Mining Common Topics from Multiple Asynchronous Text Streams *

Xiang Wang
School of Software, Tsinghua University
Beijing 100084, China
xiang_w00@mails.tsinghua.edu.cn

Kai Zhang
School of Software, Tsinghua University
Beijing 100084, China
z-k06@mails.tsinghua.edu.cn

Xiaoming Jin
School of Software, Tsinghua University
Beijing 100084, China
xmjin@tsinghua.edu.cn

Dou Shen
Microsoft Adcenter Labs
One Microsoft Way, Redmond, WA, USA
doushen@microsoft.com

ABSTRACT

Text streams are becoming more and more ubiquitous, in the forms of news feeds, weblog archives and so on, which result in a large volume of data. An effective way to explore the semantic as well as temporal information in text streams is topic mining, which can further facilitate other knowledge discovery procedures. In many applications, we are facing multiple text streams which are related to each other and share common topics. The correlation among these streams can provide more meaningful and comprehensive clues for topic mining than those from each individual stream. However, it is nontrivial to explore the correlation with the existence of *asynchronism* among multiple streams, i.e. documents from different streams about the same topic may have different timestamps, which remains unsolved in the context of topic mining. In this paper, we formally address this problem and put forward a novel algorithm based on the generative topic model. Our algorithm consists of two alternate steps: the first step extracts common topics from multiple streams based on the adjusted timestamps by the second step; the second step adjusts the timestamps of the documents according to the time distribution of the discovered topics by the first step. We perform these two steps alternately and a monotone convergence of our objective function is guaranteed. The effectiveness and advantage of our approach were justified by extensive empirical studies on two real data sets consisting of six research paper streams and two news article streams, respectively.

Categories and Subject Descriptors

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

General Terms

Algorithms

Keywords

Temporal text mining, topic model, asynchronous streams

1. INTRODUCTION

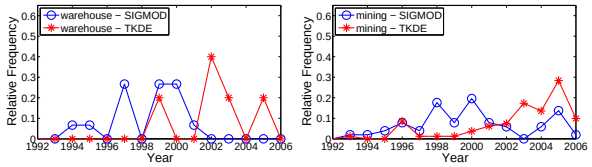
More and more text streams are being generated in various forms, such as news streams, weblog articles, emails, instant messages, research paper archives, web forum discussion threads, and so forth. To discover valuable knowledge from a text stream, a first step is usually to extract topics from the stream containing both semantic and temporal information, which are described by two distributions, respectively: a word distribution describing the semantics of the topic and a time distribution describing the topic's intensity over time [3, 5, 7, 8, 10, 11, 12, 14, 15].

In many real-world applications, we are facing multiple text streams that are correlated to each other by sharing common topics. Intuitively, the interactions among these streams could provide clues to derive more meaningful and comprehensive topics than topics found using information from each individual stream alone. The intuition was confirmed by very recent work [16], which utilized the temporal correlation over multiple streams to explore the semantic correlation among common topics. The method proposed therein relied on a critical assumption that different streams are always *synchronous* in time, or in their own term *coordinated*, which means that the common topics share the same time distribution over different streams.

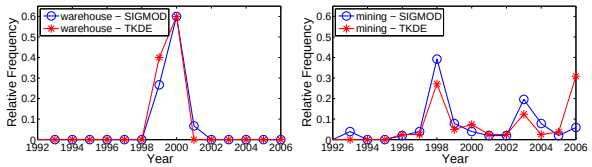
However, this assumption is too strong to hold in all cases. Rather, *asynchronism* among multiple streams, i.e. documents from different streams about the same topic have different timestamps, is actually very common in practice. For instance, in news streams, there is no guarantee that news articles covering the same topic are indexed by the same timestamps. There can be hours of delay for news agencies, days for newspapers, and even weeks for periodicals. This is because some news feeds try to provide first-hand flashes shortly after the incidents, while others provide more comprehensive reviews afterwards. Another example is research paper archives, where the latest research topics are closely followed by newsletters and communications within weeks

*The work was partly supported by NSFC 60403021, 60673140 and 863 funding 2007AA01Z156.

WSDM '09, February 9-12, 2009, Barcelona, Spain.



(a) Before synchronization



(b) After synchronization

Figure 1: An illustrative example of the asynchronism between two text streams and how it is eliminated by our method.

or months, then the extended versions may appear in conference proceedings, which are usually published annually, and at last in journals, which may sometimes take years to appear after submission. Specifically, let us see the relative frequency of the occurrences of two terms *warehouse* and *mining* respectively in the titles of all research papers published in SIGMOD (ACM International Conference on Management of Data) and TKDE (IEEE Transactions on Knowledge and Data Engineering) from 1992 to 2006. The first term identifies the topic *data warehouse* and the second *data mining*, which are two common topics shared by both streams. As shown in Fig. 1(a), the bursts of both terms in SIGMOD are significantly earlier than those in TKDE, which suggests the presence of asynchronism between these two streams. Thus, in this paper, we do not assume that given text streams are always synchronous. Instead, we deal with text streams that share common topics yet are temporally *asynchronous*.

We apparently expect multiple correlated streams to facilitate topic mining. However, the asynchronism among streams brings new challenges to conventional topic mining methods. As shown in Fig. 1(a), we may fail to discover the topic about *data mining* and/or *data warehouse* since they are relatively weak in each individual stream and the bursts in two streams do not coincide. On the other hand, as shown in Fig. 1(b), after adjusting the timestamps of documents in the two streams using our proposed method, the relative frequency of both *warehouse* and *mining* are boosted over a certain range of time, relatively. It proves that eliminating asynchronism can significantly benefit the topic discovery process. However, as desirable as it is for topic discovery to detect the temporal asynchronism among streams and eventually synchronize them, the task is difficult without knowing the topics to which the documents belong before hand. A naïve solution is to use coarse granularity of the timestamps of streams so that the asynchronism among streams can be smoothed out. This is obviously dissatisfactory as it may lead to unbearable loss in the temporal information of common topics and different topics would be inevitably mixed up. A second way, shifting or scaling the time dimension manually and empirically, may not work either because the time difference of topics among different streams can vary largely and irregularly, of which we can never have enough prior knowledge.

In this paper, we target the problem of mining common topics from multiple asynchronous text streams and propose an effective method to solve it. We formally define the problem by introducing a principled probabilistic framework, based on which a unified objective function can be derived. Then we put forward an algorithm to optimize this objective function by exploiting the mutual impact between topic discovery and time synchronization.

The key idea of our approach is to utilize the semantic and temporal correlation among streams and to build up a mutual reinforcement process. We start with extracting a set of common topics from given streams using their original timestamps. Based on the extracted topics and their word distributions, we update the timestamps of documents in all streams by assigning them to most relevant topics. This step reduces the asynchronism among streams. Then after synchronization, we refine the common topics according to the new timestamps. These two steps are repeated alternately to maximize a unified objective function, which provably converges monotonously.

Besides of theoretical justification, our method was also evaluated empirically on two real-world text streams. The first is a collection of 6 literature streams consisting of research papers on database technology from year 1975 to 2006 and the second contains 2 news streams of 61 days' news articles between April 1 and May 31, 2007. We show that our method is able to detect and eliminate the underlying asynchronism among different streams and effectively discover meaningful and highly discriminative common topics.

