Time-aware online reputation analysis

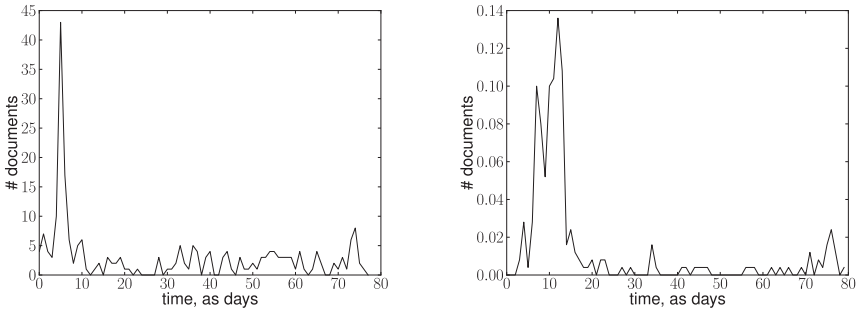
Peetz, M.-H.

6

Using Temporal Bursts for Query Modeling

A successful system to monitor online reputation relies on well-performing retrieval algorithms for social media [6, 7, 230]. Since language changes are more apparent in social media, 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 [198, 281]. 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 [207], while others incorporate metadata [126], semantic information such as entity types or Wikipedia categories [37], or synonyms [161]. 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 [66]. 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 chapter, 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 Figure 6.1a, 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* (Figure 6.1b), 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 tem-



(a) The relevant documents for query 936, *grammys*.

(b) The top ranked document retrieved in response to query 936, *grammys*.

Figure 6.1: Temporal distributions of documents for query 936, *grammys*, in the TREC Blogs06 test collection.

poral distribution of the pseudo relevant documents are relevant (see Table 6.11). 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 [74] or they compute a temporal similarity [130] to retrieve documents that are recent or diverse. Keikha et al. [129] use relevance models of temporal distributions of posts in blog feeds and Dakka et al. [62] 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:

RQ3.1 Are documents occurring within bursts more likely to be relevant than those outside of bursts?

RQ3.2 Can documents within bursts contribute more useful terms for query modeling than documents selected for relevance models?

RQ3.3 What is the impact on the retrieval effectiveness when we use a query model that rewards documents closer to the center of the bursts?

RQ3.4 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?

RQ3.5 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 chapter 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 Section 6.1 we introduce temporal query models and the baseline. We explain the setup of our experiments in Section 6.2 and our experimental results are presented and analysed in Section 6.3. We conclude in Section 6.4.

6.1 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 [246]. 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.

Consider Figure 6.1a 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 Figure 6.1b again, which shows the temporal distribution of the

Table 6.1: Notation used in the chapter.

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

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 chapter, that documents in bursts are more likely to be relevant.

Algorithm 2 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 ($\mathrm{bursts}(\mathcal{D})$). 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 2, 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

