LDA-based online topic detection using tensor factorization

Xin Guo, Yang Xiang, Qian Chen, Zhenhua Huang and Yongtao Hao

Journal of Information Science 2013 39: 459 originally published online 8 March 2013
DOI: 10.1177/0165551512473066

Article

JIS

Journal of Information Science
39(4) 459–469

DOI: 10.1177/0165551512473066
jis.sagepub.com
SAGE

LDA-based online topic detection using tensor factorization

Xin Guo
Department of Computer Science and Technology and The Key Laboratory of Embedded System and Services Computing, Ministry of Education, Tongji University, China

Yang Xiang
Department of Computer Science and Technology and The Key Laboratory of Embedded System and Services Computing, Ministry of Education, Tongji University, China

Qian Chen
Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan, China

Zhenhua Huang
Department of Computer Science and Technology and The Key Laboratory of Embedded System and Services Computing, Ministry of Education, Tongji University, China

Yongtao Hao
Department of Computer Science and Technology and The Key Laboratory of Embedded System and Services Computing, Ministry of Education, Tongji University, China

Abstract
In the information retrieval field, effective and efficient extraction of topics from large-scale online text streams is challenging because it is a fully unsupervised learning task without prior knowledge. Most previous studies have focused on how to analyse text corpus to extract topics, rarely considering time dimensions. In the present study, we approached topic detection as a temporal optimization problem. Here, we propose a novel approach to incremental topic detection, called online topic detection using tensor factorization (OTD-TF), which is based on latent Dirichlet allocation (LDA). First, topics are obtained from the corpus in current time slices using LDA. Second, a topic tensor with a time dimension is constructed to identify the correlations between pairs of topics. Then, approximate topics are merged using TF. Finally, documents are reallocated to corresponding topic bins. By executing these steps continuously and incrementally, temporal topic detection can be achieved. In theoretical analyses and simulation experiments, OTD-TF outperformed other systems in terms of space and time complexity and achieved a high precision ratio. Our experimental evaluations also revealed interesting temporal patterns in topic emergence, development, extinction, burst and transience.

Keywords
LDA; tensor factorization; topic detection; topic tensor

1. Introduction

With the rapid development of computer science and technology, an increasing amount of information is created by separate authors at different points in time, where subsequent authors build on the material of original authors. Such data bear

Corresponding author:
Xin Guo, Department of Computer Science and Technology, Tongji University, No. 4800, Caoan Rd, Shanghai 201804, China.
Email: guoxinjsj@163.com

timestamps, and because the information changes continuously over time, there are 'evolutionary patterns' hidden in the data. Indeed, the rise and rapid expansion of online forums such as email, news sites, blogs and so forth have made temporal text mining a hot topic in recent years. Currently, such text stream data are ubiquitous but unstructured [1]. Hence, it is difficult to extract and mine potential topics from such sources and explore their temporal patterns [2].

Topic detection and topic correlation analyses are basic tasks in the field of topic detection and tracking. A range of approaches have been investigated, most of which have viewed topic detection as a clustering issue [3]. Some have used timelines to represent temporal documents, transforming the issue into a topic visualization problem that can be solved by assessing the birth and death of topics [4, 5], while others have adopted probabilistic topic models (pTM) to detect topics and trends [6]. Most pTMs are based on latent Dirichlet allocation (LDA) [7], a three-layer Bayesian network model widely used for topic modelling in the fields of image tagging, topic detection and social network analysis. This model assumes that words in a document and documents in a text corpus are exchangeable, that is, the order of words can be neglected, each topic can be considered a multinomial distribution of selected words, and each document can be considered a multinomial distribution of topics [8]. However, previous studies using pTMs have been limited by, for example, employing a fixed number of topics in a document that needs to be specified beforehand, disregarding timestamps, and running algorithms requiring large amounts of memory. Such systems cannot be used for online topic detection.

Various dimension-reduction methods have been used to detect topics. Among them, latent semantic analysis (LSA) using singular value decomposition based on vector space modelling (VSM) [9] can compress a highly dimensional term–document matrix into a k-dimension subspace, greatly reducing text vector dimensions. Although this approach has overcome some shortcomings of VSM, there are still three main limitations. First, calculating large sparse matrices in LSA is very time consuming. Second, the results for the feature space dimension are difficult to explain semantically because each dimension is a linear combination of the original term space. Finally, the approach ignores the order of words in sentences and thus has the same problems associated with the bag-of-words model. While some work has focused on hot-topic detection [10], burst-topic detection [11], spatial analysis of topic detection [12] and topic tracking, we have mainly focused on temporal topic detection of news stories with standard formats of grammar and semantics.

