Topic Evolution in a Stream of Documents

André Gohr
Leibniz Institute of Plant
Biochemistry, IPB, Germany
agohr@ipb-halle.de

Alexander Hinneburg
Martin-Luther-University
Halle-Wittenberg, Germany
hinneburg@informatik.uni-halle.de

René Schult and Myra Spiliopoulou
Otto-von-Guericke-University Magdeburg, Germany
{schult,myra}@iti.cs.uni-magdeburg.de

Abstract

Document collections evolve over time, new topics emerge and old ones decline. At the same time, the terminology evolves as well. Much literature is devoted to topic evolution in finite document sequences assuming a fixed vocabulary. In this study, we propose "Topic Monitor" for the monitoring and understanding of topic *and* vocabulary evolution over an infinite document sequence, i.e. a stream. We use Probabilistic Latent Semantic Analysis (PLSA) for topic modeling and propose new folding-in techniques for topic adaptation under an evolving vocabulary. We extract a series of models, on which we detect *index-based topic threads* as human-interpretable descriptions of topic evolution.

1 Introduction

Scholars, journalists and practitioners of many disciplines use the Web for regular acquisition of information on topics of interest. Although powerful tools have emerged to assist in this task, they do not deal with the volatile nature of the Web. The topics of interest may change over time *as well as* the terminology associated with each topic. This yields text mining models obsolete and profile-based filters ineffective. We address this problem by a method that monitors the evolution of topic-word associations over a document stream. Our "`TopicMonitor`" discovers topics, adapts them as documents arrive and detects *topic threads*, while considering the *evolution of terminology*.

Most of the studies on topic evolution can be categorized into methods for (1) finite sequences and (2) for streams. Methods of the first category observe the arriving documents as a finite sequence and infer model parameters under the closed world assumption. Essentially, the document collection is assumed to be known. The vocabulary over all documents induces a fixed feature space. The second category makes the open world assumption, thus allowing that new documents arrive, new words emerge and old ones are forgotten. This implies a changing feature space for documents over time.

Stream-based approaches are intuitively closer to the volatile nature of the Web. However, evolutionary topic detection methods, including recent ones based on Probabilistic Latent Semantic Analysis and Latent Dirichlet Allocation [11, 4, 18] all assume that the documents form a finite sequence. To deal with a document stream, they take a *retrospective* view of the world: at each new timepoint, all information seen so far is used to re-build an extended model. This perspective has two disadvantages. First, it only works if the document stream is slow such that retrospection of the complete past is feasible. Second, the feature space grows with time, thus giving raise to the curse of dimensionality problem: the number of data points needed for a reliable estimate (of any property) grows exponentially with the number of features. Since a stream cannot be slow and fast at the same time, evolutionary topic monitoring requires adaptation to both changes in the data distribution and in the vocabulary/feature space.

`TopicMonitor` deals with these problems by adapting the feature space *and* the underlying model *gradually.* We invoke Probabilistic Latent Semantic Analysis (PLSA) in a window that slides across the stream: as the window slides from one timepoint to the next, we delete outdated words and documents, incorporate new documents into the previous model and then adapt the old model to new words, deriving the current model of each timepoint.

Gradual model adaptation brings an inherent advantage over re-learning at each discrete timepoint: not only the model as a whole is adapted, rather each discrete topic built at timepoint t_i transforms naturally to its followup topic at t_{i+1}. Hence, the adaptation process leads to a separation of topics into distinct *index-based topic threads* over time. These threads provide a comprehensive summary of topic and terminology evolution. In Section 3.2 we describe how we extend conventional PLSA for dynamic data, explain our incremental model adaptation over a document stream and show how index-based topic threads are built.

The evaluation of our approach is not trivial, because it involves comparing methods that learn on different feature spaces. In Section 4 we describe our evaluation framework and describe the reference method for our experiments, which we present in Section 5.

2 Related Work

Many studies on topic evolution derive topics by document clustering. Most of them consider a fixed feature space over the document stream: Morinaga and Yamanishi [12] use Expectation-Maximization to build soft clusters. A topic consists of the words with the largest information gain for a cluster. Aggarwal and Yu [1] trace droplets over document clusters. A droplet consists of two vectors that accommodate words associated with the cluster. The cluster evolution monitor MONIC [17] and its variants [15, 14, 13] treat topic evolution as a special case of cluster evolution over a feature space that may change, too.

Topic models like PLSA [9] and Latent Dirichlet Allocation (LDA) [5] characterize a topic as multinomial distribution over words rather than as cluster of documents. This is closer to the intuitive notion of "topic" in a collection. Most of evolutionary topic models based on PLSA or LDA [11, 4, 18] make the closed world assumption: they observe the arriving documents as a finite sequence with a fixed vocabulary. For example, Mei and Zhai [11] assume the complete vocabulary to be known *and* a static background model over all timepoints. The background model propagates statistics about non-specific word usage in time (forwards and backwards).

The retrospective approach of Mei and Zhai [11] builds a PLSA model at each timepoint in addition to the background model that is fixed over all timepoints. Griffiths and Steyvers [8] build a single LDA model based on collapsed Gibbs sampling to find scientific topics. They assign temporal properties to topics, such as becoming "cold" or "hot". Incremental LDA [16] updates the parameters of the LDA model as new documents arrive, but again assumes a fixed vocabulary and does neither forget the influence of past documents nor of outdated words. The distributed asynchronous version of LDA [3] allows to incrementally grow an LDA model. However, it still assumes a static vocabulary.

AlSumait et al. [2] propose Online LDA. They extend the Gibbs sampling approach suggested by Griffiths and Steyvers to handle streams of documents. Gibbs sampling at one timepoint is used to derive the hyperparameters of the topic-word associations at the next timepoint, so that successive LDA models are coupled. New words are collected as they are seen, so AlSumait et al. do not assume that the whole vocabulary is known in advance. However, the vocabulary grows over time so that the curse of dimensionality problem still occurs.

The topics over time model (TOT) [18] extends LDA to generate timestamps of documents. In this way, a topic is discovered in exactly those timepoints in which the vocabulary on it is homogeneous. In other words, the model associates a timepoint to some static topic.

Dynamic topic model (DTM) [4] uses a basic LDA model at each timepoint. Hyperparameters of these LDA models are propagated forward and backward in time via a state space model. A "dynamic topic" is a sequence of vectors over time. Each vector is a multinomial distribution over words of the fixed vocabulary.

`TopicMonitor` derives the model for the current timepoint by adapting the model of the previous timepoint to both new documents and words while forgetting old documents and obsolete words. Closest to our work is the recently published incremental probabilistic latent semantic indexing method of Chou and Chen [7]. We additionally address the issue of overfitting by investigating the maximum a posteriori estimation of the underlying PLSA model.

