Tracking and Connecting Topics via Incremental Hierarchical Dirichlet Processes

Zekai J. Gao[§‡] Yangqiu Song[§] Shixia Liu[§] Haixun Wang[§] Weiwei Cui Hao Wei Yang Chen
[§]*Microsoft Research Asia, Beijing, China.* [‡]*Zhejiang University, Hangzhou, China.*
[†]*Hong Kong University of Science and Technology, Hong Kong.*
{*v-zegao,yangqiu.song,shliu,haixun.wang*}*@microsoft.com; weiwei@cse.ust.hk*

***Abstract*—Much research has been devoted to topic detection from text, but one major challenge has not been addressed: revealing the rich relationships that exist among the detected topics. Finding such relationships is important since many applications are interested in how topics come into being, how they develop, grow, disintegrate, and finally disappear. In this paper, we present a novel method that reveal the inter-connections among topics discovered from the text data. Specifically, our method focuses on how one topic splits into multiple topics, and how multiple topics merge into one topic. We adopt the hierarchical Dirichlet process (HDP) model, and propose an incremental Gibbs sampling algorithm to incrementally derive and refine the labels of clusters. We then characterize the splitting and merging patterns among clusters based on how labels change. We propose a global analysis process that focuses on cluster splitting and merging, and a finer-granularity analysis process that helps users to better understand the content of the clusters and the evolution patterns. We also develop a visualization to present the results.**

***Keywords*-Hierarchical Dirichlet processes, Incremental Gibbs Sampling, Clustering, Mixture models**

I. Introduction

In many fields, including business analysis and academic research, it is not only important to keep track of topics of interest, but also to understand the evolution of topics. A topic has a life cycle, and to understand its life cycle is to understand how a topic comes into being, what triggers and contributes to its development and its disintegration, and how it finally dissolves into other topics, or simply disappears.

Much work has been devoted to topic detection [1]. For example, in the word cloud approach, words that appear more frequently are given greater prominence, and are used to summarize the text. Statistical methods such as the mixture of multinomial model [2] and latent Dirichlet Allocation (LDA) [3] try to find latent topics embodied by the distribution of a set of words. However, these methods do not reveal the dynamics and interconnections among the detected topics. Although it is straightforward to give a temporal dimension to topics, for instance, by detecting topics in windows over text streams, it alone is insufficient to reveal the causality among topics.

In this paper, we study the overall evolution of topics and their critical events in text streams. The critical events are a number of fundamental topic life-cycle events, including topic birth, splitting, merging, and death. Topic merging and splitting are the major relationships characterizing the connections among topics. As a result, we mainly focus on revealing how two (or more) topics are combined into one topic and how one topic is divided into several related topics.

Fig. 1(a) shows the evolution of topics extracted from a news dataset, as well as their relationships to one another. This news dataset contains 16 day Bing news related to "Obama." In the figure, each colored layer represents a derived cluster (hence a topic). The timestamps of the topic layers are associated with keyword clouds (Fig. 1(b)) and important documents (Fig. 1(c)). These summarize the content of the topic and its evolution over time. At each time point, the width of a layer represents the strength or popularity of the topic in terms of the number of documents covered by the topic at that time. With this visualization, users can observe how topics evolve over time, including its strength, content, and splitting/merging relationship. In Fig. 1, we can find that topic "Egypt" emerged on Jan. 27. Later it combines with topic "white house" together to generate a "democracy" topic. Moreover, the topic "reform" splits into Obama's "faith" and "health care" related issues around Feb. 2. Then it gradually develops into Obama quitting "smoking", First lady and "campaign", the education of his "daughters" , meeting "ambassador" and "university" speech from Feb. 6 to Feb. 10.

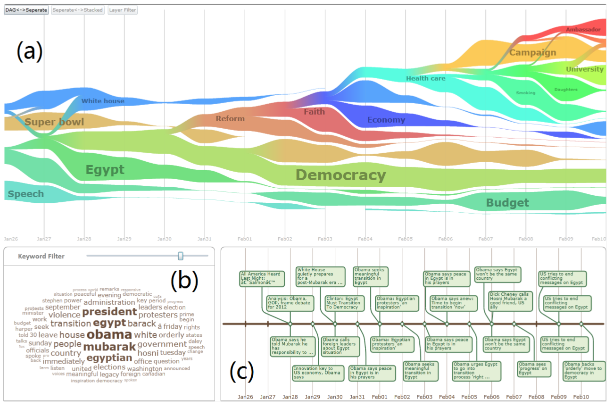


Figure 1. An example of splitting/merging of text clusters: News articles of 16 days related to "Obama".