Almost all types of data in web 2.0 applications bear timestamps. Relatively few algorithms have been designed for online text stream topic detection, where data must be processed in a small number of passes with limited storage space. In the present study, we performed incremental topic detection over time using LDA and tensor decomposition. We focused mainly on topic detection and correlation analysis to identify latent semantic structures and temporal patterns of each topic. The main contributions of this work to the field are as follows:

- We present a topic-detection method based on tensor factorization (TF) in the third dimension (i.e. time) that automatically categorizes similar online topics into groups over time.
- We show that TF can transform a topic-detection task into an optimization problem, propose an efficient algorithm based on Gibbs-sampling LDA and canonical polyadic decomposition (CPD), and demonstrate that the approach outperforms existing systems using a theoretical analysis.
- We present empirical results showing that the proposed algorithm can efficiently and effectively identify interesting patterns in temporal topics.

The remainder of this paper is organized as follows. Section 2 presents the problem, Section 3 formally defines our topic-detection tensor model and Section 4 describes experiments used to test the system and provides their empirical results. Conclusions and recommendations for future work are discussed in Section 5.

2. Problem statement

Blogs including microblogs, such as Twitter, some websites (e.g., Wikipedia), and other online sources that generate text temporally (i.e. text streams) are a rapidly growing part of the Internet. We formally define a text stream as follows.

Definition 2.1. A text stream is made up of textual data that are published continuously over time.

This can be formulated as follows:

$$Streams = \{C_1, C_2, \ldots, C_i, \ldots\} \text{ and } C_i = \{D_1, D_2, \ldots D_j, \ldots\}$$

Table 1. Notations of symbols and their corresponding descriptions.

Token	Description	Token	Description
$\circ$	Outer product	$\odot$	Khatri–Rao product
$\otimes$	Kronecker product	a	Scale variable
$\vec{a}$	Vector variable	a_i	The ith element of vector $\vec{a}$
A	Matrix variable	a_{ij}	The ijth element of matrix A
$\boldsymbol{A}$	Three-way array(tensor)	$\boldsymbol{A}(i, j, k)/a_{ijk}$	The ijkth element of tensor A
$\boldsymbol{A}(:, j, k)/a_{:jk}$	The jkth column vector	$\boldsymbol{A}(i, :, :)/a_{i::}$	The ith horizontal slice
$\boldsymbol{A}(i, :, k)/a_{i:k}$	The ikth row vector	$\boldsymbol{A}(:, j, :)/a_{:j:}$	The jth lateral slice
$\boldsymbol{A}(i, j, :)/a_{ij:}$	The ijth tube vector	$\boldsymbol{A}(:, :, k)/a_{::k}$	The kth frontal slice
$\|\|A\|\|_F$ / $\|\|\boldsymbol{A}\|\|_F$	The Frobenius norm of A or $\boldsymbol{A}$	$\times_n$	n-Mode product of a tensor and a matrix
$\boldsymbol{A}_{(n)}$	n-mode metricized version of tensor A	$\bar{\times}_n$	Contracted n-mode product of a tensor and a vector
$A^{\dagger}$	The pseudo inverse of tensor A	$\Delta(\vec{a})$	Dirichlet delta function

where $D_j = \{docID, docTitle, content, timeStamp, source\}$.

Definition 2.2. a topic is an abstract set of series of events.

We consider each topic a mixture of terms in a given vocabulary, and we denote a single topic as $T = \{E, P, N\}$, where E is a collection of events, each of which is an instance of topic T; P is a collection of properties, each of which represent a particular feature of topic T; and N is the mathematical representation of that topic. N is formulated as follows:

$$N = \sum_{v=1}^{V} \pi_v w_v \tag{1}$$

where π_v is the mixture proportion of terms in a given vocabulary, w_v is the index of that term, and V is the size of that vocabulary. Thus, we can treat N as a distribution over vocabulary.

Definition 2.3. Topic correlation is a measure evaluating the correlation cost between two individual topics.

Here, we can denote every pair of topics using the following formulation:

$$TCs(T_i, T_j) = \begin{cases} 1, & \text{if } i = j \\ \lambda \cos(T_i, T_j) + (1-\lambda)\frac{countD(T_i, T_j)}{M}, & \text{while } i \neq j \end{cases} \tag{2}$$

where $countD(T_i, T_j)$ is the number of times that T_i and T_j occur in the same document.

Definition 2.4. Online topic detection (OTD) is a method for identifying topics of online text streams.