To sum up, the main contributions of our work are:

- We address the problem of mining common topics from multiple *asynchronous* text streams. To the extent of our knowledge, this is the first attempt to solve this problem.
- We formalize our problem by introducing a principled probabilistic framework and propose an objective function for our problem.
- We develop a novel alternate optimization algorithm to solve the objective function with a theoretically guaranteed (local) optimum.
- The effectiveness and advantage of our method are validated by extensive empirical study on two real-world data sets.

The rest of the paper is organized as follows: related work is briefly discussed in Section 2; we formalize our problem and propose a generative model with a unified objective function in Section 3; we show how to optimize the objective function in Section 4; empirical results are presented in Section 5; we conclude our work in Section 6.

2. RELATED WORK

Topic mining has been extensively studied in the literature, starting with the Topic Detection and Tracking (TDT) project [1, 17], which aimed to find and track topics (events) in news streams with clustering based techniques. Later on probabilistic generative models were introduced into use, such as Probabilistic Latent Semantic Analysis (PLSA) [6], Latent Dirichlet Allocation (LDA) [4] and their derivatives [2, 9, 13].

Table 1: Symbols and their meanings

Symbols	Description
$\mathbf{d}$	document
$\mathbf{t}$	timestamp
$\mathbf{w}$	word
$\mathbf{z}$	topic
M	number of streams
T	number of different timestamps
V	number of different words
K	number of topics

In many real applications, text collections carry generic temporal information and thus can be considered as text streams. To capture the temporal dynamics of topics, various methods have been proposed to discover topics over time in text streams [3, 5, 7, 8, 10, 11, 12, 14, 15]. However, these methods were designed to extract topics from a *single* stream. For example, in [10, 15], which adopted the generative model, timestamps of individual documents were modeled with a random variable, either discrete or continuous. Then it was assumed that given a document in the stream, the timestamp of the document was generated conditionally independently from word. In [3], the authors introduced hyper-parameters that evolve over time in state transfer models in the stream. For each time slice, a hyper-parameter is assigned with a state by a probability distribution, given the state on the former time slice. In [12], the time dimension of the stream was cut into time slices and topics were discovered from documents in each slice independently. As a result, in multiple-stream cases, topics in each stream can only be estimated separately and potential correlation between topics in different streams, both semantically and temporally, could not be fully explored. In [2, 9, 13], the semantic correlation between different topics in static text collections was considered. Similarly, [18] explored common topics in multiple static text collections.

A very recent work by Wang et al. [16] firstly proposed a topic mining method that aimed to discover common (bursty) topics over multiple text streams. Their approach is different from ours because they tried to find topics that shared common *time* distribution over different streams by assuming that the streams were synchronous, or *coordinated*. Based on this premise, documents with same timestamps are combined together over different streams so that the word distributions of topics in individual streams can be discovered. As a contrast, in our work, we aim to find topics that are common in semantics, while having asynchronous time distributions in different streams.

3. PROBLEM AND OBJECTIVE FUNCTION

In this section, we formally define our problem of mining common topics from multiple asynchronous text streams. We introduce a generative topic model which incorporates both temporal and semantic information in given text streams. We derive our objective function, which is to maximize the likelihood estimation subject to certain constraints. The main symbols used throughout the paper are listed in Table 1.

First of all, we define text stream as follows:

DEFINITION 1 (TEXT STREAM). *A text stream $\mathcal{S}$ is a sequence of N documents $(d_1, \ldots, d_N)$. Each document d is a collection of words over vocabulary $\mathcal{V}$ and indexed by a unique timestamp $t \in \{1, \ldots, T\}$.*

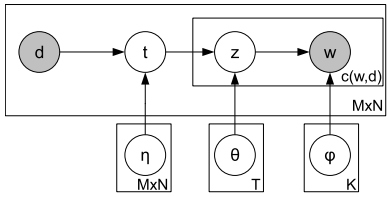


Figure 2: An illustration of our generative model. Shaded nodes mean observable variables while white nodes mean unobservable variables. Arrow indicates the generation relationship.

Note that in our definition, we allow multiple documents in the same stream to share a common timestamp, which is usually the case in real applications.

Given M text streams, we aim to extract K common topics from them (K is given by users), which are defined as:

DEFINITION 2 (COMMON TOPIC). *A common topic $\mathcal{Z}$ over text streams is defined by a word distribution over vocabulary $\mathcal{V}$ and a time distribution over timestamps $\{1, \ldots, T\}$.*

To find common topics $\{\mathcal{Z}_k : 1 \leq k \leq K\}$ over text streams $\{\mathcal{S}_m : 1 \leq m \leq M\}$, we put forward a novel generative model, derived from the topic model family that has been widely-used in topic mining tasks. Our generative model is able to capture the interaction between temporal and semantic information of topics and this interaction as shown later can be used to extract common topics from asynchronous streams with an alternate optimization process.

The documents $\{d \in \mathcal{S}_m : 1 \leq m \leq M\}$ are modeled by a discrete random variable $\mathbf{d}$. The words are modeled by a discrete random variable $\mathbf{w}$ over vocabulary $\mathcal{V}$. The timestamps are modeled by a discrete random variable $\mathbf{t}$ over $\{1, \ldots, T\}$. At last the common topics $\mathcal{Z}$ are encoded by a discrete random variable $\mathbf{z} \in \{1, 2, \ldots, K\}$. Note that semantic information of a topic is encoded by the conditional distribution $p(\mathbf{w}|\mathbf{z})$ and its temporal information by $p(\mathbf{z}|\mathbf{t})$.

The generating process is as follows (also see Fig. 2):

1. Pick a document $\mathbf{d}$ with probability $p(\mathbf{d})$.
2. Given the document $\mathbf{d}$, pick a timestamp $\mathbf{t}$ with probability $p(\mathbf{t}|\mathbf{d}) \sim \text{Mult}(\eta, \{0, 1\})$, which is a multinomial distribution with parameter η and the value of $p(\mathbf{t}|\mathbf{d})$ is either 0 or 1. It means that a given document has and only has one timestamp.
3. Given the timestamp $\mathbf{t}$, pick a common topic $\mathbf{z}$ with probability $p(\mathbf{z}|\mathbf{t}) \sim \text{Mult}(\theta)$.
4. Given the topic $\mathbf{z}$, pick a word $\mathbf{w}$ with probability $p(\mathbf{w}|\mathbf{z}) \sim \text{Mult}(\varphi)$.

According to the generative process, the probability of word $\mathbf{w}$ in document $\mathbf{d}$ is

$$p(\mathbf{w}, \mathbf{d}) = \sum_{\mathbf{t},\mathbf{z}} p(\mathbf{d}) p(\mathbf{t}|\mathbf{d}) p(\mathbf{z}|\mathbf{t}) p(\mathbf{w}|\mathbf{z}).$$

Consequentially the log-likelihood function over all streams writes:

$$\mathcal{L} = \sum_{\mathbf{w}} \sum_{\mathbf{d}} c(\mathbf{w}, \mathbf{d}) \log p(\mathbf{w}, \mathbf{d}),$$

where $c(\mathbf{w}, \mathbf{d})$ is the number of occurrences of word $\mathbf{w}$ in document $\mathbf{d}$.