Algorithm 2: 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: $\text{bursts}(\mathcal{D})$, 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 ∈ bursts(D) do
2      foreach D ∈ B ∩ D̂ do
3          foreach w ∈ D do
4              update P(w | B) by adding (1/N̂) P(D | B) · P(w | D) to it
5          end
6          W is the set of top-M terms based on P(w | B);
7          foreach w ∈ W do
8              update P(w | q) by adding P(B | bursts(D)) · P(w | B) to it
9          end
10     end
11 end
```

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 [266].

Formally,

$$\hat{P}(w \mid q) = \sum_{B \in \text{bursts}(\mathcal{D})} \frac{P(B \mid \text{bursts}(\mathcal{D}))}{\hat{N}} \sum_{D \in \mathcal{D}_B} P(D \mid B) P(w \mid D). \tag{6.1}$$

Lines 1 to 11 in Algorithm 2 provide an algorithmic view on Eq. 6.1. The key components on which we focus are the document prior $P(D \mid B)$ in Section 6.1.3 and the burst normalisation ($P(B \mid \text{bursts}(\mathcal{D}))$) in Section 6.1.4. We start by defining bursts and detailing the query model.

6.1.1 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

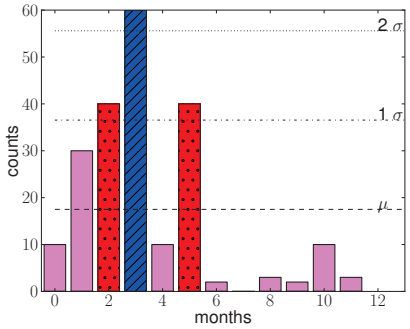


Figure 6.2: Example time series: time bins 3 and 4 form a burst and bin 3 peaks.

time-dependent, dynamic mean can easily be accommodated with a moving average estimation.

Consider the example in Figure 6.2. 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,

$$t_{\mathcal{D}}(i) = \sum_{\{D \in \mathcal{D} \,:\, \text{bin}(D) = i\}} R(D). \tag{6.2}$$

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 ($t(i) + 1\sigma > \mu$).

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

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. 6.2). An alternative approach for the estimation of the time series would be to use the counts of documents:

$$t'_{\mathcal{D}}(i) = |\{D \in \mathcal{D} : \text{bin}(D) = i\}|. \tag{6.3}$$

For the estimation of the bursts and peaks we proceed similar as for the time series introduced in Eq. 6.2. Unless stated otherwise, a time series is estimated using Eq. 6.2.

6.1.2 Term Reweighting

At the end of this section we introduce the score of a document for a query (Eq. 6.17), used in line 4 of Algorithm 2. To this end we need to determine the probability of a term being generated by a burst (Eq. 6.4 below) and how to combine the probabilities for all bursts (Eq. 6.5 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, $P(D \mid B)$:

$$P(w \mid B) = \frac{1}{\hat{N}} \sum_{D \in \mathcal{D}_B} P(D \mid B) P(w \mid D), \tag{6.4}$$

where $P(w \mid D)$ is the probability that term w is generated by document D. The summation in Eq. 6.4 is over documents in $\mathcal{D}_{\mathcal{B}}$ only to avoid topic drift.

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

$$\hat{P}(w \mid q) = \sum_{B \in \text{bursts}(\mathcal{D})} P(B|\, \text{bursts}(\mathcal{D})) P(w \mid B). \tag{6.5}$$

This is the same as Eq. 6.1. 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

$$\hat{P}^*(w \mid q) = \frac{1}{|q| + \sum_{w' \in W} \hat{P}(w' \mid q)} \begin{cases} \text{tf}(w, q) & \text{if } w \in q, \\ \hat{P}(w \mid q) & \text{if } w \in W \setminus q, \\ 0 & \text{else.} \end{cases} \tag{6.6}$$

This concludes the definition of the query model.

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

6.1.3 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. 6.4) we are missing the probability of a document generated by a burst, $P(D \mid B)$, which is introduced in this section (Section 6.1.3). Finally, we estimate the probability of a burst given other bursts, $P(B \mid \text{bursts}(\mathcal{D}))$ (Section 6.1.4).