OTD assesses and detects topics of news stories in current time slices before the next news stories arrive. The process can be illustrated as follows.

$$\begin{aligned} &\text{Input: Streams} = \{C_1, C_2, \ldots, C_i, \ldots\}, \text{where } C_i \text{ is the corpus in current period} i \text{ and } C_i = \{D_1, D_2, \ldots D_j, \ldots\}. \\ &\text{Current topic vector: } T = \{T_1, T_2, \ldots, T_n\}, \text{in time period } n. \\ &\text{Output: } T_{D_i} = \begin{cases} T_i, & i \in 1, \ldots, n \\ T_{n+1}, & \text{if new topic detected} \end{cases} \end{aligned} \tag{3}$$

The process of mining and analysing temporal pattern of online text streams is called temporal text mining. A time slice can be denoted by a matrix of size $K \times K$ in a specified period t. See Table 1 for the symbols used in this paper.

3. Topic detection using TF

We propose a novel online topic detection approach called online topic detection using tensor factorization (OTD-TF). Given documents in a current period, a topic adjacency matrix is constructed for each iteration. It involves two main

Algorithm 1. Pseudo-code of OTD-TF

```
Input: Text stream, t
Output: document-topic matrix, together with topic-term matrix and updated topic vector.
1  Global variable tensor                    //global variable tensor
2  Initialize M, K, V = readVoc();           //initialize parameters
3  For each period t = 1, …, T do
4     corpus ← readCorps(t);                 //read corpus at time t
5     lda.gibbs(corpus, K, alpha, beta);     //LDA Gibbs method
6     double phi[K][V] ← lda.getPhi();       //Φ matrix
7     double theta[M][K] ← lda.getPhi();     //Θ matrix
8     topicAdjacentMatrix ← covertAdjMatrix(theta); //convert Θ to adjacent matrix
9     if(t equals 1)
10       tensor ← topicAdjacentMatrix;       //if t is equal to 1, extend adjacent matrix
11       continue;
12    else
      //incrementally construct tensor using adjacent matrix
13      tensor ← consTensorIncrem(tensor, topicAdjacentMatrix);
      // tensor factorization method
14      topicThemeMatrix[i][j] ← threeWayFactorization(tensor);
15      compress the number of topic using topicThemeMatrix;
      //get topic vector which implies the current topics list.
16      vector ← reconstruct a compressed topic adjacent matrix;
17      print the vector result in console or hardware files.
18    End if
19 End For
```

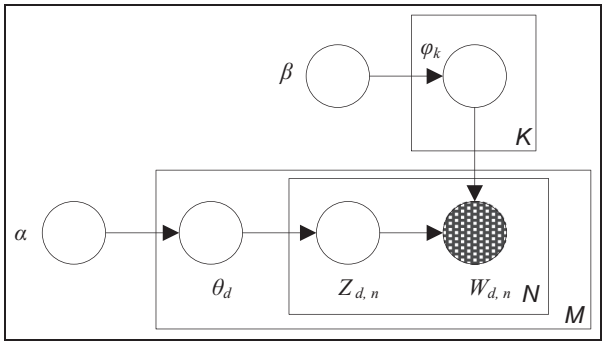

Figure 1. Graphic representation of topic extraction using a modified LDA.

steps: constructing a third-order tensor based on a modified LDA model and then detecting topics using tensor decomposition. This makes online topic detection an optimization problem. The pseudo-code of this algorithm is described below.

Parameter K is the number of topics, V denotes the size of the vocabulary, and M is the number of documents in the current period. There are three main functions in the algorithm, including topic extraction using LDA, incremental tensor construction, and three-way TF. First, the document–topic matrix is produced using LDA, and then the tensor is incrementally constructed and factorized. This results in an updated topic vector that includes the index of each topic in a corresponding text document. The topic adjacency matrix gives the implied correlations between all pairs of topics.

Algorithm 2. Generative process of LDA [7]

Online topic detection using tensor decomposition:
1 For each topic 1, 2, …, K
2 Draw $\varphi_k \sim$ Dirichlet(β);
3 End for
4 For each document D_m from a certain period time of corpus 1, 2, …, M
5 Draw topic mixture proportion $\vartheta_d \sim$ Dirichlet(α);
6 For each word from D_m
7 Draw $Z_{d,n} \sim$ Multinomial(ϑ_d);
8 Draw $W_{d,n} \sim$ Multinomial(φ-$Z_{d,n}$);
9 End for
10 End for

3.1. Topic extraction using LDA

First, we construct a corpus tensor where the first dimension represents documents, the second represents vocabulary terms and the third refers to a timeline. The salient feature in this construction process is that it is incremental. To identify the evolution of a topic over time and to determine the correlations between all pairs of topics, we use a modified LDA model to extract topics as shown in Figure 1.