There are two challenges to mine evolution patterns and the related critical events. The first challenge is how to model the evolution relationships among topics. The evolution patterns may change considerably between two time points. Consequently, it is hard to model them by using current evolutionary clustering [4], [5] or topic modeling approaches [6], [7]. The second challenge is how to allow the user to quickly and effectively examine the major reasons that trigger these evolution patterns. Understanding why is very important for the user to derive insight from a large set of text data, and it is therefore desirable to design a mechanism to extract the critical events, as well as the keyword connections to provide the related information.

To tackle these challenges, we propose an approach to tracking and connecting clusters in text data. Our approach consists of two phases: a global analysis and a local analysis. The global analysis focuses on learning the cluster merging/splitting patterns. In this phase, we propose an incremental learning procedure to learn the the hierarchical Dirichlet processes (HDP) model [8] and the splitting and merging relationships are then extracted given the incremental Gibbs sampling of cluster indicators. The local analysis aims at automatically identifying critical events and keyword connections. The keyword connections are used to represent the semantics underlying text. Compared to the top topic keywords based on the bag-of-words model, the co-occurrence analysis provides users with the second order statistics of keywords.

II. Global Analysis of Cluster Splitting/Merging

In this section, we present the probabilistic model for incremental document splitting and merging analysis. We adopt an HDP mixture model since it provides a unified view of multiple corpora analysis. When each corpus is a document, and each data point is a word, HDP is a Bayesian hierarchical modeling of LDA [3] with Dirichlet process prior. In particular, HDP can automatically determine the topic/cluster numbers [8].

We first introduce some concepts and notations which are useful for subsequent discussions. In our model, we assume the data are coming in an incremental batch-mode manner, i.e., there are multiple documents coming at each epoch (or time point, e.g. a month). We denote t as the time point, and $X^t_j = \{x^t_{j1}, \ldots, x^t_{ji}, \ldots, x^t_{jn^t_j}\}$ is the data set at time t, where x^t_{ji} is the ith data in jth corpus. n^t_j is the data number in corpus j at time t. The associated cluster indicator variables are denoted by $Z^t_j = \{z^t_{j1}, \ldots, z^t_{ji}, \ldots, z^t_{jn^t_j}\}$, where z^t_{ji} is the cluster assignment for ith data in jth corpus. Moreover, we let $X^t = \{X^t_1, \ldots, X^t_J\}$ and $Z^t = \{Z^t_1, \ldots, Z^t_J\}$ for all the corpora $1, \ldots, J$; $X = \{X^1, \ldots, X^T\}$ and $Z = \{Z^1, \ldots, Z^T\}$ for all the data.

A. HDP Modeling

We model the data as an HDP mixture, where the documents at different time epochs share the same HDP generative process. Inspired by the TDT method based on DP mixture model [9], our incremental HDP also leverages the property of Dirichlet process [10] to automatically infer the changing number of clusters. In HDP, a global measure G_0 is drawn from a DP(γ, H), with concentration parameter γ and base measure H. Then, a set of measures $\{G_j\}^J_{j=1}$ is drawn from the DP with base measure G_0 . Here, G_j models corpus j. Such a process is mathematically summarized as

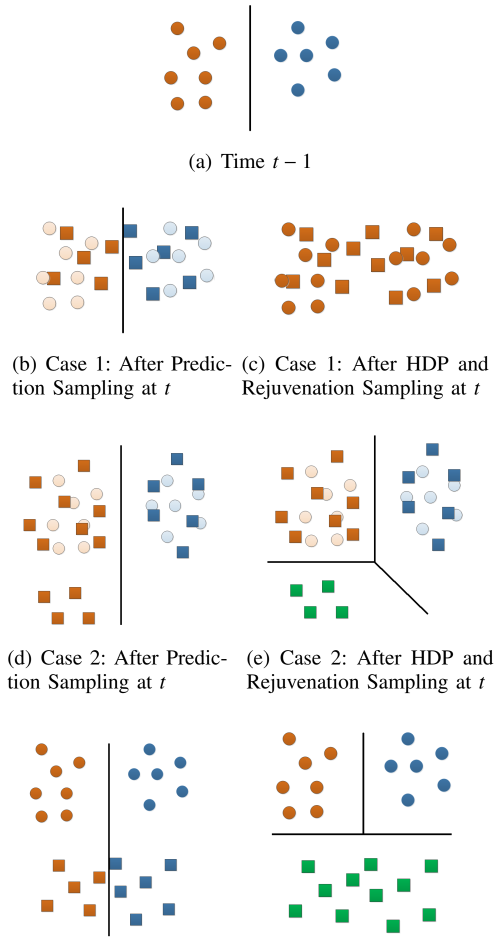
$$G_0 \sim \text{DP}(\gamma, H), \quad G_j|G_0, \alpha_0 \sim \text{DP}(\alpha_0, G_0). \qquad (1)$$

