Inf Retrieval
DOI 10.1007/s10791-013-9227-2

Using temporal bursts for query modeling

Maria-Hendrike Peetz · Edgar Meij · Maarten de Rijke

Received: 2 October 2012 / Accepted: 23 February 2013

Abstract We present an approach to query modeling that leverages the temporal distribution of documents in an initially retrieved set of documents. In news-related document collections such distributions tend to exhibit bursts. Here, we define a burst to be a time period where unusually many documents are published. In our approach we detect bursts in result lists returned for a query. We then model the term distributions of the bursts using a reduced result list and select its most descriptive terms. Finally, we merge the sets of terms obtained in this manner so as to arrive at a reformulation of the original query. For query sets that consist of both temporal and non-temporal queries, our query modeling approach incorporates an effective selection method of terms. We consistently and significantly improve over various baselines, such as relevance models, on both news collections and a collection of blog posts.

Keywords Information retrieval · Temporal information retrieval · Query modeling

An earlier version of this article appeared as Peetz et al. (2012). In this substantially extended version we add a novel, non-uniform burst prior and carefully evaluate this new prior. We extend the query models presented in Peetz et al. (2012) with a new method to estimate the temporal distribution. We incorporate this method into the query modeling approach from Peetz et al. (2012) and compare it with algorithms for temporal information retrieval. What is also new is that we evaluate the influence of different test collections.

M.-H. Peetz (✉) · E. Meij · M. de Rijke
ISLA, University of Amsterdam, Amsterdam, The Netherlands
e-mail: m.h.peetz@uva.nl

E. Meij
e-mail: edgar.meij@uva.nl

M. de Rijke
e-mail: derijke@uva.nl

Published online: 27 July 2013

1 Introduction

Query modeling is often used to better capture a user's information need and help bridge the lexical gap between a query and the documents to be retrieved. Typical approaches consider terms in some set of documents and select the most informative ones. These terms may then be reweighted and—in a language modeling setting—be used to estimate a query model, i.e., a distribution over terms for a query (Ponte and Croft 1998, Zhai and Lafferty 2001). The selection of the set of documents is crucial: a poor selection may cause topic drift and thus decrease precision with a marginal improvement in terms of recall. Typical approaches base query modeling on information pertinent to the query or the documents (Rocchio 1971), while others incorporate metadata (Kamps 2004), semantic information such as entity types or Wikipedia categories (Bron et al. 2010), or synonyms (Meij et al. 2010). In the setting of social media there have been proposals to obtain rich query models by sampling terms not from the target collection from which documents are to be retrieved, but from trusted external corpora instead (Diaz and Metzler 2006). For queries with an inherent temporal information need such query modeling and query expansion methods might be too general and not sufficiently focused on events the user is looking for.

To make matters concrete, let us consider an example taken from one of the test collections that we are using later in the paper, query 936, *grammys*, from the TREC Blogs06 collection. The Grammy awards ceremony happens once a year and is therefore being discussed mainly around this time. The information need underlying the query *grammys* is about this event and not, for example, a list of grammy awards for a starlet: relevant documents for this query are therefore less likely to be published six months after this event. The temporal distribution of relevant results reflects this observation; see Fig. 1a, in which we plot the number of relevant documents against days, ranging from the first day in the collection to the last. We see a clear peak in the temporal distribution of relevant results around the date of the Grammy Awards ceremony. The temporal distribution for the pseudo-relevant result set for the query *grammys* (Fig. 1b), i.e., the top ranked documents retrieved in response to the query, shows a similar pattern: here, we also see a temporal overlap of peaks. Indeed, in temporally ordered test collections we observe that typically between 40 and 50 % of all documents in a burst of the temporal distribution of the pseudo relevant documents are relevant (see Table 11). Query modeling based on those documents should therefore return more relevant documents without harming precision. That is, we hypothesize that distinguishing terms that occur within documents in such bursts are good candidate terms for query modeling purposes.

Previous approaches to exploiting the transient and bursty nature of relevance in temporally ordered document collections assume that the most recent documents are more relevant (Efron and Golovchinsky 2011) or they compute a temporal similarity (Keikha et al. (2011b) to retrieve documents that are recent or diverse. Keikha et al. (2011) use relevance models of temporal distributions of posts in blog feeds and Dakka et al. (2012) incorporate normalized temporal distributions as a prior in different retrieval approaches, among them relevance modeling methods. Our approach builds on these previous ideas by performing query modeling on bursts instead of recent documents.

We address the following research questions:

1. Are documents occurring within bursts more likely to be relevant than those outside of bursts?
2. Can documents within bursts contribute more useful terms for query modeling than documents selected for relevance models?

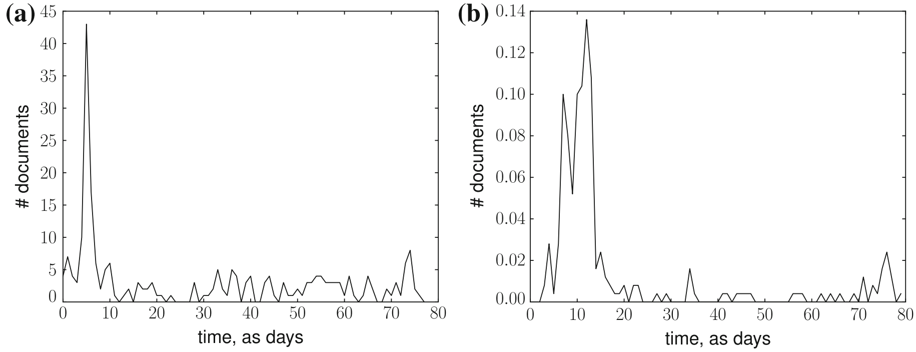


Fig. 1 Temporal distributions of documents for query 936, *grammys*, in the TREC Blogs06 test collection. **a** The relevant documents for query 936, *grammys*. **b** The top ranked document retrieved in response to query 936, *grammys*

3. What is the impact on the retrieval effectiveness when we use a query model that rewards documents closer to the center of the bursts?
4. Does the number of pseudo-relevant documents used for burst detection matter and how many documents should be considered for sampling terms? How many terms should each burst contribute?
5. Is retrieval effectiveness influenced by query-independent factors, such as the quality of a document contained in the burst or size of a burst?

To answer our research questions, we identify temporal bursts in ranked lists of initially retrieved documents for a given query and model the generative probability of a document given a burst. For this we propose various discrete and continuous models. We then sample terms from the documents in the burst and update the query model. The effectiveness of our temporal query modeling approaches is assessed using several test collections based on news articles (TREC-2, 7, and 8) and a test collection based on blog posts (TREC Blog track, 2006–2008).

The main contributions we make in this paper are novel temporal query models and an analysis of their effectiveness, both for time-aware queries and for arbitrary queries. For query sets that consist of both temporal and non-temporal queries, our model is able to find the balance between performing query modeling or not: only if there are bursts and only if some of the top ranked documents are in the burst, the query is remodeled based on the bursts. We consistently improve over various baselines such as relevance models, often significantly so.

In §2 we discuss related work. In §3 we introduce temporal query models and the baseline. We explain the setup of our experiments in §4 and our experimental results are presented and analyzed in §5. We conclude in §6.

2 Related work

A query often consists of only a few keywords which may or may not adequately represent the user's underlying information need. Query modeling aims to transform simple queries to more detailed representations of the underlying information need. Among others, those representations can have weights for terms or may be expanded with new terms. There are two main types of query modeling, global and local. Global query modeling uses collection statistics to expand and remodel the query. An example of global query modeling can be found in Qiu and Frei (1993), using thesaurus and dictionary-based expansion and (Meij and de Rijke 2010) perform semantic query modeling by linking queries to Wikipedia. Local query modeling is based on top retrieved documents for a given query. Typical local query expansion techniques used are the relevance models by Lavrenko and Croft (2001). Richer forms of (local) query modeling can be found in work by Balog et al. (2008, 2010).

For blog (post) retrieval, one often uses large external corpora for query modeling (Diaz and Metzler 2006). Several TREC Blog track participants have experimented with expansion against a news corpus, Wikipedia, the web, or a mixture of these (Java et al. 2006; Weerkamp et al. 2009, 2012; Zhang and Yu 2006). For blog retrieval, the motivation underlying most of these approaches is to improve the estimation of the query representation, often trying to make up for the unedited nature of the corpus from which posts or blogs need to be retrieved. Our approach tries to address this problem by focusing on bursts in the collection.

Temporal information retrieval (IR) is a difficult problem. Alonso et al. (2011) state the main challenges of temporal IR, ranging from extracting mentions of time within documents and linking them (like Verhagen and Pustejovsky 2008), to spatio-temporal information exploration (e.g., Martins et al. 2008), and temporal querying (such as Odijk et al. 2012). In this paper we approach two research questions raised by Alonso et al. with respect to real-time search and temporal querying:

1. What is the lifespan of the main event?
2. How can a combined score for the textual part and the temporal part of a query be calculated in a reasonable way?

Under the assumption that more recent documents are more likely to be read and deemed relevant, early work by Li and Croft (2003) creates an exponential recency prior. Corso et al. (2005) rank news articles using their publication date and their interlinkage. Jones and Diaz (2007) classify queries according to the temporal distribution of result documents into temporal and non-temporal queryies. Efron and Golovchinsky (2011) expand on Li and Croft (2003)'s recency prior by directly incorporating an exponential decay function into the query likelihood, while Peetz and de Rijke (2013) examine the performance of a range of cognitively motivated document priors. Recent work focusses not only on retrieving recent URLs (Dong et al. 2010) or tweets (Massoudi et al. 2011) but also on detecting temporally active time periods (salient events) in the temporal distribution of pseudo-relevant documents (Amodeo et al. 2011); Dakka et al. 2012; Keikha et al. 2011 Peetz et al. 2012). Berberich et al. (2010) detect temporal information need, manifested as temporal expressions, in the query and incorporate them into a language model approach. Amodeo et al. (2011) select top-ranked documents in the highest peaks as pseudo-relevant, while documents outside peaks are considered to be non-relevant. They use Rocchio's algorithm for relevance feedback based on the top-10 documents. Dakka et al. (2012) incorporate temporal distributions in different language modeling frameworks; while they do not actually detect events, they perform several standard normalizations to the temporal distributions. We, however, do detect events. Unlike our proposed work, Dakka et al. do not include the (pseudo)-relevance assessments into the creation of the temporal distribution, but use global temporal distributions as a prior.

The application of different time series analysis tools to temporal distributions in IR is not new. Chien and Immorlica (2005) analyze query logs for temporal similarities. Jones and Diaz (2007) classify queries as temporal or not by detecting bursts in the distribution of the top-N retrieved documents. They used a HMM burst detector proposed by Kleinberg (2002). Wang et al. (2007) make use of temporal correlations between collections for topic modelling. For temporally ordered collections, Amodeo et al. (2011) detect bursts and incorporate this knowledge in Rocchio feedback methods and Efron (2010) uses linear time series to estimate the collection frequency of a term. Neither approach uses the temporal distribution of pseudo-relevant result sets for query modeling.

Focusing on blog retrieval with temporal elements, the number of approaches to blog (post) retrieval that make specific use of temporal aspects is limited. Weerkamp and de Rijke (2008) use timeliness of a blog post as an indicator for determining credibility of blog posts. For blogger finding, Keikha et al. (2011b) propose a distance measure based on temporal distributions and Seki et al. (2007) try to capture the recurring interest of a blog for a certain topic using the notion of time and relevance. Under the assumption that the most recent tweets are the most relevant, Massoudi et al. (2011) use an exponential decay function for query expansion on microblogs. For blog feed retrieval, Keikha et al. (2011) use temporal relevance models based on days and the publications in the blog feeds. Peetz et al. (2012) use salient events for query modeling in news and blog data.

Similar to the approaches just listed we propose a combined score for the textual part and the temporal part of a query by selecting documents based on temporal bursts and using those documents as a basis for query modeling. This is different from Keikha et al. (2011), who do not detect bursts but combine language models based on days of a blog feed. We also differ from Amodeo et al. (2011), because we use query modeling and not query expansion and present different approaches to model the probability of a document to be in a burst. Finally, our task is different from Efron (2010) and Li and Croft (2003) because we focus on queries for a certain event as opposed to queries asking for recent events.

Based on previously published work we use various subsets of queries from test collections based on newspaper articles: Li and Croft (2003) and Efron and Golovchinsky (2011) use a recency biased subset while Dakka et al. (2012) use a general temporal subset of queries. The precise splits into temporal and recent query sets that we use can be found in Appendix 1.

3 Temporal query models

Our temporal query model is based on pseudo-relevance feedback: we aim to improve a query by first retrieving a set of documents, $\mathcal{D}$, and then identifying and weighting the most distinguishing terms from those documents; the remodeled query is used to retrieve the final ranked list of documents. We proceed in this standard fashion, but take into account the temporal distribution of the documents in $\mathcal{D}$. We consciously decided to make our model discrete. For one, aggregating time points into temporal bins is natural for these types of collections. For blogs it has been noted that the publishing volume is periodic and depends on the daytime (Tsagkias et al. 2010). A granularity less than a day will therefore introduce noise in the bursts, due to the chrono-biological idiosyncrasies of human beings. Similarly for news documents: newspapers from the time period we employ will rarely publish more than one or two articles per day. Thus, a granularity smaller than a month will lead to very few bursts. Furthermore, using a finer granularity would result in near-uniform peaks and therefore we would not be able to identify bursts. Table 1 provides an overview over the notation used in this paper.

