Semi-Supervised Dirichlet-Hawkes Process with Applications of Topic Detection and Tracking in Twitter

Wanying Ding, Yue Zhang, Chaomei Chen, Xiaohua Hu
College of Computing & Informatics,Drexel University
Philadelphia, Pennsylvaina,USA
Email: wd78@drexel.edu, yz559@drexel.edu ,cc345@drexel.edu, xh29@drexel.edu

***Abstract*—Understanding ongoing topics and their evolutions in social media is of great importance. Although topic analysis is not a novel research question, social media environment has presented new challenges. First, with insufficient co-occurrence information, short text have undermined many word co-occurrence oriented topic models' applicability. Second, real time message streams make traditional discretized topic tracking methods hard to function. Third, topics' evolution mechanisms are of great importance in social media context, but many studies have ignored them. Forth, topics have more complicated correlation among each other. Considering the existing problems, this paper has proposed a Semi-Supervised Dirichlet-Hawkes Process (SDHP) to deal with topic detection and tracking from social media. The main contributions of this paper are reflected in: (1) SDHP can handle short text problem efficiently; (2) SDHP can track topics from continuous message stream; (3) SDHP can reveal topics' underlying evolution patterns; and (4) SDHP can capture topics' correlations We have evaluated SDHP's ability in both topic detection and tracking in 8 real datasets from Twitter, and the algorithm's performances are very promising.**

***Keywords*-Topic Detection and Tracking; Twitter; Hawkes Process; Dirichlet Process;**

I. Introduction

Social Media sites have taken over our lives. They create convenient communication channels to disseminate user generated content in real time. However, the amount of social media information is massive and the length of each message is short. Taking Twitter as an example, there are approximately 500 million tweets flowing across the service everyday[1], and each tweet is no longer than 140 characters. In face of this sheer mass and fragmentation of data, many individuals and organizations feel incapable to understand it, not to mention utilizing it. The purpose of this paper is to design a systematic framework to help users detect and track ongoing topics discussed in social media. To make a consistent statement, we use Twitter as an example.

For a long time, there have been many interests in developing text mining techniques for topic detection from large volumes of tweets[1], [2], [3], [4].Among them, Latent Dirichlet Allocation(LDA)[5] is most widely-used[6]. However, a question that usually arises when using LDA is how to choose a proper number of topics K, which needs to be pre-defined and will heavily affect final results[7]. *Dirichlet Process(DP)*[8] has been proposed to cope with this problem. In DP, K is obtained directly from the dataset instead of manually defined. Once been proposed, DP has and been widely applied in various text mining tasks[7], [9], [10]. But, one problem of DP is that it relies on words' co-occurrence to make clustering inference. As we stated before, many online messages are too short to present enough co-occurrence information to generate an accurate result.

Twitter, as well as many other social media platforms, delivers a continuous stream of messages, allowing topics to grow, fade, split and emerge over time. Many dynamic topic models have been created, but most of them are *Discretized Methods*[11], [12], [13], [6]. Discretized methods require to discretize continuous time stream into discrete time episodes, and apply topic models on each of them. Time granularity definition is always tricky for such methods. On the other hand, *Stochastic Methods*[14], [15], [16], [17] treat time as a continuous variable, providing a more flexible framework. Recently, combining topic models with stochastic process to cluster continuous document stream begins to attract researchers' attention[17], [18], [16]. But, these studies either are LDA oriented,or imply independence among topics, ignoring their semantic correlations.

This paper has created a Semi-Supervised Dirichlet Hawkes Process(SDHP) to detect and track topics online. Specifically, SDHP has made contributions in four aspects.(1) SDHP could handle with *Short Text*.(2) SDHP could track topics from *Continuous Text Stream*.(3) SDHP could reveal *Topic Correlations*.(4) SDHP could underlying *Topic Evolution Mechanism*.

II. Preliminaries

Before digging into technical details, we first give a brief introduction about some preliminary knowledge associated with our model.