Differences Between PLSA and LDA. Since PLSA and LDA are frequently juxtaposed in the topic modeling literature, we provide a brief discussion of their differences here. The PLSA approach proposed by Hofmann [9] does not use prior distributions for the model-parameters. The LDA approach proposed by Blei et al. [5] uses Dirichlet priors for model-parameters: the hyperparameters are learned by Empirical Bayes from data. However, many LDA-oriented approaches [8, 3, 19, 20] on collapsed Gibbs sampling always assume fixed hyperparameters. These variants of LDA come quite close to the maximum a posteriori (MAP)-PLSA model, as we used it here. Beside inference, the only notable difference between the two approaches is that MAP-PLSA still uses empirically estimated document-probabilities[1], while LDA never uses this quantity. In our approach document-probabilities are needed to account for new words, as we describe in the next section.

3 Topic Discovery With an Adaptive Method

The core of `TopicMonitor` is an adaptive process deriving a sequence of PLSA models over time. Each PLSA model is adapted from the previous one to comprise

[1] Document-probabilities $P(d)$ are denoted by δ_d later on.

Table 1: Summary of Notation

$\lvert X\rvert$	number of elements in set X
l	size of sliding window in timepoints
D, W	set of documents, words (vocabulary)
d, w, z	denote documents, words, hidden topics
k, i	run over topics and timepoints
$n_{d,w}$	number of occurrences of (d, w)
$\zeta = (\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{\delta})$	PLSA model
K	number of hidden topics of ζ
$\boldsymbol{\theta} \in \mathbb{R}^{\lvert D\rvert \times K}$ $\boldsymbol{\omega} \in \mathbb{R}^{K \times \lvert W\rvert}$ $\boldsymbol{\delta} \in \mathbb{R}^{\lvert D\rvert}$	document-specific topic-mixture proportions topic-word associations empirical document-probabilities
$\boldsymbol{\tau} \in \mathbb{R}^{\lvert W\rvert}$	empirical word-probabilities
m	EM-iterations for recalibration
s, r	smoothing parameters controlling prior distributions of topic-mixture proportions (s) and topic-word associations (r), respectively

smooth evolution of topics over time. We use a Maximum A Posteriori (MAP) estimator of the PLSA model[2] to guard against overfitting and we propose new folding-in techniques for adaptation of PLSA models. We first present how we apply the MAP-principle to estimate a PLSA model and introduce our new folding-in techniques upon it. In Subsection 3.3 we describe how our adaptive PLSA uses MAP and performs folding-in to learn a new model gradually from a previous one. The notation is summarized in Table 1.

3.1 MAP-PLSA and Folding-In of Documents.

Conventional PLSA [9] models co-occurrences (pairs of documents and words)[3] by a mixture model with K components (topics). Modeled documents are denoted by D, modeled words by W. The number of times a (document,word) pair occurs is denoted by $n_{d,w}$.

Assuming K hidden topics z_k $(1 \leq k \leq K)$, the probability of observing document d and word w decomposes into $P(d, w) = P(d) \sum_{k=1}^{K} P(w|z_k)P(z_k|d)$. Hence, a PLSA model ζ is parametrized by a tuple $(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{\delta})$ containing:

1. $\forall\, d \in D:$ document-specific topic-mixture proportions $\boldsymbol{\theta} = [\boldsymbol{\theta}_d] = [\theta_{d,k} = P(z_k|d)]$,
   each K-dimensional row $\boldsymbol{\theta}_d$ fulfills: $\sum_{k=1}^{K} \theta_{d,k} = 1$
2. $\forall\, w \in W:$ topic-word associations $\boldsymbol{\omega} = [\boldsymbol{\omega}_k] = [\omega_{k,w} = P(w|z_k)]$,
   each row $\boldsymbol{\omega}_k$ fulfills: $\sum_{w \in W} \omega_{k,w} = 1$
3. $\forall\, d \in D:$ document-probabilities $\boldsymbol{\delta} = [\delta_d = P(d)]$

Each row $\boldsymbol{\omega}_k$ is a multinomial distribution over all modeled words and describes the k^{th} topic.

The log-likelihood of the data D given the PLSA model ζ is $\sum_{d \in D} \sum_{w \in W} n_{d,w} \cdot \log P(d, w)$ with $P(d, w) = \delta_d \sum_{k=1}^{K} \omega_{k,w}\ \theta_{d,k}$.

The parameters are usually estimated by maximizing the likelihood (cf. [9]). Document-probabilities $\boldsymbol{\delta}$ are estimated by

$$\delta_d = \sum_{w \in W} n_{d,w} / \sum_{d' \in D} \sum_{w \in W} n_{d',w} \tag{3.1}$$

The EM-algorithm is used to estimate the other parameters $\boldsymbol{\theta}, \boldsymbol{\omega}$.

MAP estimation of $\boldsymbol{\theta}, \boldsymbol{\omega}$ determines those parameters that maximize the log-a-posteriori probability. The following log-priors for topic-mixture proportions and topic-word associations are used:

$$\log P(\zeta) = \sum_{d \in D} \log \mathrm{Dir}(\boldsymbol{\theta}_d|s) + \sum_{k=1}^{K} \log \mathrm{Dir}(\boldsymbol{\omega}_k|r) \tag{3.2}$$

The hyperparameters of the Dirichlet distributions (Dir) are determined by $s > -1$ and $r > -1$ as follows: $[s+1]_{k=1,\ldots,K}$ and $[r+1]_{w \in W}$. The parameters s and r are called smoothing parameter for the estimates of $P(z|d)$ and $P(w|z)$. Setting $s = 0$, $r = 0$ makes MAP estimates equal to ML estimates because the Dirichlet priors become uniform. Note that the priors remain fixed throughout EM-learning (in contrast to LDA [5] which learns the hyperparameters itself). The EM-algorithm [10] leads to these update-equations during the $\langle t+1 \rangle$th iteration:

E-step:

$$\gamma_{d,w}^{k} = P(z_k|w, d, \zeta^{\langle t \rangle}) = \frac{\underline{\omega_{k,w}^{\langle t \rangle}}\ \theta_{d,k}^{\langle t \rangle}}{\sum_{k'=1}^{K} \underline{\omega_{k',w}^{\langle t \rangle}}\ \theta_{d,k'}^{\langle t \rangle}} \tag{3.3}$$

M-step:

$$\theta_{d,k}^{\langle t+1 \rangle} = \frac{s + \sum_{w \in W} n_{d,w}\ \gamma_{d,w}^{k}}{K \cdot s + \sum_{w \in W} n_{d,w}} \tag{3.4}$$

$$\underline{\omega_{k,w}^{\langle t+1 \rangle}} = \frac{r + \sum_{d \in D} n_{d,w}\ \gamma_{d,w}^{k}}{|W| \cdot r + \sum_{d \in D} n_{d,w'}\ \gamma_{d,w'}^{k}} \tag{3.5}$$

If $n_{d,w}$ or $\gamma_{d,w}^{k}$ become very small, the EM update-equations (3.4) and (3.5) will be problematic since the maximum of the posterior density might get undefined.

[2] A conceptually similar estimator is used in [6].

[3] Co-occurrence means that a particular document-ID appears together with a ID of a specific word type stating a word is seen in a document.