Table 1 Notation used in the paper

Notation	Explanation
q	Query
N	Number of documents to retrieve for burst detection
N_B	Number of documents to retrieve for term selection
M	Number of terms used to model a burst
$\mathcal{D}^q, \mathcal{D}$	The set of top N retrieved documents for query q
$\hat{\mathcal{D}}^q, \hat{\mathcal{D}}$	Set of top $\hat{N}$ retrieved documents for query q
D, D_j	Document
$w \in D$	Term in the document D
$w \in q$	Term in the query q
$T(D)$	Publishing time of a document D
$R(D)$	Retrieval score of a document D
l	Length of the time interval for binning the documents
$\min(\mathcal{D})$	Document in the set of documents $\mathcal{D}$ that is oldest with respect to publishing time
$\mathrm{Time}(D)$	Normalize publishing time of a document D
$\mathrm{Bin}(D)$	Time bin of a document D
$\mathrm{bursts}(\mathcal{D})$	Set of bursts in $\mathcal{D}$
W, W_B	Terms used for query modeling
$t_{\mathcal{D}}(i)$, $t(i)$	Time series based on the publishing times of the documents in $\mathcal{D}$
$t_{\mathcal{D}_B}(i)$	Time series over a subsequence $\mathcal{D}_B$
$\mathrm{bursts}(\mathcal{D})$	Bursts in the $t_{\mathcal{D}}(i)$
B	A burst
$\mathcal{D}_B$	Documents published within the burst B
$\max(B)$	Peak in a burst B with the highest value for the time series t
$\sigma(t(i))$, σ	Standard deviation of temporal distribution $t(i)$
$\mu(t(i))$, μ	Mean deviation of temporal distribution $t(i)$
α	Discrete decay parameter
γ	Decay parameter
k	Number of neighboring documents of a document in a burst

Consider Fig. 1a again, which shows the temporal distribution of relevant documents for a single query (query 936, *grammys*, from the TREC Blogs06 collection). We observe that the ground truth for the query *grammys* has more relevant documents on some days than on others and experiences *bursts*; a burst appears on days when more documents are published than usual. Some of the documents might be near duplicates: those documents provide a strong signal that their terms are relevant to the event in the burst. It is inherent to the assumptions of the algorithm, that the documents in a burst are textually close. Near-duplicate elimination might therefore remove important information. Informally, a burst in a temporal distribution is a time period where more documents are published than usual. Bursts are often related to events relevant to the query: in this case the ceremony for the

Grammy Awards triggered the publishing of relevant documents. Now consider Fig. 1b again, which shows the temporal distribution of the documents in the result set. Again, we see bursts around the time of the ceremony. This observation gives rise to the key assumption of this paper, that documents in bursts are more likely to be relevant.

Algorithm 1: QMB: Query Modeling using Bursts.

```
    Input: $q$, query
    Input: $N$, number of documents to retrieve for burst detection
    Input: $\hat{N}$, number of documents to retrieve for burst modeling
    Input: $M$, number of terms used to model a burst
    Input: $\mathcal{D}$, set of top $N$ retrieved documents for query $q$
    Input: $\hat{\mathcal{D}}$, set of top $\hat{N}$ retrieved documents for query $q$
    Input: bursts($\mathcal{D}$), the set of temporal bursts in $\mathcal{D}$
    Output: $W$, the terms used for query modeling
    Output: $P(w \mid q)$, for all $w \in W$ the reweighted probability given query $q$
 1  foreach $B \in$ bursts($\mathcal{D}$) do
 2      foreach $D \in B \cap \hat{\mathcal{D}}$ do
 3          foreach $w \in D$ do
 4              update $P(w \mid B)$ by adding $\frac{1}{\hat{N}} P(D \mid B) \cdot P(w \mid D)$ to it
 5          end
 6          $W$ is the set of top-$M$ terms based on $P(w \mid B)$;
 7          foreach $w \in W$ do
 8              update $P(w \mid q)$ by adding $P(B \mid \text{bursts}(\mathcal{D})) \cdot P(w \mid B)$ to it
 9          end
10      end
11  end
```

Algorithm 1 summarizes our approach. Given a query q, we first select a ranked list of top-N pseudo-relevant documents, $\mathcal{D}$. In $\mathcal{D}$ we identify bursts (bursts($\mathcal{D}$)). Within $\mathcal{D}$ we then select a second ranked list of top-$\hat{N}$ documents ($\hat{\mathcal{D}}$) of length $\hat{N}$. For all identified bursts we select the intersection of documents in the burst and in the top-$\hat{N}$ documents. In line 4 of Algorithm 1, those documents are used to estimate $P(w \mid B)$, the probability that a term is generated within a burst; we include different generative probabilities $P(D \mid B)$ for each document D.

In line 6, we select the top-M terms per burst with the highest probability of being generated by this burst. Finally, in line 8, we estimate the probability that a term is generated by a query $P(w \mid q)$ and we merge the terms for each burst, weighted by the quality of the documents within the burst or size of the burst $P(B \mid \text{bursts}(\mathcal{D}))$. The quality of a document is based on textual features that capture how well the document has been written (e.g., correctness of spelling, emoticons), which are typical text quality indicators (Weerkamp and de Rijke 2012).

Formally,

$$\hat{P}(w \mid q) = \sum_{B \in \text{bursts}(\mathcal{D})} \frac{P(B \mid \text{bursts}(\mathcal{D}))}{\hat{N}} \sum_{D \in \mathcal{D}_B} P(D \mid B) P(w \mid D). \qquad (1)$$

Lines 1–11 in Algorithm 1 provide an algorithmic view on Eq. 1. The key components on which we focus are the document prior $P(D \mid B)$ in Sect. 3.3 and the burst normalisation ($P(B \mid \text{bursts}(\mathcal{D}))$) in Sect. 3.4 We start by defining bursts and detailing the query model.

3.1 Bursts

Informally, a burst in a temporal distribution of documents is a set of time periods in which "unusually" many documents are published. Often, what is "normal" (or the mean) might change over time. In the collections we are considering, however, the mean is rather stable and the distribution stationary. For longer periods, estimating a time-dependent, dynamic mean can easily be accommodated with a moving average estimation.

Consider the example in Fig. 2. The blue (striped) time bin peaks and forms a burst together with the red (dotted) bin to its left. The right red (dotted) bin is not peaking as it does not contain enough documents.

Formally, let $\mathcal{D}^q$ (or $\mathcal{D}$ when the query q is clear from the context) denote the set of top-N documents retrieved for a query q. Let $R(D)$ and $T(D)$ be the relevance score and publication time point of document D, respectively.[1] Let l be the distance between two time points; l can be phrased in terms of days, months, or years. Further, let $\min(\mathcal{D})$ be the oldest publication time of a document in $\mathcal{D}$. The *time normalised publication time* of a document D is

$$\text{time}(D) = \frac{T(D) - \min(\mathcal{D})}{l},$$

and the *binned time* of D is $\text{bin}(D) = \lfloor \text{time}(D) \rfloor$.

Let $i \in \mathbb{N}$ denote a time bin, then a discrete time series $t_{\mathcal{D}}(i)$, for a set of documents $\mathcal{D}$, is the sum of ranking scores of the documents,

$$t_{\mathcal{D}}(i) = \sum_{\{D \in \mathcal{D} : \text{bin(D)=i}\}} R(D). \tag{2}$$

We write $t(i)$ instead of $t_{\mathcal{D}}(i)$ whenever $\mathcal{D}$ is clear from the context. The mean (standard deviation) μ (σ) is the mean (standard deviation) of the time series $t(i)$. A time bin i (lightly) *peaks*, when $t(i)$ is two (one) standard deviation(s) bigger than the mean.

A *burst* for a set of documents $\mathcal{D}$ is a sequence $B \subseteq \mathbb{N}$ such that

- at least one time bin $i \in B$ peaks, thus $t_{\mathcal{D}}(i)$ is at least two standard deviations bigger than the mean ($t(i) + 2\sigma > \mu$);
- and for all time bins $i \in B, t(i)$ is at least one standard deviation bigger than the mean ($t(i) + 1\sigma > \mu$).

A time series can have multiple bursts. The set of maximal bursts for $\mathcal{D}$ is denoted as $\text{bursts}(\mathcal{D})$.[2] Given a sequence of time bins B, its set of documents is denoted as $\mathcal{D}_B = \{D \in \mathcal{D} : \text{bin}(D) \in B\}$. The time series over the subsequence B is $t_{\mathcal{D}_B}(i)$.

It is sometimes useful to adopt a slightly different perspective on time series. So far, we have used the sum of ranking scores (see Eq. 2). An alternative approach for the estimation of the time series would be to use the counts of documents:

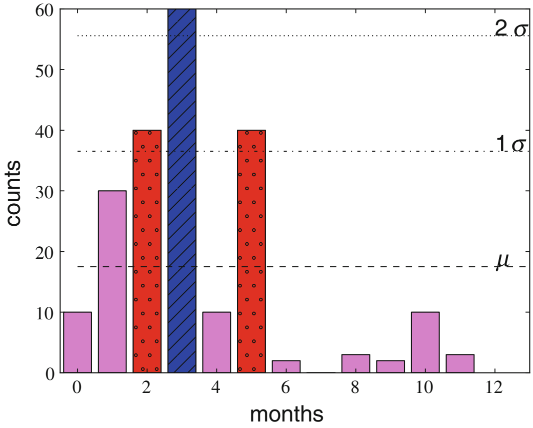
$$t'_{\mathcal{D}}(i) = |\{D \in \mathcal{D} : \text{bin}(D) = i\}|. \tag{3}$$

For the estimation of the bursts and peaks we proceed similar as for the time series introduced in Eq. 2. Unless stated otherwise, a time series is estimated using Eq. 2.

[1] We assume that $R(D)$ takes values between 0 and 1.

[2] Burst B_1 is maximal if there is no burst B_2 such that $B_1 \subseteq B_2$ and $B_1 \neq B_2$.

Fig. 2 Example time series: time bins 3 and 4 form a burst (blue and striped) and bin 3 peaks (red and dotted) (Color figure online)



3.2 Term reweighting

At the end of this section we introduce the score of a document for a query (Eq. 17), used in line 4 of Algorithm 1. To this end we need to determine the probability of a term being generated by a burst (Eq. 4 below) and how to combine the probabilities for all bursts (Eq. 5 below).

Formally, let $\hat{\mathcal{D}}^q$ (or $\hat{\mathcal{D}}$ if q is clear from the context) be the top-$\hat{N}$ documents retrieved for a query q. For a burst B, the suitability of a term w for query modeling depends on the generative probability of the documents ($D \in B$) in the burst, $P(D \mid B)$:

$$P(w \mid B) = \frac{1}{\hat{N}} \sum_{D \in \mathcal{D}_B} P(D \mid B) P(w \mid D), \tag{4}$$

where $P(w \mid D)$ is the probability that term w is generated by document D. The summation in Eq. 4 is over documents in $\mathcal{D}_B$ only to avoid topic drift.

The probability $\hat{P}(w \mid q)$ of a term w given a query q is

$$\hat{P}(w \mid q) = \sum_{B \in \text{bursts}(\mathcal{D})} P(B | \text{bursts}(\mathcal{D})) P(w \mid B). \tag{5}$$

This is the same as Eq. 1. Since we only use a subset of the possible terms for query modeling, we need to normalize. For each burst B, the set of M terms W_B used for query modeling are the terms with the highest probability of a burst B without being stopwords; the set W of all terms is denoted

$$W = \bigcup_{B \in \text{bursts}(\mathcal{D})} W_B.$$

Let $|q|$ be the number of terms in query q and $\text{tf}(w, q)$ the term frequency of term w in query q. We normalize $\hat{P}(w \mid q)$ according to

$$\hat{P}^*(w \mid q) = \frac{1}{|q| + \sum_{w' \in W} \hat{P}(w' \mid q)} \begin{cases} \text{tf}(w, q) & \text{if } w \in q, \\ \hat{P}(w \mid q) & \text{if } w \in W \setminus q, \\ 0 & \text{else.} \end{cases} \tag{6}$$

This concludes the definition of the query model.

3.3 Generative probability of a document in a burst

We continue by describing the remaining components. In particular, for the estimation of $P(w \mid B)$ (Eq. 4) we are missing the probability of a document generated by a burst, $P(D \mid B)$, which is introduced in this Sect. (3.3). Finally, we estimate the probability of a burst given other bursts, $P(B \mid \text{bursts}(\mathcal{D}))$ (Sect.3.4)

Our hypothesis is that bursts contain the most relevant documents. But how can we quantify this? We assume a generative approach, and introduce different functions $f(D, B)$ to approximate $P(D \mid B)$ in this section. One discrete approximation assumes that the most relevant documents are in the peaking time bins of a burst (i.e., (two standard deviations above mean; see Eq. 8 below). This could potentially increase the precision. However, assuming all documents in a burst to be generated uniformly (as we do in Eq. 7 below), we may find more terms, but these are not necessarily as useful as the terms estimated from the documents in the peaks of bursts (see Eqs. 8 and 9 below). To achieve a smoother transition between the peak of a burst and the rest of the burst, we consider multiple smoothing functions. We compare one discrete step function and four continuous functions. The discrete function gives lower probability to documents in bursts that are outside peaks than to documents that are inside peaks; documents outside bursts are not considered for estimation. The continuous functions should alleviate the arbitrariness of discrete functions: we introduce a function based on the exponential decay function from Li and Croft (2003) (see Eq. 10 below) and augment it with a k-nearest neighbor kernel (see Eq. 12 below). The discrete approximations for $P(D \mid B)$ are $f_{\mathrm{DB0}}(D, B)$, $f_{\mathrm{DB1}}(D, B)$ and $f_{\mathrm{DB2}}(D, B)$, while the continuous approximations are $f_{\mathrm{DB3}}(D, B)$ to $f_{\mathrm{DB6}}(D, B)$. We begin with the former.