Given the global measure G_0 and concentration parameter α_0, G_j's are conditionally dependent. Having G_j , sample x^t_{ji} at time t in corpus j is drawn from the following mixture model

$$\theta^t_{ji} \sim G_j, \quad x^t_{ji} \sim \text{Multi}(x|\theta^t_{ji}), \qquad (2)$$

We assume the distribution of each cluster is a multinomial distribution:

$$P(x^t_{ji}|\phi_k) = \text{Multi}(x^t_{ji}|\phi_k) = \frac{\sum_{d=1}^{D} x^t_{ji,d}!}{\prod_{d=1}^{D} x^t_{ji,d}!} \prod_{d=1}^{D} \phi_{k,d}^{x^t_{ji,d}} \qquad (3)$$



(a) Time $t-1$

(b) Case 1: After Prediction Sampling at t

(c) Case 1: After HDP and Rejuvenation Sampling at t

(d) Case 2: After Prediction Sampling at t

(e) Case 2: After HDP and Rejuvenation Sampling at t

(f) Case 3: After Prediction Sampling at t

(g) Case 3: After HDP and Rejuvenation Sampling at t

Figure 2. Examples of splitting/merging of clusters. (Circles represent samples at time $t-1$; rectangles encode samples at time t.)

where $x^t_{ji,d}$ is the dth dimension of document term vector, D is the number of dimension of x^t_{ji}, and ϕ_k is the cluster distribution parameter. $\theta^t_{ji} = \phi_k$ if x^t_{ji} is in the kth cluster. When applying HDP to topic modeling, x^t_{ji} is one word and then we have $D = 1$.

Following the stick-breaking construction [11], G_0 has the form

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k}, \phi_k \sim H, \boldsymbol{\beta} \sim \mathrm{GEM}(\gamma), \boldsymbol{\beta} = (\beta_k)_{k=1}^{\infty} \quad (4)$$

The discrete set of parameters $\{\phi_k\}_{k=1}^{\infty}$ is drawn from the base measure H, which is a Dirichlet distribution. GEM (γ) refers to such a process: $\tilde{\beta}_k \sim Beta(1, \gamma)$, $\beta_k = \tilde{\beta}_k \prod_{i=1}^{k-1}(1 - \tilde{\beta}_i)$. δ_{ϕ_k} is a probability measure concentrated at ϕ_k . Then it is shown in [8] that G_j can be constructed as

$$G_j = \sum\nolimits_{k=1}^{\infty} \pi_{jk} \delta_{\phi_k}, \quad \boldsymbol{\pi}_j \,|\, \boldsymbol{\beta}, \alpha_0 \sim \mathrm{DP}(\alpha_0, \boldsymbol{\beta}), \quad (5)$$

where $\boldsymbol{\pi}_j$ is a vector composed by π_{jk}. This formula indicates that different corpora share the same set of distinct atoms [8]. We then have the underlying model to generate x^t_{ji} for corpus j as

$$z^t_{ji} \sim \mathrm{Multi}(z|\boldsymbol{\pi}_j), x^t_{ji} \sim \mathrm{Multi}(x|\theta^t_{ji} = \phi_{z^t_{ji}}). \quad (6)$$

Here the second multinomial distribution is with the parameter θ^t_{ji} which is equals to $\phi_{z^t_{ji}}$ when the cluster label of x^t_{ji} is z^t_{ji}.

One of the major schemes of Gibbs sampling to infer HDP is to first sample z^t_{ji} and then sample other hyper-parameters [8]. Sampling the label z^t_{ji} of x^t_{ji} is given by:

$$\begin{aligned} p(z^t_{ji} = k|Z^{\neg tji}, X) \quad & \propto p(z^t_{ji} = k|Z^{\neg tji}) \cdot p(x^t_{ji}|Z^{\neg tji}_k, X^{\neg tji}_k) \\ & \propto \begin{cases} (n^{\neg tji}_{j,k} + \alpha_0\beta_k) f^{\neg tji}_k(x^t_{ji}) & k \leq K_{\text{active}} \\ \alpha_0\beta_u f^{\neg tji}_{new}(x^t_{ji}) & k > K_{\text{active}} \end{cases} \end{aligned} \quad (7)$$

where $Z^{\neg tji}$ and $X^{\neg tji}$ represent the cluster indicator variables and observations without z^t_{ij} and x^t_{ij} respectively, $Z^{\neg tji}_k$ and $X^{\neg tji}_k$ represent the variables in cluster k except for z^t_{ij} and x^t_{ij} respectively, K_{active} is the sampled cluster number, $n^{\neg tji}_{j,k}$ is the number of x except for x^t_{ji} that belongs to cluster k, and $\beta_u = 1 - \sum_{k=1}^{K_{\text{active}}} \beta_k$. Moreover, $f^{\neg tji}_k(x^t_{ji}) = p(x^t_{ji}|Z^{\neg tji}_k, X^{\neg tji}_k)$ and $f^{\neg tji}_{new}(x^t_{ji}) = p(x^t_{ji})$ can be computed by the marginal distribution based on the conjugate multinomial and Dirichlet distributions [8].