Conventional methods on topic mining try to maximize the likelihood function $\mathcal{L}$ by adjusting $p(\mathbf{z}|\mathbf{t})$ and $p(\mathbf{w}|\mathbf{z})$ while assuming $p(\mathbf{t}|\mathbf{d})$ is known. However, in our work, we need to consider the potential asynchronism among different streams, i.e., $p(\mathbf{t}|\mathbf{d})$ is also to be determined. Thus besides of finding optimal $p(\mathbf{z}|\mathbf{t})$ and $p(\mathbf{w}|\mathbf{z})$, we also need to decide $p(\mathbf{t}|\mathbf{d})$ to further maximize $\mathcal{L}$. In other words, we want to assign the document with timestamp t to a new timestamp $g(t)$ by determining its relevance to respective topics, so that we can obtain larger $\mathcal{L}$, or equivalently, topics with better quality.

Note that the mapping from t to $g(t)$ is not arbitrary. By the term *asynchronism*, we refer to the time distortion among different streams. The relative temporal order within each individual stream is still considered meaningful and generally correct (otherwise the current temporal information in the streams will be discarded and the problem would reduce to mining topics from a collection of texts, not text streams). Therefore, during each synchronization step, we preserve the relative temporal order of documents in each individual streams, i.e., a document with earlier timestamp before adjustment will always be assigned to earlier timestamp after adjustment as compared to its successors. This constraint aims to protect local temporal information within each individual stream while fixing the asynchronism among different streams. Formally, given two documents d_1 and d_2 in a same stream, we require that:

$$g(t_1) \leq g(t_2) \text{ iff } t_1 \leq t_2.$$

In sum we have:

DEFINITION 3 (ASYNCHRONISM). *Given M text streams $\{\mathcal{S}_m : 1 \leq m \leq M\}$, in which documents are indexed by timestamps $\{t : 1 \leq t \leq T\}$, asynchronism means that the timestamps of the documents sharing the same topic in* different *streams are not properly aligned. However, it does not involve the relative temporal order between documents within the* same *stream.*

Finally, our objective is to maximize the likelihood function $\mathcal{L}$ by adjusting $p(\mathbf{z}|\mathbf{t})$ and $p(\mathbf{w}|\mathbf{z})$ as well as $p(\mathbf{t}|\mathbf{d})$ subject to the constraint of preserving temporal order within stream. Formally it writes:

$$\begin{aligned} &\text{argmax}_{p(\mathbf{t}|\mathbf{d}),p(\mathbf{z}|\mathbf{t}),p(\mathbf{w}|\mathbf{z})}\mathcal{L}, \\ &\text{s.t. } \forall d_1, d_2 \in \mathcal{S}_m, g(t_1) \leq g(t_2) \text{ iff } t_1 \leq t_2, \end{aligned} \tag{1}$$

for $1 \leq m \leq M$, where t_1 and t_2 are the current timestamps of d_1 and d_2, respectively and $g(t_1)$ and $g(t_2)$ are the timestamps after adjustment.

4. ALGORITHM

In this section we show how to solve our objective function in Eq.(1) through an alternate (constrained) optimization scheme. The outline of our algorithm is:

Step 1 We assume the current timestamps of streams are synchronous and extract common topics from them.

Step 2 We synchronize the timestamps of all documents by matching them to most related topics respectively. Then we go back to Step 1 until convergence.

4.1 Topic Extraction

First we assume the current timestamps of all streams are already synchronous and extract common topics from them. In other words, now $p(\mathbf{t}|\mathbf{d})$ is fixed and we try to maximize the likelihood function by adjusting $p(\mathbf{t}|\mathbf{z})$ and $p(\mathbf{w}|\mathbf{z})$. Thus we can rewrite the likelihood function as follows:

$$\begin{aligned} &\sum_{\mathbf{w}}\sum_{\mathbf{d}} c(\mathbf{w}, \mathbf{d}) \log \sum_{\mathbf{t}}\sum_{\mathbf{z}} p(\mathbf{d})p(\mathbf{t}|\mathbf{d})p(\mathbf{z}|\mathbf{t})p(\mathbf{w}|\mathbf{z}) \\ =&\sum_{\mathbf{w}}\sum_{\mathbf{d}} c(\mathbf{w}, \mathbf{d}) \log p(\mathbf{d}) \sum_{\mathbf{t}} p(\mathbf{t}|\mathbf{d}) \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{t})p(\mathbf{w}|\mathbf{z}). \end{aligned}$$

Since $p(\mathbf{t}|\mathbf{d}) \sim \text{Mult}(\eta, \{0, 1\})$, above equation can be reduced to

$$\begin{aligned} &\sum_{\mathbf{w}}\sum_{\mathbf{d}}\sum_{\mathbf{t}} c(\mathbf{w}, \mathbf{d}, \mathbf{t}) \log \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{t})p(\mathbf{w}|\mathbf{z}) \\ =&\sum_{\mathbf{w}}\sum_{\mathbf{t}} c(\mathbf{w}, \mathbf{t}) \log \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{t})p(\mathbf{w}|\mathbf{z}). \end{aligned} \tag{2}$$

Here $c(w, d, t)$ denotes the number of occurrences of word w in document d at time t, and $p(d)$ is summed out because it can be considered as a constant in the formula [6].

Eq.(2) can be solved by well-established EM algorithm [6]. The E-step writes:

$$p(\mathbf{z}|\mathbf{w}, \mathbf{t}) = \frac{p(\mathbf{z}|\mathbf{t})p(\mathbf{w}|\mathbf{z})}{\sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{t})p(\mathbf{w}|\mathbf{z})}, \tag{3}$$

and the M-step writes:

$$\begin{aligned} p(\mathbf{z}|\mathbf{t}) &= \frac{\sum_{\mathbf{w}} c(\mathbf{w}, \mathbf{t})p(\mathbf{z}|\mathbf{w}, \mathbf{t})}{\sum_{\mathbf{z}}\sum_{\mathbf{w}} c(\mathbf{w}, \mathbf{t})p(\mathbf{z}|\mathbf{w}, \mathbf{t})}, \\ p(\mathbf{w}|\mathbf{z}) &= \frac{\sum_{\mathbf{t}} c(\mathbf{w}, \mathbf{t})p(\mathbf{z}|\mathbf{w}, \mathbf{t})}{\sum_{\mathbf{w}}\sum_{\mathbf{t}} c(\mathbf{w}, \mathbf{t})p(\mathbf{z}|\mathbf{w}, \mathbf{t})}. \end{aligned} \tag{4}$$

The E- and M-step repeat alternately and our objective function will converge to a local optimum after finite rounds.

4.2 Time Synchronization

Once the common topics are extracted, we match documents in all streams to these topics and adjust their timestamps to synchronize the streams.