Discrete functions. For simple approximations of $P(D \mid B)$ we view burst detection as a discrete binary or ternary filter. The approximation below only uses documents in a burst and assigns uniform probabilities to documents in bursts:

$$f_{\mathrm{DB0}}(D, B) = \begin{cases} 1 & \text{if } D \in \mathcal{D}_B, \\ 0 & \text{else.} \end{cases} \tag{7}$$

We refer to this approach as DB0.

Documents in the onset or offset of a burst may be noisy in the sense that they may only be marginally relevant. For our running example query *grammy*, documents before the event may be anticipations or event listings, but they are unlikely to contain a detailed description of actual incidents at the ceremony. Articles published long after the Grammy Awards may be imprecise and superficial as the retention of events decays over time and the author may have forgotten details or remember things differently. Also, the *event* may be very important during the time period, but later the *award* becomes more important and is mentioned more in relation to the award winners.

Compared to DB0, a more strict approach to estimating whether a document is in a burst is a binary decision if the document is in a peak of the burst or not:

$$f_{\mathrm{DB1}}(D, B) = \begin{cases} 1 & \text{if } D \in \mathcal{D}_B \text{ and } \mathrm{bin}(D) \text{ peaks}, \\ 0 & \text{else.} \end{cases} \tag{8}$$

Here, we ignore all documents that are not in a peak of a burst. Alternatively, we can assume that documents in a peak are more relevant than the documents published outside the peaks, but still published in the burst. The documents inside the peak should therefore have more influence in the query modeling process: the terms in the documents inside the peak should be more likely to be used in the remodeled query. We propose to use a simple step function that assigns lower probabilities to documents outside peaks, but inside bursts,

$$f_{\mathrm{DB2}}(D,B)=\begin{cases}\alpha & \text{if } D\in\mathcal{D}_B,\\ 1-\alpha & \text{if } D\in\mathcal{D}_B \text{ and } \mathrm{bin}(D) \text{ peaks},\\ 0 & \text{else},\end{cases} \tag{9}$$

with $\alpha < 0.5$.

Continuous functions. In previously published approaches to temporal query modeling, continuous functions are used with term reweighting with a decay or a recency function depending on the entire result set. The most commonly used decay function is exponential decay (Efron and Golovchinsky 2011; Li and Croft 2003; Massoudi et al. 2011). We use similar functions to estimate the probability of a document being generated by a burst. The approximation $f_{\mathrm{DB3}}(D, B)$ decreases exponentially with its distance to the largest peak of the burst max(B), the global maximum of the time series $t_{\mathcal{D}_B}(i)$ ($\mathrm{argmax}_i\, t_{\mathcal{D}_B}(i)$). Formally, let $\mathrm{time}(D)$ denote the normalized publishing time of document D; then

$$f_{\mathrm{DB3}}(D,B)=e^{-\gamma(|\max(B)-\mathrm{time(D)}|)}, \tag{10}$$

where γ is an (open) decay parameter.

Result sets of queries may have different temporal distributions: some bursts are wider and can last over multiple days, whereas some distributions may have short bursts lasting a single day. Using a global decay parameter may ignore documents at the fringe of the burst or include documents far outside the burst. We propose a burst-adaptive decay. This decay function is a gaussian fitted over the burst by estimating the mean and variance of the burst. We call this *adaptive* exponential decay function, and define

$$f_{\mathrm{DB4}}(D,B)=e^{-\frac{|\max(B)-\mathrm{time(D)}|}{2\sigma(t_{\mathcal{D}_B}(i))^2}}, \tag{11}$$

where $\sigma(t_{\mathcal{D}_B}(i))$ is the standard deviation for the time series $t(i), i\in B$. The power in this equation says that for wide bursts, that is, bursts with a great variance, the decay is less than for bursts with a single sharp peak.

The temporal distributions of pseudo-relevant ranked document lists can be very noisy and might not accurately express the temporal distribution of the relevance assessments. Smoothing of the temporal distribution may alleviate the effects of such noise (Hamilton 1994). As a smoothing method we propose the use of k-NN (Cover and Hart 1967), where the $\mathrm{time}(D)$ of each document D is the average timestamp of its k neighbors. Let the distance between documents D, D_j be defined as $|\mathrm{time}(D)-\mathrm{time}(D_j)|$. We say that document D_j is a *k-neighbor* of document D ($\mathrm{neighbor}_k(D, D_j)$) if D_j is among the k nearest documents to D. The smoothed probability is then calculated using the exponential decay functions (Eqs. 10 and 11) Formally,

$$f_{\mathrm{DB5}}(D,B)=\frac{1}{k}\sum_{D_j\in\mathrm{neighbor_k(D,D_j)}} f_{\mathrm{DB3}}(D_j|B) \tag{12}$$

and

$$f_{\mathrm{DB6}}(D,B)=\frac{1}{k}\sum_{D_j\in\mathrm{neighbor_k(D,D_j)}} f_{\mathrm{DB4}}(D_j|B). \tag{13}$$

3.4 Burst normalization

We now introduce two approaches to burst normalization, based on quality (Eq. 15) and size (Eq. 16). Bursts within a ranked list for a given query may be focused on one subtopic of the query, the burst can be an artifact of the temporal distribution of the document collection. Or it may be spam or irrelevant chatter related to the query. The latter is especially relevant for blog post retrieval, where it was shown that using quality priors improves retrieval performance (Weerkamp and de Rijke 2008). A burst may also be more important because it contains a large number of documents (see Eq. 16). Based on these intuitions, we propose different methods to reweight bursts.

The *uniform burst normalization* method assumes no difference between the bursts and assigns each burst the same weight

$$P(B \mid \text{bursts}(\mathcal{D})) = \frac{1}{|\text{bursts}(\mathcal{D})|}. \tag{14}$$

Unless explicitly stated otherwise, we only use the uniform normalization from Eq. 14.

When using non-uniform normalization, we assume the overall quality of a burst to be based on the quality of single documents:

$$P_C(B \mid \text{ bursts } (\mathcal{D})) = \frac{1}{|B|} \sum_{D \in \mathcal{D}_B} P(D \mid \text{bursts}(\mathcal{D})), \tag{15}$$

where $P(D)$ is the quality of the document using the best performing quality indicators from (Weerkamp and de Rijke 2008).[3]

We can assume that the quality of a burst depends on its size: the more documents are in a burst, the less probable it is for the burst to be an artifact, so

$$P_S(B \mid \text{bursts}(\mathcal{D})) = \frac{1}{|\mathcal{D}_B|}, \tag{16}$$

where $|\mathcal{D}_B|$ is the number of documents in the burst B.

3.5 Document score

In the previous sections we introduced all probabilities needed to estimate the query model $P(w \mid q)$, for a query q and term w (see Eq. 6). Indeed, we can now use the query model to estimate the document score. We use the Kullback-Leibler (KL) divergence (Manning et al. 2008) to estimate the retrieval score of document D for query q. The documents are ranked using the divergence between the query model just presented and the document model. Thus,

$$\text{Score}(q, D) = -\sum_{w \in V} P(w \mid q) \log \frac{P(w \mid q)}{P(w \mid D)}, \tag{17}$$

where V is the vocabulary, i.e., the set of all terms that occur in the collection, $P(w \mid q)$ is defined as the maximum likelihood estimate of w in the query, and $P(w \mid D)$ is the generative probability for a term as specified in Eq. 18 below.

[3] We use the following indicators: number of pronouns, amount of punctuation, number of emoticons used, amount of shouting, whether capitalization was used, the length of the post, and correctness of spelling.

This concludes the introduction of our burst sensitive query models. In the following sections we present and analyze experiments to assess their performance.

4 Experimental setup

In this section we describe experiments to answer the research questions introduced in Sect. 1. We describe collections and query sets in Sects. 4.1 and 4.2 presents our baselines. We list the parameter values in Sect. 4.3 and evaluation methods in Sect. 4.4.

4.1 Collections and topics

A summary of the collection and topic statistics can be found in Table 2. For our experiments we use three collections: (a) TREC-2: on AP data disks 1 and 2, (b) TREC-{6, 7, 8}: the LA Times and Financial Times data on disks 3 and 4, and (c) TREC-Blogs06. We only use the title field of the queries for all topics and test collections. In previous work, the construction of the training and test set and selection of temporal data for a news collection has been done in multiple ways.

For comparability with previous literature, we show the results for different subsets of queries; the precise query splits can be found in Appendix 1. We consider the following query subsets: *recent-1*, *recent-2*, *temporal-t*, and *temporal-b*. Here, *recent-1* is a subset of TREC-{7, 8}, an English news article collection, covering a period between 1991 and 1994 and providing nearly 350,000 articles; we have 150 topics for TREC-{6, 7, 8}; *recent-1* was selected by Li and Croft (2003); below, this query set was randomly split to provide training and testing data.

The query set *recent-2* consists of two parts. The first part is based on the TREC-2 data set, an English news article collection, covering the period between 1988 and 1989 and providing a total of just over 160,000 articles; we have 100 topics for TREC-2, of which 20 have been selected as recent by Efron and Golovchinsky (2011); this query subset is part of *recent-2*. The second part of *recent-2* is based on the TREC-{6, 7, 8} data set, again selected by Efron and Golovchinsky (2011). Training and testing data are the queries from TREC-6 and TREC-{7,8}, respectively.

Finally, Dakka et al. (2012) created a set of temporal queries, *temporal-t*, a subset of TREC-{6,7,8}, where again, training and testing data are the queries from TREC-6 and TREC-{7,8}, respectively.

Table 2 Summary of collection statistics for AP, LA/FT, and Blogs06, and of the various query sets that we use

	TREC-2 (disks 1, 2)	TREC-{6, 7, 8} (disks 4, 5)	TREC-Blogs06
# documents	164,597	342,054	2,574,356
Period covered	02/1988–12/1989	04/1991–12/1994	12/2005–02/2006
Topics	101–200	351–450 (test), 301–350(train)	851–950, 1001–1050
Recent-1 queries	–	7 (train), 24 (test)	–
Recent-2 queries	20	16 (train), 24 (test)	–
Temporal-t queries	–	31 (train), 55 (test)	–
Temporal-b queries	–	–	74

Our parameter analysis is based on TREC-6, the training set for the query sets *temporal-t* and *recent-2*.

The Blogs06 collection (Macdonald and Ounis 2006) is a collection of blog posts, collected during a three month period (12/2005–02/2006) from a set of 100,000 blogs and was used in the TREC Blog track (Ounis et al. 2006). As to the topics that go with the collections, we have 150 topics for the blog collection (divided over three TREC Blog track years, 2006–2008), of which *temporal-b* forms a set of temporal queries. The queries were manually selected by looking at the temporal distribution of the queries ground truth and the topic descriptions as queries that are temporally bursting. We split the blog collection dataset in two ways: (1) leave-one-out cross validation, and (2) three fold cross-validation split by topic sets over the years. One issue with the second method is that the 2008 topics have a smaller number of temporal queries, because these topics were created two years after the document collection was constructed—topic creators probably remembered less time-sensitive events than in the 2006 and 2007 topic sets.

As to preprocessing, the documents contained in the TREC data sets were tokenized with all punctuation removed, without using stemming. The Blogs06 was cleaned fairly aggressively. Blog posts identified as spam were removed. For our experiments, we only use the permalinks, that is, the HTML version of a blog post. During preprocessing, we removed the HTML code and kept only the page title and block level elements longer than 15 words, as detailed in (Hofmann and Weerkamp 2008). We also applied language identification using TextCat,[4] removing non-English blog posts. After preprocessing we are left with just over 2.5 million blog posts.

4.2 Baselines

4.2.1 Query likelihood

In order to keep our experiments comparable with previous work, we use the query likelihood model (Manning et al. 2008; Ponte and Croft 1998), both as baseline and as retrieval algorithm for the initial retrieval set. We rank documents by the likelihood $P(D \mid q)$; using Bayes' rule and the assumption that $P(q)$ is uniform, we obtain $P(D \mid q) \propto P(q \mid D)P(D)$. We set the prior $P(D)$ to be uniform and rank documents by the probability that their model (in our case the multinomial unigram language model) generates the query. More formally, $P(q \mid D) = \prod_{w \in q} P(w \mid D)$, where w is a term in a query. To obtain $P(w \mid D)$, we use Jelinek-Mercer smoothing, defined as a linear interpolation between $\hat{P}(w \mid D)$, the maximum likelihood estimate of D, and $P(w \mid C)$, the estimated probability of seeing w in the collection C (Efron and Golovchinsky 2011; Manning et al. 2008):

$$P(w \mid D) = (1 - \lambda)\hat{P}(w \mid D) + \lambda P(w \mid C). \tag{18}$$

We use the above baseline, but also Dirichlet smoothing as it generally performs better (Zhai and Lafferty 2004): the interpolation with the background corpus is document-dependent. Here,

[4] See http://odur.let.rug.nl/%7Evannoord/TextCat/.

$$P(w \mid D) = \frac{\hat{P}(w \mid D) + \mu\lambda P(w \mid C)}{|D| + \mu}, \tag{19}$$

where μ is the average document length of the collection.