A. Dirichlet Process

Generally, a Dirichlet Process(DP) is specified by a base distribution G and a positive real number α, which is

[1]http://www.internetlivestats.com/twitter-statistics/

978-1-4673-9005-7/16/$31.00 ©2016 IEEE

called *Concentration Parameter*.It can be illustrated as a Chinese Restaurant Process(CRP).Imaging a restaurant with an infinite number of tables, and customers walk in to sit around tables. The process can be described as:

1) The first customer will choose the first table.
2) For n^{th} customer:
   a) With probability $\pi_n = \frac{\alpha}{\alpha+n-1}$ to create a new table to sit.
   b) With probability $\pi_n = \frac{c_k}{a+n-1}$ to sit around table k, where c_k is the number of previous customers sitting around table k.

Mathematically, the whole generative process can be written as Equation(1), where $\delta(\cdot)$ is a mass function, and ϕ_k to describe the customer distribution over table k, G is the base distribution, α is the concentration parameter.

$$\phi_n|\phi_{1:n-1} \sim \sum_k \frac{c_k}{n-1+\alpha}\delta(\phi_k) + \frac{\alpha}{n-1+\alpha}G \quad (1)$$

The problem of DP oriented methods is that π_n is generated randomly without considering any context information or prior knowledge.This will lead to very poor results when ϕ_k could not provide enough clustering information, such as word co-occurrence.

B. Hawkes Process

Hawkes Process[20] is an important class of temporal point process, which is designed to utilize historical arrived points, $\boldsymbol{I_{t-}} = \{t_1, ..., t_i : t_i \leq t_{i+1}\}$, to predict the probability of an event occurring at time t, and usually this probability is denoted as $\lambda(t|\boldsymbol{I_{t-}})$. The distinguishing advantage of Hawkes Process is that:(1) it can model continuous arrival process; and (2)it can model different impacts that trigger an event in a unified framework. Assuming there are K events evolve at the same time, and the k^{th} event's evolution process can be formulated as Equation(2).

$$\begin{aligned}\lambda_k(t) =& \mu_k(t) + \sum_{t_{k,i}<t} \psi_{kk}(t - t_{k,i}) \\ &+ \sum_{k'=1,k'\neq k}^{K} \sum_{t_{k',j}<t} \psi_{kk'}(t - t_{k',j})\end{aligned} \quad (2)$$

where $\mu_k(t)$ is the *Background Function*, ψ_{kk} is the *Self-Excitation Function*, describing impacts coming from its own historical points, and $\psi_{kk'}$ is the *Cross-Excitation Function*, describing impacts coming from other processes. One deficiency of Hawkes Process is that it ignores the semantic relationship among different processes.

III. Model Description

Assuming that we have a tweet collection containing N tweets, and the n^{th} tweet is denoted as $d_n \sim d_n(\boldsymbol{w_n}, \boldsymbol{h_n}, t_n)$, where $\boldsymbol{w_n} =< w_{n1}, ...w_{nW} >$ represents the $W(W > 0)$ words contained in d_n, $\boldsymbol{h_n} =< h_{n1}, ..., h_{nH} >$ represents the $H(H \geq 0)$ hashtags d_n has, and t_n indicates the time stamp when this tweet is posted.

A. Topic Generation Process

Essentially, Dirichlet Process relies on word co-occurrence to cluster words, but tweets are too short to provide sufficient such information. This paper employs two kinds of supervision to facilitate topic detection from tweets: **Hashtag Supervision** and **Relevance Kernel Supervision**.

Hashtag Supervision: According to the presence of hashtags, we classify tweets into two categories: **Hashtaged Tweets**(tweets with hashtags) and **Free Tweets**(tweets without hashtags).We constrain that words in a hashtaged tweet can only distribute over its attached hashtags, while, words in free tweets can either distribute over existing hashtags, or generate new hashtags. Specifically, we have two hashtag sets in total, the original hashtag sets, denoted as $\boldsymbol{L}^*$, and generated hashtag sets, denoted as $\boldsymbol{L}^+$. The whole hashtag set $\boldsymbol{L}$ is the combination of $\boldsymbol{L}^*$ and $\boldsymbol{L}^+$, namely $\boldsymbol{L} = \boldsymbol{L}^* \cup \boldsymbol{L}^+$. For a hashtaged tweet d_h, we attach it with a binary hashtag presence/absence indicator vector $\boldsymbol{U^{d_h}} = (l_1, ..., l_u, ...l_{|\boldsymbol{L}^*|})$, where $|\boldsymbol{L}^*|$ indicates the length of $\boldsymbol{L}^*$. In $\boldsymbol{U^{d_h}}$, each $l_u \in \{0, 1\}$, where 1 indicates hashtag l_u presents, and 0 indicates hashtag l_u absents. Words in d_h can only distribute over hashtags with $l_u = 1$.On the contrary, a free tweet d_f can either distribute over any hashtags from $\boldsymbol{L}$ or generate a new hashtag.