Specifically, now $p(\mathbf{z}|\mathbf{t})$ and $p(\mathbf{w}|\mathbf{z})$ are assumed as known and we try to adjust $p(\mathbf{t}|\mathbf{d})$ to maximize our objective function. Given document d, we denote its current timestamp with t and its timestamp after adjustment with $g(t)$. Then our objective function in Eq.(1) can be rewritten as:

$$\begin{aligned} &\text{argmax}_{g(t)} \sum_{m=1}^{M}\sum_{\mathbf{w}}\sum_{s=1}^{T} \mathcal{Q}(\mathbf{w}, s) \sum_{\{d\in\mathcal{S}_m : g(t)=s\}} c(\mathbf{w}, d) \\ &\text{s.t. } \forall d_1, d_2 \in \mathcal{S}_m, g(t_1) \leq g(t_2) \text{ iff } t_1 \leq t_2, \end{aligned} \tag{5}$$

where $\mathcal{Q}(\mathbf{w}, s) = \log \sum_{\mathbf{z}} p(\mathbf{z}|s)p(\mathbf{w}|\mathbf{z})$. It is obvious that we can solve Eq.(5) by solving the following objective function for each stream respectively:

$$\begin{aligned} &\max_{g(t)} \sum_{\mathbf{w}}\sum_{s=1}^{T} \mathcal{Q}(\mathbf{w}, s) \sum_{\{d:g(t)=s\}} c(\mathbf{w}, d), \\ &\text{s.t. } \forall d_1, d_2, g(t_1) \leq g(t_2) \text{ iff } t_1 \leq t_2. \end{aligned} \tag{6}$$

And $p(\mathbf{t}|\mathbf{d})$ can be decided by $p(\mathbf{t} = g(t)|d) = 1$ and $p(\mathbf{t} \neq g(t)|d) = 0$.

Next we define following function:

$$H(1:i,1:j) = \max_{g(t)} \sum_{\mathbf{w}} \sum_{s=1}^{j} \mathcal{Q}(\mathbf{w},s) \sum_{r=1}^{i} \sum_{d(r,s)} c(\mathbf{w},d),$$

where $1 \le i,j \le T$. Here $d(r,s)$ denotes the set of all documents whose timestamps are changed from r to s, i.e., $\{d : t = r, g(t) = s\}$. It is easy to see that our objective function in Eq.(6) equals to $H(1:T,1:T)$.

Then we show how to compute $H(1:T,1:T)$ recursively. The basic idea behind our approach is that: suppose we already have j timestamps $\{1,\ldots,j\}$ and documents whose current timestamps are ranging from 1 to $i-1$, i.e., $\{d : 1 \le t \le i-1\}$; then given documents whose current timestamps are i, according to our constraint, its new timestamp $g(i)$ must be no smaller than the new timestamps of documents in $\{d : 1 \le t \le i-1\}$. Thus if the smallest timestamp of documents in $\{d : t = i\}$ is a, then documents in $\{d : 1 \le t \le i-1\}$ can only match to timestamps from 1 to a. So we can enumerate all possible matching for $1 \le a \le j$ to find an optimal a for $H(1:i,1:j)$. Formally, we have

$$\begin{aligned}
&H(1:T;1:T)\\
&= \max_{g(t)} \sum_{\mathbf{w}} \sum_{s=1}^{T} \mathcal{Q}(\mathbf{w},s) \left(\sum_{r=1}^{T-1} \sum_{d(r,s)} c(\mathbf{w},d) + \sum_{d(T,s)} c(\mathbf{w},d) \right)\\
&= \max_{1\le a\le T} \max_{g(t)}\\
&\sum_{\mathbf{w}} \left(\sum_{s=1}^{a} \mathcal{Q}(\mathbf{w},s) \sum_{r=1}^{T-1} \sum_{d(r,s)} c(\mathbf{w},d) + \sum_{s=a}^{T} \mathcal{Q}(\mathbf{w},s) \sum_{d(T,s)} c(\mathbf{w},d) \right)\\
&= \max_{1\le a\le T} \left(H(1:(T-1);1:a) + \delta(T;a:T) \right),
\end{aligned}$$

where the second term equals to

$$\delta(r;a:T) = \sum_{\{d:t=r\}} \max_{a\le s\le T} \sum_{\mathbf{w}} \mathcal{Q}(\mathbf{w},s) c(\mathbf{w},d),$$

for $1 \le r \le T$, and the first term can be computed recursively as

$$H(1:i,1:j) = \max_{1\le a\le j} \left(H(1:(i-1);1:a) + \delta(i;a:j) \right) \quad (7)$$

for $2 \le i \le T$ and $1 \le j \le T$. Specially we have

$$H(1:1,1:a) = \sum_{\{d:t=1\}} \max_{1\le s\le a} \sum_{\mathbf{w}} \mathcal{Q}(\mathbf{w},s) c(\mathbf{w},d)$$

for $1 \le a \le T$. After $H(1:T,1:T)$ is computed recursively, it gives the global optimum to our objective function in Eq.(6).

Our algorithm is summarized in Algorithm 1. K is the number of topics and specified by users. The initial values of $p(\mathbf{t}|\mathbf{d})$ and $c(\mathbf{w},\mathbf{d},\mathbf{t})$ are counted from the original timestamps in the streams.

The computational complexity of the topic extraction step (with EM algorithm) is $\mathcal{O}(KVT)$ while the complexity of time synchronization step is approximately $\mathcal{O}(VMT^3)$. Thus the overall complexity of our algorithm is $\mathcal{O}(VT(K+MT^2))$, where V is the size of vocabulary, T the number of different timestamps, K the number of topics and M the number of streams. If we take V, K and M as constants and only consider the length of stream, which is T, the complexity of Algorithm 1 becomes $\mathcal{O}(T^3)$. We will show in next section how to reduce it to $\mathcal{O}(T^2)$ with a local search strategy.

Algorithm 1: Topic mining with time synchronization

```
Input: K, p(t|d), c(w,d,t);
Output: p(w|z), p(z|t), p(t|d);
repeat
    Update c(w,t) with p(t|d) and c(w,d,t);
    Initialize p(z|t) and p(w|z) with random values;
    repeat
        Update p(z|t) and p(w|z) following Eq.(3) and (4);
    until Convergence ;
    for m=1 to M do
        for j=1 to T do Initialize H(1:1,1:j);
        for i=2 to T do
            for j=1 to T do
                Compute H(1:i,1:j) as shown in Eq.(7);
            end
        end
        Update p(t|d);
    end
until Convergence ;
```

4.3 Remarks

Constraint on Time Synchronization. During *each* synchronization step, the constraint in Eq.(6) requires that a document with an earlier timestamp can only be assigned to an earlier timestamp, as compared to its successors in the same stream. At the first glance, this may seem too strict because the original temporal order of given text streams cannot be perfect. However, the constraint in our algorithm is much more tolerant than it appears to be. Specifically, after several iterations, it is possible that two adjacent documents swap their positions along the time dimension. For instance, suppose we have document d_1 with timestamp 3 and d_2 with timestamp 5. After the first round of synchronization, both d_1 and d_2 are mapped to time 4. Now we use 4 as input value for d_1 and d_2, thus in the following round, it is possible that d_2 would be assigned to an earlier timestamp than d_1, without violating our constraint. As we will show later in the experimental results, in practice, documents tend to find new timestamps in the neighborhoods of their original positions and local swapping of documents' positions often happens, which can empirically justify the flexibility and robustness of our method.