A variant to this baseline for recency queries has been proposed by (Li and Croft 2003). Rather than having a uniform document prior $P(D)$, they use an exponential distribution (or decay function). Intuitively, documents closer to the query time time(q) has a higher chance of being read and are therefore more likely to be relevant. Therefore, the prior $P(D)$ can be approximated by $P(D \mid \text{time}(q))$, a query time time(q) dependent factor:

$$P(D \mid \text{time}(q)) = \beta e^{-\beta(\text{time(q)} - \text{time(D)})}, \tag{20}$$

where time(D) is the document time. The exponential decay parameter β indicates how quickly news grows old and less relevant. The higher it is, the steeper the curve, causing more recent documents to be rewarded.

4.2.2 Relevance models

Relevance models (Lavrenko and Croft 2001) re-estimate the document probabilities based on an initial feedback set. First, the top-N documents ($\mathcal{D}$) for a query q are retrieved using a simple retrieval method (e.g., Eq. 19). A model M_D over a document D is the smoothed maximum likelihood distribution over the term unigrams in the document D. The set of all models M_D where $D \in \mathcal{D}$ is $\mathcal{M}_D$. For all documents D, the final score is computed as

$$\text{score}(D) = \prod_{w \in D} \frac{P(w \mid R)}{P(w \mid N)}, \tag{21}$$

where R is a model of relevance and N of non-relevance. The term $P(w \mid N)$ can be based on collection frequencies. As to $P(w \mid R)$, Lavrenko and Croft (2001) assume that the query was generated from the same model as the document. The model of relevance R is then based on the query and

$$P(w \mid R) = \lambda \frac{P(w, q)}{P(q)} + (1 - \lambda) P(w \mid q), \tag{22}$$

where $P(q)$ is assumed to be uniform, $P(w \mid q)$ is defined as the maximum likelihood estimate of w in the query, and $\lambda \in [0, 1]$. Interpolation with the query model was shown to be effective (Jaleel et al. 2004). We use the first relevance model (RM-1), i.i.d. sampling of the query terms with a document prior (Lavrenko and Croft 2001), to estimate $P(w, q)$:

$$P(w, q) = \sum_{M_j \in \mathcal{M}} P(M_j) P(w \mid M_j) \prod_{w' \in q} P(w' \mid M_j). \tag{23}$$

The relevance model is then truncated to the top-N_{RM} terms. The resulting relevance model is often called RM-3 (Jaleel et al. 2004).

4.3 Parameter settings

For the parameter setting of the baseline experiments we follow (Efron and Golovchinsky 2011) and set λ = 0.4, β = 0.015, and $N_{RM} = 10$. Those parameters were optimised using grid search on TREC-6. Furthermore, as there is no query time associated with the queries in the query sets, we set the reference date to the most recent document in the collection. The granularity of time for burst estimation is months and days for the news and blog data, respectively. Initially, we return $M = 5$ terms per burst, use the top-$\hat{N}$, where $\hat{N} = 5$, documents to estimate the bursts, and use the top-N, where $N = 175$, documents for burst detection. In Sect. 5.3 we investigate the influence of varying these parameter settings on retrieval performance. Unless noted otherwise, we use the temporal distribution based on the relevance score (see Eq. 2); in Sect. 5.3 we show why it is more stable than using counts. The parameters $M, \hat{N}$, and N were selected based on an analysis of the training set (see Sect. 5.3.) An overview of the chosen parameters can be found in Table 3.

4.4 Evaluation

For all experiments, we optimize the parameters with respect to mean average precision (MAP) on the training sets and on the cross validation folds. MAP and precision at 10 (P@10) are our quantitative evaluation measures. We use the Student's t-test to evaluate the significance of observed differences. We denote significant improvements with $^{\blacktriangle}$ and $^{\vartriangle}$ ($p < 0.01$ and $p < 0.05$, respectively). Likewise, $^{\triangledown}$ and $^{\blacktriangledown}$ denote declines. Table 4 provides an overview over the acronyms used for the runs. If two methods are combined with a "-" (e.g., DB3-D), then the runs combine the two methods, as described in Sect. 3.

5 Results and discussion

In this section we seek to answer our research questions from Sect. 1. Section 5.1 discusses whether documents in bursts are more relevant than documents outside bursts. Section 5.2 analyzes if it matters when in a temporal burst a document is published. Section 5.3 investigates parameter values and, finally, Sect. 5.4 elaborates on experiments to assess our approaches to burst normalization.

5.1 Selection of relevant documents

To begin, we seek to answer the following research questions

> For a given query, are documents occurring within bursts more likely to be judged relevant for that query than those outside of bursts?

and

> Can documents within bursts contribute more useful terms for query modeling than documents selected for relevance models?

Table 3 Parameter gloss

Parameter	Value	References
λ	0.4	Eq. 18
β	0.015	Eq. 20
N_{RM}	10	Sect. 4.2.2
$\hat{N}$	5	Eq. 4
N	175	Sect. 3.1
M	5	Eq. 5

Table 4 Temporal query models examined in this paper

Name	Description	Equations
J	Jelinek Mercer Smoothing (Manning et al. (2008), Ponte and Croft (1998)	(18)
D	Dirichlet smoothing	(19)
EXP	Exponential prior, proposed by Li and Croft (2003)	(20)
RM	Relevance modeling, proposed by Lavrenko and Croft (2001)	(21)
DB0	Temporal query model with step wise decay: burst	(7)
DB1	Temporal query model with step wise decay: peaks	(8)
DB2	Temporal query model with step wise decay: burst and peaks, optimised α	(9)
DB3	Temporal query model with fixed exponential decay,	(10)
DB4	Temporal query model with variable exponential decay	(11)
DB5	Temporal query model with fixed exponential decay and *k*-NN	(12)
DB6	Temporal query model with variable exponential decay and *k*-NN	(13)
Y	Training on the respective other years	
L	Training with leave-one-out cross-validation	
LY	Training with leave-one-out cross-validation only on the same year	
C	Credibility normalisation	(15)
S	Size normalisation	(16)

We compare the performance of the baseline query model DB0 against using relevance models (RM) on news and blog data (TREC-2, TREC-7, TREC-8 and TREC-Blog06). We use Dirichlet (D) and Jelinek-Mercer (J) smoothing for the retrieval of the top-N and $\hat{N}$ documents for both relevance models and temporal query models.

Table 5 shows the retrieval results on the TREC-7 and TREC-8 query sets, comparing the baselines, query likelihood using Dirichlet and Jelinek-Mercer smoothing, with using exponential decay prior (EXP), relevance modeling (RM) and temporal query modeling (DB0). Temporal query modeling (DB0-D) based on Dirichlet smoothing obtains the highest MAP. It performs significantly better than its baseline (D) and relevance modeling using the same baseline (RM-D). Unlike for relevance models, we see that the P@10 scores increase (although not significantly so). Using Jelinek-Mercer smoothing as a baseline, the differences between the approaches are more pronounced and already significant on smaller datasets. The improvements can mainly be found on the temporal queries. Relevance modeling (RM) only helps for Jelinek-Mercer as baseline.

In the following we explain varying results for different query classes. We classify queries according to their temporal information need. To this end, we identified different classification systems. One is a crowd-sourced approach, where the classes are defined as the sub-categories of the Wikipedia category *event*.[5] TimeML (Pustejovsky et al. 2003) is a mark-up language for events, but the possible classes for the events[6] are difficult to annotate and distinguish. (Kulkarni et al. 2011) provide four classes of temporal distributions based on the number of bursts (spikes) in the distribution. This approach is data-

[5] http://en.wikipedia.org/wiki/Category:Events.

[6] These classes being *occurence*, *perception*, *reporting*, *aspectual*, *state*, *i_state*, and *i_action*.

Table 5 Retrieval effectiveness for TREC-7 and TREC-8, comparing different temporal retrieval methods and DB0

Model	Query subset									
			Temporal-t							
	Recent-1		TREC-7		TREC-8		Recent-2		All queries	
	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10
J	0.1963	0.3750	0.1406	0.3720	0.1800	0.3633	0.2007	0.3062	0.1997	0.3420
EXP-J	$0.1982^{\blacktriangle}$	0.3750	0.1413	0.3680	0.1809	0.3633	$0.2025^{\blacktriangle}$	$0.3125^{\vartriangle}$	$0.2009^{\vartriangle}$	0.3410
RM-J	0.1978	0.3708	0.1435	0.3640	0.1810	0.3667	0.2048	0.3062	0.2033	0.3420
DB0-J	$0.2117^{\vartriangle}$	0.3708	$0.1546^{\blacktriangle}_{\vartriangle}$	0.3920	$0.1914^{\blacktriangle}_{\vartriangle}$	0.3867	0.1650	0.2667	$0.2166^{\blacktriangle}_{\blacktriangle}$	$0.3580^{\vartriangle}$
D	0.2108	**0.4125**	0.1566	0.4320	0.1859	0.3633	0.2183	0.3438	0.2154	0.3710
EXP-D	0.2129	**0.4125**	0.1572	0.4320	0.1872	0.3667	0.2203	0.3563	0.2163	0.3740
RM-D	0.2105	0.3875	0.1579	0.4200	0.1854	0.3700	0.2193	0.3375	0.2158	0.3690
DB0-D	**0.2280**	0.4042	$\mathbf{0.1696}^{\vartriangle}$	**0.4360**	**0.1939**	**0.3833**	$\mathbf{0.2430}^{\vartriangle}$	**0.3750**	$\mathbf{0.2381}^{\blacktriangle}_{\blacktriangle}$	**0.3840**

The best values are in bold

Significant changes are with respect to the respective baseline (J/D), indicated using superscripts, and the relevance model (RM), indicated using subscripts

driven and not based on the information need. Finally, Vendler (1957) proposed classes for the temporal flow (aspect) of verbs. Similarly, we can distinguish queries based on the aspect of the underlying information need. The aspectual classes are: *states* (static without an endpoint), *actions* (dynamic without an endpoint), *accomplishments* (dynamic, with an endpoint and are incremental or gradual), *achievements* (with endpoint and occur instantaneously). The classes of the information need in the queries can be found in Appendix 2. The categorisation for the blog queries disregards the opinion aspect of the information need of the query.

In particular we look at the four example queries 417, 437, 410, and 408. Figure 3 shows the temporal distributions of the queries result sets and relevant documents. Query 417 asks for different ways to measure creativity. This is not temporally dependent because this does not change over time. We find four rather broad bursts with very similar term distributions; the terms *creative* and *computer* stand out. Finding several bursts for queries in the state class is therefore not a problem because the term distributions are very similar. We can also see that biggest bursts of the result set are on the same time period as for the relevant document set. Ignoring other documents leads to a higher AP for TQM-D as compared to RM-D (0.3431 vs. 0.3299).

Query 437 asks for experiences regarding the deregulation of gas and electric companies. We expected to find different actions that lead to the experiences that were reported. However, as in July 1992 the Energy Policy Act passed the Senate, while the actions took place before, the reports on the experiences centered around this date. The burst detection failed; however, the resulting query model for DB0-D is based on *all* top-$\hat{N}$ documents and thus close to RM-D: the term distributions are again very similar. Indeed, the AP for RM-D and DB0-D are very close (0.0172 vs. 0.0201).

Query 410 about the Schengen agreement was created at a time when the Schengen agreement had already been signed, but the implementation had not been successful yet. We expect to see discussions leading up to the accomplishment of the Schengen agreement. However, the Schengen agreement came into effect after last publication date included in the collection. Figure 3c shows, however, that there was one period of intense

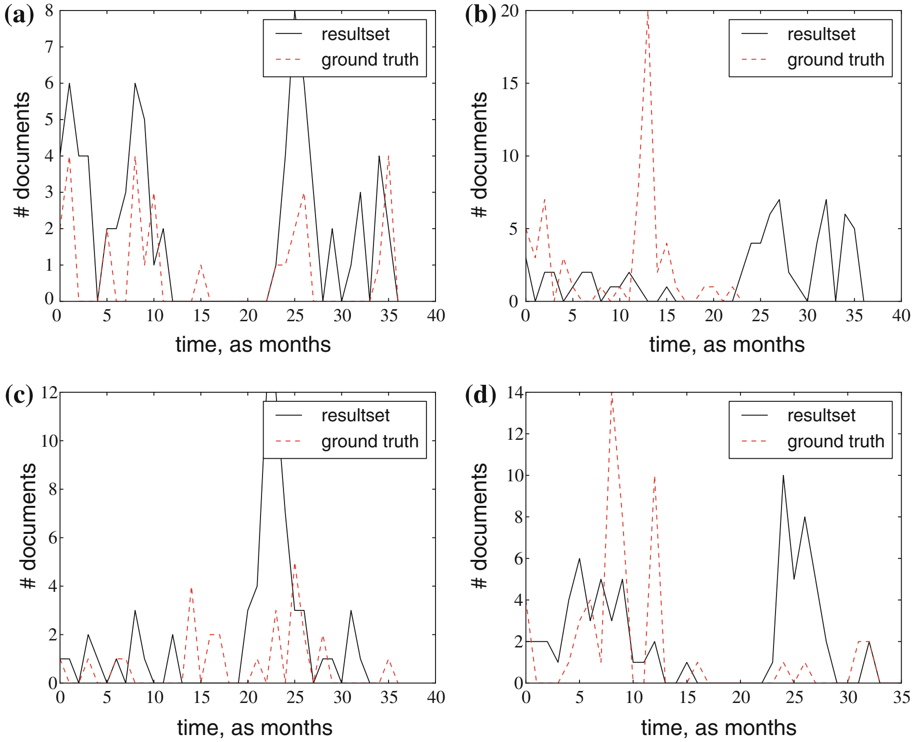


Fig. 3 Temporal distributions for example queries of aspectual classes. The *red* (*dashed*) *line* is the temporal distribution of the ground truth, while the *black* (*solid*) is the temporal distribution of the top 175 documents of the result set for *D*. **a** 417 (states). **b** 437 (actions). **c** 410 (accomplishments). **d** 408 (achievements) (Color figure online)

discussion. This is also captured in the temporal distribution of the relevant result set. And indeed, using DB0-D for this query we have an AP of 0.8213 while using relevance modeling (RM-D) yields an AP of 0.7983.