Relevance Kernel Supervision: In HDP, the probability of a word assigning to a table is proportional to the word count of that table[21]. Such word assignment mechanism ignores semantic information. When documents are long enough, ϕ_k (shown in Equation(1)), which records word co-occurrence, can help to avoid random assignment. When text is short, the whole process tends to result in a random assignment. In order to ameliorate the assignment mechanism, SDHP replaces the count oriented assignment with "Relevance" oriented assignment by implementing BM25 as a kernel to measure relevance. BM25 is a famous scoring function used by search engines to rank documents by considering *Term Frequency*(TF), *Inverse Document Frequency*(IDF) and *Document Length*. Taking word-to-hashtag assignment as an example, we treat words as queries and hashtags as documents. The probability of one word w assigning to a certain hashtag h depends on their relevance score, which can be calculated as Equation(3), where $TF(w, h)$ refers the word frequency of w in hashtag h, $IDF(w)$ is the inverse document frequency of w, $aveLw$ refers to the average hashtag length, $|\boldsymbol{L}|$ is the number of hashtags, including both original hashtags and generated hashtags, and $|\boldsymbol{L(w)}|$ is the number of hashtags that contain word w. b and k are free constants here, and we set $b = 0.75$ and $k = 1.5$ by default.

$$\begin{aligned}Score_{BM25}(w, h) &= \frac{IDF(w) \cdot TF(w, h) \cdot (k + 1)}{TF(w, h) + k \cdot (1 - b + b \cdot \frac{|\boldsymbol{L}|}{avgLw})} \\ IDF(w) &= log\frac{|\boldsymbol{L}|}{|\boldsymbol{L(w)}|}\end{aligned} \quad (3)$$

Since we detect topics along time stream, *Time* needs to be considered. We use Equation(4) to lower the assignment probabilities between a word and a hashtag who are widely apart in time line.

$$score(w,h) = \frac{Score_{BM25}}{|t_w - t_h|} \tag{4}$$

where t_w represents word w's time stamp, and t_h represents hashtags h' time stamp, which is calculated as the average value of all words' time stamp values containing in that hashtag.

Generative Process for Topic Detection: The generative process of topic detection part in SDHP is shown as follows:
Step 1: Initiate the process:
(1) Define a baseline G to generate topics(To avoid unlimited topic growth, we define G as an exponential distribution).
(2) Draw a Hashtag-Topic Distribution $\boldsymbol{\theta}$ parametrized by β_θ.
(3) Draw a Word-Hashtag Distribution $\boldsymbol{\phi}$ parametrized by β_ϕ
Step 2: For each hashtag h_l in the hashtag set $\boldsymbol{L}$, draw a topic assignment k according to G_0, which is controlled by G:

$$G_0 : p(k_{h_l} = k) \propto \begin{cases} \dfrac{score(h_l,k)}{\sum_{k'=1}^{K} score(h_l,k') + \alpha_{topic}} \cdot \delta(\theta_k) \\ \dfrac{\alpha_{topic}}{\sum_{k'=1}^{K} score(h_l,k') + \alpha_{topic}} \cdot G \end{cases} \tag{5}$$

