

# Analyzing topics and authors in chat logs for crime investigation

Abdur Rahman M. A. Basher · Benjamin C. M. Fung

Received: 1 March 2012 / Revised: 28 January 2013 / Accepted: 17 February 2013 /  
Published online: 8 March 2013  
© Springer-Verlag London 2013

**Abstract** Cybercriminals have been using the Internet to accomplish illegitimate activities and to execute catastrophic attacks. Computer-Mediated Communication such as online chat provides an anonymous channel for predators to exploit victims. In order to prosecute criminals in a court of law, an investigator often needs to extract evidence from a large volume of chat messages. Most of the existing search tools are keyword-based, and the search terms are provided by an investigator. The quality of the retrieved results depends on the search terms provided. Due to the large volume of chat messages and the large number of participants in public chat rooms, the process is often time-consuming and error-prone. This paper presents a topic search model to analyze archives of chat logs for segregating crime-relevant logs from others. Specifically, we propose an extension of the Latent Dirichlet Allocation-based model to extract topics, compute the contribution of authors in these topics, and study the transitions of these topics over time. In addition, we present a special model for characterizing authors-topics over time. This is crucial for investigation because it provides a view of the activity in which authors are involved in certain topics. Experiments on two real-life datasets suggest that the proposed approach can discover hidden criminal topics and the distribution of authors to these topics.

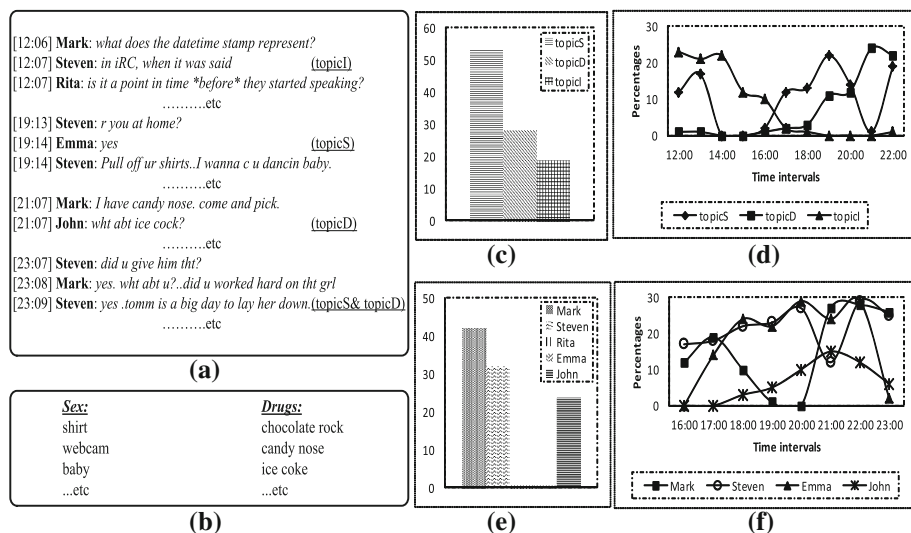
**Keywords** Latent Dirichlet Allocation (LDA) · Topic modeling · Gibbs sampling · Topic evolution · Author-topics over time · Cybercrime

## 1 Introduction

Demand for Computer-Mediated Communication, such as online chat, instant messages, blogs, and tweets, is growing tremendously, and many software applications have been developed to serve this demand. Instant messaging seems to be preferred, especially chatting,

---

A. R. M. A. Basher · B. C. M. Fung (✉)  
Concordia Institute for Information Systems Engineering, Concordia University, Montreal,  
QC H3G 1M8, Canada  
e-mail: fung@ciise.concordia.ca



**Fig. 1** **a** A detained chat log *d*. **b** Criminal topics (Sex and Drugs) with their associated terms. **c** Topics distribution in the chat log *d*. **d** Topics over time in the chat log *d*. **e** Authors distribution over topic D in the chat log *d*. **f** Authors-Topics over time in the chat log *d* for topic S

because it provides one-to-one or one-to-many instant communication and can handle video and audio calls as well. However, the widespread use of communication applications increases both legitimate and illegitimate activities. Illegitimate activities include cyber bullying, cyber drug trafficking, child pornography, and cyber sexual harassment. Traditional crimes that are conducted through the Internet pose new challenges for law enforcement agencies to prevent, detect, investigate, and prosecute perpetrators. Unfortunately, the capability of current crime-investigation software tools does not fully meet the actual needs of real-life investigations.

