Temporal Feedback for Tweet Search with Non-Parametric Density Estimation

Miles Efron[1], Jimmy Lin[2], Jiyin He[3], and Arjen de Vries[3]

[1] Graduate School of Library and Information Science, University of Illinois, Urbana-Champaign
[2] The iSchool, University of Maryland, College Park
[3] Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

mefron@illinois.edu, jimmylin@umd.edu, jiyinhe@acm.org, arjen@acm.org

ABSTRACT

This paper investigates the *temporal cluster hypothesis*: in search tasks where time plays an important role, do relevant documents tend to cluster together in time? We explore this question in the context of tweet search and temporal feedback: starting with an initial set of results from a baseline retrieval model, we estimate the temporal density of relevant documents, which is then used for result reranking. Our contributions lie in a method to characterize this temporal density function using kernel density estimation, with and without human relevance judgments, and an approach to integrating this information into a standard retrieval model. Experiments on TREC datasets confirm that our temporal feedback formulation improves search effectiveness, thus providing support for our hypothesis. Our approach outperforms both a standard baseline and previous temporal retrieval models. Temporal feedback improves over standard lexical feedback (with and without human judgments), illustrating that temporal relevance signals exist independently of document content.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—Relevance feedback

Keywords: temporal clustering; cluster hypothesis; relevance feedback

1. INTRODUCTION

Twitter has become an indispensable communications platform through which hundreds of millions of users around the world witness breaking news events. They can participate in the global conversation in real time, 140 characters at a time. To access relevant content in microblogs, people often turn to search. And naturally, time plays an important role in tweet search. We seek to improve access to microblog information by building better search systems.

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.

ACM 978-1-4503-2257-7/14/07 ...$15.00.
http://dx.doi.org/10.1145/2600428.2609575.

From a theoretical perspective, this work formulates and explores the *temporal cluster hypothesis*, stated as follows: in search tasks where time plays an important role, do relevant documents tend to cluster together in time? This parallels the "classic" cluster hypothesis [7], which is the observation that relevant documents tend to share similar content (i.e., cluster in document space). In the same way that the effectiveness of content-based relevance feedback techniques affirms the classic cluster hypothesis, the effectiveness of temporal relevance feedback techniques, which we explore in this paper, can be considered evidence supporting the temporal cluster hypothesis.

Our formulation of the tweet search problem follows a user scenario that underpins the recent Microblog evaluations at the Text Retrieval Conference (TREC): at time t, a user expresses an information need in the form of a query Q. The system's task is to return topically-relevant documents (tweets) posted before the query time. Since the temporal distribution of relevant tweets for an information need is usually non-uniform, it is important for retrieval systems to model the temporal characteristics of the query, retrieved documents, and the collection as a whole. This insight, shared by many researchers [8, 3, 5, 4, 2, 20], provides the starting point for our study.

In this paper, we propose a family of techniques for tweet search that integrates temporal signals with "classic" lexical (i.e., content-based) approaches. We adopt a feedback framework where temporal features are extracted from R, the initial list of documents retrieved by a standard query-likelihood approach, and then used to rerank R to produce a final ranked list. Let us suppose that each document $D_i \in R$ has an associated timestamp T_i: the core contribution of our work lies in novel techniques (1) to estimate $f(T|Q)$, the temporal density of relevance (i.e., for a particular information need, where we would expect relevant documents to occur in time), and (2) to integrate this signal with standard lexical features in a log-linear model.

We propose two ways to estimate f, *implicit temporal feedback* and *explicit temporal feedback*. Both methods rely on a simple non-parametric approach to estimating a distribution from data, kernel density estimation (KDE). The implicit/explicit distinction is analogous to the difference between pseudo- and true relevance feedback based on document content. Experiments using the TREC 2011 and 2012 Microblog test collections reveal three main findings:

1. Our temporal feedback approach (implicit as well as explicit feedback) improves tweet search effectiveness

over a query-likelihood baseline and over two previously proposed temporal retrieval techniques.

2. Because the effectiveness gain of our approach is additive with standard lexical feedback (with and without human judgments), we show that temporal relevance signals exist independently of document content.

3. With only a few human relevance judgments (a negligible effort), temporal feedback achieves a large effectiveness increase.

Note that although this work only studies tweet search, there is no *prima facie* reason why our proposed techniques would not generalize to other domains and retrieval tasks with a strong temporal component (e.g., news search). Substantial interest in social media today justifies a study focused only on tweets, and we leave such generalizations to future work.

2. BACKGROUND AND RELATED WORK

2.1 Temporal IR

Various papers have previously observed that the temporal distribution of relevant documents for an information need is rarely uniform [8, 3, 5, 4, 2, 20]. There is general consensus in the IR community that effective retrieval systems need to model the temporal characteristics of the query, retrieved documents, and the collection as a whole. We take this as a non-controversial starting point.

Retrieval models that incorporate temporal evidence have been explored in early work by Li and Croft [12], who proposed temporal extensions to language models. This thread has been subsequently extended by others [4, 2], and provides the basis for our formal model. Jones and Diaz [8] explored the temporal profile of queries, classifying queries as atemporal, temporally ambiguous, and temporally unambiguous. They show that the distribution of retrieved documents can provide an additional source of evidence to improve rankings. Other attempts at incorporating temporal signals in ranking include [3, 5].

An important difference between the cited papers and our own work lies in the envisioned role of the user. Most previous work relies on automatic methods for inferring temporal signals, such as the distribution of retrieved documents or term statistics time series. Our proposed implicit temporal feedback approach can be viewed as an extension of this line of work, but the inclusion of explicit temporal feedback in our study does imply a different direction. We assume an active role for the end user, which we believe is plausible for a class of sophisticated searchers (e.g., journalists or historians), and empirically demonstrate that even a small amount of user-supplied temporal "hints" can significantly improve result quality. Thus, the contribution of this paper is both technical (the methods outlined in Section 4) and conceptual (making the role of temporal evidence a type of user-directed relevance feedback).

Beyond search ranking, researchers have explored related problems that benefit from modeling temporal signals. Examples include query log analysis (mining similar web queries by examining query volume over time [28]), behavior prediction (by modeling the temporal dynamics of user activities [22]), time-sensitive query auto-completion [24], and real-time query suggestion in the context of Twitter [18].

Finally, we are also aware of researchers who have attempted to characterize and quantify temporal change of web pages [1] as well as the "churn" of queries on Twitter [16]. Although these cited works explore fundamentally different issues than the focus of our study, they demonstrate that temporal modeling is important from a variety of perspectives.