Step 3: For each tweet d_n in the dataset:
(3.1) If d_n is a hashtaged tweet:
(3.1.1)we first define a set$\boldsymbol{L^{d_n}} = \{l_i | l_i = 1, l_i \in \boldsymbol{U^{d_n}}\}$ to describe the hashtags d_n has.
(3.1.2) For each word w_{ni} draw a hashtag h from G_n:

$$G_n : p(h_{w_{ni}} = h) \propto \frac{score(w_{ni},h)}{\sum_{h' \in \boldsymbol{L^{d_n}}} score(w_{ni},h')} \cdot \delta(\phi_h) \tag{6}$$

(3.2)If d_n is a free tweet,its words $\boldsymbol{w_n}$ can distribute over all the hashtag collections $\boldsymbol{L}$. So for each word w_{ni}, draw a hashtag h from G_n, which is controlled by G_0 :

$$G_n : p(h_{w_{ni}} = h) \propto \begin{cases} \dfrac{score(w_{ni},h)}{\sum_{h' \in \boldsymbol{L}} score(w_{ni},h') + \alpha_{hashtag}} \cdot \delta(\phi_h) \\ \dfrac{\alpha_{hashtag}}{\sum_{h' \in \boldsymbol{L}} score(w_{ni},h') + \alpha_{hashtag}} \cdot G_0 \end{cases} \tag{7}$$

Step 4: Draw a word w_{ni} from ϕ_h
We use a Gibbs Sampling to iterate Step 2 Step 4 in the above process in order to optimize $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$

B. Topic Evolution Process

We implement a Multivariate Hawkes Process framework to simulate topic evolution process. In order to figure out how topics evolve over time, we first discuss three common scenarios where a tweet might be triggered.

(1) Background-Excitation.: Unexpected incidents, such as Brussels Attack, stimulate users to post tweets spontaneously and simultaneously.We use a *Background Function* $\mu_k(t)$ (shown in Equation(2)) to describe such phenomenon. In real situations, The background effect usually rises to a peak quickly, and then decays gradually. We use a Rayleigh Distribution(shown as Equation(8)) to model background effect, where ω is its parameter.

$$\mu(t) = \frac{t}{\omega^2} e^{-\frac{t^2}{\omega^2}} \tag{8}$$

(2) Self-Excitation.: According to "Matthew Effect", a hot topic will always intrigue more people to discuss it.We use a *Self-Excitation Function* $\psi_{kk}(t)$(shown in Equation(2)) to describe such phenomenon.In this paper, we use a Log-Normal distribution to model this function. The reason is that Log-Normal distribution is various shaped to model the variety of self-excitation. We use $s(t)$ to indicate the self-excitation function, which is formulated as Equation (9), where μ and σ are parameters.

$$s(t) = \frac{1}{t\sigma\sqrt{2\pi}} e^{-\frac{(lnt-\mu)^2}{2\sigma^2}} \tag{9}$$

(3) Cross-Excitation.: Discussion about one topic will stimulate discussion about the other one. We use a *Cross-Excitation Function* $\psi_{kk'}(t)$ (shown in Equation (2)) to describe such phenomenon. We apply the decreasing Exponential Distribution to describe Cross-Excitation Function. We use $c(t)$ to represent the cross-excitation function, and it can be formulated as Equation(10), where κ is its parameter.

$$c(t) = \kappa e^{-\kappa t} \tag{10}$$

Similarity Kernel Supervision: Hawkes Process only uses the distance between time points to simulate the relationships among dimensions while ignoring semantics. Tweets have semantics. Although two tweets may be published close in time, the triggering probability could still be low if they belong to two irrelevant topics. Considering this, we insert a Cosine Kernel in cross-excitation functions tuning triggering probabilities among different topics. Assume $t_{A,i}$ indicates the i^{th} time point in topic A, and $t_{B,j}$ indicates the j^{th} time point in topic B. The semantic similarity between A and B is measured by a Cosine Function(V indicates the vocabulary), shown as Equation(11), and the distance between $d_{A,i}$ and $d_{B,j}$ can be calculated as Equation(12).

$$sim(A,B) = \frac{\sum_{v=1}^{V} A_v B_v}{\sqrt{\sum_{v=1}^{V} A_v^2}\sqrt{\sum_{v=1}^{V} B_v^2}} \tag{11}$$

$$dis(t_{A,i}, t_{B,j}) = \frac{sim(A,B)}{|t_i - t_j|} \tag{12}$$

Thus cross-excitation function can be re-written as:

$$c(t_{A,i}, t_{B,j}) = \kappa e^{-\kappa \times dis(t_{A,i}, t_{B,j})} \tag{13}$$