Convergence. Both of the two steps in our algorithm guarantee a monotone improvement in our objective function in Eq.(1), the algorithm will converge to a local optimum after finite numbers of iterations. Note that there is a trivial solution to the objective function, which is to assign all documents to a single (arbitrary) timestamp and our algorithm would terminate at this local optimum. This local optimum is apparently meaningless since it is equivalent to discard all temporal information of text streams and treat them like a collection of documents. Nevertheless, this trivial solution only exists theoretically. In practice, our algorithm will not converge to this trivial solution, as long as we use the original timestamps of text streams as initial value and have $K > 1$, where K is the number of topics. As shown in Section 5, the adjusted timestamps of documents always converge to more than K different time points.

The Local Search Strategy. In some real-world appli-

cations, we can have a quantitative estimation of the asynchronism among streams so it is unnecessary to search the entire time dimension when adjusting the timestamps of documents. This gives us the opportunity to reduce the complexity of time synchronization step without causing substantial performance loss, by setting a upper bound for the difference between the timestamps of documents before and after adjustment. Specifically, given document d with time t, we now look for an optimal $g(t)$ within the ϵ-neighborhood of t, where ϵ is the user-specified search range. Accordingly, Eq.(6) becomes:

$$\max_{g(t)} \sum_{\mathbf{w}} \sum_{s=1}^{T} \mathcal{Q}(\mathbf{w}, s) \sum_{\{d:g(t)=s\}} c(\mathbf{w}, d),$$

$$\text{s.t. } \forall d,\ g(t) \in [t-\epsilon, t+\epsilon] \wedge \forall d_1, d_2,\ g(t_1) \leq g(t_2) \text{ iff } t_1 \leq t_2.$$

This objective function can be solved by Eq.(7) with slight modification, which we do not show in detail here due to limited space. We can see that the complexity of the synchronization step has been reduced to $\mathcal{O}(\epsilon V M T^2)$, thus the overall complexity is reduced from $\mathcal{O}(T^3)$ to $\mathcal{O}(T^2)$.

5. EMPIRICAL EVALUATION

We evaluated our method on two sets of real-world text streams, a set of 6 research paper streams and a set of 2 news article streams. The goal is to see if our method is able to:

1. Explore the underlying asynchronism among text streams and fix it with our time synchronization techniques;
2. Find meaningful and discriminative common topics from multiple text streams;
3. Consistently outperform the baseline method (without time synchronization).

5.1 Data Sets

The first data set used in our experiment is six research paper collections extracted from DBLP[1], namely DEXA, ICDE, Information Systems (journal), SIGMOD, TKDE (journal) and VLDB. All of these collections mainly consist of research papers on database technology. Each collection is considered as a single text stream where each document is represented by the title of the paper and indexed by its publication year. The second data set is two news articles streams, which consist of the full texts of daily news reports published on the web sites of International Herald Tribune[2] and People's Daily Online[3] respectively from April 1, 2007 to May 31, 2007. Each document is indexed by its publication date.

Text streams are preprocessed by stemming and removing stop words. Words that appear too many or too few times are also removed. After preprocessing, the literature streams have a vocabulary of 1686 words and news streams 3358 words. The basic statistics of the data sets after preprocessing is shown in Table 2 and 3.

[1] http://www.informatik.uni-trier.de/~ley/db/
[2] http://www.iht.com/
[3] http://english.peopledaily.com.cn/

Table 2: Statistics of the literature streams

ID	Year	#Docs	#Words/doc
DEXA	1990 - 2006	1477	6.03
ICDE	1984 - 2006	1957	5.90
IS	1975 - 2006	939	5.93
SIGMOD	1975 - 2006	1877	5.40
TKDE	1989 - 2006	1457	6.29
VLDB	1975 - 2006	2329	5.67

Table 3: Statistics of the news streams

ID	#Days	#Docs	#Words/doc
IHT	61	2488	271.9
People	61	6461	65.8

5.2 The Baseline Method and Implementation

For the simplicity of description, in Section 4, we use standard PLSA [6] method as the topic extraction step of our algorithm. Yet in the experiments, we introduced two additional techniques as used by [12, 16] and this modified version of PLSA algorithm was used as a baseline method for topic extraction.

The first technique is to introduce a background topic $p(\mathbf{w}|B)$ into our generative model so that background noise can be removed and we can find more bursty and meaningful topics. Specifically, the objective function in Eq.(2) is rewritten as

$$\sum_{\mathbf{w}} \sum_{\mathbf{t}} c(\mathbf{w}, \mathbf{t}) \log \left(\lambda_B p(\mathbf{w}|B) + (1-\lambda_B) \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{t}) p(\mathbf{w}|\mathbf{z}) \right),$$

where $p(\mathbf{w}|B) = \sum_{\mathbf{t}} c(\mathbf{w}, \mathbf{t}) / \sum_{\mathbf{w}} \sum_{\mathbf{t}} c(\mathbf{w}, \mathbf{t})$ is a background topic whose distribution is independent from time and $\lambda_B \in (0, 1)$ is a weighting parameter that decides the strength of the background topic. Empirically we have $\lambda_B \in [0.9, 0.95]$, as suggested by previous work [12, 16]. In our experiments, we empirically had $\lambda_B = 0.9$ for literature streams and $\lambda_B = 0.95$ for news streams, according to their respective characteristics.

The second technique is to impose time dependency on $p(\mathbf{z}|\mathbf{t})$ by smoothing the time distribution of topic between adjacent timestamps, which writes:

$$p(\mathbf{z}|t) \leftarrow \frac{\mu p(\mathbf{z}|t-1)}{2(1+\mu)} + \frac{p(\mathbf{z}|t)}{1+\mu} + \frac{\mu p(\mathbf{z}|t+1)}{2(1+\mu)},$$

where μ is a smoothing factor. In our experiment, we empirically chose $\mu = 0.1$, following [16]. Note that the introduction of background topic and smoothing factor does not affect the time synchronization step of our algorithm.

In sum, we implemented two different methods in our experiments, one was the baseline method described above (labeled as **no_sync**) and the other was our method with time synchronization (labeled as **sync**).

5.3 Evaluation Metrics

We evaluated the performance of our method using several different metrics.

Recall that in order to optimize our objective function, as shown in Eq.(1), we have three parameters to estimate, namely $p(\mathbf{t}|\mathbf{d})$, $p(\mathbf{z}|\mathbf{t})$ and $p(\mathbf{w}|\mathbf{z})$. Here $p(\mathbf{t}|\mathbf{d})$ gives the new timestamps of documents after adjustment, $p(\mathbf{z}|\mathbf{t})$ indicates the time distribution of extracted topics while $p(\mathbf{w}|\mathbf{z})$ gives

the word distribution. These parameters were all examined in our experiments. Specifically:

- For $p(\mathbf{w}|\mathbf{z})$, we evaluate the meaningfulness of extracted topics by examining their top-ranked topical words. We also compute the pairwise KL-divergence between topics to evaluate how discriminative they are. (In practice, we normally expect meaningful topics that can be easily understood by human users and we want these topics to be as discriminative as possible, in order to avoid redundant information.)
- For $p(\mathbf{z}|\mathbf{t})$, we want to see if, after synchronization, our method is able to separate different topics along the time dimension, which would eventually improve the quality of extracted topics.
- For $p(\mathbf{t}|\mathbf{d})$, we demonstrate how our method adjusts documents' timestamps and fixes the synchronization among given text streams.