2.2 Microblog Evaluations

The context for our study is the recent Microblog tracks at TREC [19, 27]. The 2011 and 2012 evaluations used the Tweets2011 corpus,[1] which consists of an approximately 1% sample (after some spam removal) of tweets from January 23, 2011 to February 7, 2011 (inclusive), totaling approximately 16 million tweets. Major events that took place within this time frame include the massive democracy demonstrations in Egypt as well as the Super Bowl in the United States. There are 49 topics for TREC 2011 and 60 topics for TREC 2012. Each topic consists of a query and an associated timestamp, which indicates when the query was issued. Using a standard pooling strategy, NIST assessors evaluated a total of 114K tweets and assigned one of three judgments to each: "not relevant", "relevant", and "highly relevant". For the purpose of our experiments, we considered both "relevant" and "highly relevant" tweets relevant.

The premise of this work is that the temporal distribution of relevant tweets is not uniform, and that a retrieval model should take this signal into account. More precisely, we seek to estimate $f(T|Q)$, the temporal density of relevance—where in time we would expect relevant documents for a query to show up. To confirm our intuitions, we began by creating simple visualizations that characterize the distribution of relevant documents for TREC Microblog topics from 2011 and 2012 [15]. The results of three topics are shown in Figure 1: in each timeline, the query time is anchored to the right edge; the x-axis shows time *prior* to the query time, in days. Dots show tweets that were retrieved by participating teams and evaluated by assessors (i.e., the pools): green dots are relevant, red dots are highly relevant. The vertical position of the dots has no meaning; jitter is added only to prevent overlap. The underlying blue bars show the distribution of relevant and highly-relevant tweets as a histogram. Due to space limitations, only three topics are shown here, but these timelines are representative of the shapes of the distributions we see across all topics. For topic 29 "global warming and weather", relevant tweets are distributed relatively evenly from a temporal perspective; for topic 30 "Keith Olbermann new job", with one exception all relevant tweets are very close to the query time; and for topic 37 "Giffords recovery", most relevant tweets are clustered in two temporal intervals that occur several days prior to the query time.

These visualizations confirm our intuition that the temporal distribution of relevant tweets is highly non-uniform—which means that any retrieval model that does not take into account document timestamps is potentially "missing out" on an important relevance signal. Furthermore, the temporal distributions appear to be query specific, which means that one-size-fits-all strategies are unlikely to be effective for all topics. For example, we accept that topic 30 would benefit from recency priors, but it is unclear how the same

[1] `http://twittertools.cc/`

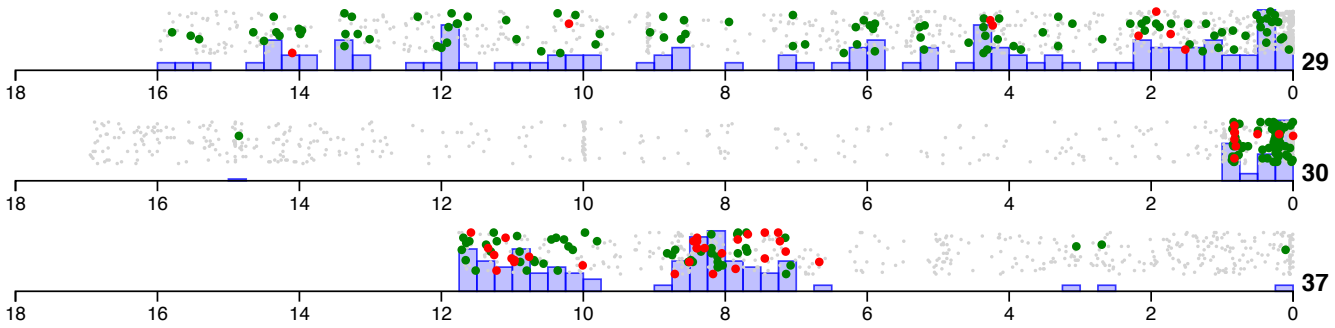

Figure 1: Visualizations illustrating the temporal distribution of retrieved documents and relevant documents for three topics from the TREC 2011 Microblog track: topic 29, "global warming and weather", topic 30 "Keith Olbermann new job", and topic 37 "Giffords recovery". The timeline is measured in days, anchored by the query time on the right edge. Green dots represent relevant documents, red dots represent highly-relevant documents, and gray dots represent non-relevant documents. The bar graphs show bucketed distributions of the relevant and highly-relevant documents.

technique could help topic 37, whose relevant documents are mostly concentrated a week before the query time.

Finally, note that these visualizations are created *post hoc*, i.e., after the assessment process has completed, so it is not immediately obvious what features are available at *query time*. In this paper, we propose techniques that exactly address this issue—using no user input and a limited amount of user input.

3. FORMAL MODEL

In this section, we present a formal model for integrating traditional (i.e., lexical) models of relevance with temporal relevance signals—specifically, $f(T|Q)$, the temporal density of relevance that we will estimate from an initial list of retrieved documents. Details about the estimation procedure are described in the next section. By way of comparison, we discuss alternative formulations that attempt to integrate temporal information into standard retrieval models.

We use as a starting point the query-likelihood approach in the language modeling framework [21], where documents are ranked on

$$P(D|Q) \propto P(Q|D)P(D) \tag{1}$$

where $P(Q|D)$ is the likelihood that the language model that generated document D would also generate the text of query Q, and $P(D)$ is a prior distribution over documents.

Recency Priors. One of the simplest way to let time influence the ranking model was given by Li and Croft [12], who proposed a document prior that favors recently published documents. If T_D is the timestamp associated with document D, they propose modeling $P(D)$ in Eq. (1) via an exponential distribution:

$$P(D) = \lambda e^{-\lambda T_D} \tag{2}$$

where $\lambda \geq 0$ is the rate parameter of the exponential distribution. We refer to this as a *recency prior* and refer to runs that use it as "Recency".

Independent Evidence. Though previous studies have shown that recency priors increase overall effectiveness across a set of topics, by definition they are query-independent. This could be problematic insofar as the dependencies between time and relevance vary from query to query [8]. Figure 1 clearly shows that this is the case: we would expect recency priors to be effective for topic 30, but such techniques are not likely to be effective for information needs represented by topic 37, where the relevant documents are not clustered close to the query time.