Assume the i^{th} word in the n^{th} tweet has been assigned to topic k, and t_{ni} is the r^{th} time point ordered in topic k, the complete intensity rate function to trigger t_{ni} can be written

as follows:

$$\lambda_k(t_r) = \eta_k^B \frac{t}{\omega_k^2} e^{-\frac{t^2}{\omega_k^2}} + \eta_k^S \sum_{t_j<t_r} \frac{1}{(t_r - t_j)\sigma_k\sqrt{2\pi}} e^{-\frac{(ln(t_r-t_j)-\mu_k)^2}{2\sigma_k^2}} + \sum_{k'=1,k'\neq k}^{C} \eta_{kk'}^C \sum_{t_{k',l}<t_{k,r}} \kappa_{kk'} e^{-\kappa_{kk'} \times dis(t_{k,r},t_{k',l})} \tag{14}$$

Parameter Inference: We use an EM algorithm to infer parameters in Equation(14). For the r^{th} time point in topic k, denoting as $t_{k,r}$, we create a vector $\boldsymbol{Z_{t_{k,r}}} = (Z_{k,0,0}, Z_{k,k,1}, Z_{k,k,2}, ..., Z_{k,k,N_k}, Z_{k,k',1}..Z_{k,k',N_{k'}}...)$ to indicate its affiliation. If $Z_{k,k',i} = 1$ indicates $t_{k,r}$ is evoked by the i^{th} time point in the k'^{th} topic, and $Z_{k,0,0} = 1$ indicates that this time point is triggered by background.

In the **E** step, the posterior probability of Z can be obtained from Bayes' rule. The posterior probability of that $t_{k,r}$ is triggered by the i'^{th} in the k'^{th} process can be calculated as:

$$p(Z_{k,k',i} = 1|\boldsymbol{\Pi}, \boldsymbol{I}) = \frac{\lambda_{Z_{k',i}} t(k,r)}{\sum_{m\in \boldsymbol{Z_{t_{k,r}}}} \lambda_m(t_{k,r})} \tag{15}$$

where $\boldsymbol{\Pi}$ indicates all the parameters contained in the whole process, and $\boldsymbol{I}$ indicates the preceding time points.

In the **M** step, we have simplified $\Lambda_z(T)$ with assumption from [22], [23], and our final log-likelihood function over time period $[0, T]$ can be written as Equation(16).

$$Q_k(\Pi_k) = \sum_{Z in Z_k} \sum_{r=1}^{N_k} (ln(\eta_k^B \mu(t))) \delta(Z_{k,0,0} = 1) + \sum_{t_{k,j}<t_{k,r}} ln(\eta_k^S s(t_{k,r} - t_{k,j})) \delta(Z_{k,k,j} = 1) + \sum_{k'=1,k'\neq k}^{K} \sum_{t_{k'l}<t_{k,r}} ln(\eta_{kk'} c(t_{k,r}, t_{k',l})) \delta(Z_{k,k',l} = 1) - \eta_k^B - \eta_k^S \times N_k - \sum_{k'=1,k'\neq k}^{K} \eta_{kk'}^C \times N_{k'} \tag{16}$$

By taking derivatives of Equation(16), we can get each parameter's optimized value.

IV. Evaluation

A. Data Set

Since there is no proper ground truth available for us to conduct evaluations, we collect data through Twitter Search API[2]. We choose 8 popular events as our research subjects.The detailed data information is shown as in Table 1. Twitter data are noisy, so before we conduct evaluations and experiments, we made some data pre-processes, including:

[2]https://dev.twitter.com/rest/public/search

Table I
Data Set Description

Event	Time Span	Count
AlphaGo	20160316-20160401	13099
Batman v Superman	20160323-20160330	565418
Zootopia	20160317 - 20160401	110163
Iphone SE	20160316-20160331	112070
Kobe	20160416-20160502	208766
Brussels Attack	20160318 -20160326	886964
Hilary	20160426 - 20160505	56358
Trump	20160428 -20160505	708348

(1) **Abbreviation Process**. According to a dictionary[3], we replace these abbreviations with their original phrase.
(2) **Repeat Word Process**. We convert words, like looooooooove to their original from.
(3) **Emoticons Process**. We transform emoticons, such as "O:-)" into words by referring an emoticons dictionary[4].
(4) **Stop Word Process**.Words occur in more than 10% tweets or less than 100 times as stop words are treated as stopwords, and removed.