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. 6.8 below). This could potentially increase the precision. However, assuming all documents in a burst to be generated uniformly (as we do in Eq. 6.7 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 Eq. 6.8 and Eq. 6.9 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 [144] (see Eq. 6.10 below) and augment it with a k-nearest neighbor kernel (see Eq. 6.12 below). The discrete approximations for $P(D \mid B)$ are $f_{\text{DB0}}(D, B)$, $f_{\text{DB1}}(D, B)$ and $f_{\text{DB2}}(D, B)$, while the continuous approximations are $f_{\text{DB3}}(D, B)$ to $f_{\text{DB6}}(D, B)$. 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:

$$f_{\text{DB0}}(D, B) = \begin{cases} 1 & \text{if } D \in \mathcal{D}_B, \\ 0 & \text{else.} \end{cases} \tag{6.7}$$

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:

$$f_{\text{DB1}}(D, B) = \begin{cases} 1 & \text{if } D \in \mathcal{D}_B \text{ and bin}(D) \text{ peaks}, \\ 0 & \text{else.} \end{cases} \tag{6.8}$$

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,

$$f_{\text{DB2}}(D,B) = \begin{cases} \alpha & \text{if } D \in \mathcal{D}_B, \\ 1-\alpha & \text{if } D \in \mathcal{D}_B \text{ and bin}(D) \text{ peaks}, \\ 0 & \text{else}, \end{cases} \tag{6.9}$$

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 [74, 144, 157]. We use similar functions to estimate the probability of a document being generated by a burst. The approximation $f_{\text{DB3}}(D,B)$ decreases exponentially with its distance to the largest peak of the burst $\max(B)$, the global maximum of the time series $t_{\mathcal{D}_B}(i)$ ($\text{argmax}_i\, t_{\mathcal{D}_B}(i)$). Formally, let $\text{time}(D)$ denote the normalized publishing time of document D; then

$$f_{\text{DB3}}(D,B) = e^{-\gamma(|\max(B)-\text{time}(D)|)}, \tag{6.10}$$

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

$$f_{\text{DB4}}(D,B) = e^{-\frac{|\max(B)-\text{time}(D)|}{2\sigma(t_{\mathcal{D}_B}(i))^2}}, \tag{6.11}$$

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 [98]. As a smoothing method we propose the use of k-NN [59], where the $\text{time}(D)$ of each document D is the average timestamp of its k neighbors. Let the distance between documents D, D_j be defined as $|\text{time}(D) - \text{time}(D_j)|$. We say that document D_j is a *k-neighbor* of document D ($\text{neighbor}_k(D, D_j)$) if D_j is among the k nearest documents

to D. The smoothed probability is then calculated using the exponential decay functions (Eq. 6.10 and Eq. 6.11) Formally,

$$f_{\text{DB5}}(D, B) = \frac{1}{k} \sum_{D_j \in \text{neighbor}_k(D, D_j)} f_{\text{DB3}}(D_j|B) \tag{6.12}$$

and

$$f_{\text{DB6}}(D, B) = \frac{1}{k} \sum_{D_j \in \text{neighbor}_k(D, D_j)} f_{\text{DB4}}(D_j|B). \tag{6.13}$$

6.1.4 Burst Normalization

We now introduce two approaches to burst normalization, based on quality (Eq. 6.15) and size (Eq. 6.16). 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 [265]. A burst may also be more important because it contains a large number of documents (see Eq. 6.16). 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

$$P(B \mid \text{bursts}(\mathcal{D})) = \frac{1}{|\text{bursts}(\mathcal{D})|}. \tag{6.14}$$

Unless explicitly stated otherwise, we only use the uniform normalization from Eq. 6.14.

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

$$P_C(B \mid \text{bursts}(\mathcal{D})) = \frac{1}{|B|} \sum_{D \in \mathcal{D}_B} P(D \mid \text{bursts}(\mathcal{D})), \tag{6.15}$$

where $P(D)$ is the quality of the document using the best performing quality indicators from [265].[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

$$P_S(B \mid \text{bursts}(\mathcal{D})) = \frac{1}{|\mathcal{D}_B|}, \tag{6.16}$$

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

[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.

6.1.5 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. 6.6). Indeed, we can now use the query model to estimate the document score. We use the Kullback-Leibler (KL) divergence [155] 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,

$$\text{Score}(q, D) = -\sum_{w \in V} P(w \mid q) \log \frac{P(w \mid q)}{P(w \mid D)}, \tag{6.17}$$