B. Incremental Splitting/Merging Computing

In this section, we present an incremental Gibbs sampling algorithm to sample the incoming documents as well as to model the splitting and merging process of the clusters. The major steps of the algorithm are illustrated in Algorithm 1. In the algorithm, we incrementally sample the latent cluster indicator variable z^t_{ji} for x^t_{ji}. At each time t, we introduce three sampling steps and obtain three summarization results.

1) Sampling: The sampling procedure of the incremental HDP model targets at extracting the connections between dynamic clusters. We have three samplers including "prediction sampler", "HDP sampler", and "rejuvenation sampler". The "prediction sampler" is a simulation of supervised classifier, which predicts the labels of x^t_{ji} at time t based on the previous HDP model. The "HDP sampler" is a simulation of semi-supervised clustering, which infers the labels of X^t while fixing the old labels of $X^{\{1:t-1\}}$, where $X^{\{1:t-1\}} = \{X^1, \ldots, X^{t-1}\}$. The "rejuvenation sampler" works as a pure unsupervised clustering, which re-samples the data $X^{\{t-T_{win}+1:t\}}$ within a time window T_{win}.

Prediction Sampler. As shown in Fig. 2(a), before time t, we have some samples, as well as an HDP model. We first apply a prediction sampler to predict the labels of new coming data at time t based on the previous HDP model (shown in Figs. 2(b), 2(d) and 2(f)). The prediction sampler is defined as:

$$\begin{aligned} & p(z^t_{ji} = k|Z^{\{1:t-1\},old}, X^{\{1:t-1\}}) \\ \propto \quad & p(z^t_{ji} = k|Z^{\{1:t-1\},old}) \cdot p(x^t_{ji}|Z^{\{1:t-1\},old}_k, X^{\{1:t-1\}}_k) \\ \propto \quad & (n^{\{1:t-1\},old}_{j,k} + \alpha^t_0\beta^t_k) f^{\{1:t-1\},old}_k(x_{ji}) \;\; k \leq K_{\text{active}} \end{aligned} \quad (8)$$

where $Z^{\{1:t-1\},old} = \{Z^{1,old}, \ldots, Z^{t-1,old}\}$, $Z^{\{1:t-1\},old}$ is the old data label set before predicting z^t_{ji}, $n^{\{1:t-1\},old}_{j,k}$ is the number of documents that belong to cluster k from time 1 to $t-1$, and $f^{\{1:t-1\},old}_k(x_{ji}) = p(x^t_{ji}|Z^{\{1:t-1\},old}_k, X^{\{1:t-1\}}_k)$ is the marginal distribution based on the previous model. The prediction sampler neither modifies the HDP model, nor generates new clusters. It mainly targets at predicting the labels of the samples at time t based on the HDP model at time $t-1$. We denote the predicted labels of x^t_{ji} as $z^{t,old}_{ji}$.

HDP Sampler. After sampling by the prediction sampler, we have a set of labels of the new incoming data X^t. However, the labels are only based on the previous data and model. They may fail to clearly convey the content of the new data. To tackle this problem, we apply an HDP sampler based on the property of DP, to re-sample the document labels of X^t

$$\begin{aligned} & p(z^t_{ji} = k|Z^{\{1:t-1\},old}, Z^{t,new,\neg tji}, X^{\{1:t\},\neg tji}) \\ \propto \quad & p(z^t_{ji} = k|Z^{\{1:t-1\},old}, Z^{t,new,\neg tji}) \\ & \cdot p(x^t_{ji}|Z^{\{1:t-1\},old}_k, Z^{t,new,\neg tji}_k, X^{\{1:t\},\neg tji}_k) \\ \propto \quad & \begin{cases} (n^{\{1:t\},new,\neg tji}_{j,k} + \alpha^t_0\beta^t_k) f^{\{1:t\},new,\neg tji}_k(x_{ji}) & k \leq K_{\text{active}} \\ \alpha_0\beta_u f^{\{1:t\},new,\neg tji}_{new}(x_{ji}) & k > K_{\text{active}} \end{cases} \end{aligned} \quad (9)$$