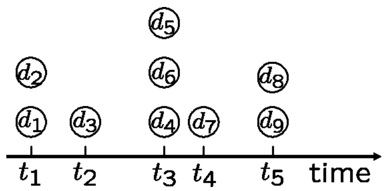

Figure 1: Document stream inducing partial ordering on documents

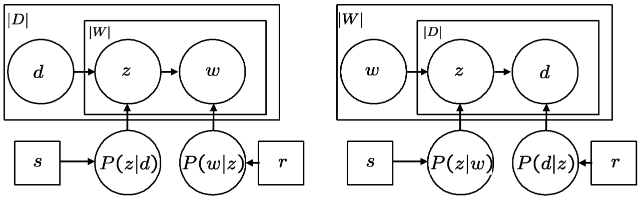

Figure 2: MAP-PLSA-model allowing to fold-in new documents (left) and inverted MAP-PLSA-model allowing to fold-in new words (right). Bayesian calculus is used to transform between both forms.

This might result in negative parameter estimates [6]. To avoid that problem, we restrict the smoothing parameters to $s \geq 0$ and $r \geq 0$.

The folding-in procedure by Hofmann [9] allows to incorporate new documents into an already trained model. MAP-folding-in of a new document d' into an existing model ζ estimates the topic-mixture proportions $\boldsymbol{\theta}_{d'}$: the EM-algorithm continues to estimate $\boldsymbol{\theta}_{d'}$ having fixed the topic-word associations $\boldsymbol{\omega}$ (underlined in Eq. 3.3 and 3.5). The extended model comprises the concatenation of $\boldsymbol{\theta}_{d'}$ row-wise to the already derived ones: $[\boldsymbol{\theta}\ \boldsymbol{\theta}_{d'}]$. The document-probabilities are re-estimated for all modeled documents using Eq. 3.1. It is required that d' contains some words modeled by ζ since d' is reduced to those words for folding-in. To gradually adapt a PLSA model according to a document-stream, we use MAP-folding-in of new arriving documents into an existing model, as explained in Section 3.2.

3.2 Adaptive PLSA for Topic Monitoring – Overview. Streaming documents $d_1, d_2, \ldots$ arrive at (possibly irregularly spaced) timepoints $t_1, t_2, \ldots$. We allow that several documents arrive at the same time. Hence, time induces a partial ordering on documents. Figure 1 depicts a possible stream of documents.

Topic-detection at timepoint t_i implies building a PLSA model ζ^i upon all documents that arrive in the interval $(t_i - l, t_i]$. The parameter l affects the size of the interval and allows for taking some documents from the near past into account. The result is a sequence of PLSA models $\zeta^1, \zeta^2, \ldots$, one at each point in time.

Almost all topic monitoring methods so far as explained in Section 2 exploit the old model ζ at timepoint t_{i-1} to build $\overline{\zeta}$ at t_i assuming an unchanged vocabulary from one timepoint to the next. We judge this assumption questionable for real stream data because future vocabulary is unknown at each timepoint. If the vocabulary does change, the baseline approach will be to build models ζ^i independently of each other taking the vocabulary of $(t_i - l, t_i]$ into account. To uncover threads of topics over time, one would measure similarity between each detected topic at successive timepoint and "connect" those being most similar according to some similarity function. For PLSA-based topic monitoring, we call this baseline "independent PLSA".

In contrast thereto, `TopicMonitor` builds the new model $\overline{\zeta}$ by adapting the one of the previous timepoint ζ to new words and new documents – it *evolves* the old model to the new one. Because successive PLSA models have been adaptively learned, the kth analyzed topic at t_i evolves from the previously analyzed kth topic at t_{i-1}. Hence, we formally define for each k an *index-based topic thread* that consists of the sequence of word-topic associations $\omega_k^1, \omega_k^2, \ldots$ at timepoints $t_1, t_2, \ldots$. Each *index-based topic thread* describes the evolution of hidden topics z_k over time.

Essential to our approach are the two equivalent forms of an PLSA model which assume the following different decompositions:

$$P(d, w) = P(d) \sum_{k=1}^{K} P(w|z_k) P(z_k|d) \tag{3.6}$$

$$= P(w) \sum_{k=1}^{K} P(d|z_k) P(z_k|w) \tag{3.7}$$

Plate models of these two forms are given in Figure 2. If a PLSA model was given in its document-based form (Eq. 3.6), new documents would conveniently be folded-in into that model. In the same way, new words would conveniently be folded-in into an existing model if the model was given in its word-based form (Eq. 3.7).

Since not only the documents but also the vocabulary changes over time, we want the adaptive process to learn model $\overline{\zeta}$ at timepoint t_i from model ζ at timepoint t_{i-1} by adapting to both: new documents and new words. This includes to incorporate new and to "forget" out-dated words. Forgetting out-dated words prevents `TopicMonitor` from accumulating all words ever seen and hence `TopicMonitor` does not unnecessarily blow up the feature space over time.

Roughly, the adaptive process that evolves the model $\overline{\zeta}$ at time t_i from the model ζ at time t_{i-1} works as follows.

1. remove documents modeled by ζ but not covered by $(t_i - l, t_i]$

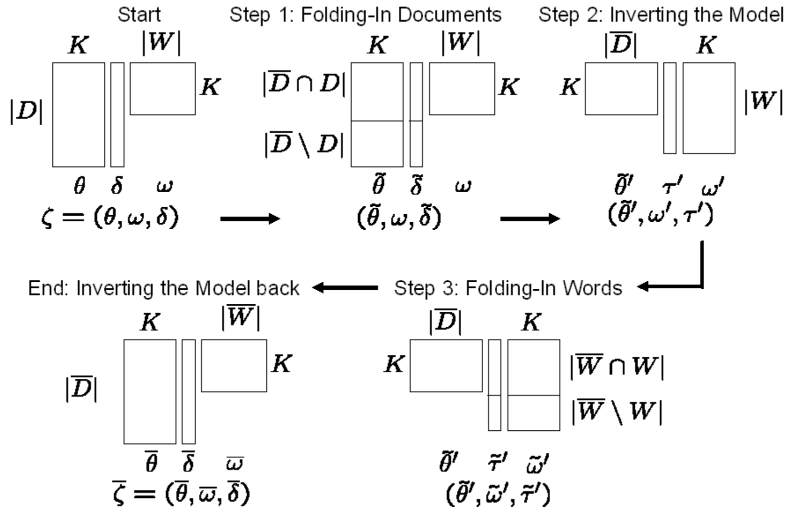
Start
Step 1: Folding-In Documents
Step 2: Inverting the Model
K
$|W|$
$|D|$
K
$|\overline{D} \cap D|$
$|\overline{D} \setminus D|$
$|\overline{D}|$
$|W|$
θ δ ω
$\zeta = (\theta, \omega, \delta)$
$\tilde{\theta}$ $\tilde{\delta}$ ω
$(\tilde{\theta}, \omega, \tilde{\delta})$
$\tilde{\theta}'$ τ' ω'
$(\tilde{\theta}', \omega', \tau')$
End: Inverting the Model back
Step 3: Folding-In Words
$|\overline{W}|$
$|\overline{D}|$
$|\overline{W} \cap W|$
$|\overline{W} \setminus W|$
$\overline{\theta}$ $\overline{\delta}$ $\overline{\omega}$
$\overline{\zeta} = (\overline{\theta}, \overline{\omega}, \overline{\delta})$
$\tilde{\theta}'$ $\tilde{\tau}'$ $\tilde{\omega}'$
$(\tilde{\theta}', \tilde{\omega}', \tilde{\tau}')$

Figure 3: Overview of the steps necessary to adapt a model ζ at timepoint t_{i-1} to $\bar{\zeta}$ at timepoint t_{i+1}.