The modified version of LDA shown in Figure 1 is also presented in Algorithm 2 as follows. Topic extraction is performed by Gibbs sampling.

where φ_k and ϑ_d are both Dirichlet distributed, and the conjugacy property between a multinomial distribution and a Dirichlet distribution can largely simplify the process of inference on parameters. The Dirichlet distribution can be written as follows:

$$Dir(\vec{x}|\vec{\alpha}) = \frac{1}{\Delta(\vec{\alpha})}\prod_{k=1}^{K} x_k^{\alpha_k - 1} \tag{4}$$

where $\Delta(\vec{\alpha})$ is a Dirichlet function.

$$\Delta(\vec{\alpha}) = \int\left(\prod_{k=1}^{K} x_k^{\alpha_k}\right)\mathrm{d}\vec{x} = \frac{\prod_{k=1}^{K}\Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^{K}\alpha_k)} \tag{5}$$

Then, the joint distribution of all parameters then can be derived from Figure 1, as follows:

$$p(\Theta, \Psi, W_d, Z_d|\vec{\alpha}, \vec{\beta}) = \prod_{k=1}^{K} p(\varphi_k|\vec{\beta})\prod_{d=1}^{M} p(\theta_d|\vec{\alpha})\prod_{n=1}^{N} p(z_n|\theta_d)p(w_d|\varphi_{z_n}) \tag{6}$$

We integrate out z and regard vectors α and β as hyperparameters. Because it is difficult to estimate all parameters exactly, we adopt collapsed Gibbs sampling [13] for estimation. The target distribution of z is conditioned on observed values w.Thus, we have

$$p(\vec{z}|\vec{w}) = \frac{p(\vec{z},\vec{w})}{p(\vec{w})} = \frac{p(\vec{w}|\vec{z},\vec{\beta})p(\vec{z}|\vec{\alpha})}{p(\vec{w})} \tag{7}$$

where

$$p(\vec{w}|\vec{z},\vec{\beta}) = \int p(\vec{w}|\vec{z},\vec{\varphi})p(\vec{\varphi}|\vec{\beta})d\vec{\varphi}$$

Using the collapsed LDA Gibbs sampling algorithms [14], we get

$$\varphi_{k,t} = \frac{n_k^{(t)} + \beta_t}{\sum_{t=1}^{V} n_k^{(t)} + \beta_t}, \quad \theta_{m,k} = \frac{n_m^{(k)} + \alpha_k}{\sum_{k=1}^{K} n_m^{(k)} + \alpha_k} \tag{8}$$

where $n_k^{(t)}$ denotes the number of times that term t is observed with topic k, $n_m^{(k)}$ refers to the number of times that topic k is observed with a word of document m, and β_t and α_k denote the tth and kth elements in vector β and α respectively.

We ultimately get matrix Φ of size $K \times V$ and matrix Θ of size $M \times K$. Θ is a document–topic matrix. We transform it into a topic–topic adjacency matrix using a cosine measure, which reflects the probability of two topics existing in the same document, that is, the correlation between each pair of topics.

$$adj_{ij} = adj_{ji} = \begin{cases} 1, & \text{if } k = m; \\ \dfrac{\sum_{m=1}^{M} \varphi_{mi}\varphi_{mj}}{\sqrt{\sum_{m=1}^{M} \varphi_{mi}^2}\sqrt{\sum_{m=1}^{M} \varphi_{mj}^2}}, & \text{otherwise} \end{cases} \tag{9}$$

where adj_{ij} is the ijth element of the topic–topic adjacency matrix. This equation allows us to incrementally construct our topic tensor with a third dimension of time.

3.2. Incremental construction of topic tensor

We incrementally construct a topic tensor during each iteration process, that is, we adopt the slide-window technique for each period. Here, we select month as the time span. If the topic matrix is $A_{(t-1)}$ of size $k_{(t-1)} \times k_{(t-1)}$ after time period $t-1$, and the number of topics at period t is K, then topic tensor Γ_t at period t can be constructed by extending each frontal slice to $k_{(t-1)} + K$, including the adjacency matrix at period t. For each frontal slice of Γ_t,

$$\Gamma(i;j;0) = \begin{cases} 1 & \text{if } i = j \\ a_{ij} & \text{if } i \le k_{(t-1)} \text{ and } j \le k_{(t-1)} \\ 0 & \text{if } i > k_{(t-1)} \text{ and } j > k_{(t-1)} \end{cases} \tag{10}$$

and

$$\Gamma(i;j;0) = \begin{cases} 1 & \text{if } i = j \\ a_{ij} & \text{if } i \le k_{(t-1)} \text{ and } j \le k_{(t-1)} \\ 0 & \text{if } i > k_{(t-1)} \text{ and } j > k_{(t-1)} \end{cases} \tag{11}$$

where adj_{ij} is the ijth element of Adj of size $K \times K$.

