
Q16	cancel a new car contract	0.316	0.553	0.485	0.684	0.640
Q65	low income housing	0	0	0	0	0.360
Q89	reality TV stars	0	0	0.182	0.531	0.704
Q57	keyboard reviews	0.246	0.390	0.246	0.710	0.805
Q90	retail sales index	0	0	0.301	0.316	0.655
Q95	smartphone reviews	0.699	0.684	0.699	0.699	0.830
Q15	California state parks jobs	0.246	0.390	0.637	0.805	0.805
Q79	newest tablet	0	0	0.202	0.296	0.704
Q17	celebrity gossips	0	0	0	0.146	0.316
Q75	NBA game schedule	0.170	0.146	0.146	0.214	0.616
Q76	NBA scores	0	0	0.131	0.170	0.655
Q14	California lottery results	0	0	0	0	0

Sensitivity of Retrieval Quality wrt. Document Volume Distribution. An interesting observation on our data set is that even if we follow a non time-based ranking (e.g., BM25), the time distribution of the most relevant documents is skewed towards the more recent ones. Specifically, in the top-500 documents of BM25, the average number of documents per query is: 62, 71, 83, 92, 101, 90 for years 2007-2011 and the first half of 2012 respectively. The distributions for some queries are more skewed than the average; we identified 44 out of the 119 queries where the top-500 results contain more that 40% documents that have been published in the last 1.5 year, vs. 60% of documents from 2007-2010. One possible explanation for this is that, since the size of the web grows faster over time, recent documents have a larger number; hence the probability for a recent document to be relevant is higher. Further, some older relevant web pages are no longer accessible or might be penalized with lower scores by commercial search engines. Recall that previous algorithms, especially BEX [3], leverage the time distribution in their rankings and thus could benefit from this skewed distribution.

We studied the effect of the document distribution on retrieval quality. In particular, we computed the retrieval quality for two sets of queries; 75 queries that exhibit a quite steady time distribution in relevant documents and 44 queries with more skewed distributions towards recent documents. For the former subset, we got an average Prec@5 equal to 0.296 for BEX and 0.352 for TAR, i.e., TAR outperforms BEX (which is the best performing competitor) by 18.9%. For the 44 queries with more skewed distributions, the calculated average Prec@5 for BEX and TAR is 0.373 and 0.409 respectively, which is 9.6% improvement for TAR. The smaller improvement on this subset is expected, since BEX performs better for highly skewed time distributions. For both query sets, the improvement of TAR over BEX is statistically significant with p-value < 0.05.

Summary. All time-sensitive ranking algorithms outperform BM25 with significant improvements on both Prec@5 and NDCG@5 for timely queries. Further, consistent with former research [3], BEX generally exhibits better retrieval quality than EXP.

TAR achieves the best performance among all ranking algorithms. In particular, TAR improves over 10% in terms of both Prec@5 and NDCG@5 over BEX ranking algorithm on our complete query workload. TAR can satisfy queries with different timeliness requirements better than other time-sensitive ranking algorithms as shown in the experiments (Figures 7 and 8). Further, if we remove the effect of the skewness in the time distribution of the documents, TAR achieves even higher improvement (18.9%) over BEX (which delivers the best ranking quality among the competitor rankings).

The retrieval quality results on queries from different timeliness groups validate our proposed model, i.e., the timeliness requirements can be predicted accurately based on the degree of content change in relevant documents and it is used in an effective way in our proposed ranking model.

6. DISCUSSION

Limitations. In this paper, we focus on timely queries that do not exhibit clear or significant variance in query or document popularity over time, but where recent results are preferred. The proposed model is not meant to handle other types of queries such as those targeting specific events. As a direction for future work, we will study a principled way to combine our model with previous works that focus on other types of time-sensitive queries, such as [4] which studies the volume distribution of relevant documents. In other words, we will study how to propose a unified model which considers different signals to estimate the temporal requirements for a broader set of queries.

Also note that because of a lack of public benchmarks that provide both the topical and temporal relevance of results for timely queries, it's hard to conduct experiments on a very large query workload; however we believe that 119 queries is a reasonably large workload.

Practical Issues. In a real-world scenario, instead of conducting a user survey, the timeliness requirements of each query ($TR(Q)$) can be extracted based on clickthrough data, for instance by observing the timestamps of results that are clicked or not-clicked by the users after each web search.