B. Evaluation

Topic Detection Evaluation: The measurement traditionally used for topic model evaluation is **Perplexity** The formulation of perplexity calculation is shown as Equation(17).

$$perplexty(\boldsymbol{w}) = exp\{-\frac{\sum_d logp(\boldsymbol{v}^d|\Phi, \theta)}{countsoftokens}\} \tag{17}$$

where $\boldsymbol{v}$ is the words in test dataset, $\boldsymbol{\Phi}$ is the trained model and $\boldsymbol{\theta}$ indicates the parameters.

We want to check whether our proposed Hashtag Supervision and Relevance Kernel Supervision can help to improve model's performance in topic detection from short text. We compare three models:
(1)**Traditional HDP (TP)**. Traditional HDP clusters words into topics without any supervision.
(2) **Hashtag Supervised HDP (HP)**. We only add hashtag supervision(detailed in Section 3.1) on traditional HDP, and it still utilize count oriented word assignment.
(3) **SDHP(KP)**. We further add relevance kernel supervision on HP, and get our SDHP model, which has been elaborated in Section 3.1

In addition to the models, models' parameters may also affect final results. In Dirichlet Process, the only influential parameter is its concentration parameter. In HP and KP, their concentration parameters are α_{hash}, which controls hashtag generation, and α_{topic}, which controls topic generation. In TP, we treat its first level concentration parameter, which controls table generation, as α_{hash}, and the second level concentration parameter, which controls topic generation,

[3]https://blog.bufferapp.com/social-media-acronyms-abbreviations
[4]http://textmeanings.com/emoticons/

as α_{topic}. We vary the values of α_{topic} and α_{hash} through $\{0.1, 0.2, 0.5, 0.8, 1, 2, 5, 8, 10\}$ respectively, and conduct a 5-fold cross validation on each of our datasets with each pairs of α_{topic} and α_{hash}.

Since perplexity varies dramatically in different datasets. In order to present a concise comparison result, we calculate a p-score (shown in Equation(18)) to measure the improvement percentage achieved by HP or KP when compared to TP.

$$p - score = \frac{(TP_{ppx} - model_{ppx})}{TP_{ppx}} \tag{18}$$

TP_{ppx} indicates TP's perplexity value, and we replace $model_{ppx}$ with HP's perplexity and KP's perplexity respectively.

Figure 1(a) shows perplexity variance with α_{hash} goes through $\{0.1, 0.2, 0.5, 0.8, 1, 2, 5, 8, 10\}$ and α_{topic} with value 1,and Figure 1(b) shows perplexity variance with α_{topic} goes through $\{0.1, 0.2, 0.5, 0.8, 1, 2, 5, 8, 10\}$ with α_{hash} set to be 1. All p-scores are larger than 0, indicating hashtag supervision indeed helps in improving model's performance in topic detection. Comparing HP and KP, we find that on the same dataset, KP always performs better than HP. This result implies that a kernel oriented assignment mechanism has imposed a positive effect on the model's performance.

From Figure1(a) and (b), we also find that concentration parameters have very small impacts on perplexity values. Even so, if we set their values too small, the model will lose the ability to generate new topics, and if we set their values too large, the model will generate overwhelming number of topics. Thus, although concentration parameters have very little impact on perplexity value, it still requires users to select a proper concentration parameter value in order to get an acceptable generated topic number.

Topic Evolution Evaluation: We use plain log-likelihood to evaluate the SDHP's performance in topic tracking. Again, we want to check whether our adaption on Hawkes Process have improved the model's performance. Therefore, we designed four models to compare:

(1)**Plain Hawkes Process (PPL)**. PPL assumes the background effect is a constant, and use Exponential distributions to model both self-excitation and cross-excitation processes.