Dakka et al. [2] proposed a query-specific way to combine lexical and temporal evidence in the language modeling framework by separating the lexical and temporal signals into two components: W_D, the words in the document and T_D, the document's timestamp. This leads to the following derivation:

$$P(D|Q) = P(W_D, T_D|Q) \tag{3}$$
$$= P(T_D|W_D, Q)P(W_D|Q) \tag{4}$$
$$\sim P(W_D|Q)P(T_D|Q) \tag{5}$$

where the last step follows from Eq. (4) if we assume independence between lexical and temporal information. The resulting formula is identical to the standard query-likelihood model, but with the addition of the probability of observing a time T_D given the query Q.

Dakka et al. proposed several ways to estimate $P(T_D|Q)$. In our experimental analyses (see Section 5) we use one of their methods, the moving window (WIN) approach, as a point of comparison to our own techniques, so we describe it here. With WIN, documents retrieved for Q are allocated among b bins according to their timestamps. For each bin b_t, we count $n(b_t)$, the number of retrieved documents in b_t. Next, bin counts are smoothed by averaging x bins into the past and x bins into the future (where x is the window width). Let $n(b_{tx})$ be the average number of documents in the $2x$ bins surrounding b_t and b_t itself. Finally, bins are arranged in decreasing order of $n(b_{tx})$. The quantity $P(T_D|Q)$ depends on the bin associated with T_D. If T_D is in the n^{th} ordered bin, then $P(T_D|Q) = \phi(n, \lambda)$ where ϕ is an exponential distribution with rate parameter λ.

Log-Linear Temporal Integration. Following Dakka et al., we take the view that two distinct distributions arise during retrieval that we wish to integrate into a single ranking. There is $P(R|W_D, Q)$, the word-based (i.e., lexical) probability of relevance given Q. We also have $P(R|T_D, Q)$, which is the probability of relevance to Q given temporal considerations.

Defining the word-based distribution is a well-studied problem. Lafferty and Zhai [10] have argued that $P(R|W_D, Q)$ is not substantially different from the standard query-likelihood estimate. Accepting that view, we may assume that

$$P(R|W_D, Q) \stackrel{def}{=} P(Q|D) \tag{6}$$

where, by assuming term independence and a multinomial language model, we have:

$$P(Q|D) = \prod_{i=1}^{c(Q)} P(q_i|\theta_D) \tag{7}$$

for the language model θ_D, where $c(Q)$ is the number of terms in the query. Using Bayesian updating with a Dirichlet prior parameterized by the real vector $\mu P(w|C)$, we have the estimator:

$$\hat{P}(w|D) = \frac{c(w, D) + \mu P(w|C)}{c(w, D) + \mu} \tag{8}$$

where $P(w|C)$ is the term probability given the language model of the entire corpus, and $c(w, D)$ is the count of term w in document D.

Now consider $P(R|T_D, Q)$, the probability of the relevance of document D to Q given temporal information. To combine the temporal and lexical evidence, we assume a log-linear model. For a parameter $\alpha \in [0, 1]$, we have

$$\begin{aligned} \log P_\alpha(R|D, Q) = Z_\alpha \quad & +(1-\alpha) \log P(R|W_D, Q) \\ & +\alpha \log P(R|T_D, Q) \end{aligned} \tag{9}$$

where Z_α is a normalization constant. Since Z_α does not depend on D for ranking, we can ignore it. We estimate α from a set of training topics by finding the value that maximizes mean average precision (MAP); see Section 5 for more details.

The resulting log-linear retrieval model is equivalent to ranking documents based on Eq. (10):

$$P_\alpha(R|D, Q) \sim P(R|W_D, Q)^{1-\alpha} \cdot P(R|T_D, Q)^\alpha \tag{10}$$

A log-linear interpolation (in the sense of Klakow [9]) is appealing because it allows us to combine temporal and lexical evidence multiplicatively, as in Eq. (5); time becomes simply another feature in our ranking model. A log-linear approach allows us to express the strength of the temporal evidence explicitly via the interpolation parameter α. This is in contrast to most previous work, where the influence of temporal information is controlled indirectly, by parameterizing a distribution such as an exponential to optimize retrieval metrics. Lexical and temporal evidence may differ inherently in importance, but this should not be controlled via the temporal model itself. The log-linear combination provides the advantage that the relative importance of lexical and temporal evidence is controlled independently from the way we capture the temporal relevance information.

Summarizing, log-linear models provide a flexible means to combine heterogeneous evidence and integrate arbitrary features. As a final remark, a different way to understand our approach is to think of it as a very simple linear learning-to-rank model [17] with only two features.

4. TEMPORAL FEEDBACK

The theoretical motivation for this work is what we call the *temporal cluster hypothesis*: in search tasks where time plays an important role (such as tweet search), we hypothesize that relevant documents tend to cluster together in time, and that this property can be exploited to improve search effectiveness. Just as van Rijsbergen's "classic" cluster hypothesis suggests that documents relevant to a query Q will form clusters in a term space, we argue that documents relevant to a query will form clusters along a timeline. Analysis shown in Figure 1 suggests that this is indeed the case. Note that although this intuition is implicit in most prior work in temporal IR, to our knowledge we are the first to explicate such a hypothesis as an underlying principle of how time impacts retrieval.

More formally, we define $P(R|T_D, Q)$ in Eq. (10) as the distribution of documents relevant to Q over time. That is, we assume that there is a density f_Q over the time span of the corpus, such that f_Q is large for times where relevant documents are likely to appear and small during times where we are unlikely to find relevant documents. Intuitively, we want to promote documents whose timestamps coincide with large values of f_Q, i.e., temporal regions where relevant documents "cluster together". This section focuses on the problem of estimating f_Q.

4.1 Kernel Density Estimation

To estimate f_Q, we take advantage of kernel density estimation (KDE), which is a non-parametric method to approximate a density by analyzing data generated from that density. Let $\{x_1, x_2, \ldots, x_n\}$ be an i.i.d. sample drawn from some distribution with an unknown density f. We are interested in estimating the shape of this function f. Its kernel density estimator is:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=0}^{n} K\left(\frac{x - x_i}{h}\right) \tag{11}$$

where $K(\cdot)$ is the kernel—a symmetric but not necessarily positive function that integrates to one—and $h > 0$ is a smoothing parameter called the bandwidth. Though many kernel functions are viable, we use the common Gaussian distribution, such that:

$$K\left(\frac{x - y}{h}\right) = \mathcal{N}\left(\frac{x - y}{h}, 0, h\right) \tag{12}$$

where $\mathcal{N}$ is the normal density. We chose the Gaussian kernel for two reasons. First, as shown below, it gives a ready plug-in value for the optimal bandwidth h. Second, experimentally we found that the choice of kernels has almost no effect on the effectiveness of our methods.