If a new query Q for which the timeliness requirement is unknown is issued to the search engine, we can use the timeliness requirements of similar queries in order to estimate a TDC value for Q. Similar queries can be found by considering keyword text similarity, or based on a query likelihood approach, etc. Further, in terms of implementation, one alternative approach to estimate $TDC(Q)$ on query time would be to first compute the top-k results in a time-insensitive way for query Q, then compute $TDC(Q)$ using these k results, and then rerank them using Equation 5.

With regard to learning an optimal value for α parameter, in a practical use case a search engine can train the model based on user feedback. Different values of α might be suitable for queries that exhibit different timeliness requirements; this is also an interesting direction that we aim to explore as our future work.

In a real-world web search system, our model can be applied as a complement to previous work studying news queries. Previous works have proposed methods to classify queries as news-related or not news-related [6, 10]. If the query is not news-related, our proposed TAR algorithm can be used, otherwise a news ranking approach [5, 6, 7, 10] can be applied.

7. CONCLUSIONS

In this paper we studied the freshness factor for a class of queries that we refer to as timely queries. We show that previous works on news queries cannot be applied effectively for predicting the timeliness requirement of queries if the query popularity from document publishers or consumers does not vary significantly over time. We propose a method to estimate query timeliness with high accuracy using the terms distribution change of a query's relevant documents over time. Further, we present a ranking model that incorporates the timeliness factor in order to improve the results freshness for timely queries, and we experimentally show that our ranking improves upon previous methods over 10% in terms of both precision and NDCG. In our future work we plan to explore methods to automatically learn the *TDC* scores and to combine our proposed ranking model with other signals in order to support a broader set of queries.

8. ACKNOWLEDGMENTS

This work was supported in part by National Science Foundation grants IIS-1216032 and IIS-1216007.

9. REFERENCES

[1] X. Li and W. B. Croft, "Time-based language models," in *CIKM*, pp. 469–475, 2003.

[2] R. Jones and F. Diaz, "Temporal profiles of queries," *TOIS*, vol. 25, no. 3, 2007.

[3] M. Efron and G. Golovchinsky, "Estimation methods for ranking recent information," in *SIGIR*, pp. 495–504, 2011.

[4] W. Dakka, L. Gravano, and P. G. Ipeirotis, "Answering general time-sensitive queries," *TKDE*, vol. 24, no. 2, pp. 220–235, 2012.

[5] F. Diaz, "Integration of news content into web results," in *WSDM*, pp. 182–191, 2009.

[6] A. Dong, Y. Chang, Z. Zheng, G. Mishne, J. Bai, R. Zhang, K. Buchner, C. Liao, and F. Diaz, "Towards recency ranking in web search," in *WSDM*, pp. 11–20, 2010.

[7] A. Dong, R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, Z. Zheng, and H. Zha, "Time is of the essence: improving recency ranking using Twitter data," in *WWW*, pp. 331–340, 2010.

[8] A. Kulkarni, J. Teevan, K. M. Svore, and S. T. Dumais, "Understanding temporal query dynamics," in *WSDM*, pp. 167–176, 2011.

[9] K. Radinsky, K. M. Svore, S. T. Dumais, J. Teevan, A. Bocharov, and E. Horvitz, "Modeling and predicting behavioral dynamics on the web," in *WWW*, pp. 599–608, 2012.

[10] N. Dai, M. Shokouhi, and B. D. Davison, "Learning to rank for freshness and relevance," in *SIGIR*, pp. 95–104, 2011.

[11] A. Styskin, F. Romanenko, F. Vorobyev, and P. Serdyukov, "Recency ranking by diversification of result set," in *CIKM*, pp. 1949–1952, 2011.

[12] Y. Chang, A. Dong, P. Kolari, R. Zhang, Y. Inagaki, F. Diaz, H. Zha, and Y. Liu, "Improving recency ranking using twitter data," *ACM Trans. Intell. Syst. Technol.*, vol. 4, pp. 4:1–4:24, 2013.

[13] A. C. König, M. Gamon, and Q. Wu, "Click-through prediction for news queries," in *SIGIR*, pp. 347–354, 2009.

[14] J. L. Elsas and S. T. Dumais, "Leveraging temporal dynamics of document content in relevance ranking," in *WSDM*, pp. 1–10, 2010.