3.3. Three-way tensor decomposition

Next, we perform topic detection using TF. A tensor, which can be seen as an N-way array, is the tensor product of N vector spaces. Here, we adopt a third-order tensor ($N = 3$), where there are three indices, and the third index refers to the timeline. We treat each point of the third index as a time slot within which the tensor can be referred to as a time slice, during which a certain number of documents instantly arrives.

Standard matrix factorization approaches and their variants are powerful for two-way-representation feature selection and dimensionality reduction. However, because they are limited when processing multi-way arrays such as third-order tensors, we adopt the CPD approach, the core of which is the alternating least squares (ALS) method [15,16]. TFs first appeared in the psychometrics literature [17].

Given a third-order topic tensor Γ of size $K \times K \times T$, the goal is to compute a CPD with R components that best approximates Γ. This can be seen as an optimization problem, as described below.

$$A,B,C = \underset{A,B,C}{\arg\min} \left\| \mathbf{\Gamma} - \sum_{r=1}^{R} \lambda_r \vec{a}_r \circ \vec{b}_r \circ \vec{c}_r \right\|_{\mathrm{F}} \tag{12}$$

where $\|.\|_{\mathrm{F}}$ is the Frobenius norm of a tensor. We first fix B and C to solve for A and then permute for B and C, respectively, and iteratively repeat the whole procedure until a certain convergence criterion is satisfied. After having fixed all matrices but one, the problem can be reduced to a linear least squares problem. Formally, assuming that A and B are fixed, we can rewrite the above optimization problem as follows:

Algorithm 3. OTD-TF

Procedure of ALS:

Input: tensor Γ

Output: int *Arr*[]

0 Begin

1 Initialize $A(n)$ for $n = 1, 2, 3$

2 Repeat

3 For $n = 1, \ldots, N$ do

4 Update $A_{ip} \leftarrow A_{ip} \frac{(\Gamma^{I \times JK} F)_{ip}}{(AF^T F)_{ip} + \varepsilon}$, $F = (C \odot B)$;

5 Update $B_{jp} \leftarrow B_{jp} \frac{(\Gamma^{J \times IK} F)_{jp}}{(BF^T F)_{jp} + \varepsilon}$, $F = (C \odot A)$;

6 Update $C_{kp} \leftarrow C_{kp} \frac{(\Gamma^{K \times IJ} F)_{kp}}{(CF^T F)_{kp} + \varepsilon}$, $F = (B \odot A)$;

7 End for

8 Until fit ceases to improve or maximum iterations exhausted

9 $\min \| \Gamma^{I \times JK} - A(C \odot B) \|_F$

10 For $\vec{a}_k k = 1, \ldots, K$ //for each row vector of matrix A

11 *Arr*[k] = maxIndex(a_k[]) //return the largest column index of vector a_k

12 End for

13 Return *Arr*[]

14 End

$$\hat{A} = \arg\min_{\hat{A}} \| \mathbf{\Gamma} - \hat{A}(C \odot B)^T \|_{\mathrm{F}} \tag{13}$$

where $\odot$ is a Khatri–Rao product of two matrices.

$$A = \mathbf{\Gamma}_{(1)}[(C \odot B)^{\mathrm{T}}]^{\dagger} = \mathbf{\Gamma}_{(1)}[(C \odot B)(C^{\mathrm{T}} C^{\mathrm{T}} * B^{\mathrm{T}} B)]^{\dagger} \tag{14}$$

and $\hat{A} = A \cdot \mathrm{diag}(\vec{\lambda})$, so the only step remaining is to compute the pseudo-inverse of matrix $(C \odot B)$. The result of this decomposition is three low-rank matrices.

Based on the above formulations, our proposed OTD-TF algorithm can be reformulated as shown in Algorithm 3.

Algorithm 3 performs dimension reduction, providing an array, *Arr*. As for A of size $K \times R$, where $R \ll K$, we have compressed the size of K to that of R, which can be seen as topic clustering.

3.4. Time and space complexity

OTD requires less time and less memory storage than classic topic detection task. For time complexity, we suppose that the size of the corpus in time slice k is m_k, so the time complexity of executing LDA in each time slice is O(a). Hence, the complexity when using a traditional method is O$[(0.5n^2 + 0.5n)a]$ as time increases, whereas that of our method is O(na) because we run each iteration in an incremental way. For space complexity, when topics are detected after LDA, we only need to construct the topic tensor using compressed topics, so the number of topics in a tensor is much less than usual. As time goes by, the space used for topic detection and storage becomes much smaller than that of traditional methods. Indeed, as shown in Section 4, OTD-TF reduces the time and space requirements by almost half compared with traditional methods.