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. 2.1.

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

6.2 Experimental Setup

In this section we describe experiments to answer the research questions introduced earlier. Section 6.2.2 presents our baselines. We list the collections and query sets for the experiments in Section 6.2.1. We list the parameter values in Section 6.2.3 and evaluation methods in Section 6.2.4.

6.2.1 Data

For our experiments we use three collections. Two collections are the news collections TREC-2 and TREC-{6,7,8} (see Section 3.1). The third collection is the blog dataset TREC-Blogs06 (see Section 3.2). We use all query subsets introduced for the datasets. Our parameter analysis is based on TREC-6, the training set for the query sets *temporal-t* and *recent-2*.

6.2.2 Baselines

In order to keep our experiments comparable with previous work, we use the query likelihood model [155, 198] (see Section 2.3) and relevance models [139] (see Section 2.3) both as baseline and as retrieval algorithm for the initial retrieval set. The temporal extension based on recency priors Li and Croft [144] (Equation 2.6) functions as a baseline.

6.2.3 Parameter Settings

For the parameter setting of the baseline experiments we follow Efron and Golovchinsky [74] and set $\lambda = 0.4$, $\beta = 0.015$, 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 Section 6.3.3 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. 6.2); in Section 6.3.3 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 Section 6.3.3.) An overview of the chosen parameters can be found in Table 6.2.

Table 6.2: Parameter gloss.

parameter	value	reference
λ	0.4	Eq. 2.1
β	0.015	Eq. 2.6
N_{RM}	10	Section 2.3
$\hat{N}$	5	Eq. 6.4
N	175	Section 6.1.1
M	5	Eq. 6.5

6.2.4 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 $^{\triangle}$ ($p < 0.01$ and $p < 0.05$, respectively). Likewise, $^{\triangledown}$ and $^{\blacktriangledown}$ denote declines. Table 6.3 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 Section 6.1.

6.3 Results and Discussion

In this section we seek to answer our research questions from the introduction. Section 6.3.1 discusses whether documents in bursts are more relevant than documents outside bursts. Section 6.3.2 analyses if it matters when in a temporal burst a document is published. Section 6.3.3 investigates parameter values and, finally, Section 6.3.4 elaborates on experiments to assess our approaches to burst normalization.

6.3.1 Selection of Relevant Documents

To begin, we seek to answer the following research questions:

RQ3.1 For a given query, are documents occurring within bursts more likely to be judged relevant for that query than those outside of bursts?

Table 6.3: Temporal query models examined in this chapter.

Name	Description	Equation
J	Jelinek Mercer Smoothing [155, 198]	2.1
D	Dirichlet smoothing	2.2
EXP	Exponential prior, proposed by Li and Croft [144]	2.6
RM	Relevance modeling, proposed by [139]	2.3
DB0	Temporal query model with step wise decay: burst	6.7
DB1	Temporal query model with step wise decay: peaks	6.8
DB2	Temporal query model with step wise decay: burst and peaks, optimised α	6.9
DB3	Temporal query model with fixed exponential decay,	6.10
DB4	Temporal query model with variable exponential decay	6.11
DB5	Temporal query model with fixed exponential decay and k-NN	6.12
DB6	Temporal query model with variable exponential decay and k-NN	6.13
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	6.15
S	size normalisation	6.16

and

RQ3.2 Can documents within bursts contribute more useful terms for query modeling than documents selected for relevance models?

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 6.4 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*.[4] TimeML [201] is a mark-up

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