where $Z^{t,new,\neg tji}$ is the sampled data label set of X^t except for x^t_{ji}, $X^{\{1:t\},\neg tji}$ is $X^{\{1:t\}}$ without x^t_{ji}, $n^{\{1:t\},new,\neg tji}_{j,k}$ is the number of documents that belong to cluster k from time 1 to t except for x^t_{ji}. Similar to the computation of $f^{\neg ji}_k(x_{ji})$ and $f^{\neg ji}_{new}(x_{ji})$, $f^{\{1:t\},new,\neg tji}_k(x_{ji})$ and $f^{\{1:t\},new,\neg tji}_{new}$ are calculated based on the new data and labels from time 1 to t. The HDP sampler both modifies the HDP model and generates new clusters for the new coming data. We denote the predicted labels of x^t_{ji} as

$z_{ji}^{t,new}$ here. After this step, we have $Z^{\{1:t-1\},old}$ and $Z^{t,new}$ for $X^{\{1:t-1\}}$ and X^t.

Rejuvenation Sampler. Inspired by the incremental Gibbs sampler for LDA [12], we also provide a rejuvenation sampler for historical data. In this sampler, we bound the rejuvenation set in a certain time window, to fix the memory cost of the inference algorithm. We select a time window T_{win} to do the rejuvenation sampling, which means we only sample the labels z_{ji}^{τ} from $t - T_{win} + 1$ to t for better fitness to the HDP model:

$$\begin{aligned} & p(z_{ji}^{\tau} = k | Z^{\{1:t-T_{win}\},old}, Z^{\{t-T_{win}+1:t\},new,\neg\tau ji}, X^{\{1:t\},\neg\tau ji}) \\ \propto\ & p(z_{ji}^{\tau} = k | Z^{\{1:t-T_{win}\},old}, Z^{\{t-T_{win}+1:t\},new,\neg\tau ji}) \\ & \cdot p(x_{ji}^{\tau} | Z_k^{\{1:t-T_{win}\},old}, Z_k^{\{t-T_{win}+1,t\},new,\neg\tau ji}, X_k^{\{1:t\},\neg\tau ji}) \\ \propto\ & (n_{j,k}^{\{1:t\},new,\neg\tau ji} + \alpha_0^t \beta_k^t) f_k^{\{1:t\},new,\neg\tau ji}(x_{ji}) \quad k \leq K_{\text{active}} \end{aligned} \tag{10}$$

where $Z^{\{t-T_{win}+1:t\},new,\neg\tau ji}$ is the sampled data label set of $X^{\{t-T_{win}+1:t\}}$ except for x_{ji}^{τ}, $n_{j,k}^{\{1:t\},new,\neg\tau ji}$ is the number of documents that belong to cluster k from time 1 to t based on the old labels $Z^{\{1:t-T_{win}\},old}$ and new labels $Z^{\{t-T_{win}+1:t\},new,\neg\tau ji}$. The rejuvenation sampler modifies the HDP model but does not generate new clusters. After this step, we have $Z^{\{1:t-T_{win}\},old}$ and $Z^{\{t-T_{win}+1:t\},new}$ for $X^{\{1:t-T_{win}\},old}$ and $X^{\{t-T_{win}+1:t\},new}$. In the incremental setting, $X^{\{1:t-T_{win}\}}$ and $Z^{\{1:t-T_{win}\},old}$ can be saved and removed from memory. As shown in Fig. 2(c), the clusters in the adjacent times epochs merge into one cluster, and one of the previous clusters dies after re-sampling. In Fig. 2(e), the left cluster splits into two clusters from time $t-1$ to t, and one of the clusters is new while another remains unchanged. Moreover, in Fig. 2(g), both left and right clusters split, while the bottom documents merge into one new cluster from time $t-1$ to t.

2) Summarization: After sampling at each time point, we summarize the splitting and merging relationships of the related clusters. Typically, there are three types of splitting/merging statistics, in terms of "merging input at time t", "splitting output at time $t-1$", and "cluster content at time t" as shown in Algorithm 1. The summarization of "merging input at time t" measures how many documents in a cluster at current time t are coming from different clusters based on the previous HDP model. The summarization of "splitting output at time $t-1$" measures how many documents will be sampled into different clusters for a specific cluster. The summarization of "cluster content at time t" shows the top keywords of a specific cluster at time t. We compute them respectively as follows. The merging and splitting probabilities are measured based on both the data at time t and the historical data in a time window with size T_{win}.