Table 2. The training datasets for topic detection.

Topic id	Topic
01	The Royal Wedding
02	Earthquake Hits Japan
03	South Sudan New Nation
04	Wen's Japan Tour
05	French Lagarde named IMF chief

Table 3. Online topic detection results.

Period	Topic no.: Top 10 words of each topic
February	T2-1: Prince William Kate Middleton couple royal wedding love romance Harry
March	T3-1: Japan tsunami earthquake disaster hit March Friday damage magnitude catastrophe
	T3-2: nuclear plant power radiation reactor Japan Fukushima leak crisis evacuation
April	T4-1: music procession fanfare anthem piece Peter Edward organ Hastings hymn
	T4-2: Chinese China student product stone carve tea study zhu Beijing
	T4-3: wedding royal William Prince London Kate Middleton Westminster Abbey Britain
	T4-4: Japan nuclear plant tsunami radiation Fukushima power earthquake Tepco disaster
	T4-5: Sudan south Abyei government march Juba benjamin khartoum Dinka war
May	T5-1: Sudan southern Wau Chinese peacekeeper CNPC refinery hospital Khartoum medical
	T5-2: wedding prince royal sale William warm Obama Zealand rise pound
	T5-3: plant nuclear Japan reactor power Tokyo tsunami March government Tepco
	T5-4: IMF Europe strauss Kahn Minister country finance Lagarde candidate bank
	T5-5: Japan China Wen Jiabao south premier minister disaster cooperate visit
June	T6-1: IMF Lagarde country Minister Europe support finance candidate fund French
	T6-2: plant Japan nuclear worker reactor water Fukushima tsunami power radiation

4. Experiment and result analysis

We built a data set of five popular topics taken from the website of China Daily (http://www.chinadaily.com.cn). To obtain the text corpus, we employed the Web crawler tool Heritrix 4.2 to crawl files on the website. We extracted 1369 news stories from the period 14 February to 29 June 2011. Our experiment was performed on an Intel® Core 2.27 GHz CPU with 3.00 GB RAM, a 160 GB hard disk, and a Microsoft Windows 7 Professional operating system. The data were pre-processed before being used in the experiment. We removed HTML tags, punctuation and other non-informative text. Experiments were run using Java 1.6.

In this experimental, we focus mainly on topic detection and topic correlation analysis rather than on text categorization. A common standard measure for model quality is the *F*-score [18]. Therefore, we compared our approach to Online LDA (OLDA) in terms of precision, recall and *F*-score. Unless specifically stated, we initialized some parameters in our algorithms with the following values: the number of topics in every document $k = N^{1/2}$, where N is the number of documents in each period; the hyperparameters $\alpha = 0.01$ and $\beta = 0.5$; and the number of iteration in the sampling stage $n = 1000$.

The corpus was labelled beforehand according to the source URL and the name of the website. Moreover, we used the label as a baseline. The topics in the corpus are listed in Table 2.

4.1. Comparison with OLDA

We extracted topics at each period individually using OTD-TF; the results are shown in Table 3. We represent each topic using a discrete distribution of vocabulary; only the top 10 words are shown in this table for simplicity.

Table 4 shows the initial topic list, the list after extension used as the topic tensor, and the topic extraction results after TF at each iteration stage. Note that, once we get the index value that corresponds to the maximum value of each topic vector, subtopics are allocated to existing topics according to that index value.

From Table 4, we can see that a total of six topics were detected in the text stream corpus: the royal wedding, earthquake hits Japan, music procession and hymn, South Sudan new nation, French Lagarde named IMF chief, and Wen's Japan tour. We compared OTD-TF with OLDA in terms of *F*-score based on these baseline data.

We use *F*-score as our performance evaluation measure, as follows:

Table 4. Online topic detection results using OTD-TF.