Table 6.4: Retrieval effectiveness for TREC-7 and TREC-8, comparing different temporal retrieval methods and DB0. Significant changes are with respect to the respective baseline (J/D), indicated using superscripts, and the relevance model (RM), indicated using subscripts.

	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
J	.1963	.3750	.1406	.3720	.1800	.3633	.2007	.3062	.1997	.3420
EXP-J	.1982$^{\blacktriangle}$	.3750	.1413	.3680	.1809	.3633	.2025$^{\blacktriangle}$	.3125$^{\triangle}$	.2009$^{\triangle}$	.3410
RM-J	.1978	.3708	.1435	.3640	.1810	.3667	.2048	.3062	.2033	.3420
DB0-J	.2117$^{\triangle}$	.3708	.1546$^{\blacktriangle}_{\triangle}$	.3920	.1914$^{\blacktriangle}_{\triangle}$	.3867	.1650	.2667	.2166$^{\blacktriangle}_{\blacktriangle}$	.3580$^{\triangle}$
D	.2108	**.4125**	.1566	.4320	.1859	.3633	.2183	.3438	.2154	.3710
EXP-D	.2129	**.4125**	.1572	.4320	.1872	.3667	.2203	.3563	.2163	.3740
RM-D	.2105	.3875	.1579	.4200	.1854	.3700	.2193	.3375	.2158	.3690
DB0-D	**.2280**	.4042	**.1696**$^{\triangle}$	**.4360**	**.1939**	**.3833**	**.2430**$^{\triangle}$	**.3750**	**.2381**$^{\blacktriangle}_{\blacktriangle}$	**.3840**

language for events, but the possible classes for the events[5] are difficult to annotate and distinguish. Kulkarni et al. [134] provide four classes of temporal distributions based on the number of bursts (spikes) in the distribution. This approach is data-driven and not based on the information need. Finally, Vendler [257] 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 6.B. 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 6.3 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

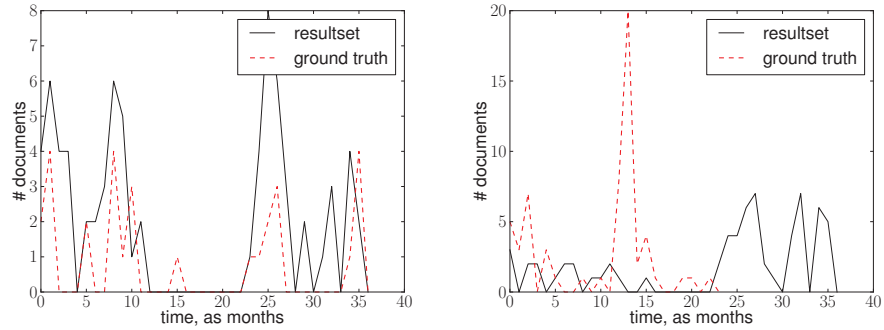
[5]These classes being *occurence*, *perception*, *reporting*, *aspectual*, *state*, *i_state*, and *i_action*.

Table 6.5: Retrieval effectiveness for TREC-2, comparing different temporal retrieval methods and DB0. Significant changes are with respect to the respective baseline (J/D), indicated using superscripts, and the relevance model (RM), indicated using subscripts.

	query set					
	recent-2		non-recent-2		all queries	
Model	MAP	P@10	MAP	P@10	MAP	P@10
J	.2444	.4100	.1647	.3100	.1806	.3300
EXP-J	.2450	.4100	.1648	.3088	.1808	.3290
RM-J	.2487	**.4250**	.1717	.3200	.1871	.3410
DB0-J	.2488	.3950	**.1796**$^{\blacktriangle}_{\triangle}$	**.3475**$^{\blacktriangle}_{\triangle}$	**.1934**$^{\blacktriangle}$	**.3570**$^{\blacktriangle}$
D	.2537	.4050	.1683	.3263	.1854	.3420
EXP-D	**.2541**	.4050	.1684	.3287	.1856	.3440
RM-D	.2522	.4100	.1679	.3312	.1848	.3470
DB0-D	.2488	.3950	.1775$^{\triangle}_{\triangle}$	.3425	.1917	.3530

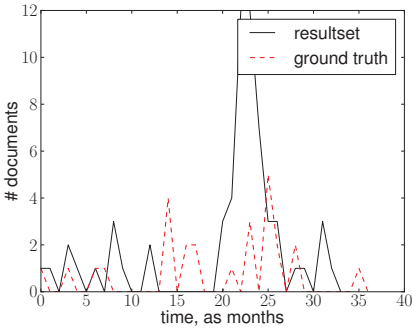
Table 6.6: Retrieval effectiveness for TREC-Blog06, 2006–2008, comparing different temporal retrieval methods and DB0. Significant changes are with respect to the respective baseline (J/D), indicated using superscripts, and the relevance model (RM), indicated using subscripts.

	query set					
	temporal-b		non-temporal-b		all queries	
Model	MAP	P@10	MAP	P@10	MAP	P@10
J	.2782	.5041	.2909	.4697	.2846	.4867
EXP-J	.2784	.5054	.2914	.4750	.2850	.4900
RM-J	.3029	.4946	.2903	.4632	.2965	.4787
DB0-J	.3373$^{\blacktriangle}_{\blacktriangle}$	.5162	.3261$^{\blacktriangle}_{\blacktriangle}$	.4895	.3316$^{\blacktriangle}_{\blacktriangle}$	.5027$_{\triangle}$
D	.3707	.6838	.3692	.6553	.3699	.6693
EXP-D	.3705	.6919	.3699	.6579	.3702	.6747
RM-D	**.3965**	**.7041**	.3627	.6184	.3793	.6607
DB0-D	.3923$^{\blacktriangle}$	.6973	**.3746**	**.6539**	**.3833**$^{\blacktriangle}$	**.6753**

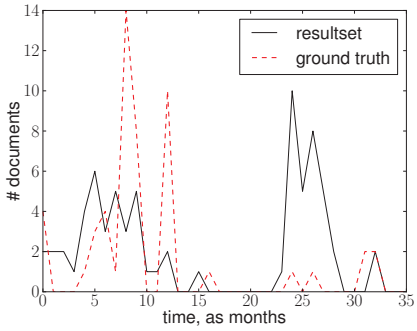


(a) 417 (states)

(b) 437 (actions)



(c) 410 (accomplishments)



(d) 408 (achievements)

Figure 6.3: 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.

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 6.3c shows, however, that there was one period of intense 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 6.3d 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