Algorithm 1 Incremental HDP Gibbs Sampling.

```
Input: Initial cluster number $K_{init}$. Document sets at each time
  $X^1, \ldots, X^T$. Maximum sampling iteration number $MaxIter$.
  Record window size $T_{win}$.
  {\\Initialize time 1.}
Random initialize cluster IDs for time 1.
for $i = 1, \ldots, n^1$ do
    Sample $z_{ji}^1$ according to Eq. (9).
end for
Sample hyper-parameters $\alpha_0$ and $\gamma$ [13].
for $t = 2, \ldots, T$ do
    {\\Prediction Sampler.}
    for $j = 1, \ldots, J$ and $i = 1, \ldots, n_j^t$ do
        Predict $z_{ji}^t$ (denoted as $z_{ji}^{t,old}$) according to Eq. (8).
    end for
    {\\HDP Sampler.}
    for $iter < MaxIter$, $j = 1, \ldots, J$ and $i = 1, \ldots, n_j^t$ do
        Sample $z_{ji}^t$ (denoted as $z_{ji}^{t,new}$) according to Eq. (9).
        Update cluster models $\phi_k$.
    end for
    {\\Rejuvenation Sampler.}
    Let $T_{win} = \max(t - T_{win} + 1, 1)$.
    for $\tau = (t - T_{win} + 1), \ldots, t$ do
        for $iter < MaxIter$, $j = 1, \ldots, J$ and $i = 1, \ldots, n_j^{\tau}$ do
            Re-sample $z_{ji}^{\tau}$ (denoted as $z_{ji}^{\tau,new}$) according to Eq. (10).
        end for
    end for
    Sample hyper-parameters $\alpha_0$ and $\gamma$ [13].
    Summarize merging input at time $t$ according to Eq. (11).
    Summarize splitting output at time $t-1$ according to
      Eq. (12).
    Summarize cluster at time $t$ according to Eq. (13).
    Let $z_{ji}^{\tau,old} = z_{ji}^{\tau,new}$ for all $\tau < t, j, i$.
end for
```

Merging Input at Time t. For the merging input at time t, the proportion of cluster r coming from cluster s is measured by the difference between $z_{ji}^{t,old}$ and $z_{ji}^{t,new}$ from time $t - T_{win} + 1$ to t:

$$P_t^{in}(s \to r) \triangleq \frac{\sum_{\tau=t-T_{win}+1}^{t} \sum_{i,j} I(z_{ji}^{\tau,old} = s \ \&\ z_{ji}^{\tau,new} = r)}{\sum_{\tau=t-T_{win}+1}^{t} \sum_{i=1}^{n^t} I(z_{ji}^{\tau,new} = r)} \tag{11}$$

where $I(\cdot)$ is an indicator function that flips between binary values, i.e., $I(true) = 1$ and $I(false) = 0$. As shown in Figs. 2(c) and 2(g), we can have two basic patterns of merging. The first case happens when two or more clusters become combined into one (Figs. 2(c)). The second one is more complex, the new cluster is merged from the two (or more) branches which are split from the previous clusters (Fig. 2(g)).

Splitting Output at Time $t-1$. For the splitting output at time $t-1$, the proportion of cluster s flowing to cluster r is measured by the difference between $z_{ji}^{t,old}$ and $z_{ji}^{t,new}$ from time $t - T_{win} + 1$ to t:

$$P_{t-1}^{out}(s \to r) \triangleq \frac{\sum_{\tau=t-T_{win}+1}^{t} \sum_{j,i} I(z_{ji}^{\tau,old} = s \ \&\ z_{ji}^{\tau,new} = r)}{\sum_{\tau=t-T_{win}+1}^{t} \sum_{j,i} I(z_{ji}^{\tau,old} = s)}. \tag{12}$$

In cluster s, if some documents are clustered into different clusters when incrementally processing more documents, then we can regard that the current cluster cannot describe the content of the inside documents anymore. Consequently, the cluster is actually split into several smaller clusters, based on the new documents and model. As shown in Fig. 2(e), the left cluster splits into two clusters when incrementally handling new documents. In Fig. 2(g) both left and right sides split, the historical data is then useful for extracting such splitting relationships.

Cluster Content at Time *t*. For each time t, we need to summarize the cluster content based on the top keywords. Since the documents are represented by the vectors of term frequencies, the cluster center can be regarded as the histogram of term frequencies in the cluster. The posterior of the cluster parameter ϕ is computed by:

$$p(\phi_k^t \big| \{x_{ji}^t, z_{ji}^t = k, \forall\, j, i\}, H) \propto p(\phi_k|H) p(x_{ji}^t|\phi_k, \{z_{ji}^t = k, \forall\, j, i\}). \tag{13}$$

Here, $p(\phi_k|H)$ is a Dirichlet distribution and $p(x_i^t|\phi_k, z_i^t = k)$ is a multinomial distribution. As a result, the posterior distribution is also a Dirichlet distribution, and ϕ_k^t is considered as the pseudo-count of the cluster term frequency vector at time t. Accordingly, the top keywords can be extracted based on ϕ_k^t to represent the cluster at time t.