(2) **Background Hawkes Process(BGL)**. BGL relaxes the constant background effect assumption, and implements a Rayleigh distribution to model background effect. But it still use Exponential distribution to model both self-excitation and cross-excitation processes.

(3) **Triple Hawkes Process (TPL)**. Based on BGL, TPL differentiates self-excitation function from cross-excitation functions. TPL utilizes a Log-Normal Distribution to model self-excitation process.

(4) **SDHP (KPL)**. Further, KPL imposes similarity kernels on TPL to adjust the triggering relationship among events from different topics. KPL has been detailed in Section 3.2.

Here, we define N as the number of preceding events that allows to trigger the current one, and we change N value through $\{1, 5, 10, 50, 100, 500, 1000, 500, 1000\}$ to check four models' log-likelihood. For each N, we conduct a 5 cross-validation on all the dataset respectively. Again, the log-likelihood value varies dramatically, and we also apply p-score to calculate the performance improvement achieved by the models.

$$p - score = \frac{(model_{lhd} - PPL_{lhd})}{PPL_{lhd}} \tag{19}$$

where PPL_{lhd} indicates PPL's log-likelihood, and we replace $model_{lhd}$ with log-likelihood value of BGL, TPL, and KPL. Figure 1(c) presents the comparison result.

In Figure 1(c), we find that when N value is small, both BGL and TPL provide no improvement compared to PPL. What is worse, when N is smaller than 1000, BGL presents a worse performance. However, with N grows, BGL's performance becomes better. This result indicates that, within a short period of time, background effect should be assumed as a constant, but if we want to analyze a long-term evolution, it is better to assume background effects as a decreasing function. Compared to BPL, TPL's performances are always better at each dataset. The result indicates that self-excitation has a very different evolution pattern form cross-excitation, and they should be treated differently. KPL always performs the best, and improves PPL's performance dramatically. This means a similarity kernel supervision is necessary when we want to simulate a multi-dimension topic evolutions.

V. Conclusion

This paper has created a complete framework for topic detection, tracking and visualization from Twitter. This paper has proposed a SDHP model to realize topic detection and tracking from continuous short text streams. It exerts two kinds of supervision, Hashtag Supervision and Relevance Kernel Supervision to assist topic detection. Evaluation results show that both of them can provide positive effects on model's performance. In addition, SDHP has implemented a Multivariate Hawkes Process to simulate topic evolutions over time. SDHP has impose a Similarity Kernel Supervision to adjust the triggering relationships among different topics. Evaluation results show that SKDH achieves very promising results. Finally, this paper has created a decent visualization to present a concise view about our analyzed results to users.

References

[1] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li, "Comparing twitter and traditional media using topic models," in *Advances in Information Retrieval*. Springer, 2011, pp. 338–349.

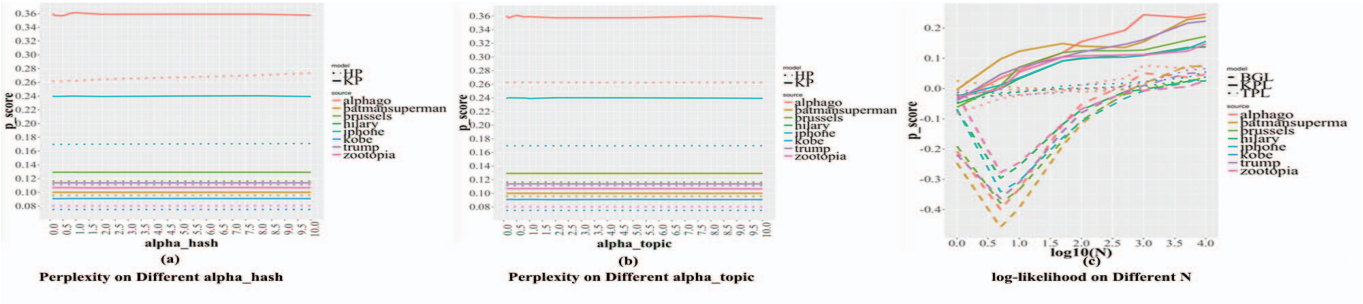

Figure 1. Evaluation Results

[2] D. Yajuan, C. Zhimin, W. Furu, Z. Ming, and H.-Y. Shum, "Twitter topic summarization by ranking tweets using social influence and content quality," in *Proceedings of the 24th International Conference on Computational Linguistics*, 2012, pp. 763–780.