We also computed the log-likelihood of our method and compared it to that of the baseline method.

In order to show the stability of our method against random initialization, we repeated our method for 100 times and compared it to the baseline method under two different metrics: log-likelihood and pairwise KL-divergence between the words distributions of different topics.

5.4 Results and Analysis

5.4.1 Literature Streams

First we performed our method as well as the baseline method on the literature streams data set. We extracted 10 common topics from the streams. For each topic, 10 topical words with highest probability ($p(\mathbf{w}|\mathbf{z})$) were shown in Fig. 3 and 4. We can see that all topics extracted by our method (sync) were meaningful and easy to understand. For example, #7 includes research topics like *data mining*, *high-dimensional/multidimensional data*, *data warehouse*, *association rule*, *workflow*, etc., while #10 includes *sensor network*, *privacy preserving*, *classification*, *ontology*, *top-k query*, etc. All of these topical words accurately suggest most important research topics in the database area. Comparing the topics extracted by our method to those by the baseline method (no_sync), we can see that our method provided highly discriminative topics. As a contrast, the baseline method suffered from the asynchronism in the streams and extracted many duplicated topical words (see Fig. 4). In asynchronous streams, documents related to different topics may be indexed by the same timestamp, and documents related to the same topic may appear at different timestamps. As a result, common topics discovered by conventional method contain redundant information, whereas our method is able to fix the asynchronism and discover highly discriminative topics.

To further prove that our time synchronization technique helped to generate more discriminative topics, we computed the pairwise KL-divergence between topics as follows:

$$KL(z_1, z_2) = \sum_{\mathbf{w}} p(w|z_1) \log \frac{p(w|z_1)}{p(w|z_2)}.$$

Note that larger KL-divergence indicates the two topics are more discriminative to each other and 0 divergence means two topics are identical. We present the results in Fig. 5, where darker blocks mean smaller KL-divergence values. We can see that our method extracted much more discriminative topics than those extracted by the baseline method. As discussed above, this was due to the fact that our method successfully fixed the asynchronism in the data set.

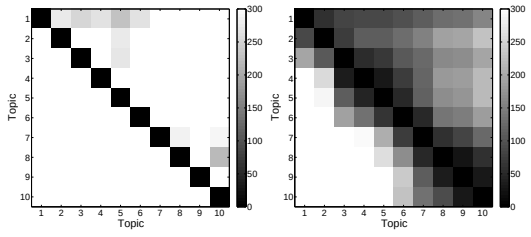
The time distribution of extracted topics is shown in Fig. 6. We can see that without synchronization, the extracted topics overlapped significantly over time (Fig. 6(b)), while our method substantially reduced the overlapping area between

	Top-10 topical words (sorted by probability)
1.	file data language abstract relational program model base access user
2.	design schema theory conceptual methodology CODASYL specific paper tool practice
3.	distribute concurrency control relational hash performance extend recursive evaluation depend
4.	knowledge expert transaction transit replicate closure protocol product intelligence hypertext
5.	object orient deductive parallel database multi-database language model buffer persistent
6.	active server multimedia heterogenous time real constraint architecture maintain federal
7.	mining spatial warehouse association dimension workflow high business scalable video
8.	web search similarity cache service sequence multi-dimensional mobile nearest extract
9.	XML stream peer pattern document continuous adaptive approximate XQuery move
10.	network privacy sensor preserve match XPath ranking classification ontology top-K

Figure 3: Common topics extracted by our method (sync) from literature streams ($K = 10$).

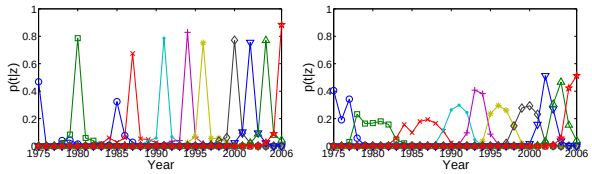
	Top-10 topical words (sorted by probability)
1.	data base file abstract relational language level large conversation structural
2.	base design data theory paper relational CODASYL practice methodology language
3.	database relational design distribute file recursive hash concurrency control extend
4.	object knowledge orient system expert transaction transit parallel hypertext deductive
5.	object orient parallel knowledge database deductive multi-database system expert language
6.	object rule active orient server parallel heterogenous database multimedia transaction
7.	mining web warehouse multimedia spatial index workflow scalable dimension high
8.	XML cache web efficiency service similarity search mobile mining association
9.	XML stream web peer mining service XQuery P2P adaptive pattern
10.	XML network stream efficiency privacy pattern peer classification web clustering

Figure 4: Common topics extracted by the baseline method (no_sync) from literature streams ($K = 10$). Some of the duplicated topical words are underlined.



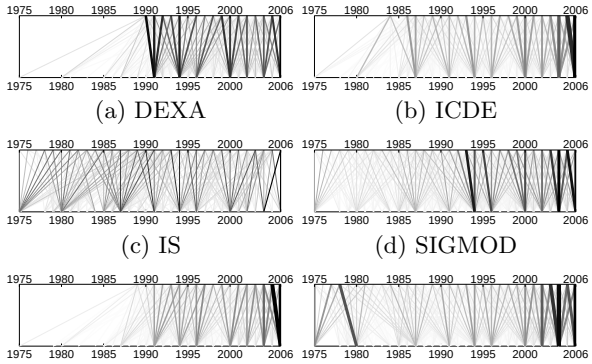
(a) sync (b) no_sync

Figure 5: The pairwise KL-divergence between topics extracted from the literature streams ($K = 10$).



(a) sync (b) no_sync

Figure 6: The time distribution of topics extracted from the literature streams ($K = 10$).



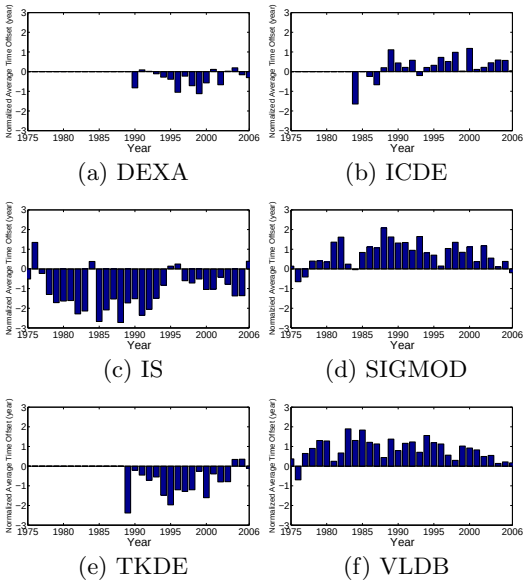
(a) DEXA (b) ICDE (c) IS (d) SIGMOD (e) TKDE (f) VLDB

Figure 7: The mapping from documents' original timestamps (upper axis) to those determined by our method (lower axis) in literature streams. The boldness of lines indicates the number of documents belonging to that mapping.

topics by fixing the asynchronism (Fig. 6(a)). This explains why our method was able to find more discriminative topics. We further provide a detailed view of how our method adjusted the timestamps of documents. Fig. 7 shows the mapping from documents' original timestamps to the ones assigned by our method (sync). We can see that our synchronization technique on one hand preserved the temporal order in original text streams, and on the other hand, it discovered temporally adjacent documents belonging to the same topic and assigned them to same timestamps.