A kernel density estimate is very similar to a histogram. However, KDE requires no binning of data, offloading the bias/variance tradeoff to the choice of bandwidth, which has well-defined methods of selection. One key advantage in using KDE versus histograms for estimating f is KDE's ability to handle weighted observations naturally. If we have $\{\omega_1, \omega_2, \ldots, \omega_n\}$, a vector of non-negative weights on our observed X's such that $\sum \omega_i = 1$, then

$$\hat{f}_\omega(x) = \frac{1}{nh} \sum_{i=0}^{n} \omega_i K\left(\frac{x - x_i}{h}\right) \tag{13}$$

is also a proper density: $\hat{f}_\omega$ is similar to $\hat{f}$, except that we allocate different weights to the kernels. As noted by Hall and Turlach [6], ω_i can be interpreted as the probability

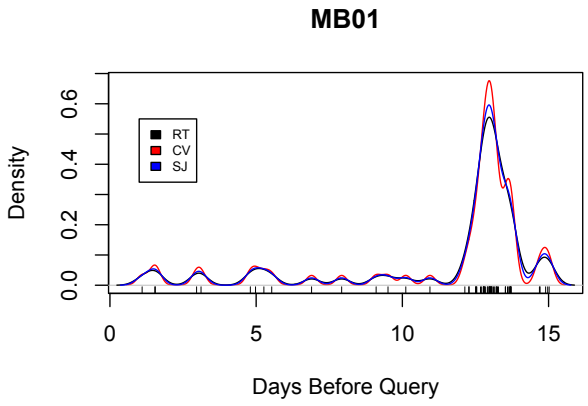

Figure 2: Kernel density estimates for the topic MB01. Each curve corresponds to a different bandwidth selection method.

associated with x_i. Unless otherwise specified, in this paper, the phrase *kernel density estimate* refers to Eq. (13).

Bandwidth Selection. If we choose a Gaussian kernel, as we do here, then as Silverman [25] has shown, the optimal bandwidth is:

$$h^* = \left(\frac{4\hat{\sigma}^5}{3n}\right)^{-\frac{1}{5}} \tag{14}$$

where $\hat{\sigma}$ is the sample standard deviation. It is important to note that the choice of a kernel function is mainly a matter of convenience, carrying with it no implications of the underlying parametric forms of the data. We select the Gaussian due to its wide use and its ready definition of an optimal bandwidth. We refer to this bandwidth as *RT* because it is often called Silverman's *R*ule of *T*humb (*RT*).

The notion of the "optimal" bandwidth has seen much attention in the statistical literature. Many state-of-the-art bandwidth selection approaches are based on some type of cross validation. For instance, a common approach is to minimize the mean integrated squared error (MISE):

$$\text{MISE}(\hat{f}) = E\int (f(x) - \hat{f}(x))^2 dx \tag{15}$$

where E is the expectation. It can be shown that an asymptotically correct approximation MISE can be calculated analytically, yielding a simple algorithm for bandwidth selection that can be optimized via *c*ross *v*alidation [26]; we refer to this method as *CV*. Related methods based on penalized log-likelihood have been proposed and evaluated by Sheather and Jones [23], which we refer to as *SJ*.

Figure 2 shows different kernel density estimates for document timestamps of relevant documents for TREC Microblog topic MB01. A mode at day 14 is clearly visible under all bandwidth selection methods. Overall, the bandwidths given by the three methods yield nearly identical densities. We found that in general, the bandwidth selection method—among those mentioned here—made little difference in effectiveness. Because of its wide adoption in current statistical practice, for the remainder of this paper, we rely on the *SJ* estimate.

4.2 Weighting Schemes for KDE

KDE, via Eq. (13), presents a simple framework for weighting observations (document timestamps) during density estimation. The intuition behind the weight ω_i for document D_i is that this quantity corresponds to our prior belief that the corresponding timestamp T_i was truly generated by f_Q. However, this approach leaves the matter of defining these weights unspecified. In this section we propose four alternative weighting schemes:

Uniform Weights. The simplest approach to weighting for density estimation is simply to give all documents in the initial retrieval equal weight during estimation. Thus for all $D_i \in R$, we have $\omega_i^u = \frac{1}{|R|}$, where $|R|$ is the number of documents retrieved. We call this approach *uniform weighting*.

Score-Based Weights. The simplicity of uniform weighting ignores the information that we have from the score of each document D_i with respect to its lexical similarity to Q, expressed by Eq. (8). Thus, we define a second weighting method, *score-based weights*, where document weights are proportional to their language model-derived probabilities of relevance:

$$\omega_i^s = \frac{P(Q|D_i)}{\sum_{j=1}^n P(Q|D_j)}. \tag{16}$$

Rank-Based Weights. A reasonable objection to score-based weights is their reliance on lexical similarity, which we are ostensibly measuring in tandem with temporal probabilities. In other words, ω_i^s is tied to the retrieval scores of the initial run, while in theory, $f_Q(T_i)$ should be independent of any retrieval model. To remove this coupling, we can make an assumption common in lexical feedback settings. Given an initial lexical ranking R of documents against Q, we can assume that documents near the front of R have a higher probability of relevance than documents ranked lower in R. While traditional pseudo-relevance feedback requires us to choose a hard cutoff of putatively relevant documents k in the context of weighting, we can be less restrictive. Thus, we define *rank-based weights* via an exponential distribution:

$$\omega_i^r = \frac{\lambda e^{-\lambda r_i}}{\sum_{j=1}^n \lambda e^{-\lambda r_i}} \tag{17}$$

where $\lambda > 0$ is the rate parameter of the exponential density and r_i is the rank of document D_i in R.

Though we could leave λ as a tuneable parameter, a simpler way to approach rank-based weights is to use the maximum likelihood estimate. If R contains n documents, the MLE of λ is simply $\frac{1}{\bar{r}}$, where $\bar{r}$ is the mean of the ranks $1, 2, \ldots, n$. This is the approach we use in Eq. (17) and in our implementation of rank-based weights.