2. fold-in new documents covered by $(t_i - l, t_i]$ into ζ using its document-based form
3. use Bayesian calculus to invert ζ into its word-based form
4. remove words modeled by ζ not seen in $(t_i - l, t_i]$
5. fold-in word firstly seen in $(t_i - l, t_i]$
6. use Bayesian calculus to invert ζ again into its document-based form
7. recalibration gives the next PLSA model $\overline{\zeta}$

Figure 3 graphically describes the adaptive process of `TopicMonitor` . Mathematical details of that process are given in Section 3.3.

3.3 Adaptive PLSA for Topic Monitoring – Mathematics. For ease of presentation we denote by $\overline{\zeta}$ the PLSA model at timepoint t_i which should be evolved from model ζ at timepoint t_{i-1}. In addition, $\overline{W}$ denotes all words seen in $(t_i - l, t_i]$ and W all words seen in $(t_{i-1} - l, t_{i-1}]$. In the same way $\overline{D}$ and D are used.

Recall that a model $\zeta = (\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{\delta})$ in its document-based form consists of three parameter sets: document-specific topic-mixture proportions $\boldsymbol{\theta}$, topic-word associations $\boldsymbol{\omega}$ and document-probabilities $\boldsymbol{\delta}$. The first two parameter sets can be seen as matrices of dimensions $|D| \times K$ for $\boldsymbol{\theta}$ and $K \times |W|$ for $\boldsymbol{\omega}$. Folding-in new documents $d \in \overline{D} \setminus D$ translates to adding rows to the first matrix as explained in Section 3.1. Next, topic-mixture proportions of documents which are not covered anymore by the current window at timepoint t_i are not needed and deleted. The current document-specific topic-mixture proportions are given by: $\widetilde{\boldsymbol{\theta}} = [\boldsymbol{\theta}_d \ \boldsymbol{\theta}_{d'}]$ with $d \in D \cap \overline{D}$ and $d' \in \overline{D} \setminus D$. The new document-probabilities $\widetilde{\boldsymbol{\delta}} = [\widetilde{\delta}_d]$ with $d \in \overline{D}$ are re-estimated as shown in Equation 3.1 but using the old vocabulary W only.

New documents may introduce new words not contained in the current vocabulary W. The topic-word associations $\boldsymbol{\omega}$ of model ζ describe only words $w \in W$.

To handle new words $w \in \overline{W} \setminus W$, they will be fold-in into the current PLSA model. For that purpose that model is inverted into its word-based form by Bayesian calculus. Note that the elements of $\widetilde{\boldsymbol{\theta}}$ and $\boldsymbol{\omega}$ estimate conditional probabilities $P(z_k|d)$ and $P(w|z_k)$, respectively, which are inverted by the following formulae: $\widetilde{\theta}'_{d,k} = P(d|z_k) = {}^{P(z_k|d)P(d)}/_{P(z_k)}$ and $\omega'_{w,k} = P(z_k|w) = {}^{P(w|z_k)P(z_k)}/_{P(w)}$. The word probabilities for $w \in W$ are derived by marginalization $\tau'_w = P(w) = \sum_{d \in \overline{D}} \sum_{k=1}^{K} P(d)P(z_k|d)P(w|z_k) = \sum_{d \in \overline{D}} \sum_{k=1}^{K} \widetilde{\delta}_d \cdot \widetilde{\theta}_{d,k} \cdot \omega_{w,k}$. The inversion of the current model into its word-based form is derived by:

$$\widetilde{\theta}'_{k,d} = \frac{\widetilde{\theta}_{d,k} \cdot \widetilde{\delta}_d}{\sum_{d' \in \overline{D}} \widetilde{\theta}_{d',k} \cdot \widetilde{\delta}_{d'}} \quad \text{and}$$

$$\omega'_{w,k} = \frac{\omega_{k,w} \sum_{d' \in \overline{D}} \widetilde{\theta}_{d',k} \cdot \widetilde{\delta}_{d'}}{\tau'_w}$$

with $d \in \overline{D}$ and $w \in W$. This inversion gives $\zeta' = (\widetilde{\boldsymbol{\theta}}', \boldsymbol{\omega}', \boldsymbol{\tau}')$. Note that documents and words have changed their roles. Thus, $\boldsymbol{\omega}'$ denotes word-specific mixture proportions and $\widetilde{\boldsymbol{\theta}}'$ denotes document-topic associations. Figure 2 schematically depicts the models.

The word-based form ζ' allows folding-in words in a similar way the document-based form allows folding-in documents. New words $w \in \overline{W} \setminus W$ are folded-in into ζ' using their occurrences in documents of $\overline{D}$. In addition, mixture proportions of words which do not appear in $\overline{W}$ are deleted.

The resulting word-specific topic-mixture proportions are given by: $\widetilde{\boldsymbol{\omega}}' = [\boldsymbol{\omega}'_w \ \boldsymbol{\omega}'_{w'}]$ with $w \in W \cap \overline{W}$ and $w' \in \overline{W} \setminus W$. Word-probabilities $\widetilde{\boldsymbol{\tau}}' = [\widetilde{\tau}'_w]$ with $w \in \overline{W}$ are empirically re-estimated: $\widetilde{\tau}'_w = \sum_{d \in \overline{D}} n_{d,w} / \sum_{w' \in \overline{W}} \sum_{d \in \overline{D}} n_{d,w'}$.

Having incorporated new and deleted out-dated words, the current extended model $\widetilde{\zeta}' = (\widetilde{\boldsymbol{\theta}}', \widetilde{\boldsymbol{\omega}}', \widetilde{\boldsymbol{\tau}}')$ is inverted back into its document-based form using Bayesian calculus again.

Parameters according to new documents and words are derived separately by folding-in. In order to couple the influence of both: new documents, and new words, we run the full EM-algorithm for a small number m of iterations using all data in $(t_i - l, t_i]$. The result is the adapted model $\bar{\zeta} = (\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\omega}}, \bar{\boldsymbol{\delta}})$ at timepoint t_i.

Successively adapting ζ^i from ζ^{i-1} as new data arrive, gives a sequence of PLSA models $\zeta_1, \zeta_2, \ldots$. That sequence models K topics at each timepoint and thus their evolution over time. Tracing the topics having the same index over time is possible because the adaptive approach evolves the kth topic-word association $\bar{\boldsymbol{\omega}}_k$ at timepoint t_i from the kth topic-word association $\boldsymbol{\omega}_k$ at timepoint t_{i-1} without renumbering the topic indexes.