Moreover, for documents indexed by each timestamp, we computed the difference between their original timestamps and final timestamps after synchronization ($g(t) - t$). The offsets was then normalized so that they added up to 0 at each timestamp. At last the average offset for each timestamp was shown in Fig. 8. Note that positive time offset means that most documents at this timestamp were assigned to a later timestamp after synchronization. In other words, documents with positive time offset addressed common topics earlier than documents with negative time offset. In Fig. 8 we can see that papers from ICDE, SIGMOD and VLDB had positive time offsets at most timestamps while papers from IS and TKDE mostly had negative time offsets. This means that common topics were addressed earlier in ICDE, SIGMOD and VLDB than IS and TKDE, which conforms to our knowledge that latest research results in this area normally first appear in conference proceedings years before they appear in journals.

At last we studied the robustness and stability of our method against random initialization and parameter K (the

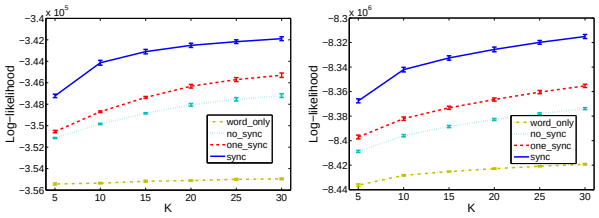


(a) DEXA (b) ICDE (c) IS (d) SIGMOD (e) TKDE (f) VLDB

Figure 8: Normalized average time offset of papers at each year. Positive offset indicates that most papers in the corresponding year were assigned to a later timestamp, which means that they addressed common topics earlier than those papers with negative offset.

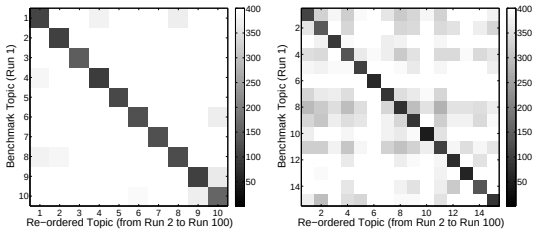
number of topics). Fig. 9(a) is the log-likelihood curves of our method (sync) and the baseline method (no_sync). We used different K ranging from 5 to 30, and for each K, we ran all methods for 100 times with random initialization. The log-likelihood was defined as Eq.(1). For the baseline method, we simply used the original timestamps of documents. We can see in Fig. 9(a) that our method (sync) consistently outperformed the baseline method (no_sync) by a large margin. In addition, we show that our method outperformed the one_sync method, which is the one-time synchronization version of our method. This on the other hand verified the improvement in objective function due to iterations of synchronization step. We also introduced the word_only method, which discards all the temporal information and handles given streams as a static collection of documents. It performed the worst in terms of likelihood and this suggests that temporal information can indeed facilitate the topic mining procedure.

We also examined semantically the stability of topics extracted by our method against random initialization. Specifically, we chose the topics extracted with $K = 10$ and 100 rounds of random initialization. The 10 topics from Run 1 was chosen as benchmark, and topics from other 99 rounds were re-ordered to match topics from Run 1 using a greedy algorithm, i.e., we matched a given topic to its most similar topic in Run 1, with similarity function defined by KL-divergence. Thus, we obtained 99 similarity matrices constructed by the KL-divergence values between re-ordered topics and benchmark topics. Then we averaged the 99 similarity matrices into one matrix. We repeated above process 100 times so that every run was chosen once as the benchmark run. The average KL-divergence is shown in Fig. 10(a). This matrix suggests that a large percentage of topics have similar word distributions over different rounds of random



(a) Literature ($K = 10$) (b) News ($K = 15$)

Figure 9: The log-likelihood curves of our method and the baseline method, with different K and 100 rounds of random initialization.



(a) Literature ($K = 10$) (b) News ($K = 15$)

Figure 10: The average pairwise KL-divergence between topics extracted by our method (sync) over 100 rounds of random initialization.

initialization. In other words, the topics extracted by our method are stable in semantics.

5.4.2 News Streams

Now we present the performance of our method on the news streams data set. We extracted 15 common topics ($K = 15$) from two news streams consisting of 61 days' news reports with full texts. Note that in consideration of efficiency, here we used the local search strategy for time synchronization, as described in Section 4. The local search radius was set to be 3, as we assumed that time difference between (online) news articles belonging to the same topic normally will not exceed 3 days. The topic extraction step remained the same.

We list in Fig. 11 the topical words of all 15 common topics extracted by our method (sync) and those by baseline method (no_sync) in Fig. 12. Comparing these two sets of results, we can see that both methods discovered some common topics in the streams, e.g. *British sailors captured in Iran*, *Campus shooting at VT*, *France presidential election*, *Darfur problem*, etc. Besides, our method was able to find better focused and more discriminative topics, while the baseline method found some confusing and duplicated topics. For instance, #10 of our method clearly and uniquely describes the France presidential election. As a contrast, relevant topical words appear repeatedly in several different topics found by the baseline method (#8, #9, #11 and #12). Similarly, #12 of our method discusses mideast situation, which is also discussed by #14 and #15 of the baseline method, and these two topics are basically duplicated. Besides of duplicated topical words, some of the topics found by the baseline method contain keywords about different (and irrelevant) news events, which may confuse the users. For example, #8 of the baseline method mentions both campus shooting at VT and France presidential election. As a contrast, topics extracted by our method are much better focused. Moreover, since our method is able to fix the asynchronism in the streams and discover better focused and discriminative topics, it can eventually extract more information than the baseline method. In our case, given the same number of common topics ($K = 15$), our method found in #9 the resignation of President of the World Bank, which was not properly addressed by the baseline method. Fig. 13 proves in quantity that topics extracted by our method (sync) are much more discriminative to each other than those extracted by the baseline method (no_sync).