True Feedback-Based Weights. All of the weighting methods we have discussed so far assume no knowledge of which documents in R are actually relevant to Q. But if some sort of user interaction gives us true (i.e., human) relevance judgments, it makes sense that we take advantage of the corresponding timestamps to influence $\hat{f}_Q$.

If we know that document D_i is relevant (e.g., based on user input), it makes sense to give that document extra weight. For k relevant documents, $D_{r1}, D_{r2}, \ldots, D_{r_k}$, it is sensible that they be given the same weight (absent graded relevance judgments). Thus, given true relevance

Table 1: Tuning parameters for the temporal retrieval models. Separate values for each parameter were estimated from training topics for runs with no lexical relevance feedback and runs with lexical feedback.

Method	Parameter
Recency	Rate parameter of the exponential prior
WIN	Bin size; window width; rate parameter for exponential bin weighting
KDE	log-linear mixing parameter from Eq. (10)

judgments, we define the *true feedback-based weights* with respect to the score-based approach as follows:

$$\bar{\omega}_i^s = \begin{cases} c, & \text{if } D_i \text{ is relevant} \\ \omega_i^s & \text{otherwise} \end{cases} \tag{18}$$

The final weight z_i^s is arrived at after renormalization:

$$z_i^s = \frac{\bar{\omega}_i^s}{\sum_{j=1}^{n} \bar{\omega}_j^s}. \tag{19}$$

Note that true feedback-based weights can also be computed with respect to document ranks, per Eq. (17), to produce z_i^r in a similar manner.

To eliminate the need to tune one more parameter, in our experiments we simply set $c = 1$ (arbitrarily). Because the value one is almost always much larger than $P(Q|D_i)$, this scheme amplifies the influence of relevant documents.

5. EXPERIMENTAL EVALUATION

Experiments were performed on the Tweets2011 corpus using test collections from the Microblog tracks at TREC 2011 and 2012 (described in Section 2.2). Relevance judgments for the test collections were made on a 3-point scale ("not relevant", "relevant", "highly relevant"), but in this work we ignored the different degrees of relevance and use both higher grades as "relevant".

During collection preparation, we eliminated all retweets since they are by definition not relevant according to the assessment guidelines. No stemming was used, and no stoplist was applied at index time. However, a Twitter-specific stoplist was used when estimating relevance models to reduce the dominance of common terms such as "RT" and "http" during query expansion. All three temporal retrieval models required parameter tuning, which is described in Table 1. Parameters were trained with respect to mean average precision on even-numbered topics (54 total); odd-numbered topics were used for testing (55 total). We report mean average precision (MAP) and precision at rank 30, which was the primary metric used in the TREC 2011 Microblog evaluation. In our experiments, the statistical significance of effectiveness differences were determined using one-sided paired t-tests; results are reported using the symbols shown in Table 2.

All experiments were performed with the Indri search engine.[2] The baseline condition (QL) uses the standard query likelihood approach with Dirichlet smoothing ($\mu = 2500$), retrieving no more than 1000 results per topic. All temporal retrieval models were implemented by reranking the originally returned documents. This means that they can be applied even if we do not have direct access to the entire document collection, as is the case with the "evaluation as a service" approach implemented in the TREC 2013 Microblog track [14, 13].

[2] `http://www.lemurproject.org/indri/`

Table 2: Symbols indicating statistically significant change for data reporting.

Symbol	Description
$\circ$	$p < 0.05$: improve against the QL baseline
$\bullet$	$p < 0.01$: improve against the QL baseline
$\triangle$	$p < 0.05$: improve against the recency prior
$\blacktriangle$	$p < 0.01$: improve against the recency prior
$\dagger$	$p < 0.05$: improve against the WIN method
$\ddagger$	$p < 0.01$: improve against the WIN method

Table 3: Effectiveness measures on held-out test data (odd-numbered topics). Results show mean average precision (MAP) and precision at 30 (P30).

	MAP	P30
QL	0.2363	0.3473
Recency	0.2467°	0.3642°
WIN	0.2407	0.3515
KDE (uniform)	0.2457°	0.3618°
KDE (score-based)	$0.2505^{\bullet\dagger}$	0.3606°
KDE (rank-based)	$0.2546^{\bullet\triangle\dagger}$	$0.3709^{\bullet\ddagger}$
KDE (oracle)	$0.2843^{\bullet\blacktriangle\ddagger}$	$0.4024^{\bullet\blacktriangle\ddagger}$

5.1 Temporal Feedback

Table 3 compares the effectiveness of our KDE approach using the three different weighting schemes described in Section 4.2. Our techniques were compared against the following methods:

- QL: Simple query likelihood (no feedback and no temporal conditioning).
- Recency: Li and Croft's recency prior method [12]. When tuning λ, we measure time in fractions of a day (cf. [20]).
- WIN: The moving window method proposed by Dakka et al. [2].

The results suggest that our KDE approach improves retrieval effectiveness significantly over a purely lexical baseline (QL) and at least one of the previously published temporal models (WIN). However, it unclear whether any of the weighting schemes for KDE systematically yields better results than the others. As we might expect, non-uniform weights appear to boost effectiveness over an estimate that does not take advantage of lexical evidence. However, it is unclear whether score-based or rank-based weighting is more effective. Lacking strong evidence one way or the other, the remainder of this paper relies on the score-based weights, as this approach eliminates an extra free parameter.