Iteration	Items	Result
Iteration 1	Topics	T2-1
	Result	T1(T2-1)
Iteration 2	Topics	T3-1, T3-2
	Extension	T1, T3-1, T3-2
	Factorization	T1(T2-1), T2(T3-1, T3-2)
Iteration 3	Topics	T4-1, T4-2, T4-3, T4-4, T4-5
	Extension	T1, T2, T4-1, T4-2, T4-3, T4-4, T4-5
	Factorization	T1(T2-1, T4-2, T4-3), T2(T3-1, T3-2, T4-4), T3(T4-1), T4(T4-5)
Iteration 4	Topics	T5-1, T5-2, T5-3, T5-4, T5-5
	Extension	T1, T2, T3, T4, T5-1, T5-2, T5-3, T5-4, T5-5
	Factorization	T1(T2-1, T4-2, T4-3, T5-2), T2(T3-1, T3-2, T4-4, T5-3), T3(T4-1), T4(T4-5, T5-1), T5(T5-4), T6(T5-5)
Iteration 5	Topics	T6-1, T6-2
	Extension	T1, T2, T3, T4, T5, T6, T6-1, T6-2,
	Factorization	T1(T2-1, T4-2, T4-3, T5-2), T2(T3-1, T3-2, T4-4, T5-3, T6-2), T3(T4-1), T4(T4-5, T5-1), T5(T5-4, T6-1), T6(T5-5)

Table 5. The comparison of *F*-score, precision and recall between OLDA and OTD-TF.

	T1	T2	T4	T5	T6	Total
p(OLDA)	0.8857	0.8814	0.8889	1.0000	1.0000	0.9106
r(OLDA)	0.9841	1.0000	1.0000	1.0000	0.9600	0.9941
F(OLDA)	0.9323	0.9369	0.9412	1.0000	0.9796	0.9505
p(OTD-TF)	0.9286	0.9915	0.8889	1.0000	1.0000	0.9892
r(OTD-TF)	0.9701	1.0000	1.0000	1.0000	0.9600	0.9918
F(OTD-TF)	0.9489	0.9957	0.9412	1.0000	0.9796	0.9905

$$F - \text{score} = \frac{\text{precision} \times \text{recall}}{\lambda\ \text{precision} + (1 - \lambda)\ \text{recall}} \tag{15}$$

where

$$\text{precision} = \frac{|\{cr\} \cap \{rc\}|}{|\{rc\}|},\ \text{recall} = \frac{|\{cr\} \cap \{rc\}|}{|\{cr\}|} \tag{16}$$

and $\{cr\}$ is a collection of documents on a topic that were detected by our prototype system, $\{rc\}$ is a baseline collection of documents on that topic, and |.| denotes the size of a collection. Here, we set λ to 0.5. Table 5 compares the performances of our system and the OLDA method.

As depicted in the table, our method outperformed OLDA in terms of both precision and *F*-score, although both systems had similar recall.

4.2. Topics over time

We demonstrate the topics' strength curves in Figure 2, where we briefly represent the number of documents as the topic strength. For simplicity, we only demonstrate the topic strength computed monthly.

Figure 2 illustrates each topic's strength curve, where topic strength reflects the number of documents of that topic. For simplicity, we present it by month. As shown in Figure 2, the topic *the royal wedding* began in February and peaked in April when the wedding was held. The topic *earthquake hits Japan*, which was a burst topic, almost reaching its peak when it emerged in March 11. Thus, it had a great impact and lasted for a long time. The topic *music procession and hymn* was detected by mistake, as it is a document under *the royal wedding* topic. *South Sudan new nation* and *French Lagarde named IMF chief* were both in the initial stage of coverage, and their curves extended in a relatively stable

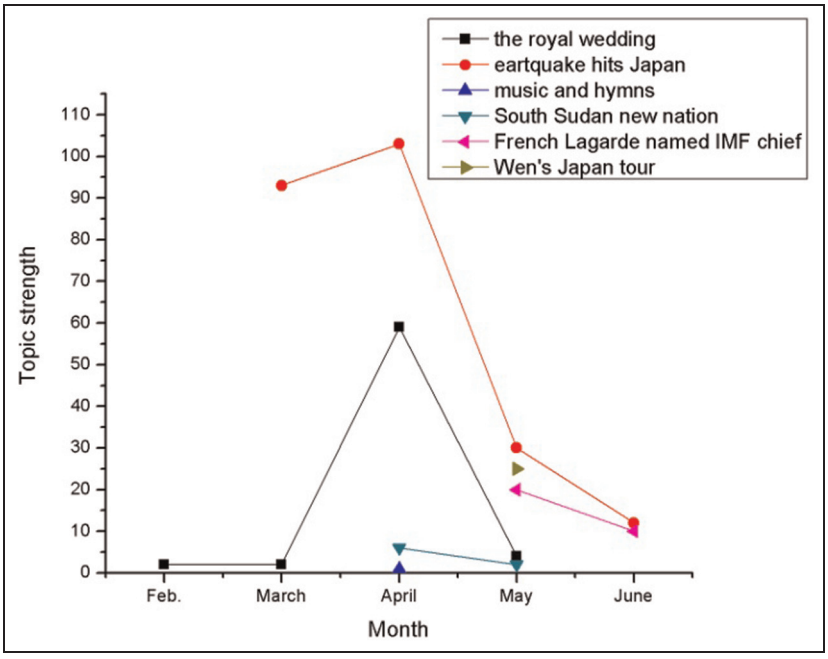

Figure 2. Strengths of topics over time.

manner. In fact, in early February, each party leader in South Sudan agreed to found a new nation and formally declare independence. Finally *Wen's Japan tour* was a transient topic that emerged in May.