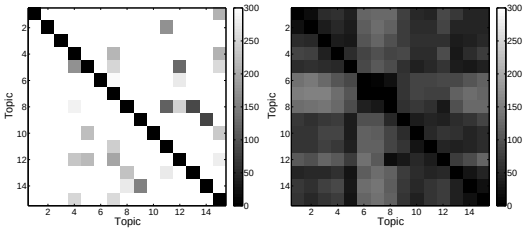
Fig. 14 and 15 show how our method adjusted the timestamps of documents in both news streams, which is consistent to its behavior on literature streams: it automatically discovered documents related to the same topic after considering their semantic as well as temporal information and

	Top-10 topical words (sorted by probability)
1.	British Iranian Iran sailor Britain water captive marine personnel seize
2.	church Somalia prison Somali Mogadishu tax Ethiopian ship Timor muslim
3.	English language company China learn test oil watch native speaker
4.	student shoot Virginia campus Tech Cho gunman university victim classroom
5.	gun Korean mental Korea Cho blame firearm happen society kid
6.	company billion share market price stock game Hong Kong sale
7.	Arab Nigeria Baghdad Maliki car gate wall Sunny Sadr neighborhood
8.	Russia missile Russian Putin Moscow Yeltsin NATO Japan ab Czech
9.	bank Wolfowitz bill senate Republican Olmert resign committe board Turkey
10.	Sarkozy France French Royal socialist Bayrou Nicolas Segolene candidate voter
11.	Afghan Taliban Blair Afghanistan Pakistan Pakistani church Musharraf abort justice
12.	Palestinian Hamas Gaza Isra Israel Fatah rocket camp Lebanese Lebanon
13.	Syria climate Pelosi emission Yushchenko warm Damascus Yanukovich environment water
14.	Iraqi Iran Baghdad nuclear wound Sadr Shiite insurgency Sunni explosion
15.	Darfur African Africa Sudan Sudanese rebel DPRK peacekeeper north Thai

Figure 11: Common topics extracted by our method (sync) from news streams ($K = 15$).

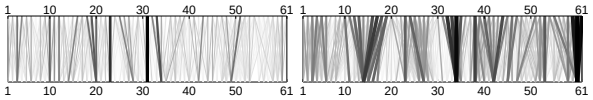
	Top-10 topical words (sorted by probability)
1.	water Syria Pelosi emission Damascus sailor environment music diplomat gas
2.	British Iranian Iran sailor water Britain marine personnel captive seize
3.	Baghdad church tax Sadr Timor desert prison ship gas catholic
4.	English language learn native speaker speak oil culture method gas
5.	Darfur nuclear Sudan Sudanese Africa north Arab bank Thai tribune
6.	student shoot campus Virginia gunman gun Tech bear hall classroom
7.	gun Korean Cho mental Korea student Virginia blame killer happen
8.	gun France mental thing Bayrou (Le)Pen video man Cho Don
9.	wall Royal round voter Bayrou Nigeria candidate ballot (Le)Pen Sunni
10.	Yeltsin Russian rose George treaty Putin ab Soviet Chinese Japanese
11.	Olmert debate Royal oil labor Mccain resign governor candidate veto
12.	Sarkozy France French Royal socialist Nicolas Segolene Chirac voter Paris
13.	Afghan Cheney abort Taliban Kosovo depart drug justice church (Ramos-)Horta
14.	Hamas Fatah camp Gaza Lebanese rocket Palestinian Lebanon military Islam
15.	Hamas Isra Iran Iraqi Palestinian Gaza rocket camp Israel arrest

Figure 12: Common topics extracted by the baseline method (no_sync) from news streams ($K = 15$). Some of the duplicated topical words are underlined.



(a) sync (b) no_sync

Figure 13: The pairwise KL-divergence between topics extracted from the news streams ($K = 15$).



(a) IHT (b) People

Figure 14: The mapping from documents' original timestamps (upper axis) to those determined by our method (lower axis) in news streams. The boldness of lines indicates the number of documents belonging to that mapping.

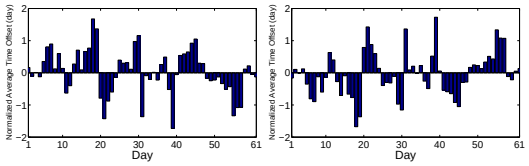
then assigned them to the same timestamp. Fig. 9(b) shows the log-likelihood curves of our method with K ranging from 5 to 30 and 100 rounds of random initialization. Again we can see that our method consistently outperformed the baseline method and its performance was robust against different K and random initialization. Similarly, Fig. 10(b) shows that the semantics of topics extracted by our method with different random initial values were stable.

Results on news streams show that our method performs well on different kinds of data. It has also proved that the local search strategy, which reduces the complexity of our method from $\mathcal{O}(T^3)$ to $\mathcal{O}(T^2)$, would not harm the performance of the method, as long as we have a rough estimation for the level of asynchronism.

6. CONCLUSION AND FUTURE WORK

In this paper we tackle the problem of mining common topics from multiple asynchronous text streams. We propose a novel method which can automatically discover and fix potential asynchronism among streams and consequentially extract better common topics. The key idea of our method is to introduce a self-refinement process by utilizing correlation between the semantic and temporal information in the streams. It performs topic extraction and time synchronization alternately to optimize a unified objective function. A local optimum is guaranteed by our algorithm. We justified the effectiveness of our method on two real-world data sets, with comparison to a baseline method. Empirical results suggest that 1) our method is able to find meaningful and discriminative topics from asynchronous text streams; 2) our method significantly outperforms the baseline method, evaluated both in quality and in quantity; 3) the performance of our method is robust and stable against different parameter settings and random initialization.

In the future we plan to further reduce the computational complexity of our time synchronization algorithm so that our method can be applied to real-time text stream processing.



(a) IHT (b) People

Figure 15: Normalized average time offset of news articles at each day.

7. REFERENCES

[1] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *SIGIR*, pages 37–45, 1998.

[2] D. M. Blei and J. D. Lafferty. Correlated topic models. In *NIPS*, 2005.

[3] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML*, pages 113–120, 2006.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. In *NIPS*, pages 601–608, 2001.

[5] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *VLDB*, pages 181–192, 2005.

[6] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57, 1999.

[7] J. M. Kleinberg. Bursty and hierarchical structure in streams. In *KDD*, pages 91–101, 2002.

[8] A. Krause, J. Leskovec, and C. Guestrin. Data association for topic intensity tracking. In *ICML*, pages 497–504, 2006.

[9] W. Li and A. McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML*, pages 577–584, 2006.

[10] Z. Li, B. Wang, M. Li, and W.-Y. Ma. A probabilistic model for retrospective news event detection. In *SIGIR*, pages 106–113, 2005.

[11] Q. Mei, C. Liu, H. Su, and C. Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *WWW*, pages 533–542, 2006.

[12] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *KDD*, pages 198–207, 2005.

[13] D. M. Mimno, W. Li, and A. McCallum. Mixtures of hierarchical topics with pachinko allocation. In *ICML*, pages 633–640, 2007.

[14] R. C. Swan and J. Allan. Automatic generation of overview timelines. In *SIGIR*, pages 49–56, 2000.

[15] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *KDD*, pages 424–433, 2006.

[16] X. Wang, C. Zhai, X. Hu, and R. Sproat. Mining correlated bursty topic patterns from coordinated text streams. In *KDD*, pages 784–793, 2007.

[17] Y. Yang, T. Pierce, and J. G. Carbonell. A study of retrospective and on-line event detection. In *SIGIR*, pages 28–36, 1998.

[18] C. Zhai, A. Velivelli, and B. Yu. A cross-collection mixture model for comparative text mining. In *KDD*, pages 743–748, 2004.