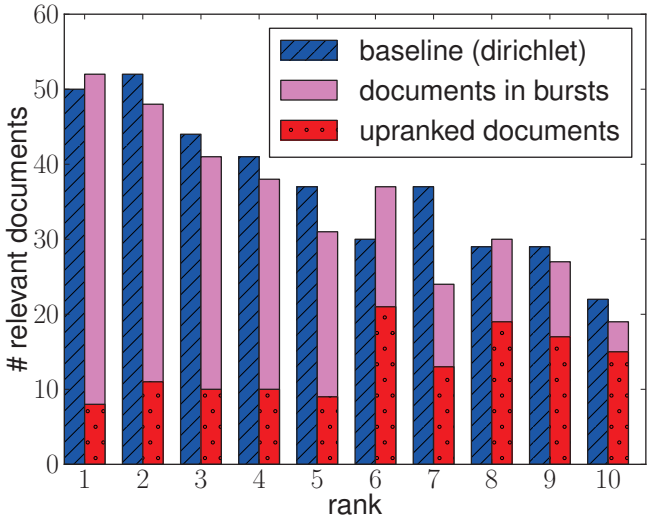


Figure 6.4: The number of relevant documents at rank X using the baseline compared to only retrieving documents in bursts and the number of documents that are new at this rank (upranked documents).

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 6.5 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 6.6 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 (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 Section 6.3.2. Unlike for the other datasets, for the TREC-Blog06 collection, RM improves the results.

Table 6.11 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 6.4 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 6.4 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 [31].[6] 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).

6.3.2 Document Priors

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

RQ3.3 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 the optimal parameters for k, γ and α.[7] 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 Table 6.7 and Table 6.8 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 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.

Table 6.10 and Table 6.12 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

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

[7] 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 $\alpha \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$

Table 6.7: Retrieval effectiveness for TREC-7 and TREC-8, comparing the use of different document priors. We report on significant differences with respect to DB0-D.

	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	.2105	.3875	.1580$^{\triangledown}$	.4200	.1854	.3700	.2193$^{\triangledown}$	.3375	.2158$^{\blacktriangledown}$	.36900
DB0-D	**.2280**	.4042	**.1696**	**.4360**	**.1939**	.3833	**.2430**	**.3750**	**.2381**	**.3840**
DB1-D	.2102	**.4083**	.1567$^{\triangledown}$	.4240	.1858	.3600	.2182	.3375	.2165$^{\blacktriangledown}$	.3700
DB2-D	**.2280**	.4042	**.1696**	**.4360**	**.1939**	.3833	**.2430**	**.3750**	**.2381**	**.3840**
DB3-D	.2275	.3958	.1686	.4320	.1919	.3767	.2419	**.3750**	.2333	.3800
DB4-D	.2275	.3958	.1690	.4320	.1920	.3767	.2430	**.3750**	.2358	.3830
DB5-D	.2275	.3958	.1685	.4320	.1922	.3800	.2419	**.3750**	.2354	**.3840**
DB6-D	.2274	.3958	.1690	.4320	.1921	.3800	.2419	**.3750**	.2359	**.3840**

Table 6.8: Retrieval effectiveness for TREC-2, comparing the use of different document priors. We report on significant differences with respect to DB0-D.

	query set					
	recent-2		non-recent-2		all queries	
Model	MAP	P@10	MAP	P@10	MAP	P@10
RM-D	.2522	**.4100**	.1679$^{\triangledown}$	.3312	.1848	.3470
DB0-D	.2488	.3950	.1775	.3425	.1917	.3530
DB1-D	**.2534**	.4050	.1725	.3387	.1887	.3520
DB2-D	.2472	.4000	**.1789**	.3425	**.1926**	**.3540**
DB3-D	.2491	.3950	.1777	.3437	.1920	.3540
DB4-D	.2463	.4000	.1788	**.3438**	.1923	.3550
DB5-D	.2488	.3950	.1777	.3412	.1919	.3520
DB6-D	.2472	.4000	**.1789**	.3425	**.1926**	**.3540**

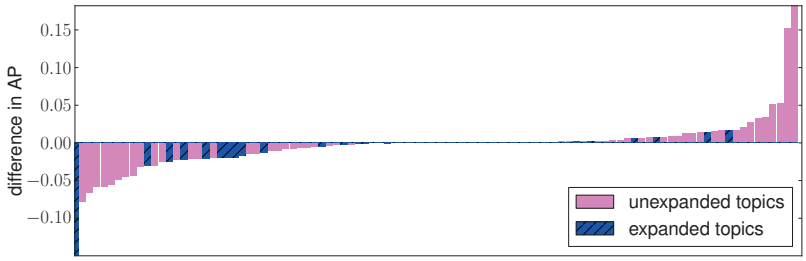
Table 6.9: Retrieval effectiveness for TREC-Blog06, 2006–2008, comparing the use of different document priors. We report on significant differences with respect to DB0-D. We add shading to improve readability.

		query set					
		temporal-b		non-temporal-b		all queries	
Model	training	MAP	P@10	MAP	P@10	MAP	P@10
RM-D		.3965	.7041	.3627	.6184	.3793	.6607
DB0-D		.3923	.6973	.3746	.6539	.3833	.6753
DB1-D		**.4040**$^{\Delta}$	.6811	**.3838**	**.6566**	**.3938**$^{\Delta}$	.6687
	Y	.3928	**.7068**	.3734	.6513	.3829	**.6787**
DB2-D	LY	.3905	.6932	.3722	.6408	.3812	.6667
	L	.3905	.6932	.3722	.6408	.3812	.6667
	Y	.3930	.7014	.3739	.6513	.3834	.6760
DB3-D	LY	.3901	.6851	.3728	.6434	.3813	.6640
	L	.3898	.6838	.3727	.6421	.3812	.6627
DB4-D		.3928	**.7068**	.3734	.6513	.3829	**.6787**
	Y	.3930	.7000	.3740	.6513	.3833	.6753
DB5-D	LY	.3901	.6838	.3727	.6434	.3813	.6633
	L	.3897	.6838	.3727	.6421	.3811	.6627