III. Local Analysis at a Finer Granularity

We have shown how to incrementally infer an HDP model, and described topic dynamics and their splitting and merging relationships. Now we illustrate how to analyze document corpora at a finer granularity. Splitting and merging represent connections among topics over time. Knowing what are the most critical events during splitting/merging is of great interest. Besides the critical events, users are also interested in why clusters split and merge. To facilitate the above analysis tasks, we propose a method based on the co-occurrence of semantic words to discern the hidden splitting/merging reasons. Furthermore, to present the most salient content to the user, we develop a keyword ranking approach to show users the content evolution along time.

A. Critical Event Detection

The first type of critical events is cluster birth and death. The birth of a cluster denotes an emerging topic in the text stream, while the death of a cluster indicates a disappeared topic. We can then detect the new topics through finding the new generated clusters in HDP and the death topics by identifying the disintegration of the clusters. In our incremental Gibbs sampling framework, we maintain a hash table for each time epoch, and the critical events of birth and death can be easily detected by comparing the hash tables between the adjacent epochs.

Another non-trivial type of critical events is significant cluster splitting/merging over time. To extract this type of critical events, we first rank the splitting/merging events by using both the number of branches at the related time points and the entropy of the splitting/merging proportions. Then we select the ones with the largest ranking scores as the critical events. Mathematically, the ranking score of the merging event is formulated as:

$$\begin{aligned} R(r,t) &= |\mathcal{N}_r| \cdot H[P_t^{in}(s \to r)] \\ &= |\mathcal{N}_r| \cdot \kappa_B \textstyle\sum_{s \in \mathcal{N}_r} P_t^{in}(s \to r) \ln P_t^{in}(s \to r) \end{aligned} \tag{14}$$

where $R(r,t)$ is the ranking score of cluster r at time t, $H[\cdot]$ denote the entropy score of a distribution, and κ_B is the Boltzmann constant. $\mathcal{N}_r$ is the neighborhood set of cluster r. It consists of the branch clusters that flow into r. and $|\mathcal{N}_r|$ is the number of elements in $\mathcal{N}_r$. Similarly, the ranking score of the splitting event is defined as:

$$\begin{aligned} R(s,t) &= |\mathcal{N}_s| \cdot H[P_{t-1}^{out}(s \to r)] \\ &= |\mathcal{N}_s| \cdot \kappa_B \textstyle\sum_{r \in \mathcal{N}_s} P_{t-1}^{out}(s \to r) \ln P_{t-1}^{out}(s \to r) \end{aligned} \tag{15}$$

where $R(s,t)$ is the ranking score of cluster s at time t. $\mathcal{N}_s$ is the neighborhood set of cluster s; its elements are the branch clusters that flows out of s. $|\mathcal{N}_s|$ is the number of elements in $\mathcal{N}_s$.

In each cluster, the time point with more equivalent branches are more likely to be selected as a critical event (see Fig. 3).

B. Keyword Ranking

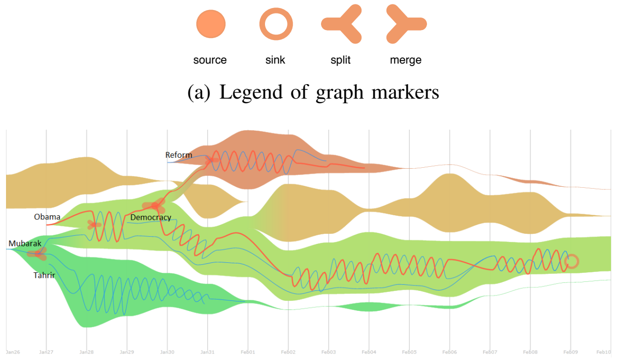
Previous study on keyword ranking methods [14], [15] has shown that the following two criteria are very useful in selecting the interesting keywords to represent the topic content at each time point. First, the keywords at each time should reflect distinctive content, thus we could observe the evolving and developing of the topic (distinctiveness). Second, the keyword sets along time together will cover the total content of the topic (completeness). In our work, we follow these two criteria and slightly modify them to adapt to our incremental batch-mode manner. The rank of a keyword w in cluster k at time t is given by:

$$\mathrm{Weight}(w)_k^t = \frac{\mathrm{TF}(w)_k^t}{\sum_k \mathrm{TF}(w)_k^t} \cdot \exp\left(-\lambda \cdot \mathrm{Weight}(w)_k^{t-1}\right) \tag{16}$$

where w represents a word, $\mathrm{TF}(w)_k^t$ is the term frequency of w in cluster k at time t, $\sum_k \mathrm{TF}(w)_k^t$ is the sum of $\mathrm{TF}(w)_k^t$ among different clusters, $\mathrm{Weight}(w)_k^{t-1}$ is the weight of w at time $t-1$, and λ is a coefficient which is set to 0.9 in our system. Note that $\exp\left(-\lambda \cdot \mathrm{Weight}(w)_k^{t-1}\right)$ can be regarded as a decay factor of each keyword. If a keyword appears at the last time point with a very high score, it will be ranked lower at the current time point. Contrarily, if it has not been shown before, it should be emphasized.