Figure 3d shows the temporal distribution for query 408. The query asks for tropical storms. Tropical storms are sudden events that occur and we can see that in the result set as well as in the set of relevant documents there are specific time periods that feature a lot of documents. The AP is low (0.0509) for both RM-D and DB0-D. However, we do find that DB0-D manages to identify 27.7 % more relevant documents than RM-D.

To conclude the class-based analysis, we can see that DB0-D performs either better or similar to RM, depending on the situation.

Table 6 shows the retrieval results on the TREC-2 query set, comparing the baselines (J and D) with EXP, RM, and DB0. Here, improvements are only significantly better than RM-D and RM-J on the non-temporal set. We see the tendency that DB0 performs better than RM modeling, for both baseline J and D. We also have an increase in precision. Again, RM-J helps, whereas RM-D does not.

Table 7 shows the retrieval results on the TREC-Blog06 query set. We observe significant improvements of DB0 in terms of MAP over RM only for the weak baseline

Table 6 Retrieval effectiveness for TREC-2, comparing different temporal retrieval methods and DB0

Model	Query set					
	Recent-2		Non-recent-2		All queries	
	MAP	P@10	MAP	P@10	MAP	P@10
J	0.2444	0.4100	0.1647	0.3100	0.1806	0.3300
EXP-J	0.2450	0.4100	0.1648	0.3088	0.1808	0.3290
RM-J	0.2487	**0.4250**	0.1717	0.3200	0.1871	0.3410
DB0-J	0.2488	0.3950	$\mathbf{0.1796}^{\blacktriangle}_{\vartriangle}$	$\mathbf{0.3475}^{\blacktriangle}_{\vartriangle}$	$\mathbf{0.1934}^{\blacktriangle}$	$\mathbf{0.3570}^{\blacktriangle}$
D	0.2537	0.4050	0.1683	0.3263	0.1854	0.3420
EXP-D	**0.2541**	0.4050	0.1684	0.3287	0.1856	0.3440
RM-D	0.2522	0.4100	0.1679	0.3312	0.1848	0.3470
DB0-D	0.2488	0.3950	$0.1775^{\vartriangle}_{\vartriangle}$	0.3425	0.1917	0.3530

The best values are in bold

Significant changes are with respect to the respective baseline (J/D), indicated using superscripts, and the relevance model (RM), indicated using subscripts

(J) and significant improvements over the baselines for both. The P@10 score using DB0 is better than for RM, and significantly so for DB0-J and RM-J. For the temporal query set, relevance modeling is better (but not significantly); we elaborate on this in Sect. 5.2 Unlike for the other datasets, for the TREC-Blog06 collection, RM improves the results.

Table 11 shows that around 30 % of the documents judged to be relevant are published in a peaking time bin. However, does this mean that documents inside bursts are actually more likely to be relevant than outside of bursts?

Figure 4 compares the early precision of the baselines with the same ranked list, but removing documents outside of bursts. We see that the early precision decreases, for all ranks but P@1 (precision at rank 1). The increase in performance is thus not just based on the precision of the selected documents. Obviously, with documents pruned from the list, new documents move up in rank. Figure 4 shows that a great deal of the documents retrieved at a certain rank indeed moved up. But how different are the ranked result lists? We clustered the documents in each of the two ranked lists using LDA (Blei et al. 2003).[7] The average size of clusters is the same, but the clusters are more varied for the result list using the pruned list: the standard deviation of the document coverage of the clusters is 4.5 % (4.0 %) for the pruned list (baseline). The number of clusters with at least one relevant document is 3.34 (4.02) for the pruned list (baseline) and together those clusters cover 45.0 % (37.5 %) of the documents respectively. All clusters with at least one relevant document cover more documents for the pruned set for the baseline. Therefore, the two ranked lists are indeed different. Naturally, the better performance comes from changing the topic models and choosing a more varied or less varied set of documents for query modeling.

We conclude that DB0 brings significant improvements over our baselines and relevance models. The better the baseline, the less prominent this improvement is. Unlike other approaches based on relevance modeling however, DB0 does not harm precision (P@10) but increases recall (as reflected in the MAP score).

[7] We used the standard settings of GibbsLDA++ (http://gibbslda.sourceforge.net/), with 10 clusters.

Table 7 Retrieval effectiveness for TREC-Blog06, 2006–2008, comparing different temporal retrieval methods and DB0

Model	Query set					
	Temporal-b		Non-temporal-b		All queries	
	MAP	P@10	MAP	P@10	MAP	P@10
J	0.2782	0.5041	0.2909	0.4697	0.2846	0.4867
EXP-J	0.2784	0.5054	0.2914	0.4750	0.2850	0.4900
RM-J	0.3029	0.4946	0.2903	0.4632	0.2965	0.4787
DB0-J	$0.3373^{\blacktriangle}_{\blacktriangle}$	0.5162	$0.3261^{\blacktriangle}_{\blacktriangle}$	0.4895	$0.3316^{\blacktriangle}_{\blacktriangle}$	$0.5027_{\triangle}$
D	0.3707	0.6838	0.3692	0.6553	0.3699	0.6693
EXP-D	0.3705	0.6919	0.3699	0.6579	0.3702	0.6747
RM-D	**0.3965**	**0.7041**	0.3627	0.6184	0.3793	0.6607
DB0-D	$0.3923^{\blacktriangle}$	0.6973	**0.3746**	**0.6539**	$\mathbf{0.3833}^{\blacktriangle}$	**0.6753**

The best values are in bold

Significant changes are with respect to the respective baseline (J/D), indicated using superscripts, and the relevance model (RM), indicated using subscripts

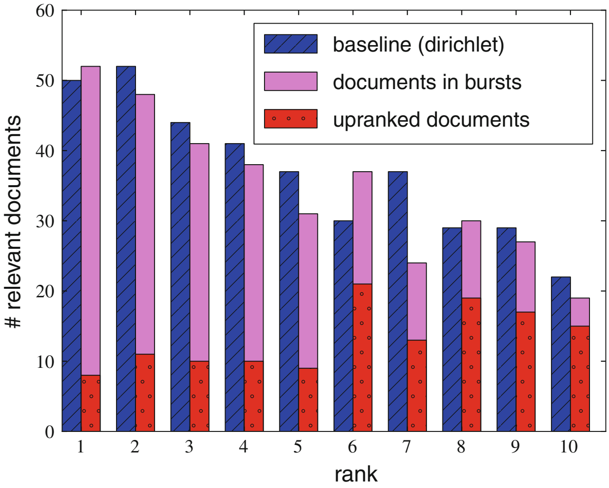


Fig. 4 The number of relevant documents at rank *X* using the baseline compared to only retrieving documents in bursts and the number of documents that are new at this rank (upranked documents)

5.2 Document priors

A document in a burst might still be far away from the actual peaking time period. We address the research question:

> What is the impact on the retrieval effectiveness when we use a query model based on an emphasis on documents close to the center of bursts?

For a quantitative analysis we compare the different temporal priors DB0–DB6 with the simplest approach DB0: using documents in a burst for query modeling. For the query models DB2, DB5, and DB6, we perform parameter optimization using grid search to find

Table 8 Retrieval effectiveness for TREC-7 and TREC-8, comparing the use of different document priors

	Query subset									
			Temporal-t							
	Recent-1		TREC-7		TREC-8		Recent-2		All queries	
Model	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10
RM-D	0.2105	0.3875	0.1580$^{\triangledown}$	0.4200	0.1854	0.3700	0.2193$^{\triangledown}$	0.3375	0.2158$^{\blacktriangledown}$	0.36900
DB0-D	**0.2280**	0.4042	**0.1696**	**0.4360**	**0.1939**	0.3833	**0.2430**	**0.3750**	**0.2381**	**0.3840**
DB1-D	0.2102	**0.4083**	0.1567$^{\triangledown}$	0.4240	0.1858	0.3600	0.2182	0.3375	0.2165$^{\blacktriangledown}$	0.3700
DB2-D	**0.2280**	0.4042	**0.1696**	**0.4360**	**0.1939**	0.3833	**0.2430**	**0.3750**	**0.2381**	**0.3840**
DB3-D	0.2275	0.3958	0.1686	0.4320	0.1919	0.3767	0.2419	**0.3750**	0.2333	0.3800
DB4-D	0.2275	0.3958	0.1690	0.4320	0.1920	0.3767	0.2430	**0.3750**	0.2358	0.3830
DB5-D	0.2275	0.3958	0.1685	0.4320	0.1922	0.3800	0.2419	**0.3750**	0.2354	**0.3840**
DB6-D	0.2274	0.3958	0.1690	0.4320	0.1921	0.3800	0.2419	**0.3750**	0.2359	**0.3840**

The best values are in bold

We report on significant differences with respect to DB0-D

Table 9 Retrieval effectiveness for TREC-2, comparing the use of different document priors

	Query set					
	Recent-2		Non-recent-2		All queries	
	MAP	P@10	MAP	P@10	MAP	P@10
RM-D	0.2522	**0.4100**	0.1679$^{\triangledown}$	0.3312	0.1848	0.3470
DB0-D	0.2488	0.3950	0.1775	0.3425	0.1917	0.3530
DB1-D	**0.2534**	0.4050	0.1725	0.3387	0.1887	0.3520
DB2-D	0.2472	0.4000	**0.1789**	0.3425	**0.1926**	**0.3540**
DB3-D	0.2491	0.3950	0.1777	0.3437	0.1920	0.3540
DB4-D	0.2463	0.4000	0.1788	**0.3438**	0.1923	0.3550
DB5-D	0.2488	0.3950	0.1777	0.3412	0.1919	0.3520
DB6-D	0.2472	0.4000	**0.1789**	0.3425	**0.1926**	**0.3540**

The best values are in bold

We report on significant differences with respect to DB0-D

the optimal parameters for k, γ and α.[8] For the news data, we do this on the dedicated training sets. For the blog data, as we do not have a dedicated training set, we evaluate on one year and train on the other years: we also use a leave-one-out cross-validation (LV1) set-up, training on queries from the same year and on all years.

In Tables 8 and 9 we compare the results using different document priors (DB3–6) with relevance modeling (RM) and the binary burst prior DB0 for TREC-{7,8} and TREC-2. For TREC-{7,8}, only using documents from peaks (DB1) decreases the MAP significantly compared to DB0. For TREC-2, DB1 performs worse than DB0, though not significantly. For both approaches and using the training data, we could not report differences

[8] We considered the following ranges: $\gamma \in \{-1, -0.9, \ldots, -0.1, -0.09, \ldots, -0.01, \ldots, -0.001, \ldots, -0.0001\}, k \in \{2, 4, 6, 8, 10, 20, 30, 50\}$, and $\alpha \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$.

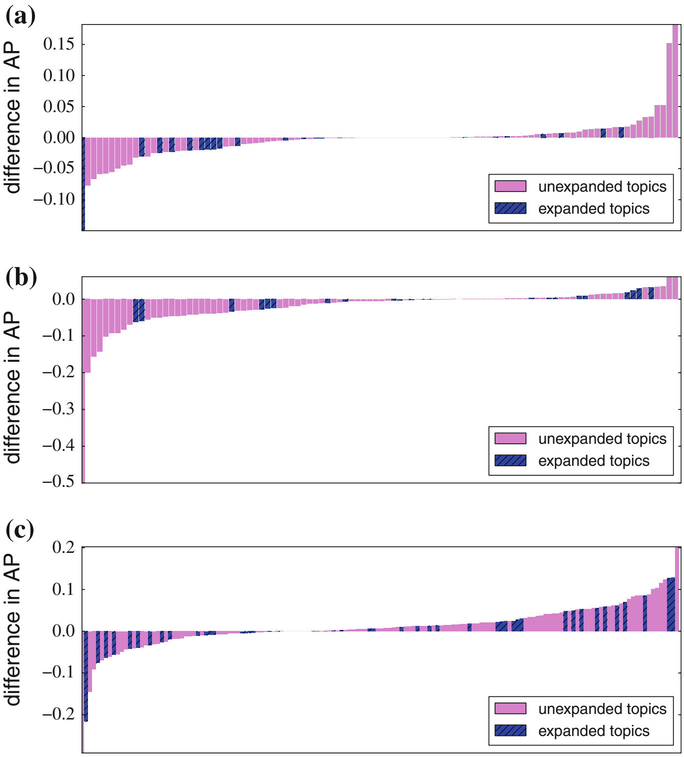


Fig. 5 Per topic comparision of AP for DB0-D and DB1-D. Queries that were expanded using DB0-D are in *blue*, queries that remained unexpanded are *pink*. The x-axis indicates each topic sorted by decreasing difference in AP. A *positive* difference indicates that DB1-D outperforms DB0-D; a *negative* difference indicates the opposite. **a** TREC-2. **b** TREC-7, 8. **c** TREC-Blog06 (Color figure online)

for different α in DB2. For TREC-6, the documents selected for burst estimation were mostly in the peak. We set α to 0.25.