4 Evaluation Framework

Topic evolution with simultaneous adaptation to both an evolving document stream *and* an evolving vocabulary has not been intensively studied in the past. Thus, there is no generally accepted ground truth available, which would allow a cross model evaluation with TOT [18] or DTM [4]. Therefore, objective of our evaluation is to show the improvements of our approach with respect to the "independent PLSA model" as described in Subsection 3.2, which is the natural baseline to tackle the problem discussed here. Independent PLSA derives for each position of the sliding window a new PLSA model from scratch. If the influence of the background component is set to zero, the approach of Mei and Zhai [11] will be equal to independent PLSA. That background component uses global knowledge and thus cannot be used in a stream scenario. Hence, independent PLSA imitates the approach of Mei and Zhai as much as possible in a stream scenario.

We use a real data document stream, namely the proceedings of the ACM SIGIR conferences from 2000 to 2007.

4.1 Model Comparison. In the absence of ground truth, our adaptive PLSA approach can be compared to independent PLSA by computing perplexity on hold-out data. Informally, perplexity measures how surprised a probabilistic model is seeing hold-out data. Hence, it assesses the ability of model generalizing to unseen data.

Hold-out data are constructed by splitting the set of word-occurrences of the current data into a training part (80%) and a hold-out part (20%). We maintain two counters for occurrences of word w in document d: $n_{d,w}$ for the training part and $\hat{n}_{d,w}$ for the hold out part. Both counters sum up to the total number of the occurrences of (d, w) in the given data. During training, the learning algorithm does not see that documents contain actually some more words. Hold-out perplexity is computed on the hold-out part of the data unused during training. Hold-out perplexity according to an arbitrary PLSA-model $\bar{\zeta}$ at timepoint t_i is computed as:

$$\text{perplex}(\bar{\zeta}) = \exp\left(- \frac{\sum_{d \in \bar{D}, w \in \bar{W}} \hat{n}_{d,w} \log\left[\sum_{k=1}^{K} \theta_{d,k} \omega_{k,w}\right]}{\sum_{d \in \bar{D}, w \in \bar{W}} \hat{n}_{d,w}} \right) \tag{4.8}$$

The computation of hold-out perplexity does not require any folding-in of new documents. Thus it is not distorted in the way explained in [19].

Firstly, we analyze the impact of the smoothing parameters s and r used to define the prior distributions of the topic-mixture proportions and topic-word associations on the generalization ability of the model. All SIGIR documents are used as a single batch. We verify by this experiment the advantage of MAP-PLSA over the maximum-likelihood estimator of PLSA.

Further, we compare the performance of adaptive PLSA and independent PLSA by average hold-out perplexity. Our adaptive method derives the model at each timepoint by adapting the model of the previous timepoint: it folds-in new documents and words, forgets old documents and words, and recalibrates the parameters. Note that both models differ only by initialization. At each timepoint, document-word-pairs contained in the sliding window are held out in order to compute its perplexity. We analyze the average hold-out perplexity over all points in time versus the number of learning-steps. Learning-steps are either the number of EM-iterations for independent PLSA or the number of EM-recalibration steps m for adaptive PLSA. In addition, the number of EM-steps of adaptive PLSA to fold-in documents and words are also limited to m. For the purpose of a fair comparison, independent PLSA is restarted three times at each timepoint; only the best results are reported. This experiment analyzes whether coupling models by initialization indeed propagates useful information along the sequence of models and helps the models to better fit the data.

The third experiment investigates to which degree the natural order of the stream contains important latent information that helps to predict future documents. In addition, we analyze how the natural order influences prediction-power of PLSA models determined by our proposed adaptive approach compared to the independent one. We compute at each timepoint the predictive perplexity of documents from the next timepoint. To that end, documents at time t_i have to be folded-in into the model computed at timepoint t_{i-1}. Afterward, these predictive perplexities are averaged over all timepoints. Averaged predictive perplexities are computed for the natural stream order and for a randomly permuted stream. If our assumptions hold, the average predictive perplexity according to the permuted stream should be significantly larger. We follow the idea of half-folding-in [19] to avoid over-optimistic estimates of predictive perplexity. The idea is to fold-in new documents based on only a part of their words, while the rest of them is held out. The predictive perplexity according to these documents is computed by (4.8) only using the held-out part unseen during folding-in.

4.2 Comparison of Topic Threads.

We study the semantic meaningfulness of index-based topics threads over time obtained by our adaptive PLSA method. We have defined index-based topic threads for a given sequence of PLSA models in Section 3.2. These topic threads are primarily defined via a technical parameter: the index of the topic.

The independent PLSA model computes topics at different timepoints independently. Thus, topic indexes at different timepoints are arbitrary. Hence, semantically close topics at different timepoints have to be matched using some similarity measure in order to define threads of topics over time. Best-match topic threads are built as follows: each topic-word association from model ζ^i is matched to that topic-word association from model ζ^{i+1} being maximal similar according to cosine-similarity subject to minimum similarity threshold MinSim. The pairs of matching topic-word associations are called best-matchings.

Best-match topic threads constitute a more intuitive definition of topic threads that match human's intuition about meaningful semantics. In case of very well matching topologies of both, we argue that index-based topic threads indeed constitute semantic meaningfulness.

Year	2000	2001	2002	2003
$\lvert D\rvert$	57	71	85	87
Year	2004	2005	2006	2007
$\lvert D\rvert$	115	121	133	194

Table 2: Number of documents per year of the SIGIR data set

5 Experiments

Here, we describe details of the used data and the results of our experiments. Additionally, we demonstrate the ability of our approach to find topic threads which have meaningful semantics.

5.1 Data Sets.

We use a real-world data set, which contains articles published at the ACM SIGIR conferences from 2000 to 2007, as they appear in the ACM Digital Library. Only titles and abstracts of the documents are considered. Stemming and stopword removal was applied to all documents. Posters are not included because they often do not have abstracts. The amount of included documents per year is shown in Table 2. Further statistics of this data set are given in Figure 4. We refer to this data set as SIGIR hereafter.

We analyze the SIGIR data set at yearly timepoints. The sliding window spans the current point in time and, if it is of length $l > 1$, the previous $l - 1$ ones. Words constituting the feature space at a certain timepoint are those appearing in at least two documents covered by the respective sliding window.

5.2 Model Comparison.

5.2.1 Effect of Priors on PLSA.

MAP-PLSA allows in contrast to ML-PLSA to specify priors, which differ from a flat Dirichlet. We analyze the impact of the smoothing parameters s and r, which define the priors for topic-mixture proportions and topic-word associations respectively, (see Section 3.1). The influence of smoothing is estimated for $s, r \in \{0.01, 0.1, 1, 10, 100\}$. MAP-PLSA is trained with $K = 10$ on the total SIGIR data without time stamps. Average perplexities (50 repetitions) on 20% hold-out data are shown in Figure 5. Curves for $r = \{10, 100\}$ are not shown, as they have much larger perplexities. Setting $s = 0.1$, $r = 0.1$ yields significantly lower hold-out perplexities than others. Using $s = 0.01$, $r = 0.01$ yields a nearly flat Dirichlet prior, which corresponds to ML-PLSA. Thus, the results show: i) that MAP-PLSA opens room for improvement over ML-PLSA, and ii) that a careful study to adapt hyperparameters is necessary in order to yield good performance.