[3] R. Zhang, W. Li, D. Gao, and Y. Ouyang, "Automatic twitter topic summarization with speech acts," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 3, pp. 649–658, 2013.

[4] L. Hong and B. D. Davison, "Empirical study of topic modeling in twitter," in *Proceedings of the first workshop on social media analytics*. ACM, 2010, pp. 80–88.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[6] X. Fu, J. Li, K. Yang, L. Cui, and L. Yang, "Dynamic online hdp model for discovering evolutionary topics from chinese social texts," *Neurocomputing*, vol. 171, pp. 412–424, 2016.

[7] E. Zavitsanos, G. Paliouras, and G. A. Vouros, "Non-parametric estimation of topic hierarchies from texts with hierarchical dirichlet processes," *The Journal of Machine Learning Research*, vol. 12, pp. 2749–2775, 2011.

[8] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the american statistical association*, 2012.

[9] A. Ahmed and E. P. Xing, "Timeline: A dynamic hierarchical dirichlet process model for recovering birth/death and evolution of topics in text stream," *arXiv preprint arXiv:1203.3463*, 2012.

[10] T. Xu, Z. Zhang, P. S. Yu, and B. Long, "Evolutionary clustering by hierarchical dirichlet process with hidden markov state," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 2008, pp. 658–667.

[11] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 113–120.

[12] C. Wang, D. Blei, and D. Heckerman, "Continuous time dynamic topic models," *arXiv preprint arXiv:1206.3298*, 2012.

[13] A. Ahmed and E. P. Xing, "Dynamic non-parametric mixture models and the recurrent chinese restaurant process: with applications to evolutionary clustering." in *SDM*. SIAM, 2008, pp. 219–230.

[14] X. Wang and A. McCallum, "Topics over time: a non-markov continuous-time model of topical trends," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 424–433.

[15] A. Dubey, A. Hefny, S. Williamson, and E. P. Xing, "A nonparametric mixture model for topic modeling over time." in *SDM*. SIAM, 2013, pp. 530–538.

[16] N. Du, M. Farajtabar, A. Ahmed, A. J. Smola, and L. Song, "Dirichlet-hawkes processes with applications to clustering continuous-time document streams," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 219–228.

[17] X. He, T. Rekatsinas, J. Foulds, L. Getoor, and Y. Liu, "Hawkestopic: A joint model for network inference and topic modeling from text-based cascades," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 871–880.

[18] J. C. L. Pinto and T. Chahed, "Modeling multi-topic information diffusion in social networks using latent dirichlet allocation and hawkes processes," in *Signal-Image Technology and Internet-Based Systems (SITIS), 2014 Tenth International Conference on*. IEEE, 2014, pp. 339–346.

[19] T. S. Ferguson, "A bayesian analysis of some nonparametric problems," *The annals of statistics*, pp. 209–230, 1973.

[20] A. G. Hawkes, "Spectra of some self-exciting and mutually exciting point processes," *Biometrika*, vol. 58, no. 1, pp. 83–90, 1971.

[21] D. M. Blei and P. I. Frazier, "Distance dependent chinese restaurant processes," *The Journal of Machine Learning Research*, vol. 12, pp. 2461–2488, 2011.

[22] P. F. Halpin and P. De Boeck, "Modelling dyadic interaction with hawkes processes," *Psychometrika*, vol. 78, no. 4, pp. 793–814, 2013.

[23] J. F. Olson and K. M. Carley, "Exact and approximate em estimation of mutually exciting hawkes processes," *Statistical Inference for Stochastic Processes*, vol. 16, no. 1, pp. 63–80, 2013.