C. Keyword Connection Discovery

Although some text mining problems, such as document clustering and classification, can be solved by using the bag-of-words representation [16], the semantics in the text is still very important for further analyzing and understanding documents. We represent the semantics in terms of word co-occurrence in clusters at different time point. This provides



(a) Legend of graph markers

(b) Visualization of keywords threads and co-occurrence in topic splitting and merging. (Red thread represents principle selected keywords, and blue threads represent related keywords.)

Figure 3. Illustration of critical points and keywords threads.

an intuitive way to help users better understand the clustering results, as well as why the clusters are connected.

As shown in Fig. 3, each keyword is encoded by a thread evolving along the topic layer, and a bundle with height represents the co-occurrence frequency of the related words.

IV. Conclusion

In this paper, we focus on characterizing the relationships among clusters detected from text streams. We first incrementally derive clusters in text, then we connect the clusters using splitting and merging patterns. Next, we develop an incremental HDP Gibbs sampling algorithm to balance the significance of splitting and merging. Finally, to better understand why clusters split and merge, we provide a set of finer granular analysis methods. Specifically, we identify the critical events and show the co-occurrence of syntactic or semantic patterns on the trend of clusters. A visualization is also developed to help user easily interact with the analysis results and find interesting patterns. In the future, we will introduce more semantic information into the clustering results and make the text clusters more interpretable. Moreover, we would like to study corpora comparison using the techniques developed in this work.

Acknowledgment

References

[1] J. Allan, Ed., *Topic detection and tracking: event-based information organization*. Norwell, MA, USA: Kluwer Academic Publishers, 2002.

[2] K. Nigam, A. K. McCallum, S. Thrun, and T. M. Mitchell, “Text classification from labeled and unlabeled documents using EM,” *Machine Learning*, vol. 39, no. 2/3, pp. 103–134, 2000.

[3] D. Blei, A. Ng, M. Jordan, and J. Lafferty, “Latent Dirichlet allocation,” *Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2003.

[4] D. Chakrabarti, R. Kumar, and A. Tomkins, “Evolutionary clustering,” in *KDD*, 2006, pp. 554–560.

[5] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng, “Evolutionary spectral clustering by incorporating temporal smoothness,” in *KDD*, 2007, pp. 153–162.

[6] D. Blei and J. Lafferty, “Dynamic topic models,” in *ICML*, 2006, pp. 113–120.

[7] X. Wang and A. McCallum, “Topics over time: A non-Markov continuous-time model of topical trends,” in *KDD*, 2006, pp. 424–433.

[8] Y. Teh, M. Jordan, M. Beal, and D. Blei, “Hierarchical Dirichlet processes,” *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006.

[9] J. Zhang, Z. Ghahramani, and Y. Yang, “A probabilistic model for online document clustering with application to novelty detection,” in *NIPS*, L. K. Saul, Y. Weiss, and L. Bottou, Eds., 2005, pp. 1617–1624.

[10] Y. W. Teh, “Dirichlet processes,” in *Encyclopedia of Machine Learning*. Springer, 2010.

[11] J. Sethuraman, “A constructive definition of Dirichlet priors,” *Statistica Sinica*, vol. 4, no. 2, pp. 639–650, 1994.

[12] K. R. Canini, L. Shi, and T. L. Griffiths, “Online inference of topics with latent dirichlet allocation,” in *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

[13] M. D. Escobar and M. West, “Bayesian density estimation and inference using mixtures,” *Journal of the American Statistical Association*, vol. 90, pp. 577–588, 1995.

[14] D. Blei and J. Lafferty, *Topic Models*. Taylor and Francis, 2009, chapter: Topic Models, (in Press).

[15] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang, “TIARA: a visual exploratory text analytic system,” in *KDD*, 2010, pp. 153–162.

[16] M. W. Berry and J. Kogan, Eds., *Text Mining: Applications and Theory*. Hoboken, NJ: Wiley, 2010. [Online]. Available: http://www.kyb.tuebingen.mpg.de/ssl-book