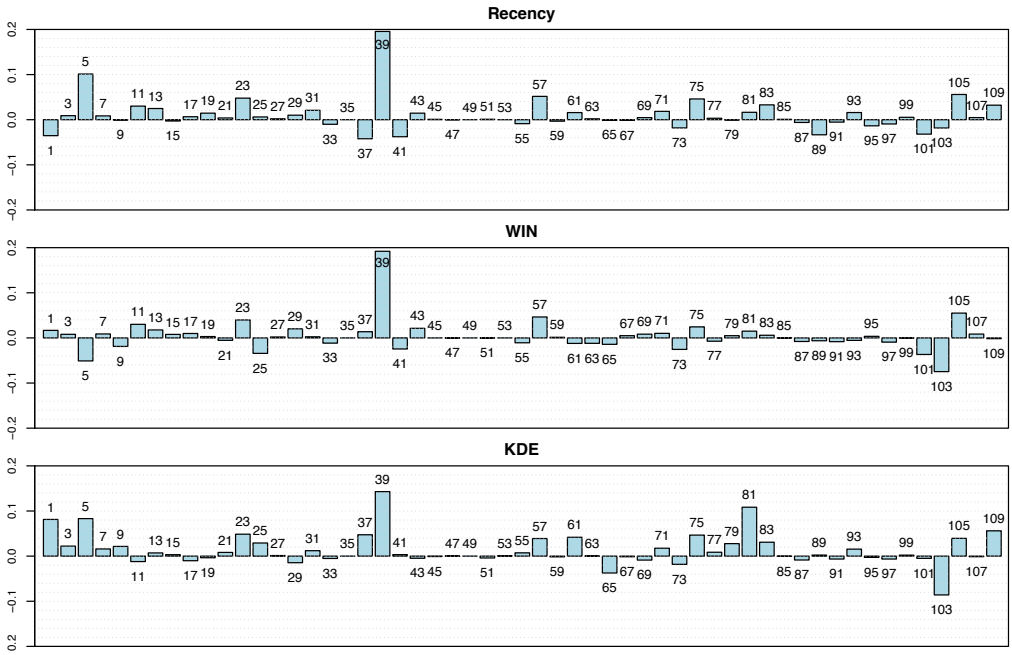

Figure 3: Per-query differences in average precision for each temporal model vs. the query-likelihood baseline.

Figure 3 plots per-topic differences in average precision obtained by each temporally-informed method versus the baseline QL run (data in the KDE panel uses score-based weights). The magnitude of change is query-dependent, suggesting that query-specific analyses are important. Consider topic 37 "Giffords recovery", which we examined in the visualization in Figure 1: effectiveness decreased with recency priors, which is expected since our visualization does not show any relevant tweets near the query time. In contrast, both the WIN and our KDE approach increase effectiveness, although our approach beats the WIN approach.

On the other hand, for topic 103 "Tea Party Caucus", all temporally-informed techniques are less effective than the QL baseline, with the kernel method incurring the biggest decline. This case is interesting insofar as relevance for the query could have both temporal and non-temporal aspects. The Tea Party political movement has an ongoing presence on the American political stage, but its influence is punctuated by time-bound news stories. On the whole, however, improvements from our KDE technique over the QL baseline are consistently higher than the other methods (Table 3). Aside from topic 103, even when KDE hurts effectiveness, its effect is usually smaller than either recency priors or WIN.

Given the results shown in Table 3, a natural question is: what is the upper bound on improvements obtainable by using f_Q, the density of actual relevant documents? To answer this question, we created an oracle condition, where f_Q was estimated using all known relevant documents in the test collection, according to the *true feedback-based weights* method discussed in Section 4.2 (using the score-based approach z_i^s). As discussed, we arbitrarily set $c = 1$ in Eq. (19) without any tuning. Note that this experimental condition still relies on KDE, so it caps the effectiveness upper bound for this particular estimation technique.

Results of the oracle condition are shown in the last row of Table 3. As we would expect, the oracle run outperforms all

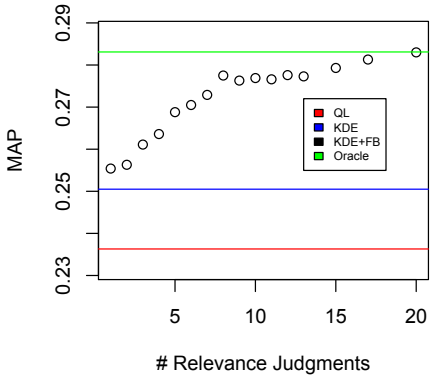

Figure 4: Effectiveness of KDE given an increasing number of user-supplied relevance judgments.

others at statistically significant levels. But the gulf between the oracle and more realistic runs is smaller than the table suggests. Figure 4 compares the results of our KDE-based temporal feedback to the oracle condition. In the figure, horizontal lines indicate the effectiveness of various techniques in terms of MAP: baseline query likelihood is shown in red, KDE with score-based weights in blue, and the oracle in green (i.e., estimating f_Q with all human relevance judgments). With the black dots, we simulate the effectiveness of an interactive retrieval system whereby users can provide relevance judgments. We emphasize that the x-axis denotes *total judgments*, not the number of *positive judgments*. This is important because, for many queries, the top k (say, ten) retrieved documents contained no relevant examples. For these queries, since we do not take advantage of negative judgments, the average precision in the feedback condition is unchanged from simple, "unsupervised" KDE. Note that ef-

fectiveness is computed over all topics, even for those where the addition of human relevance judgments made no impact.

Nevertheless, a clear trend emerges from Figure 4. Not surprisingly, obtaining more relevance judgments increases the effectiveness of our density estimates. But what is surprising is the rate of improvement: sixteen judgments allow us to reach roughly oracle effectiveness. Furthermore, a modest number of judgments (e.g., five) advances the KDE approach almost halfway to oracle effectiveness.

5.2 Integration with Lexical Feedback

Results from the previous section suggest that our formulation of temporal feedback based on kernel density estimation is effective, compared to both a query-likelihood baseline and two other temporal retrieval models. However, these results do not answer a related question: is the improvement that we see due to a signal that is different from information gleaned from document content? Perhaps traditional (lexical) relevance feedback implicitly captures whatever signal we obtain from temporal feedback. In other words, is the temporal cluster hypothesis distinct from the classic cluster hypothesis?

In this section, we experimentally show that the answer is *yes*. Effectiveness improvements from temporal feedback are additive with improvements from lexical feedback, which shows that the temporal signal we are exploiting exists independently of document content.

In our experiments, we supplemented a standard lexical feedback method with different temporal retrieval models, in the context of pseudo-relevance feedback and simulated "true" relevance feedback. In both cases, the lexical feedback method is Lavrenko and Croft's relevance model RM3 [11]:

> *RM3:* Simple relevance models with $k = 50$ pseudo-relevant documents and $n = 20$ feedback terms. The feedback model is interpolated against the original query terms with weight γ.

For clarity, we review the definition of relevance models:

$$P(w|R_Q) \quad = \quad \sum_{D \in \mathcal{D}} P(D)P(w|D) \prod_{i=1}^{n} P(q_i|D). \quad (20)$$

where q_i is the i^{th} query term in a query that is n words long. The relevance model $P(w|R_Q)$ for Q is simply a weighted average of the terms in all documents, where the weights are the query likelihood scores. In the RM3 variant, the quantity in Eq. (20) is interpolated with the observed query according to a mixing parameter γ. We report results with $\gamma = 0.5$, the Indri default.[3]

When using feedback with temporal information, we followed this sequence:

1. Retrieve initial set of documents.
2. Apply the temporal retrieval model (Recency, WIN, or KDE) to rerank results.
3. From the top k reranked documents, estimate feedback models: in the pseudo-relevance feedback case, this involves selecting the top k documents and assuming that they are relevant.
4. Run retrieval with feedback model.
5. Rerank final results using the same temporal model.