[15] S. Kullback and R. Leibler, "On information and sufficiency," *Annals of Mathematics and Statistics*, vol. 22, pp. 79–86, 1951.

[16] D. Metzler, S. Dumais, and C. Meek, "Similarity measures for short segments of text," in *Advances in Information Retrieval*, vol. 4425, pp. 16–27, 2007.

[17] A. Huang, "Similarity measures for text document clustering," in *NZCSRSC*, 2008.

[18] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *SIGIR*, pp. 275–281, 1998.

[19] J. Pérez-Iglesias, J. R. Pérez-Agüera, V. Fresno, and Y. Z. Feinstein, "Integrating the probabilistic models BM25/BM25F into Lucene," *CoRR*, vol. abs/0911.5046, 2009.

[20] "TREC Web Track Datasets." http://trec.nist.gov/data/webmain.html.

[21] "Data set." http://dblab.cs.ucr.edu/projects/TimelyQueries.

[22] O. Alonso, M. Gertz, and R. A. Baeza-Yates, "Clustering and exploring search results using timeline constructions," in *CIKM*, pp. 97–106, 2009.

[23] T. Campos and R. Nuno, "Using top-k retrieved web snippets to date temporal implicit queries based on web content analysis," in *SIGIR*, pp. 1325–1326, 2011.

[24] M.-H. Peetz, E. Meij, M. de Rijke, and W. Weerkamp, "Adaptive temporal query modeling," in *ECIR*, pp. 455–458, 2012.

[25] "Amazon Mechanical Turk." http://www.mturk.com/.

[26] "Google Insights." http://www.google.com/insights/search/.

[27] C. Kohlschütter, P. Fankhauser, and W. Nejdl, "Boilerplate detection using shallow text features," in *WSDM*, pp. 441–450, 2010.

[28] O. Zamir and O. Etzioni, "Web document clustering: a feasibility demonstration," in *SIGIR*, pp. 46–54, 1998.

[29] H.-J. Zeng, Q.-C. He, Z. Chen, W.-Y. Ma, and J. Ma, "Learning to cluster web search results," in *SIGIR*, pp. 210–217, 2004.

[30] S. M. Beitzel, E. C. Jensen, A. Chowdhury, and O. Frieder, "Varying approaches to topical web query classification," in *SIGIR*, pp. 783–784, 2007.

[31] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *TOIS*, vol. 20, no. 4, pp. 422–446, 2002.

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

[33] "Apache Lucene." http://lucene.apache.org/core/.

[34] E. M. Voorhees and D. K. Harman, *TREC: Experiment and Evaluation in Information Retrieval*. The MIT Press, 2005.

[35] I. Soboroff, "A comparison of pooled and sampled relevance judgments," in *SIGIR*, pp. 785–786, 2007.

[36] R. Fagin, R. Kumar, and D. Sivakumar, "Comparing top-k lists," *SIAM J. Discrete Math.*, vol. 17, no. 1, pp. 134–160, 2003.

APPENDIX

Filtering Timeliness Judgments. We detected that some workers' responses are very dissimilar from the rest in Amazon Mechanical Turk service. Thus, in order to remove the low-quality workers we applied the following filtering strategy. Based on the survey results from the 10 students, we picked 7 queries that receive the most consistent responses (i.e., having the lowest standard deviation in their timeliness values). For each query, we consider as valid timeliness response from a worker any value between the minimum and the maximum timeliness values given by the students. If a response from a worker yields a value that is not in the valid range, we treat this as an invalid response. We accept only those workers who did at most two invalid choices out of the responses for the 7 selected queries. Using these criteria, we got 20 valid Amazon Mechanical Turk responses for each query from unique workers.

Filtering Relevance Judgments. We perform a postprocessing of the survey results in order to make sure careless judgments are filtered out. For each query group (each survey has about 20 queries), we select two timely queries. For these two queries, we get the 3 most frequently and least frequently selected results from Amazon Mechanical Turk workers (more than 3 if there are ties). If the selections of a worker have one miss in the most selected results or one hit in the least selected results, we increase the count of the worker's inconsistent selections. We accept the worker's selections as long as the worker has at most 6 inconsistent selections in total for the two queries. We got 126 worker responses in total, i.e., for each query we got relevant judgments from 21 workers on average. From this set we finally selected 104 valid ones (17.3 per query).