Tables 12 and 13 show a sample of queries, their expansion terms, and their information need. The topics were selected based on a big difference in average precision of their expanded models under DB0 and DB1. For most cases we observed that whenever there is a strong difference in MAP between DB0 and DB1, this happens because there is no query expansion based on DB1, as there are no documents in peaks of bursts. Consider, for example, query 430 in TREC-{7,8}, with a big difference in average precision (AP) between DB0 and DB1. The expansion did not help but caused topic drift to the more general topic about bees. For query 173 in TREC-2 DB0 performs better than DB1. DB0 introduces more terms equivalent to *smoking* and *ban*. In this instance, DB2 improves the query even more by adding the term *domestic* (and down weighting terms that may cause topic drift). Figure 5a and b show the per topic analysis on TREC-2 and TREC-{7,8}. The figures show that for queries of TREC-2 (TREC-{7,8}), when DB0 performs better than DB1, 20.6 % (20.4 %) of the queries are expanded. For queries where DB1 is better than DB0, 24.4 % (32.6 %) are expanded.

In general, non-significant changes for TREC-2 are not surprising, because it is an entirely different dataset, but we used parameters trained on the query set for TREC-6 and a different corpus. The difference is explained in Table 11. We show that it has few (about

Table 10 Retrieval effectiveness for TREC-Blog06, 2006–2008, comparing the use of different document priors

	Training	Query set					
		Temporal-b		Non-temporal-b		All queries	
		MAP	P@10	MAP	P@10	MAP	P@10
RM-D		0.3965	0.7041	0.3627	0.6184	0.3793	0.6607
DB0-D		*0.3923*	*0.6973*	*0.3746*	*0.6539*	*0.3833*	*0.6753*
DB1-D		**0.4040**$^{\vartriangle}$	0.6811	**0.3838**	**0.6566**	**0.3938**$^{\vartriangle}$	0.6687
DB2-D	*Y*	*0.3928*	*0.7068*	*0.3734*	*0.6513*	*0.3829*	*0.6787*
	LY	*0.3905*	*0.6932*	*0.3722*	*0.6408*	*0.3812*	*0.6667*
	L	*0.3905*	*0.6932*	*0.3722*	*0.6408*	*0.3812*	*0.6667*
DB3-D	*Y*	*0.3930*	*0.7014*	*0.3739*	*0.6513*	*0.3834*	*0.6760*
	LY	0.3901	0.6851	0.3728	0.6434	0.3813	0.6640
	L	0.3898	0.6838	0.3727	0.6421	0.3812	0.6627
DB4-D		*0.3928*	*0.7068*	*0.3734*	*0.6513*	*0.3829*	*0.6787*
DB5-D	Y	0.3930	0.7000	0.3740	0.6513	0.3833	0.6753
	LY	0.3901	0.6838	0.3727	0.6434	0.3813	0.6633
	L	0.3897	0.6838	0.3727	0.6421	0.3811	0.6627
DB6-D	*Y*	*0.3926*	*0.7054*	*0.3737*	*0.6500*	*0.3830*	*0.6773*
	LY	*0.3901*	*0.6892*	0.3*721*	*0.6395*	*0.3810*	*0.6640*
	L	*0.3903*	*0.6905*	0.3*722*	*0.6382*	*0.3811*	*0.6640*

The best values are in bold

We report on significant differences with respect to DB0-D. We add italics to improve readability

3), narrow (about 5 bins) bursts, with relatively many documents in a burst. This data set is thus more temporal and needs less selection in the relevance modeling.

Table 10 compares the results using DB3–6 with RM and DB0 for TREC-Blog06. On this blog dataset, we observe the exact opposite to the previously used news data: DB1 is the only prior which performs weakly significantly better than DB0 and the RM. The natural question is why using DB1 performs so much better on blogs than on news. As we explain below, bursts in the temporal distribution of TREC-Blog06 queries are noisier and documents in the peaks are more suitable for query modeling.

In the following we explain why some collections perform better using different approximations to document priors. Table 11 shows temporal characteristics, number and size of bursts and peaks, of the different query sets. In general, there are not many documents from the top-$\hat{N}$ ($\hat{\mathcal{D}}$) in a peak, namely between 0.26 and 0.6 documents. However, we also see that about half of those documents in a peak are relevant for the news data sets and that still a lot of documents in the bursts are relevant as well. The picture is slightly different for the TREC-Blog06 collection: while there are more documents in the peak, only 10–20 % of the documents in the peak are relevant. As relevance modeling seems to have harmed on non-temporal topics in general (see Table 7), using only those highly specific documents (or none at all) does not cause a problematic topic drift. For example query 1045, *women numb3rs*.[9] Here, the drift caused by DB0 is to one

[9] *Numb3rs* was an American crime drama television series that ran in the US between 2005 and 2010.

Table 11 Temporal characteristics of query sets: the average number of documents in a peak and burst, the percentage of relevant documents that were published within a peaking (2 std) or lightly peaking (1 std) time bin, the average size of the burst and the average number of bins in a burst, which is roughly the width of the burst

Dataset	# documents in peak (% relevant)	# documents in burst (% relevant)	% rel in 1std	% rel in 2std	$\|\mathcal{B}\|$	Avg. # bins in B
TREC-2	0.45 (37.7)	2.91 (41.0)	45.3	36.9	2.98	5.23
TREC-6	0.29 (48.3)	3.43 (39.6)	23.9	22.6	6.12	10.02
TREC-7, TREC-8	0.26 (61.5)	3.50 (47.4)	34.5	30.8	6.67	10.6
TREC-Blog 2006	0.34 (23.5)	3.10 (23.2)	51.3	29.7	6.20	4.48
TREC-Blog 2007	0.46 (13.0)	3.12 (21.8)	49.1	27.8	5.94	4.25
TREC-Blog 2008	0.60 (13.3)	3.20 (16.3)	48.2	28.7	6.82	4.84

Table 12 Expansion terms for example queries and models with a strong difference in performance (MAP) for DB0–DB2. Query is in 173 in TREC-2, 430 in TREC-{7,8} and 430, 914, and 1045 are in TREC-Blogs06

Model	ID	Query	Expansion terms
DB0	430	killer bee attacks	pearson, developed, quarantine, africanized, honey, perhaps, bees, laboratory, mating, queens
DB1	430	killer bee attacks	–
DB0	173	smoking bans	figueroa, tobacco, airways, ordinance, oste, legislation, public, flights, smokers, years, its, areas
DB1	173	smoking bans	figueroa, ordinance, public, smoking, restaurants
DB2	173	smoking bans	figueroa, tobacco, airways, ordinance, oste, legislation, public, flights, smokers, years, its, areas, domestic
DB1	914	northernvoice	last, nothern, year, voice, email
DB2	914	northernvoice	february, 10th last, norther, clarke, scoble, jpc, session, year, jacon, voice, email, pirillo
DB2	1045	numb3rs women	love, utc, channel, charlie, tonight, amita, link, cute, im, epic, really
DB1	1045	numb3rs women	utc, tonight, cute, epic, link

specific woman (`Amita`) who is the leading actress in the series. DB1 only expands with terms from one burst and focusses on more general terms. The topic drift is now towards generally cute women on numb3rs. Careful expansion is the key: Looking at the topic analysis in Fig. 5c, for queries where DB0 performs better than DB1 in terms of MAP, 33.3 % of the queries are expanded, whereas for queries where DB1 is better, 32.2 % are expanded.

For the continuous approaches to estimating the probability of a document being generated by a burst (DB1–DB3) there is not much difference between using them in terms of performance, as can be seen in Tables 8, 9 and 10. For TREC-7,8 and TREC-2 we observe that the difference is usually on one or two queries only. For all three approaches we see a tendency to have better results for the adaptive continuous prior.

Table 13 The example queries from Table 12 and Sect. 5.1 with their information needs and Vendler class

ID	Query	Class	Information need
430	Killer bee attacks	Achievement	Identify instances of attacks on humans by Africanized (killer) bees.
173	Smoking bans	Actions	Document will provide data on smoking bans initiated worldwide in the public and private sector workplace, on various modes of public transportation, and in commercial advertising.
914	Northernvoice	Actions	Opinions about the Canadian blogging conference "NorthernVoice."
1045	Numb3rs women	Actions	Opinions about the TV show Numb3rs with regard to women.
417	Creativity	States	Find ways of measuring creativity
410	Shengen agreement	Accomplishments	Who is involved in the Schengen agreement to eliminate border controls in Western Europe and what do they hope to accomplish?
408	Tropical storms	Achievement	What tropical storms (hurricanes and typhoons) have caused significant property damage and loss of life?
413	Deregulation, gas, electric	Actions	What has been the experience of residential utility customers following deregulation of gas and electric?

In general, we can see a different temporality of the datasets in Table 11. The lifespan of a burst for blogs is usually four to five days, while the lifespan of a burst in TREC-{6,7,8} and TREC-2 is around ten months and five months respectively. This makes sense, events for the news are much longer and stretch over different months.

To conclude, it depends on the dataset if we should use DB0, DB1, or DB2: on the blog data set, which has narrow and noisy bursts, DB1 is a useful model, whereas for the news data sets, DB0 and DB2 are a better choice (Tables 12, 13).

5.3 Parameter optimisation

Temporal query models depend on three parameters: the number of documents for burst identification (N), the number of documents for query expansion ($\hat{N}$), and the number of expansion terms to return per burst (M). Additionally, the temporal distribution can be created using the raw counts of documents in a bin (Eq. 3) or the retrieval score (Eq. 2).

> Does the number of pseudo-relevant documents (N) for burst detection matter and how many documents ($\hat{N}$) should be considered for sampling terms? How many terms (M) should each burst contribute?

Given that we only have a training set for the news data, we analyze the questions on TREC-6. Based on the training data we analyzed

- the influence of the number of documents selected for burst identification (N),
- the number of documents to estimate the distribution of bursts ($\hat{N}$), and
- the number of terms sampled per burst (M).

Having two free parameters ($\hat{N}$ and N) to estimate the two pseudo-relevant result lists leads to the obvious question if either they are related or one of them is not important. In particular, using the two approaches for estimating the underlying temporal distribution (based on counts (Eq. 3) and based on the normalized retrieval score of documents (Eq. 2)) we would like to know if there is a difference for the parameter selection that leads to more stable but still effective results.

For both approaches—using the counts and the retrieval score—we expect to see a decrease in precision for high values of $\hat{N}$, since the lower the rank of documents, the less likely they are to be relevant. Using Eq. 3, documents with lower ranks may form spurious bursts and we expect the precision to drop for high N. As for Eq. 2 documents with a low score have much less influence; we expect the precision to be harmed much less for high N. The MAP score should increase for higher $\hat{N}$ for both approaches, but decrease for lower values of N: for very low values of N we have a lot of "bursts" containing two or three documents.

We generated heatmaps for different parameter combinations. By way of example we include a comparison of how the MAP score develops with respect to different values of N and N_B in Fig. 6. Other visualizations of the number of bursts, P@10, and the number of bursts with one document are available in Appendix 3, Fig. 9. Based on a number of these visualizations, we come to the following conclusions. For Eq. 3, with an increasing value N the P@10 and MAP scores decrease. With $3 < \hat{N} < 8$ and $100 < N < 250$, the performance is relatively stable. In this area of the parameter space, most detected bursts do not contain any documents that are in the top-$\hat{N}$ and vice versa, not every top-$\hat{N}$ document is part of a burst. With a value of $\hat{N} < 10$, leaving out one or two documents already has quite an influence on the term selection.

For Eq. 2 and a fixed $\hat{N}$ with $3 < \hat{N} < 10$, the MAP score does not change much with an increasing N, as long as $N > 100$, which seems to be the smallest number of documents required to effectively perform burst detection. The major difference between using Eq. 3 and Eq. 2 is that as long as there are more than 100 documents used for burst detection, using Eq. 2 does not depend on an optimization of N, while Eq. 3 does. For Eq. 2 using a high value of N, burst detection works well enough that the model with a low $\hat{N}$ can select the useful bursts. For both approaches, while the number of detected bursts is more than five, the selected documents are actually only in one or two bursts.