In many cases, an investigator seizes a suspect's computer that has an enormous amount of chat logs from, e.g., Windows Live Messenger or IRC chat rooms. The chat logs sometimes contain important information that is directly or indirectly related to the criminal activities under investigation. Figure 1a presents a general form of a chat log that contains information about criminal activities, such as Sex and Drugs.

The challenge is how to effectively and efficiently extract relevant information and evidence from a large volume of chat messages. In this paper, we propose a discovery method, in a context of chat log-topic and topic-author relations, to answer the following questions that are frequently raised by investigators:

**Q 1.** How can an investigator determine which logs are crime-relevant? In identifying a crime-relevant log, what are the contributed topics in the log file? How have they evolved over time? Moreover, how can an investigator extract the crime-relevant topics from the identified crime-relevant log files?

**Q 2.** Who are the contributors to a topic in a given chat log? How can an investigator track the activity of authors in a log file?

In general, we are concerned with generating Fig. 1c–f as results from our research questions. We would like to emphasize that the existing topics discovery methods [1–4] cannot be directly applied to address the problem illustrated in the context of crime investigation

due to the differences in characteristics of chat messages from traditional, well-structured documents [5]:

- Chat is informal and its content is not well structured. Chat often contains spoken languages with a lot of grammatical and spelling mistakes.
- The contents (topics) in chat logs change frequently and implicitly over time as consequences of incoherence of message sequences.
- Messages on chat logs are often short, ranging from a few words to a few lines.
- Transliteration is often used and refers to writing, words, or letters in a language written in a different alphabet or script.
- Authors within these short messages use many deceptive techniques for covert communication. For example, they use emoticons to express human facial behavior that complements a text message. Moreover, the semantic of the words used within a chat log may be different from their apparent meaning; street terms are more frequently used in the context of illegitimate activities. For example, the word “snow” used in drug trafficking means cocaine.

As a result, criminal-topics extraction from log files requires special handling, and the analytical techniques widely used for mining texts of literary and historic documents may not achieve the same accuracy when applied to online documents. Furthermore, these techniques do not collect information about authors composing criminal topics.

In this paper, we introduce a method for forensics investigators to precisely capture various characteristics of chat logs. Our study focuses on the first three aforementioned differences. Although some recent papers as Banerjee and Agarwal [6] discusses the prediction of the future behavior of a large population after modeling the collective behavior from observed data using swarm intelligence during the training phase, our study focuses on the first three aforementioned differences and the last two differences will be addressed in our future work.

Our proposed method has four phases: searching crime-relevant logs, discovering crime-relevant topics from identified criminal logs, estimating the contribution of authors in the discovered topics, and representing transitions of the crime-relevant topics over time. We first identify whether or not a given chat log is crime-relevant based on the predefined criminal topics. Then, we deploy a probabilistic topic model to extract the hidden semantic from the crime-relevant chat logs. Next, the authors' contributions within the discovered topics are estimated. Finally, an evolution of topics under some specific time intervals is generated. In certain cases, investigators want to distinguish some authors from others within a period of time. This is achieved by including another phase to compute the bond relationship composed of authors-topics trends over time.

**Contributions:** The contributions of this paper can be summarized as follows: First, we present a *Criminal Topic* model to identify crime-relevant chat logs. Second, we propose two topic models, namely *LDA-Topics over Time (LDA-TOT)* and *Author-Topics over Time (A-TOT)*, to extract criminal topics, the distributions of authors with topics, temporal information in these topics, and author-topic relationships within a period of time. We also propose a distance measure to group authors according to their activity.

The rest of the paper is organized as follows: In Sect. 2, we discuss the related work. In Sect. 3, we formally define the problem. In Sect. 4, we describe the background information relevant to our proposed methodology. In Sect. 5, we present our proposed method. In Sect. 6, we evaluate the effectiveness, efficiency, and scalability of our proposed approach on real-life crime-related datasets. Finally, we conclude with discussion and present possible future research directions in Sect. 7.