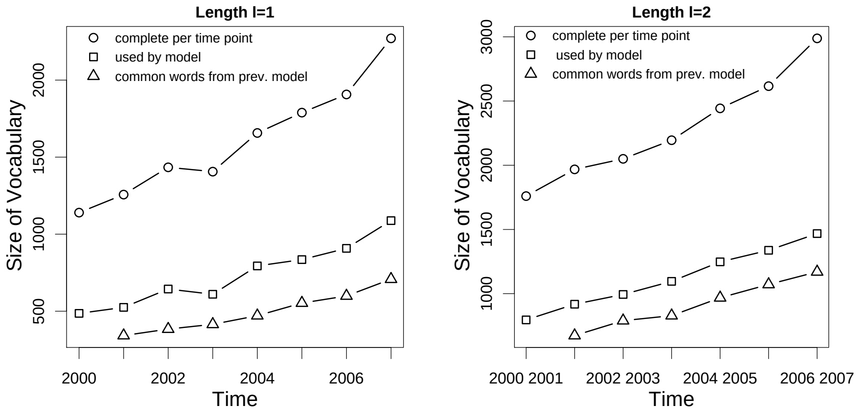

Figure 4: SIGIR word statistics for different window-lengths.

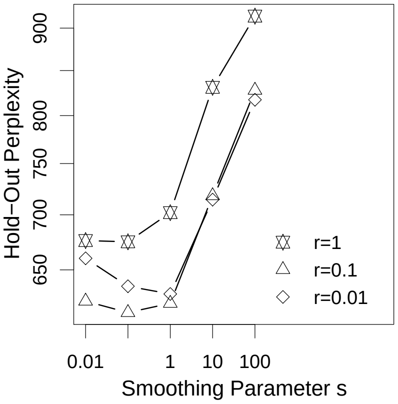

Figure 5: Influence of smoothing versus hold-out perplexity (lower values are better). Smoothing-parameters s and r control the Dirichlet hyperparameters for the estimates of $P(z|d)$ and $P(w|z)$, respectively.

5.2.2 Comparison of Adaptive and Independent PLSA. In this experiment, we compare our adaptive PLSA with independent PLSA trained at each sliding window independently. Technically speaking, the two methods differ in the information used during the initialization phase: our method folds-in documents and words, while the independent PLSA performs a random initialization. To omit side-effects of randomness, we restart the independent PLSA three times with different random initializations. We allow each method to perform m iterations.

Figure 6 shows the hold-out perplexities for different values of $m \in \{1, 2, 4, 8, 16, 32, 64, 128\}$ and different lengths of the sliding window: $l \in \{1, 2\}$. Each individual experiment is repeated 50 times and we present averages of the hold-out perplexities. The results show that our approach needs less iterations to reach the minimal hold-our perplexity. More importantly, the best reached hold-out perplexity for adaptive PLSA is much better than for independent PLSA. This demonstrates that the initialization by folding-in new documents and words helps the model to better fit the data.

5.2.3 Influence of Natural Stream Order. The third experiment evaluates the effect of relying on the natural order of the document stream. We use the best parameter settings estimated in the previous hold-out experiments. We evaluate how well the topic models may predict documents from the next timepoint by estimating predictive perplexity on the new documents. As explained in Section 4.1, we follow the concept of half-folding-in [19].

To assess the influence of the natural stream order we compare average predictive perplexity (50 repetitions) on the natural stream order versus random stream order. In particular, experiments for random streams use a different order each time the experiment is repeated. In addition we compare adaptive PLSA with independent PLSA. The results are shown in Figure 7.

In case of $l = 1$, the natural order of the stream significantly helps to predict documents from the next timepoint. The adaptive process to evolve PLSA models from each other can exploit this latent information much better than independent PLSA. In case of $l = 2$, the impact of the natural order is more blurred. Since SIGIR concepts do not change drastically from year to year, it is quite expected that a larger window size will blur the difference between natural order and random order.

5.3 Topic Threads and Their Semantics. In the following we experimentally analyze whether index-

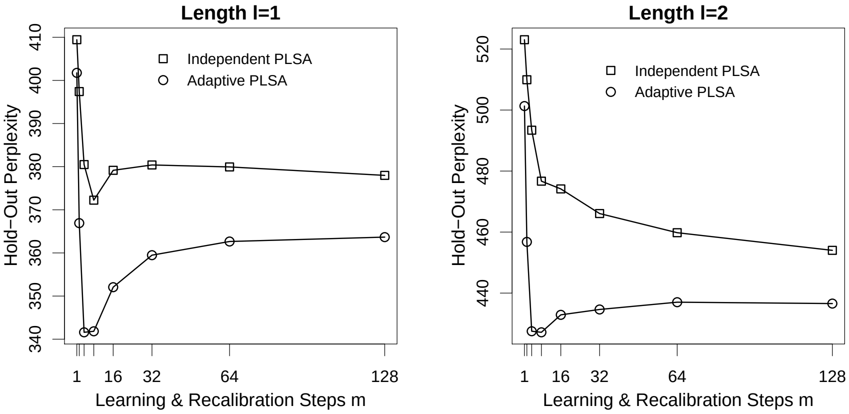

Figure 6: Impact of number of recalibration-steps on the achieved hold-out perplexity (lower values are better).

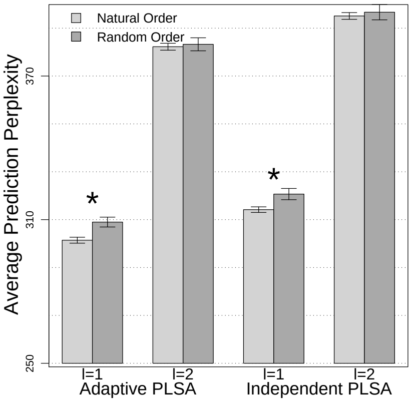

Figure 7: Impact of natural stream order on predicting new documents from next timepoint using different sizes of sliding window. The parameter l is the length of the sliding window used for training models. The star notes significant differences using t-test with significance level 0.05.

based topic threads show semantic meaningfulness and present a concrete example of index-based topic threads determined by TopicMonitor using the SIGIR data.

5.3.1 Comparing Topic Threads. Following our evaluation framework we compute index-based and best-match topic threads on a sequence of PLSA models obtained by TopicMonitor with 10 topics. Multinomial distributions (topic-word associations) at different timepoints are made comparable by filling in zero-probabilities for the differing words. This is necessary, since we allow differing vocabularies over time. We use cosine similarity, which in contrast to KL-divergence [11] can cope with zero-probabilities. Given that sequence of PLSA models, we compute the percentage of best-matchings which connect two topic-word associations having the same index. A high value indicates that both heuristics generate threads which have similar local topological structure. This means that index-based topic threads are similar to those that are constructed following human intuition and thus are semantically meaningful.

The results are shown in Figure 8 for different lengths of the sliding window $l \in \{1, 2\}$. The percentage is never below 94.5% ($l = 1$) even if all best-matchings are included (MinSim $= 0$). This demonstrates that matching-based threads, similar to those used in [11], can be approximated by index-based threads, which are build by concatenating the topics with the same index. This demonstrates further that our initialization-based propagation of word statistics keeps the sequence of models consistent, meaning index-based threads are similar in structure to dynamic topics found by DTMs [4].