Figure 7 shows how the number of expansion terms M affects the MAP score for either using a temporal distribution based on scores or on counts. We see that using the retrieval

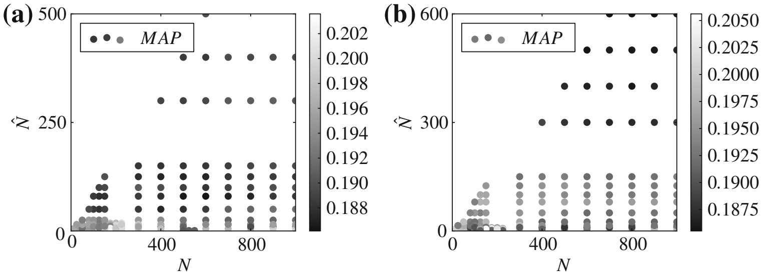


Fig. 6 Changes in MAP score for varying values for the number of documents used to estimate the temporal distribution (N) and the number documents used for query modeling (N_B), based on DB0-D. **a** Retrieval score. **b** Counts

Fig. 7 The development of the MAP score basing the temporal distribution on counts and on retrieval score, with N and $\hat{N}$ being the same and in [0,1000]

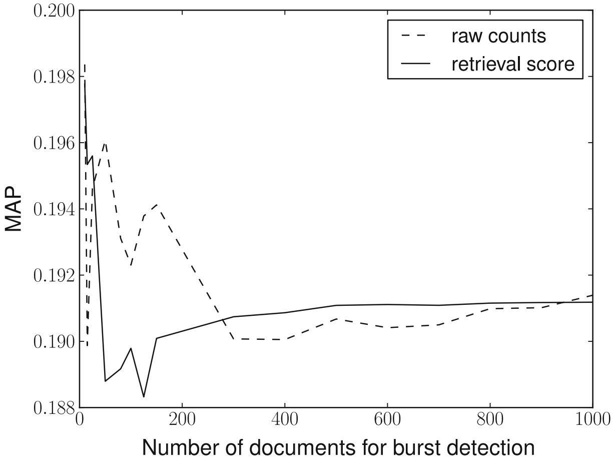
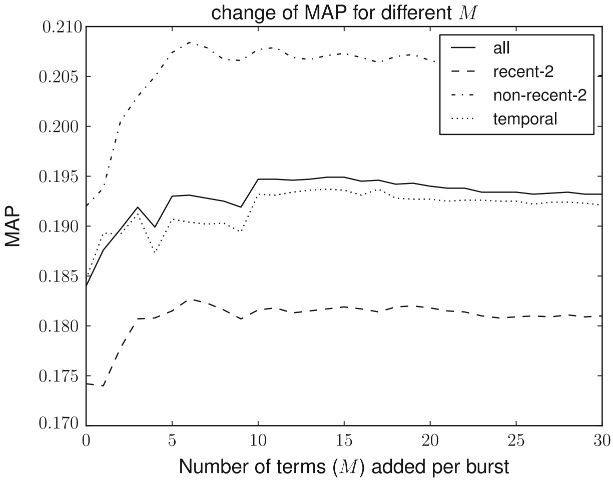


Fig. 8 The development of the MAP score over increasing values of M, the number if terms added per burst, over different splits of the training set TREC-6



scores, the graph stabilizes from around 170 documents onwards, whereas using the counts to estimate the temporal distribution is less stable over the entire graph. Hence, it seems advisable to use Eq. 2 to estimate the temporal distribution.

Figure 8 shows that for different values of M, the MAP score first increases and then stabilizes; while there is a steep increase for low values of M, the MAP score converges quickly. With increasing values of M, retrieval takes more time. It is therefore advisable to choose a low value of M. We chose $M = 5$.

To summarize, the combination of low values of $\hat{N}$ and the restriction to documents in bursts helps to select appropriate terms for query modeling. Unlike using raw counts, when we the retrieval score it does not matter how many documents (N) we use for burst estimation, as long as N is big enough. Finally, the effectiveness of our approach is stable with respect to the number of terms we sample.

Table 14 Retrieval effectiveness for TREC-Blog06, 2006–2008, comparing approaches to burst normalisation $(P(B \mid \mathcal{B}))$

Model	Query set					
	Temporal-b		Non-temporal-b		All queries	
	MAP	P@10	MAP	P@10	MAP	P@10
RM-D	0.3965	0.7041	0.3627	0.6184	0.3793	0.6607
DB0-D	0.3923	0.6973	0.3746	0.6539	0.3833	0.6753
DB0-D-C	0.3923	0.6973	0.3746	0.6539	0.3833	0.6753
DB0-D-S	0.3923	0.6973	0.3746	0.6539	0.3833	0.6753

None of the observed differences are statistically significant ($p < 0.01$)

5.4 Burst quality

Social media data is user-generated, unedited, and possibly noisy. Weerkamp and de Rijke (2008) show that the use of quality indicators improves retrieval effectiveness. We discuss the following question:

> Is the retrieval effectiveness dependent on query-independent factors, such as quality of a document contained in the burst or size of a burst?

We analyze whether some bursts are of bad quality and therefore not useful for query expansion, by comparing the basic temporal query model DB0 with its credibility expansion (see Eq. 15). Additionally, a bigger burst may indicate that it is more important. To address this intuition we compare the basic temporal query model DB0 with using a size normalization (see Eq. 16).

Table 14 shows the results for normalizing bursts on TREC-Blog06: DB0-D-C denotes using DB0-D with credibility normalisation (see Eq. 15) and DB0-D-S denotes using DB0-D using size normalisation (see Eq. 16). We see that there is no difference at all between normalizing or not. If we look at the differences in credibility of the documents, there are hardly any differences in the values. This is surprising because (Weerkamp et al. 2009) reported strong improvements using such document priors—however, unlike us they used the earlier data sets without prior spam detection. Additionally, as we explained earlier in Sect. 5.3: the documents we use for query modeling are already based on one or two bursts. Burst normalization only impacts query term weights if there are more than two bursts. Additionally, for queries where the initial result set has more than one burst, the credibility and size differences are very small and result in a low difference in the final query term weights.

Using more documents for query estimation leads to a bigger difference for the generation of terms, because documents from other, spurious, bursts are also selected. For the parameter value $\hat{N} = 100$ we have more bursts. Here, we can observe differences in the query terms generated by DB0-D-C and DB0-D-S: the query terms only have an overlap of 85 %. For a very noisy pre-selection of bursts, the size and credibility normalization does have an impact.

We conclude that as there are only few bursts to begin with, using normalization for bursts does not have an influence on the retrieval results.

6 Conclusion

We proposed a retrieval scoring method that combines the textual and the temporal part of a query. In particular, we explored a query modeling approach where terms are sampled from bursts in temporal distributions of documents sets. We proposed and evaluated different approximations for bursts—both continuous and discrete. Over query sets that consist of both temporal and non-temporal queries, most of the burst-based query models are able to arrive at an effective selection of documents for query modeling. Concerning the different approaches to approximating bursts, we found the effectiveness of the burst priors to be dependent on the data set. For example, the TREC-Blog06 data set has narrow, noisy bursts. For this dataset, using documents from the peaks of bursts yields higher MAP scores than using documents from the entire burst. In particular, we found that if there is training data, using discrete burst priors performs best. Without training data, a query-dependent variable temporal decay prior provides reliably better performance.

We found that the effectiveness of temporal query modeling based on burst detection depends on the number of documents used to estimate descriptive terms. Using less documents to model descriptive terms of a burst than for burst detection, this preselection selects very few bursts (between one and two) and causes the burst normalization to have no influence on the results.

The shortcomings of the approaches with a fixed discrete and continuous decay are the frequently missing training data and the query-independent estimation of parameter. Future work should focus on query-dependent estimation of parameters.

A benefit of the approaches is the efficient estimation of the bursts that does not add much more complexity to relevance modeling. We also provide variable and fixed parameters, thus a flexible option for situations with and without training sets.

Future work focuses on estimating temporal distributions based on external corpora, but base the query modeling on the original corpus. This should help especially for the noisy blog domain. Furthermore, temporal queries with an event-type information need are useful for, e.g., historians. An important future direction is therefore the incorporation and testing of temporal search in e-Science applications. We propose a user edited query modeling with visual feedback based on bursts. Instead of listing potential terms for query expansion, the interface would show a temporal distribution of the top-100 documents. It would exhibit burst detection, where every burst has a list of key terms associated. The terms in the bursts can be selected and used for query expansion. This allows e-Scientists to select terms related to specific time periods and queries.

Acknowledgments We are grateful to our reviewers for providing valuable feedback and suggestions. This research was partially supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 258191 (PROMISE Network of Excellence) and 288024 (LiMoSINe project), the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.061.814, 612.061.815, 640.004.802, 727.011.005, 612.001.116, HOR-11-10, the Center for Creation, Content and Technology (CCCT), the BILAND project funded by the CLARIN-nl program, the Dutch national program COMMIT, the ESF Research Network Program ELIAS, the Elite Network Shifts project funded by the Royal

Dutch Academy of Sciences (KNAW), and the Netherlands eScience Center under project number 027.012.105.

Appendix 1: Query sets used

Recent-1

The query set used by Li and Croft (2003), named *recent-1* in this work:

- TREC-7, 8 test set: 346, 400, 301, 356, 311, 337, 389, 307, 326, 329, 316, 376, 357, 387, 320, 347;
- TREC-7, 8 training set: 302, 304, 306, 319, 321, 330, 333, 334, 340, 345, 351, 352, 355, 370, 378, 382, 385, 391, 395, 396.

Recent-2

The query set used by Efron and Golovchinsky (Efron and Golovchinsky 2011), named *recent-2* in this work:

- TREC-2: 104, 116, 117, 122, 132, 133, 137, 139, 140, 148, 154, 164, 174, 175, 188, 192, 195, 196, 199, 200;
- TREC-6, training set: 06, 307, 311, 316, 319, 320, 321, 324, 326, 329, 331, 334, 337, 339, 340, 345, 346;
- TREC-7/TREC-8, test set: 351, 352, 357, 373, 376, 378, 387, 389, 391, 401, 404, 409, 410, 414, 416, 421, 428, 434, 437, 443, 445, 446, 449, 450.

Temporal

The query set used by Dakka et al. (Dakka et al. 2012), named *temporal-t* in this work:

- TREC-6, training set: 301, 302, 306, 307, 311, 313, 315, 316, 318, 319, 320, 321, 322, 323, 324, 326, 329, 330, 331, 332, 333, 334, 337, 340, 341, 343, 345, 346, 347, 349, 350;
- TREC-7, test set: 352, 354, 357, 358, 359, 360, 366, 368, 372, 374, 375, 376, 378, 383, 385, 388, 389, 390, 391, 392, 393, 395, 398, 399, 400;
- TREC-8, test set: 401, 402, 404, 407, 408, 409, 410, 411, 412, 418, 420, 421, 422, 424, 425, 427, 428, 431, 432, 434, 435, 436, 437, 438, 439, 442, 443, 446, 448, 450.

Manually selected queries with an underlying temporal information need for TREC-Blog, named *temporal-b* in this work:

- Blog06: 947, 943, 938, 937, 936, 933, 928, 925, 924, 923, 920, 919, 918, 917, 915, 914, 913, 907, 906, 905, 904, 903, 899, 897, 896, 895, 892, 891, 890, 888, 887, 886, 882, 881, 879, 875, 874, 871, 870, 869, 867, 865, 864, 862, 861, 860, 859, 858, 857, 856, 855, 854, 853, 851, 1050, 1043, 1040, 1034, 1032, 1030, 1029, 1028, 1026, 1024, 1021, 1020, 1019, 1017, 1016, 1015, 1014, 1012, 1011, 1009.

Appendix 2: Vendler classes of the queries

The classes are based on the verb classes introduced by Vendler (Vendler 1957).

TREC-2

- *State:* 101, 102, 103, 106, 107, 109, 112, 113, 116, 117, 118, 120, 124, 126, 132, 133, 134, 135, 143, 147, 151, 153, 157, 158, 160, 161, 163, 166, 169, 171, 177, 179, 184, 185, 186, 189, 193, 194
- *Action:* 104, 108, 115, 119, 123, 125, 136, 138, 139, 150, 152, 164, 165, 168, 173, 176
- *Achievement:* 105, 114, 121, 122, 128, 130, 137, 141, 142, 145, 146, 155, 156, 159, 162, 167, 170, 172, 174, 180, 182, 183, 187, 188, 191, 192, 196, 197, 198
- *Accomplishment:* 110, 111, 127, 129, 131, 140, 144, 148, 149, 154, 175, 178, 181, 190, 195, 199, 200

TREC-6

- *State:* 302, 304, 305, 307, 308, 310, 313, 315, 316, 318, 320, 321, 333, 334, 335, 338, 339, 341, 344, 346, 348, 349, 350
- *Actions:* 301, 312, 314, 319, 324, 325, 327, 330, 331, 340, 345, 347
- *Achievement:* 303, 306, 309, 317, 329, 332, 337
- *Accomplishments:* 311, 322, 323, 326, 328, 336, 342, 343

TREC-{7, 8}

- *State:* 356, 359, 360, 361, 366, 368, 369, 370, 371, 372, 373, 377, 378, 379, 380, 383, 385, 387, 391, 392, 396, 401, 403, 413, 414, 415, 416, 417, 419, 420, 421, 423, 426, 427, 428, 432, 433, 434, 438, 441, 443, 444, 445, 446, 449
- *Actions:* 351, 353, 357, 381, 382, 386, 388, 394, 399, 400, 402, 406, 407, 409, 411, 412, 418, 435, 437, 440, 448, 450
- *Achievement:* 352, 355, 365, 376, 384, 390, 395, 398, 410, 425, 442
- *Accomplishments:* 354, 358, 362, 363, 364, 367, 374, 375, 389, 393, 397, 404, 405, 408, 422, 424, 429, 430, 431, 436, 439, 447

Blog06

- *State:* 851, 854, 855, 862, 863, 866, 872, 873, 877, 879, 880, 882, 883, 885, 888, 889, 891, 893, 894, 896, 897, 898, 899, 900, 901, 902, 903, 904, 908, 909, 910, 911, 912, 915, 916, 917, 918, 919, 920, 924, 926, 929, 930, 931, 934, 935, 937, 939, 940, 941, 944, 945, 946, 947, 948, 949, 950, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1014, 1016, 1017, 1019, 1020, 1022, 1023, 1024, 1025, 1026, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1038, 1039, 1040, 1041, 1043, 1044, 1046, 1047, 1049, 1050
- *Action:* 852, 853, 857, 858, 859, 860, 861, 864, 868, 869, 870, 871, 874, 875, 876, 881, 884, 886, 887, 890, 892, 895, 905, 906, 907, 913, 914, 921, 922, 925, 927, 928, 933, 936, 938, 942, 1001, 1018, 1021, 1036, 1037, 1045, 1048
- *Accomplishments:* 865, 878, 932, 943, 1013, 1015, 1027
- *Achievement:* 856, 867, 923, 1028, 1042

Appendix 3: Additional graphs

See Fig. 9.