[3]Training γ led to inter-system comparisons very similar to those reported here. Thus, we chose to fix γ at the Indri default during experimentation.

Table 4: Retrieval effectiveness in the context of lexical pseudo-relevance feedback, with RM3 as the baseline. Each temporal retrieval model augments lexical feedback via re-ranking both before and after estimating relevance models.

	MAP	P30
RM3	0.2897	0.3843
Recency	0.2898	0.3873
WIN	0.2901	0.3927
KDE	$0.3014^{\bullet\blacktriangle\ddagger}$	$0.4079^{\circ\triangle\ddagger}$

The results of integrating temporal retrieval models with RM3 are shown in Table 4. We see that neither recency priors nor the WIN variant yields much in addition to simply performing pseudo-relevance feedback. Our KDE approach, on the other hand, is significantly more effective than RM3. This result is reinforced by Figure 5, where we plot per-topic average precision differences between each temporal model and RM3. Aside from the obvious differences in magnitude of change obtained by our KDE method versus Recency or WIN, it is also notable that our KDE method's effectiveness with pseudo-relevance feedback is "safer" than KDE reranking without pseudo-relevance feedback, as in Figure 3. In each case, although the number of topics that our KDE method either helped or hurt is the same (32 helped and 21 hurt), the magnitude of decline in cases where KDE hurt is smaller when using lexical feedback.

As a final summative evaluation and to deepen our understanding of the interaction between temporal and lexical signals during feedback, we tested effectiveness in the presence of true (i.e., human) relevance judgments, as opposed to the pseudo-relevance feedback methods described above. Results in Figure 4 suggest that temporal feedback by KDE benefits immensely from human relevance judgments, but those results were on a query-likelihood baseline without lexical feedback. We would like to examine the effect of introducing RM3 into the experimental setup.

Methodologically, the experiments with true feedback mirrored the pseudo-relevance feedback runs above, except that relevance information was injected into a run for documents that had been judged by NIST assessors as relevant among the top five documents retrieved (these judgments are introduced in step two and used in all subsequent steps). In other words, this true feedback condition modeled the case where the user judges the first five tweets retrieved by a model. Relevance models were estimated only from explicitly judged documents. If no relevant tweets appeared in a run's top five results, no lexical feedback was performed.

Table 5 summarizes the true feedback results in terms of *residual* effectiveness measures, i.e., each measure was calculated after removing the documents ranked in the top five during the initial run since the (simulated) user had already "read" them. In this case, we see that both Recency and WIN do not significantly improve over RM3, whereas the gains exhibited by our KDE approach are statistically significant. Note that in all cases, the effectiveness metrics were computed over *all* topics, even those that did not con-

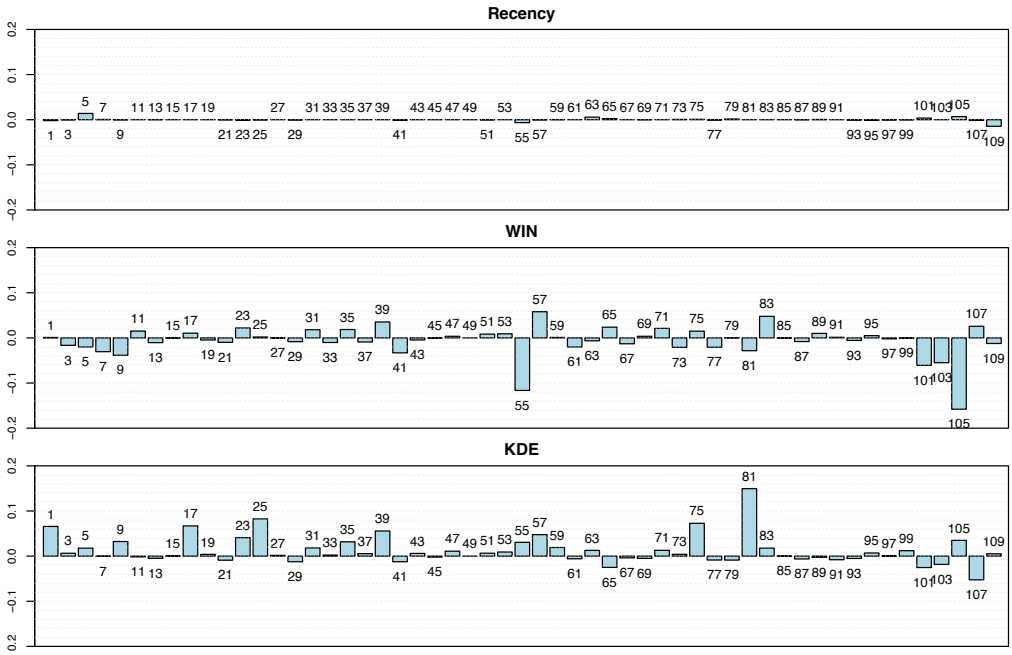

Figure 5: Per-query differences in average precision for each temporal model vs. RM3.

Table 5: Results obtained using five human relevance judgments. Measures are *residuals* (omitting results seen during feedback).

	MAP	P30
RM3	0.2363	0.3534
Recency	0.2320	0.3594
WIN	0.2332	0.3642^{Δ}
KDE	$0.2404^{\Delta\dagger}$	$0.3745^{\bullet\Delta\dagger}$

tain a relevant tweet in the top five initial results. This is a conservative evaluation approach and quantifies effectiveness improvements over a broad range of topics, not just those which are particularly beneficial for our technique.

6. DISCUSSION

Implications of the Temporal Cluster Hypothesis. The temporal cluster hypothesis frames temporal retrieval as a matter of estimating proximity along a single dimension—in this case, via the density of $f(T|Q)$, the temporal density of relevance. This amounts to treating time as simply another feature in a ranking method. This approach is simple and has intuitive appeal, not to mention many options for practical implementation in the learning to rank framework.