	Y	.3926	.7054	.3737	.6500	.3830	.6773
DB6-D	LY	.3901	.6892	.3721	.6395	.3810	.6640
	L	.3903	.6905	.3722	.6382	.3811	.6640

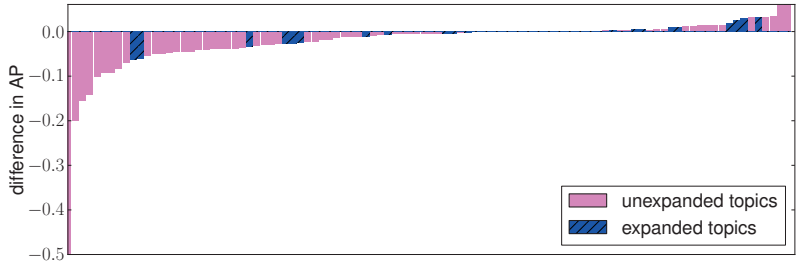
there is no query expansion based on DB1, as there are no documents in peaks of bursts. Consider, for example, query 430 in TREC-{7,8}, with a big difference in average precision (AP) between DB0 and DB1. The expansion did not help but caused topic drift to the more general topic about bees. For query 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). Figures 6.5a and 6.5b 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 6.11. We show that it has few (about 3), narrow (about 5 bins) bursts, with relatively many documents in a burst. This dataset is thus more temporal and needs less selection in the relevance modeling.

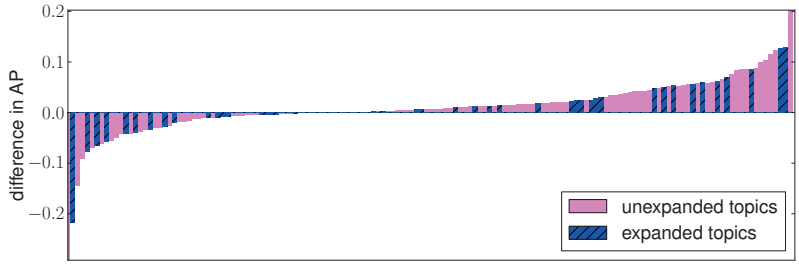
Table 6.9 compares the results using DB3–6 with RM and DB0 for TREC-Blog06. On this blog dataset, we observe the exact opposite to the previously used news data: DB1 is the only prior which performs weakly significantly better than DB0 and the RM. The natural question is why using DB1 performs so much better on blogs than on news.



(a) TREC-2



(b) TREC-{7,8}



(c) TREC-Blog06

Figure 6.5: 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.

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 6.11 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 datasets 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 6.6), using only those highly specific documents (or none at all) does not cause a problematic topic drift. For example query 1045, *women numb3rs*.[8] Here, the drift caused by DB0 is to one 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 Figure 6.5c, 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 Table 6.7–6.9. 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.

In general, we can see a different temporality of the datasets in Table 6.11. 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 dataset, which has narrow and noisy bursts, DB1 is a useful model, whereas for the news datasets, DB0 and DB2 are a better choice.

6.3.3 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. 6.3) or the retrieval score (Eq. 6.2).

RQ3.4 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 analyse the questions on TREC-6. Based on the training data we analysed

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

Table 6.10: 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

- 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. 6.3) and based on the normalized retrieval score of documents (Eq. 6.2)) 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. 6.3, documents with lower ranks may form spurious bursts and we expect the precision to drop for high N. As for Eq. 6.2 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.

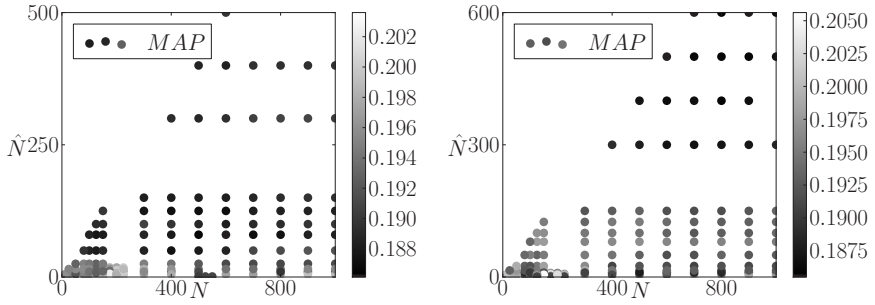
We generated heatmaps for different parameter combinations. By way of example we include a comparison of how the MAP score develops with respect to different values of N and N_B in Figure 6.6. Other visualizations of the number of bursts, P@10, and the number of bursts with one document are available in Appendix 6.A, Figure 6.9. Based

Table 6.11: 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 2std	$\lvert\mathcal{B}\rvert$	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 6.12: The example queries from Table 6.10 and Section 6.3.1 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?



(a) Retrieval score (b) Counts

Figure 6.6: 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 ($\hat{N}$), based on DB0-D.

on a number of these visualizations, we come to the following conclusions. For Eq. 6.3, 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. 6.2 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. 6.3 and Eq. 6.2 is that as long as there are more than 100 documents used for burst detection, using Eq. 6.2 does not depend on an optimization of N, while Eq. 6.3 does. For Eq. 6.2 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 6.7 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 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. 6.2 to estimate the temporal distribution.