## 2 Related work

We summarize the state of the art in the literature of topics discovery and modeling. Blei et al. [1] proposed the *Latent Dirichlet Allocation (LDA)* model to extract topics and summarize a document corpus. The general idea of LDA is to generate a discrete distribution of words per topic and a discrete distribution over topics per document. Although LDA is expressive enough to reveal topics in a document, it does not provide a way of including labels in its learning procedure. Hence, LDA has been adapted in applications for topic labeling, as in [4, 7–9]. Blei et al. [7] proposed *Supervised LDA (sLDA)*, where a label is generated from the empirical topic mixture distribution of each document. Lacoste et al. [8] proposed *Discriminate variation on Latent Dirichlet Allocation (DiscLDA)*, where a document is related to a categorical variable or class label, and a topic mixture distribution is associated with each label. However, these models use single labeling to a document and do not provide multiple labels to each document. *Multi-Multinomial LDA (MM-LDA)* [9] assigns multiple labels for each document. Unfortunately, the topics learned by MM-LDA do not link directly with the label. Therefore, Ramage et al. [4] proposed the *Labeled-LDA (L-LDA)* model to directly associate the label set of each observed document with one topic.

*Prior-LDA* [10] introduces an approach to address the variations in label frequencies and the *Dependency-LDA* [10] extends the Prior-LDA to interpret the different dependencies between the labels.

Our way to solve topics labeling is by introducing a *Criminal Topic* model that includes predefined terms, with their distributions associated with each criminal topic. The discovered topics are labeled as crime-relevant whenever the distributions of these topics and topics from the Criminal-Topic model are assumed to be relevant through some distance measurement.

Several extensions of LDA models have been proposed to identify authors and the proportion of each author in a document. For example, Rosen-Zvi et al. [2] introduced an *Author-Topic (AT)* model, a generative model for authors and their corresponding topic distributions. In their experiments, AT seems to outperform LDA when the test documents contain few observed words. Other works have been extended further to deduce the social networks between entities in different types of documents [3, 11]. Chang et al. [11] presented a probabilistic topic model to describe the relationships between pairs of entities encoded in a collection of text documents. McCallum et al. [3] proposed a *Group-Topic (GT)* model to cluster entities into groups with relations between them. In their model, the discovery of groups is guided by the emerging topics and the discovery of topics is guided by emerging groups. In addition, the model is able to capture the language attributes being used within entities and this helps to assign group memberships. Their experimental results suggest that the inference of joint probability improves the performance of both groups and topics discovery. Song et al. [12] proposed a method called *CommunityNet* to predict the behavior of authors in receiving and sending information by analyzing the contact and content of personal communications. In our approach, we modify the AT model to accommodate the evolution of topics discovered and the proportion of authors to these topics over time.

Studying the evolution of topics over time is valuable, because it reveals different characteristics of topics and their authors. Wang et al. [13] proposed the *Topics Over Time (TOT)* model, a non-Markov continuous time model of topical transitions. TOT models time stamps by parameterizing a continuous beta distribution over time with each topic. They assume that the meaning of a particular topic can be relied upon as constant, but its occurrence and correlations change significantly over time. The *Continuous Time Dynamic Topic Model (cDTM)* [14] replaces the discrete state space model of the DTM [15] with its continuous form, called *Brownian motion*. The topics are modeled through a sequential collection of

documents, where a topic is a pattern of word use that is expected to change over the course of the collection. Significantly, *cDTM* generalizes the *DTM* in that the only discretization it models is the resolution at which the time stamps of the documents are measured. AlSumait et al. [16] used an online version of the LDA model (*OLDA*), where topics are evolved through incremental updates for new data based on the current position. On the other hand, the *Sequential latent Dirichlet allocation (SeqLDA)* [17] captures the topic evolutions by detecting how topic distributions vary among segments, which can be a chapter, section, paragraph, or sentence. This model applies the sequential structure of each document, which is the position of each segment, that the LDA model ignores. Although this model considers the document structure in the hierarchical modeling, it does not account the time intervals in the documents. To collect the distribution of topics over time, we employed an extended combination of three models (LDA, AT, and TOT) where discretization of time slots is used, since the time intervals in a chat log are relatively short, from a few minutes to a few hours.

The topic models discussed in most of the current literature are applied to structured documents, which are quite different from chat logs. As a result, it becomes very difficult to obtain an accurate model from logs. Hong et al. [5] focused on online messages, particularly Twitter. They conducted an empirical study of different strategies to aggregate tweets, based on the existing models. In contrast, our work focuses on four major aspects: criminal-topics discovery, authors' proportions with respect to topics, evolution of topics with respect to time, and evolution of authors-topics over time.

### 3 Problem definition

In this paper, we assume the user of our method is a crime investigator who has access to a collection of chat log documents and who would like to analyze the relationship between the topics discussed and the participating authors. We formally define an abstract representation of chat log documents, user-specified criminal topics, and some basic notions of topics and authors, followed by a problem statement.