5.3.2 Semantics of Selected Topic Threads. In Table 3 we show an illustrative example of index-based topic threads determined using SIGIR data. In order to produce a small example fitting paper size, we chose the number of topics $K = 5$ and the length of the sliding window $l = 1$. Each thread consists of a sequence of topic-word associations over time having the same index. The top 30 words with largest topic-word association are printed for each timepoint and for each topic-index. As seen so far, adaptive learning of the sequence of PLSA models enables for semantically meaningful threads. If PLSA was applied to each sliding window anew, this continuity would not haven been achieved.

Even without a deep inspection of the SIGIR conference subjects, we find (in the SIGIR context) mean-

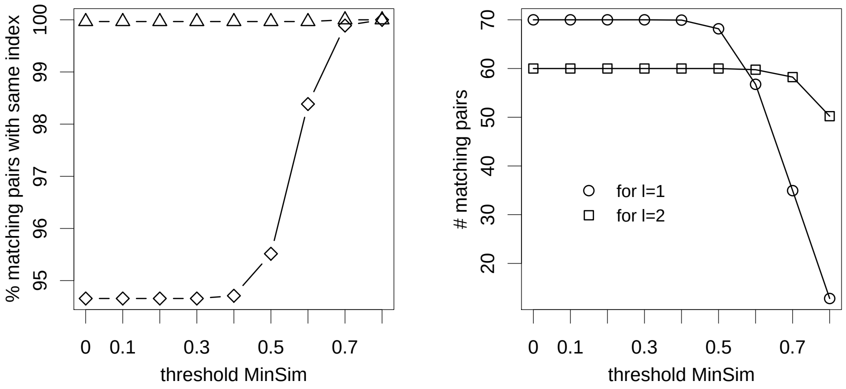

Figure 8: Percentage of best-matchings with same index (left). Total number of matching pairs. MinSim is the lower bound of similarity between consecutive topics of a thread (right).

ingful index-based topic threads:

Thread 1: main theme is “Evaluation”; the early subarea “Multilingual IR” disappears later

Thread 2: “Presentation”; later with emphasis on “Multimedia”

Thread 3: “Supervised Machine Learning”; originally “Classification”; later more elaborate aspects such as “Feature Selection” and “SVM”s

Thread 4: “Web”; originally from viewpoint “Information Extraction” and “Link traversal”; later “Web Search” and “User Queries”

Thread 5: “Document Clustering”

We see some strongly evolving index-based topic threads while others are remarkably stable. For example, Thread 5 is clearly on document clustering. Thread 3 is rather on supervised machine learning (a multiterm that does not appear among the words), originally focusing on classification and later elaborating on specialized methods like support vector machines. Another concept of Thread 3 is feature selection. The importance of concepts changes in Thread 2, too. It is about presentation, visualization and frameworks. Later, much words are about multimedia (early years: audio) for which presentation forms are indeed expected to be important.

We study Thread 4 a bit closer in Figure 9. This index-based topic thread is about Web, originally studied in the context of information extraction and link traversal. Later, web search and user queries become dominant.

Further experiments investigating longer and more diverse streams are planned for future work.

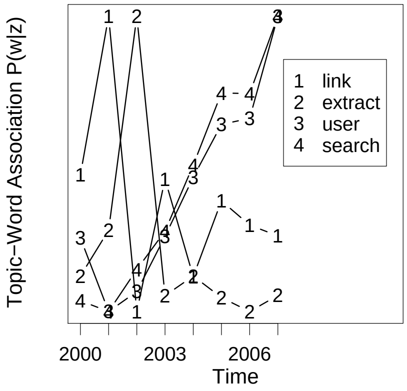

Figure 9: Evolution of topic-word associations over time for the Web index-based topic thread (4).

6 Conclusions

We study topic monitoring over a stream of documents in which both: topics and vocabulary may change. We waive the assumption that the complete vocabulary is known in advance. Our approach allows to study fast streams and alleviates the negative impact of large vocabularies with many words of only temporary importance.

`TopicMonitor` discovers and traces topics by using a new, adaptive variant of the well-known PLSA. Our *adaptive PLSA* derives the model at a certain timepoint by adapting the model found at the previous point in time to new documents and unseen words.

We evaluate `TopicMonitor` on a document-stream of ACM SIGIR conference papers (2000–2007). Our evaluation concerns the predictive performance of our approach to adaptively learn PLSA models in comparison to a baseline. Additionally, the semantic compre-

Table 3: Index-based topic threads for the SIGIR from 2000-2007. Shown are top-30 words per thread and timepoint.