Figure 6.8 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

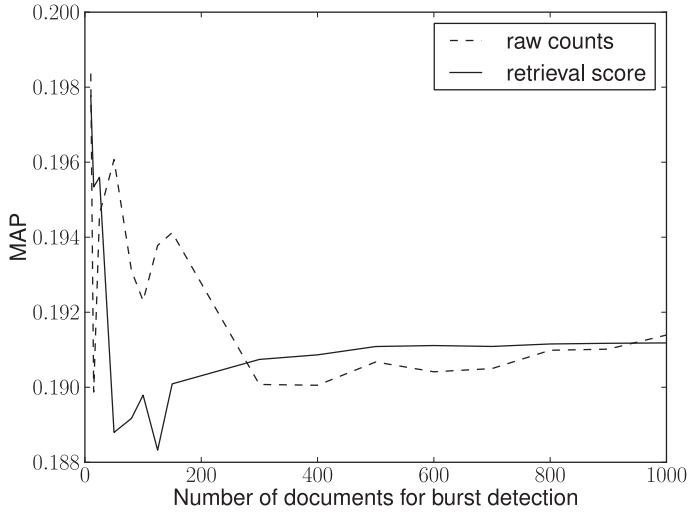


Figure 6.7: The development of the MAP score basing the temporal distribution on counts and on retrieval score, with N and $\hat{N}$ being the same and in $[0, 1000]$.

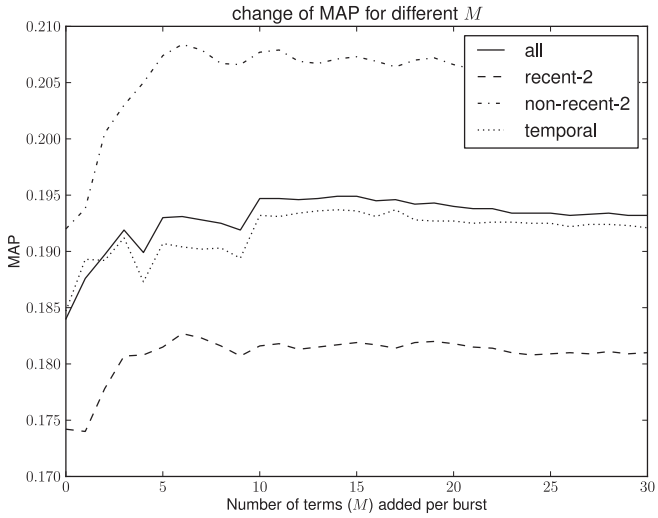


Figure 6.8: The development of the MAP score over increasing values of M, the number if terms added per burst, over different splits of the training set TREC-6.

Table 6.13: Retrieval effectiveness for TREC-Blog06, 2006–2008, comparing approaches to burst normalisation ($P(B \mid \mathcal{B})$). None of the observed differences are statistically significant ($p < 0.01$).

	query set					
	temporal-b		non-temporal-b		all queries	
Model	MAP	P@10	MAP	P@10	MAP	P@10
RM-D	.3965	.7041	.3627	.6184	.3793	.6607
DB0-D	.3923	.6973	.3746	.6539	.3833	.6753
DB0-D-C	.3923	.6973	.3746	.6539	.3833	.6753
DB0-D-S	.3923	.6973	.3746	.6539	.3833	.6753

stable with respect to the number of terms we sample.

6.3.4 Burst Quality

Social media data is user-generated, unedited, and possibly noisy. Weerkamp and de Rijke [265] show that the use of quality indicators improves retrieval effectiveness. We discuss the following question:

RQ3.5 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 analyse 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. 6.15). 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. 6.16).

Table 6.13 shows the results for normalizing bursts on TREC-Blog06: DB0-D-C denotes using DB0-D with credibility normalisation (see Eq. 6.15) and DB0-D-S denotes using DB0-D using size normalisation (see Eq. 6.16). 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 and de Rijke [265] reported strong improvements using such document priors—however, unlike us they used the earlier datasets without prior spam detection. Additionally, as we explained earlier in Section 6.3.3: 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.

6.4 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 dataset. For example, the TREC-Blog06 dataset 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.

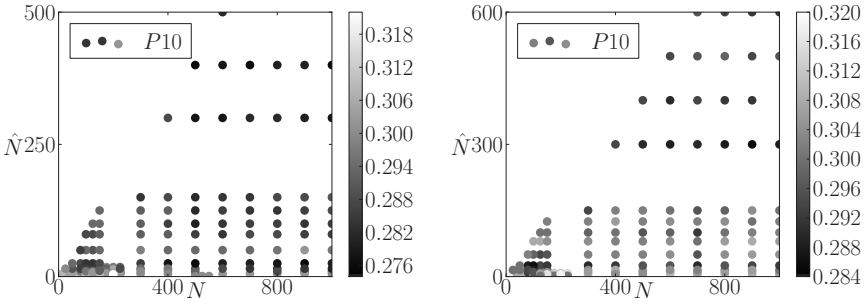
We found that the effectiveness of temporal query modeling based on burst detection depends on the number of documents used to estimate descriptive terms. Using less documents to model descriptive terms of a burst than for burst detection, this preselection selects very few bursts (between one and two) 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.

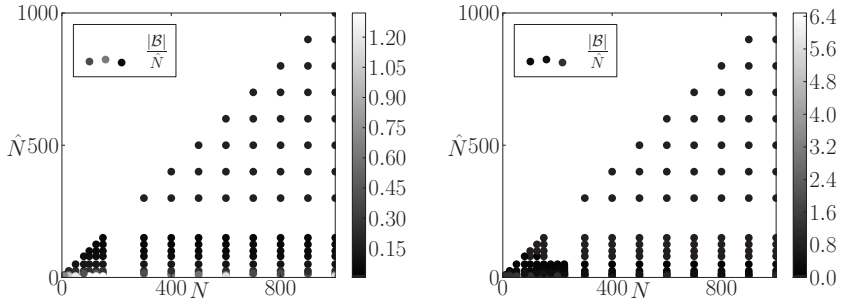
A benefit of the approaches is the efficient estimation of the bursts that does not add much more complexity to relevance modeling. We also provide variable and fixed parameters, thus a flexible option for situations with and without training sets.

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 digital humanities 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 experts in digital humanities to select terms related to specific time periods and queries.

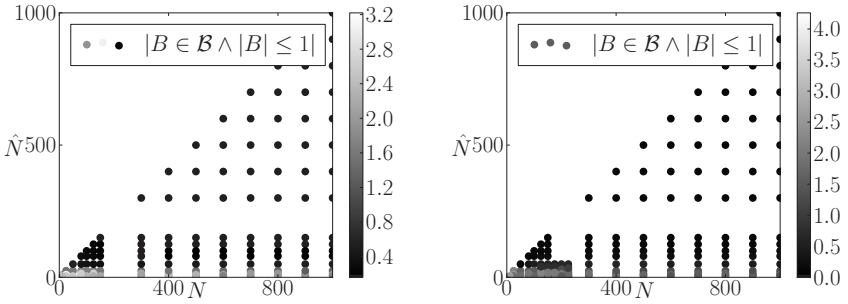
6.A Additional Graphs



(a) P@10



(b) Number of bursts



(c) Number of very small bursts

Figure 6.9: Changes of the (a) precision at 10, (b) number of bursts, (c) number of bursts which contain ≤ 1 document that is in N_B. The changes are 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. Figures on the left and right are based on temporal distributions using the retrieval score and counts, respectively.

6.B Vendler Classes of the Queries

The classes are based on the verb classes introduced by Vendler [257].

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