Beyond what we have explored here, there are other ways in which a notion of proximity could inform an assessment of similarity as a feature for ranking. For example, we might inform a query classifier of the (dis-)similarity of f_Q from some null distribution f_0 (such as the distribution of the entire collection). Another way to think about this is as follows: a broader interpretation of the classic cluster hypothesis is that documents with similar content profiles are likely relevant to the same information need. We could replace "content profile" with "temporal profile" as a more general formulation of the temporal cluster hypothesis—timeline proximity is merely one example of a temporal profile, but there are other temporal signals that could be valuable for ranking.

Model Generalizations. Although our work is couched in the broader context of the temporal cluster hypothesis, in this paper we focused on improving the effectiveness of tweet search as a concrete application. This choice was motivated by the temporal nature of tweet search and the substantial interest in social media by researchers today.

However, there is no reason to think that the methods we proposed would not generalize to other timestamped document collections. Thus, an interesting avenue of future work is to extend our analysis to other domains such as collections of news articles. In particular, two factors make this extension appealing:

1. Diverse temporal nature of queries

2. Diversity of collection timespans.

With respect to the first point, relevance in tweet search is, almost by definition, temporally conditioned. Moreover, the semantics of this temporality often lends itself to a simple promotion of recent information. But in other collections, it is likely that we would find test queries that have quite different temporal dynamics, including topics where time is not a helpful signal. A core problem that remains to be addressed (not only in our work, but in the literature at large) is how best to approach the varied nature of temporality that bears on relevance in heterogeneous retrieval settings.

The second point speaks to the fact that the Tweets2011 corpus spans a relatively small window of time—roughly two weeks. Many of the TREC news collections, on the other hand, span months or years. How temporally-informed retrieval methods generalize to longer time horizons is an important, open question.

On both of these counts, we suspect that the flexibility of the kernel-based approach will be an asset. Since the bandwidth selection methods for KDE depend on observable features of the data (i.e., timestamps) themselves, it seems likely that longer windows will not pose a problem for our general approach. This is in contrast to the use of recency priors, where the exponential rate parameter and the unit used to represent time are tightly coupled. Likewise, bandwidth selection allows temporal influence to shrink in the face of evidence that the density f_Q lacks any pronounced modes. In theory this should allow KDE-informed ranking to scale its influence when faced with non-temporal queries.

7. CONCLUSION

For much of the history of information retrieval, researchers have treated queries and document collections as mostly static. Recently, however, there is a growing recognition that time plays an important role in many aspects of search—in this paper, we have explored temporal feedback using kernel density estimation for tweet search and demonstrated its effectiveness under a variety of experimental conditions. Although the specific application domain is independently interesting, we hope that our formulation of the temporal cluster hypothesis will be a more lasting contribution, in providing a general principle for future explorations in temporal retrieval.

8. ACKNOWLEDGMENTS

This work was supported in part by the U.S. National Science Foundation under Grant Nos. 1144034, 1217279, and 1218043. Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect the views of the sponsor. Additional support was provided by the Dutch National Institute for Mathematics and Computer Science (CWI).

9. REFERENCES

[1] E. Adar, J. Teevan, S. T. Dumais, and J. L. Elsas. The web changes everything: Understanding the dynamics of web content. In *WSDM*, pages 282–291, 2009.

[2] W. Dakka, L. Gravano, and P. G. Ipeirotis. Answering general time-sensitive queries. *IEEE Transactions on Knowledge and Data Engineering*, 24(2):220–235, 2012.

[3] M. Efron. Linear time series models for term weighting in information retrieval. *Journal of the American Society for Information Science and Technology*, 61(7):1299–1312, 2010.

[4] M. Efron and G. Golovchinsky. Estimation methods for ranking recent information. In *SIGIR*, pages 495–504, 2011.

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

[6] P. Hall and B. A. Turlach. Reducing bias in curve estimation by use of weights. *Computational statistics & data analysis*, 30(1):67–86, 1999.

[7] N. Jardine and C. J. van Rijsbergen. The use of hierarchic clustering in information retrieval. *Information storage and retrieval*, 7(5):217–240, 1971.

[8] R. Jones and F. Diaz. Temporal profiles of queries. *ACM Transactions on Information Systems*, 25(3):Article 14, 2007.

[9] D. Klakow. Log-linear interpolation of language models. In *ICSLP*, 1998.

[10] J. Lafferty and C. Zhai. Probabilistic relevance models based on document and query generation. In *Language modeling for information retrieval*, pages 1–10. Springer, 2003.

[11] V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR*, pages 120–127, 2001.

[12] X. Li and W. B. Croft. Time-based language models. In *CIKM*, pages 469–475, 2003.

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

[14] J. Lin and M. Efron. Overview of the TREC-2013 Microblog Track. In *TREC*, 2013.

[15] J. Lin and M. Efron. Temporal relevance profiles for tweet search. In *SIGIR Workshop on Time-aware Information Access*, 2013.

[16] J. Lin and G. Mishne. A study of "churn" in tweets and real-time search queries. In *ICWSM 2012*, pages 503–506, 2012.

[17] D. Metzler and W. B. Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274, 2007.

[18] G. Mishne, J. Dalton, Z. Li, A. Sharma, and J. Lin. Fast data in the era of big data: Twitter's real-time related query suggestion architecture. In *SIGMOD*, pages 1147–1157, 2012.

[19] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff. Overview of the TREC-2011 Microblog Track. In *TREC*, 2011.

[20] M.-H. Peetz and M. de Rijke. Cognitive temporal document priors. In *ECIR*, pages 318–330, 2013.

[21] J. M. Ponte and W. Croft. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281, 1998.

[22] K. Radinsky, K. Svore, S. Dumais, J. Teevan, A. Bocharov, and E. Horvitz. Modeling and predicting behavioral dynamics on the web. In *WWW*, pages 599–608, 2012.

[23] S. J. Sheather and M. C. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 683–690, 1991.

[24] M. Shokouhi and K. Radinsky. Time-sensitive query auto-completion. In *SIGIR*, pages 601–610, 2012.

[25] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Monographs on Statistics and Applied Probability. Chapman & Hall, Boca Raton, 1996.

[26] J. S. Simonoff. *Smoothing Methods in Statistics*. Springer Series in Statistics. Springer, New York, 1998.

[27] I. Soboroff, D. McCullough, J. Lin, C. Macdonald, I. Ounis, and R. McCreadie. Evaluating real-time search over tweets. In *ICWSM 2012*, pages 579–582, 2012.

[28] M. Vlachos, C. Meek, Z. Vagena, and D. Gunopulos. Identifying similarities, periodicities and bursts for online search queries. In *SIGMOD*, pages 131–142, 2004.