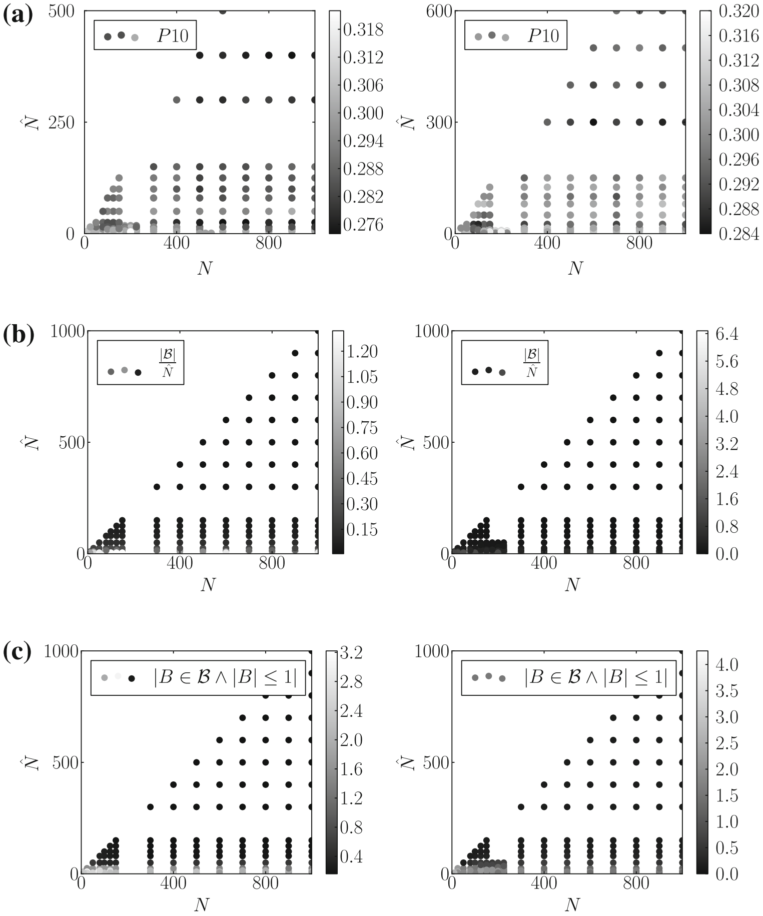


Fig. 9 Changes of **a** the precision at 10, **b** the number of bursts, **c** the number of bursts that contain ≤1 document that is in N_B. The changes are for varying values of the number of documents used to estimate the temporal distribution (N) and the number documents used for query modeling (N_B), based on DB0-D. Figures on the left and right are based on temporal distributions using the retrieval score and counts, respectively

References

Alonso, O., Strötgen, J., Baeza-Yates, R., & Gertz, M. (2011). Temporal information retrieval: Challenges and opportunities. In *Proceedings of the 1st international temporal web analytics workshop (TWAW 2011)*, pp. 1–8.

Amodeo, G., Amati, G., & Gambosi, G. (2011). On relevance, time and query expansion. In *CIKM '11: Proceedings of the 20th ACM international conference on Information and knowledge management* (pp. 1973–1976). New York, NY: ACM.

Balog, K., Weerkamp, W. & de Rijke, M. (2008). A few examples go a long way: Constructing query models from elaborate query formulations. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval*, (pp. 371–378). New York, NY: ACM. ISBN 978-1-60558-164-4.

Balog, K., Bron, M., & de Rijke, M. (2010). Category-based query modeling for entity search. In *ECIR 2010: 32nd European conference on information retrieval*, pp. 319–331.

Berberich, K., Bedathur, S., Alonso, O., & Weikum, G. (2010). A language modeling approach for temporal information needs. In *ECIR 2010: 32nd European conference on information retrieval*, Berlin: Springer .

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research, 3*(4-5), 993–1022.

Bron, M., Balog, K., & de Rijke, M. (2010). Ranking related entities: Components and analyses. In *CIKM '10: 19th ACM international conference on information and knowledge management*, Toronto: ACM.

Chien, S., & Immorlica, N. (2005). Semantic similarity between search engine queries using temporal correlation. In *Proceedings of the 14th international conference on World Wide Web (WWW '05)*, (pp. 2–11). New York, NY: ACM.

Corso, G. M. D., Gullí, A., & Romani, F. (2005). Ranking a stream of news. In *Proceedings of the 14th international conference on the World Wide Web (WWW '05).*

Cover, T. M., & Hart, P. E. (1967). Nearest neighbour pattern classification. In *Institute of electrical and electronics engineers transactions on information theory, 13*, pp. 21–27

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

Diaz, F. & Metzler, D. (2006). Improving the estimation of relevance models using large external corpora. In *SIGIR '06: 29th annual international ACM SIGIR conference on research & development on information retrieval*, pp. 154–161.

Dong, A., Zhang, R., Kolari, P., Bai, J., Diaz, F., Chang, Y., Zheng, Z., & Zha, H. (2010). Time is of the essence: improving recency ranking using twitter data. In *Proceedings of the 19th international conference on World wide web (WWW '10)*, (pp. 331–340). New York, NY: ACM.

Efron, M. (2010). Linear time series models for term weighting in information retrieval. *Journal of the American Society for Information Science and Technology, 6*(7), 1299–1312.

Efron, M. & Golovchinsky, G. (2011) Estimation methods for ranking recent information. In *SIGIR '11: 34th annual international ACM SIGIR conference on research & development on information retrieval*, pp. 495–504.

Hamilton, J. D. (1994). *Time-series analysis*, 1 edn. Princeton, NJ: Princeton Univerity Press.

Hofmann, K. & Weerkamp, W. (2008). Content extraction for information retrieval in blogs and intranets. Technical report, University of Amsterdam .

Jaleel, N. A., Allan, J., Croft, W. B., Diaz, F., Larkey, L. S., Li, X., Smucker, M. D., & Wade, C. (2004). UMass at TREC 2004: Novelty and hard. In *TREC 2004.*

Java, A., Kolari, P., Finin, T., Joshi, A. & Martineau, J. (2006) The BlogVox opinion retrieval system. In *TREC 2006.*

Jones, R. & Diaz, F. (2007). Temporal profiles of queries. *ACM Transaction Informayion Systems*, 25.

Kamps, J. (2004). Improving retrieval effectiveness by reranking documents based on controlled vocabulary. In *Advances in information retrieval: 26th European conference on IR research (ECIR 2004)*, (pp. 283–295). Heidelberg: Springer.

Keikha, M., Gerani, S., & Crestani, F. (2011a) Time-based relevance models. In *SIGIR '11: Proceedings of the 34th international ACM SIGIR conference on research and development in Information*, (pp. 1087–1088). New York, NY: ACM.

Keikha, M., Gerani, S., & Crestani, F. (2011b). Temper: a temporal relevance feedback method. In *ECIR 2011: 33rd European conference on information retrieval.*

Kleinberg, J. M. (2002). Bursty and hierarchical structure in streams. In *KDD '02: The eighth ACM SIGKDD international conference on knowledge discovery and data mining*, ACM, pp. 91–101.

Kulkarni, A., Teevan, J., Svore, K. M., & Dumais, S. T. (2011). Understanding temporal query dynamics. In *WSDM 2011: The fourth ACM international conference on Web search and data mining*, WSDM '11. ACM, 2011.

Lavrenko, V., & Croft, W. B. (2001). Relevance based language models. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*, (pp. 120–127). New York, NY: ACM.

Li, X., & Croft, W. B. (2003). Time-based language models. In *CIKM '03: International conference on information and knowledge management.*

Macdonald, C., & Ounis, I. (2006). The TREC blogs06 collection: Creating and analyzing a blog test collection. Technical report TR-2006-224, U. Glasgow.

Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval.* Cambridge: Cambridge University Press.

Martins, B., Manguinhas, H., & Borbinha, J. (2008). Extracting and exploring the geo-temporal semantics of textual resources. In *Proceedings of the 2008 IEEE international conference on semantic computing*, (pp. 1–9). Washington, DC: IEEE Computer Society.

Massoudi, K., Tsagkias, E., de Rijke, M., & Weerkamp, W. (2011). Incorporating query expansion and quality indicators in searching microblog posts. In *ECIR 2011: 33rd European conference on information retrieval.*

Meij, E., & de Rijke, M. (2010) Supervised query modeling using wikipedia. In *SIGIR '10: Proceedings of the 33rd annual international ACM SIGIR conference on research and development in information retrieval*, ACM.

Meij, E., Trieschnigg, D., de Rijke, M., & Kraaij, W. (2010). Conceptual language models for domain-specific retrieval. *Information Processing and Management, 46*(4), 448–469.

Odijk, D., de Rooij, O., Peetz, M.-H., Pieters, T., de Rijke, M., & Snelders, S. (2012). Semantic document selection. Historical research on collections that Span multiple centuries. In *Research and advanced technology for digital libraries—international conference on theory and practice of digital libraries, TPDL 2012*, Cypres.

Ounis, I., de Rijke, M., Macdonald, C., Mishne, G., & Soboroff, I. (2006). Overview of the TREC-2006 blog track. In *TREC 2006*, Gaithersburg.

Peetz, M.-H., & de Rijke, M. (2013). Cognitive temporal document priors. In *34th European conference on information retrieval (ECIR'13).*

Peetz, M.-H., Meij, E., de Rijke, M., & Weerkamp, W. (2012). Adaptive temporal query modeling. In *ECIR 2012: 34th European conference on information retrieval.*

Ponte, J. M., & Croft, W. B. (1998). A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval*, pp. 275–281.

Pustejovsky, J., Castaño, J. M., Ingria, R., Sauri, R., Gaizauskas, R. J., Setzer, A., Katz, G., & Radev, D. R. (2003). Timeml: Robust specification of event and temporal expressions in text. In *New directions in question answering*, pp. 28–34.

Qiu, Y., & Frei, H.-P. (1993). Concept based query expansion. In *SIGIR '93: Proceedings of the 16th annual international ACM-SIGIR conference on research and development in Iinformation retrieval*, ACM, pp. 160–169.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The SMART retrieval system—experiments in automatic document processing*, (pp. 313–323). Prentice Hall, Englewood Cliffs, NJ.

Seki, K., Kino, Y., Sato, S., & Uehara, K. (2007). TREC 2007 blog track experiments at Kobe University. In *TREC 2007.*

Tsagkias, M., Weerkamp, W., & Rijke, M. (2010). News comments: Exploring, modeling, and online prediction. In C. Gurrin, Y. He, G. Kazai, U. Kruschwitz, S. Little, T. Roelleke, S. Rüger, & K. Rijsbergen (Eds.), *Advances in information retrieval. Lecture notes in computer science* (Vol. 5993, pp. 191–203). Berlin, Heidelberg: Springer.

Vendler, Z. (1957). Verbs and times. *The Philosophical Review, 66*(2).

Verhagen, M., & Pustejovsky, J. (2008). Temporal processing with the TARSQI toolkit. In *22nd international conference on on computational linguistics: Demonstration papers*, COLING '08, (pp. 189–192). Stroudsburg, PA: Association for Computational Linguistics.

Wang, X., Zhai, C., Hu, X., & Sproat, R. (2007). Mining correlated bursty topic patterns from coordinated text streams. In *KDD '07: The 13th ACM SIGKDD international conference on knowledge discovery and data mining.*

Weerkamp, W., & de Rijke, M. (2008). Credibility improves topical blog post retrieval. In *Proceedings of ACL-08: HLT*, (pp. 923–931). Columbus, OH: ACL.

Weerkamp, W., & de Rijke, M. (2012). Credibility-inspired ranking for blog post retrieval. *Information Retrieval Journal, 15*(3–4), 243–277.

Weerkamp, W., Balog, K., & de Rijke, M. (2009). A generative blog post retrieval model that uses query expansion based on external collections. In *Joint conference of the 47th annual meeting of the association for computational linguistics and the 4th international joint conference on natural language processing of the Asian Federation of Natural Language Processing (ACL-ICNLP 2009)*, pp. 1057–1065.

Weerkamp, W., Balog, K., & de Rijke, M. (2012). Exploiting external collections for query expansion. *ACM Transactions on the Web, 6*(4):Article 18.

Zhai, C., & Lafferty, J. (2001). Model-based feedback in the language modeling approach to information retrieval. In *CIKM 01: Tenth international conference on information and knowledge management*, pp. 403–410.

Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transaction on Information Systems, 22*(2), 179–214.

Zhang, W., & Yu, C. (2006). UIC at TREC 2006 blog track. In *TREC 2006.*