5. Conclusions and future work

Our novel approach to online topic detection, OTD-TF, can identify topic trends over time as well as the correlations among them. It shows good time and space complexity and outperforms other approaches such as OLDA in terms of time cost and *F*-score.

As for future work, there are many potential directions. It would be interesting to design a detection algorithm for a distributed environment or based on various types of text stream corpus. It would be also valuable to extend the current approach to other fields such as image/video retrieval and social network analysis.

Acknowledgements

This work was partially supported by the NSFC under grant no. 71171148, 61103069 and 51075306 and by the National Technology Plan Project under grant no. 2012BAD35B01. The work was also funded by the National High-Tech Research and Development Plan of China under grant no. 2012AA062203 and by the Project of special funds for the Informatization Development of Shanghai Municipality under grant no. 200901015, as well as the Innovation Action Program of Shanghai Science and Technology Commission under grant no. 11dz1501703 and 11dz1210600.

References

[1] Kleinberg J. Temporal dynamics of on-line information streams. In: Garofalakis M, Gehrke J and Rastogi R (eds) *Data stream management: Processing high-speed data streams*. Berlin: Springer, 2008.

[2] Kleinberg J. Bursty and hierarchical structure in streams. *Proceedings of the 8th ACM SIGKDD international conference on knowledge discovery and data mining*, 2002, pp. 1–25.

[3] Zhang J, Ghahramani Z and Yang Y. A probabilistic model for online document clustering with application to novelty detection. *Nineteenth annual conference on neural information processing systems, NIPS*, 2005, pp. 1–8.

[4] Ahmed A and Xing EP. Timeline: A dynamic hierarchical dirichlet process model for recovering birth/death and evolution of topics in text stream. *Proceedings of the 26th international conference on uncertainty in artificial intelligence*, 2010, pp. 1–10.

[5] Ahmed A and Xing EP. Timelines: Recovering birth and evolution of topics in scientific literature using dynamic non-parametric Bayesian models, http://www.umiacs.umd.edu/~jbg/nips_tm_workshop/19.pdf

[6] AlSumait L, Barbará D and Domeniconi C. On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking. *2008 eighth IEEE international conference on data mining*, 2008, pp. 3–12.
[7] Anthes G. Topic models vs. unstructured data. *Communications of the ACM* 2010; 53: 16–18.
[8] Blei DM, Ng AY and Jordan MI. Latent Dirichlet allocation. *Journal of Machine Learning Research* 2003; 3: 993–1022.
[9] Deerwester S, Dumais ST, Furnas GW, Landauer TK and Harshman R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 1990; 41: 391–407.
[10] Zhang C, Fan X and Chen X. Hot topic detection on Chinese short text. *Communications in Computer and Information Science* 2011; 176: 207–212.
[11] Ma HF and Ma HL. Combining burst detection for hot topic extraction. *Advanced Materials Research* 2011; 268–270: 1283–1288.
[12] Pan CC and Mitra P. Event detection with spatial latent Dirichlet allocation. *Proceedings of the 11th annual international ACM/IEEE joint conference on digital libraries*, 2011.
[13] Griffiths TL and Steyvers M. Finding scientific topics. *Proceedings of the National Academy of Sciences* 2004; 101(suppl.): 5228–5235.
[14] Heinrich G. Parameter estimation for text analysis. *Technical Report*, 2005.
[15] Stegeman A and Tenberge J. Kruskal's condition for uniqueness in Candecomp/Parafac when ranks and k-ranks coincide. *Computational Statistics and Data Analysis* 2006; 50: 210–220.
[16] Harshman RA. Foundations of the PARAFAC procedure: Models and conditions for an 'explanatory' multimodal factor analysis. *UCLA Working Papers in Phonetics* 1970; 16: 1–84.
[17] Cichocki A, Zdunek R, Phan AH and Amari SI. *Nonnegative matrix and tensor factorizations – Applications to exploratory multi-way data analysis and blind source separation*. Chichester: John Wiley & Sons, 2009.
[18] Azzopardi L, Girolami M and van Risjbergen K. Investigating the relationship between language model perplexity and IR precision-recall measures. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, 2003, pp. 369–370.