K	2000	2001	2002	2003	2004	2005	2006	2007
1	evalu search queri document relev approach result method effect retriev predict base experi ir user measur filter compar test collect improv number recal demonstr text translat visual sentenc paper experiment	queri document relev evalu search retriev languag result translat summari effect engin term techniqu approach system summar method rank inform highli gener paper select better propos user feedback sentenc probabl	queri retriev search term document result relev techniqu inform translat effect system perform improv approach show arab collect ir evalu paper base present statist test trec word stem resourc automat	retriev queri system effect evalu search collect word inform techniqu result perform relev xml automat term document expans interact translat test match rank languag differ show demonstr present improv gener	retriev queri relev document collect inform system feedback evalu term translat rank topic perform improv measur xml result show effect test search word blind experi phrase precis clir method length	queri retriev term inform feedback perform system document evalu relev translat result approach test measur effect user collect paper precis trec show improv word averag expans statist experi maximum crosslanguag	queri retriev document relev result evalu term feedback improv precis collect rank perform method system measur user set effect estim show averag test statist expans demonstr judgment element techniqu mean	queri retriev relev system rank evalu perform collect measur effect document term method result test precis improv techniqu user ir set compar judgment topic approach feedback differ propos select two
2	inform user retriev model document system base approach process studi differ cognit structur topic need develop search present ir type on techniqu session poster contribut represent propos gener build integr	model retriev inform user languag system document estim feedback problem probabilist studi present concept approach experiment search process provid markov queri produc framework word method perform manag brows collect implement	model inform languag retriev user document queri collect method paramet perform present propos base estim smooth music paper tradit precis differ interfac categori framework access integr data probabl cooccurr describ	model inform retriev imag user languag video perform annot show type document probabilist probabl describ base task gener visual index propos queri develop tool paper experi distribut music estim predict	model retriev inform languag imag approach document framework distribut propos video perform show ir process gener paper term present result paramet set context belief empir us provid level annot automat	model retriev imag inform languag approach annot gener propos show function ir term perform probabilist estim exist present framework paper concept set deriv result relationship document experi relev novel depend	model retriev inform document languag approach probabl base space framework probabilist topic translat structur ir measur knowledg show sampl oper paper better relev smooth logic result on propos contextu relat	model retriev inform languag base term imag propos approach show context gener depend paper expert document weight perform estim data import collect semant word algorithm present probabl improv frequenc match
3	perform queri select classifi method classif text vector system databas result train data retriev model hierarch distribut improv effect machin boolean similar collect good categor categori find accuraci support central	text perform classif similar improv stori approach combin featur distribut data index classifi categor consist structur xml strategi imag new train differ support three machin base corpora result learn svm	text index classifi result classif data corpu approach time invert learn categor process method paper size show algorithm effici experi applic file evalu collect space perform structur frequenc filter train	text classif classifi method featur result structur categor term approach extract learn paper algorithm experi machin vector improv effici show content accuraci effect gener categori sourc segment train imag optim	featur classif text method learn algorithm perform achiev automat train name improv entiti classifi machin accuraci result structur problem paper detect collect semant techniqu effect propos show pattern gener categor	method text classif featur detect extract base automat show data summar document titl new summari techniqu collect classifi problem articl improv train match pattern question propos inform paper relat categor	text question document featur classif approach method task sentenc summar answer paper learn system classifi train base compar problem relat inform propos set three hierarch new present novel brows svm	featur learn text classif answer approach paper task classifi base method rank train sentenc problem question propos detect inform segment svm label new result algorithm event data perform opinion differ
4	web qualiti page system document techniqu question retriev answer paper task high gener link propos automat profil present user learn avail collect investig statist larg manual metric dictionari show problem	web page topic answer link automat text question task gener system algorithm data show find inform analysi method differ perform two identifi extract paper larg queri import content experi approach	web extract data system item text analysi search page separ set collect question rank perform tool redund problem merg engin find design recommend cluster form answer gener techniqu specif differ	ir web research search system question answer inform data task field databas link perform commun area import user engin evalu find algorithm page combin present goal work structur need specif	web search page system user task document answer evalu result person inform present engin perform analysi techniqu provid effect base similar commun structur ir databas link current applic time data	search web user page engin result algorithm inform collect rank data link system paper relationship studi base task analysi relev answer locat question structur document interfac two order provid effect	search web user inform page result network rank algorithm structur task engin link social system gener find interest behavior queri pagerank present differ provid on paper problem commun content interact	search web user queri inform page engin result relev log present studi develop system rank link find provid task content person suggest gener support behavior similar ad sourc snippet file
5	document cluster word method retriev algorithm inform perform system improv featur learn filter event set text result combin precis relev track paper base threshold similar probabilist topic model term collect	document method base threshold index segment score term distribut algorithm retriev topic relev describ investig detect optim inform perform cluster weight metasearch automat paper model represent scheme task effect present	document cluster topic algorithm set method score evalu model similar weight result averag precis class differ comput gener compar inform function select base featur sentenc obtain initi improv track detect	document topic cluster method base algorithm set approach similar propos semant object find concept result estim sentenc relev score select show filter measur term qualiti improv experiment resourc evalu databas	document cluster filter method algorithm user data semant index differ collabor propos paper lsi effect result base similar two structur weight rate topic inform combin problem perform factor approach normal	method document topic algorithm optim semant similar data index result approach problem latent base inform time propos util set differ filter event user effici collabor show comput paper track two	cluster document data method similar semant propos algorithm index rate show filter time base result user combin topic space graph analysi two object techniqu item latent text larg paper approach	document index cluster method algorithm result propos optim perform problem time filter effici similar data show combin multipl number inform topic spam paper detect base text gener comput cach experi

hensiveness of the topics being monitored is analyzed. Since no ground truth for such studies exists, nor a baseline for topic monitoring with a vocabulary that is not known in advance, we introduce a new evaluation framework. The evaluation is based on the concepts of *perplexity* and of *topic threads*. The former captures the extent to which a model is able to generalize to unseen data. The latter reflects whether semantically related topics found at different timepoints are observed as part of the same "thread". Our experiments show that adaptation of models by `TopicMonitor` leads to better results than construction of models from scratch at each timepoint with respect to prediction performance. In addition, the experiments indicate that index-based topic threads monitored by `TopicMonitor` are indeed semantically meaningful.

Our `TopicMonitor` is appropriate for fast streams, since it does not require to pause the stream in order to recompute the feature space before building models. We intend to study the speed and space demands of our approach and devise methods that allow topic monitoring on very fast and large streams, e.g. online blogs. Further research issues are the identification of recurring topics and the detection of groups of correlated topics that occur at distant moments across time.

Acknowledgments

We thank Jan Grau for valuable discussions and comments on the manuscript.

References

[1] Charu C. Aggarwal and Philip S. Yu. A framework for clustering massive text and categorical data streams. In *SDM*, 2006.

[2] L. AlSumait, D. Barbara, and C. Domeniconi. On-line LDA: adaptive topic models for mining text streams with applications to topic detection and tracking. In *ICDM*, 2008.

[3] A. Ascuncion, P. Smyth, and M. Welling. Asynchronous distributed learning of topic models. In *NIPS*, 2008.

[4] D. Blei and J. Lafferty. Dynamic topic models. In *ICML*, 2006.

[5] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[6] J.-T. Chien and M.-S. Wu. Adaptive bayesian latent semantic analysis. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(1):198–207, Jan. 2008.

[7] Tzu-Chuan Chou and Meng Chang Chen. Using incremental PLSI for threshold-resilient online event analysis. *IEEE Trans. on Knowl. and Data Eng.*, 20(3):289–299, 2008.

[8] T.L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101(suppl_1):5228–5235, 2004.

[9] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196, 2001.

[10] Geoffrey J. McLachlan and Thiriyambakam Krishnan. *EM Algorithm and Extensions*. Wiley, 1997.

[11] Qiaozhu Mei and ChengXiang Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *SIGKDD*, pages 198–207, New York, NY, USA, 2005. ACM.

[12] Satoshi Morinaga and Kenji Yamanishi. Tracking dynamics of topic trends using a finite mixture model. In *SIGKDD*, pages 811–816. ACM, 2004.

[13] Rene Schult. Comparing clustering algorithms and their influence on the evolution of labeled clusters. In *DEXA*, pages 650–659, 2007.

[14] Rene Schult and Myra Spiliopoulou. Discovering emerging topics in unlabelled text collections. In *ADBIS*, pages 353–366, 2006.

[15] Rene Schult and Myra Spiliopoulou. Expanding the taxonomies of bibliographic archives with persistent long-term themes. In *SAC*, pages 627–634, 2006.

[16] Xiaodan Song, Ching-Yung Lin, Belle L. Tseng, and Ming-Ting Sun. Modeling and predicting personal information dissemination behavior. In *SIGKDD*, pages 479–488. ACM, 2005.

[17] Myra Spiliopoulou, Irene Ntoutsi, Yannis Theodoridis, and Rene Schult. Monic: modeling and monitoring cluster transitions. In *SIGKDD*, pages 706–711. ACM, 2006.

[18] Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *SIGKDD*, pages 424–433. ACM, 2006.

[19] Max Welling, Chaitanya Chemudugunta, and Nathan Sutter. Deterministic latent variable models and their pitfalls. In *SDM*, 2008.

[20] Max Welling, Y.W. Teh, and B. Kappen. Hybrid variational-MCMC inference in bayesian networks. In *UAI 2008*, 2008.