**Definition 1** (*Chat log document*) A chat message is a triplet  $(a, \mu, \tau)$ , representing a textual message  $\mu$  written by author  $a$  at time  $\tau$ . A chat log document, denoted by  $d$ , is a sequence of chat messages ordered by  $\tau$ .

*Example 1* In Fig. 1a, Mark wrote the text message “I have candy nose. come and pick.” at time [21:07]. This chat message is represented by a triplet (Mark, “I have candy nose. come and pick.”, [21:07]). The chat log document is a sequence of chat messages ordered by time.

The following definition formally describes the notion of topic in a chat log document.

**Definition 2** (*Topics in chat log document*) Let  $T$  be the universe of topics. Let  $D$  be a set of chat log documents. A chat log document  $d \in D$  contains a set of topics  $T_d \subseteq T$ . A topic  $t \in T_d$  is a probability distribution over a set of vocabularies  $V$ . Specifically, the probability distribution of a topic  $t$  is a collection of positive real numbers over  $V$  with sum equal to 1.

An investigator wants to identify the crime-relevant topics discussed in a chat log document and the authors participated in the discussion of the topics. Thus, the set of vocabularies  $V$  consists of  $V_D \cup W_c$ , where  $V_D$  is a set of distinct words in  $D$  and  $W_c$  is a set of crime-related words described in the following definition.

**Definition 3** (*Crime-relevant topic*) Let  $C \subseteq T$  be a set of investigator-specified criminal topics. A criminal topic  $c \in C$  consists of a set of crime-related words  $W_c$ . Let  $distance(t_1, t_2)$  be a function that describes the dissimilarity of two topics  $t_1$  and  $t_2$ . A topic  $t$  is *relevant* to a criminal topic  $c$  if  $distance(t, c) \leq \gamma$ , where  $\gamma$  is a threshold specified by an investigator.

*Example 2* The chat log document  $d$  in Fig. 1a contains three topics  $T_d = \{topicI, topicS, topicD\}$ . Figure 1b illustrates two investigator-specified criminal topics  $C = \{Sex, Drugs\}$ . Suppose  $\gamma = 0.7$  and  $distance(topicD, Drugs) = 0.56$ . The  $topicD$ , discussed in  $d$ , is relevant to the criminal topic  $Drugs$  if  $distance(topicD, Drugs) \leq \gamma$ . The  $distance$  function will be defined in Sect. 6.

To identify relevant criminal information from a large collection of chat log documents, an investigator first has to identify the crime-relevant documents, and then the topics' distribution with respect to authors over time. The following definitions formally capture these notions.

**Definition 4** (*Crime-relevant document*) A chat log document  $d$  is *crime-relevant* if  $d$  contains at least one crime-relevant topic such that  $T_d \cap C = \{c_i \mid c_i \in C\}$ , where  $T_d$  and  $C$  are defined in Definitions 2 and 3, respectively.

**Definition 5** (*Active topic*) Let  $[\tau_i^s, \tau_i^f]$  be a time interval of topic  $t$  discussed in a chat log document. The *active* level of  $t$  over the time interval  $[\tau_i^s, \tau_i^f]$  is described by function  $F(t)_{\tau_i^s}^{\tau_i^f}$ .

An investigator can define his/her own instantiation of function  $F$ . One instantiation is given in Sect. 6.3.

**Definition 6** (*Active author*) Let  $\Lambda_d$  be a set of authors participating in chat log document  $d$ . Let  $\Lambda_d^t$  be a set of authors participating in a topic  $t$  in  $d$ , where  $\Lambda_d^t \subseteq \Lambda_d$ . The *active* level of an author  $a^t \in \Lambda_d^t$  is defined by  $F(a_d^t)_{\tau_i^s}^{\tau_i^f}$  provided  $t$  is active during  $[\tau_i^s, \tau_i^f]$ .

*Example 3* Figure 1d depicts the active levels of  $topicI$ ,  $topicS$ , and  $topicD$  between 12:00 and 22:00. For example,  $topicD$  is actively discussed between 20:00 and 22:00, but is relatively inactive between 12:00 and 13:00. Figure 1f defines the evolution (active level) of authors over the previous time intervals.

The problems studied in this paper are formally defined as follows:

**Definition 7** (*Authors-Criminal topics activity over time in a chat log*) Given a collection of chat log documents  $D$ , a set of criminal topics  $C$ , and a relevance threshold  $\gamma$ , the problems are as follows:

1. to identify all crime-relevant documents from  $D$ ,
2. to identify all crime-relevant topics in each document  $d \in D$  with respect to  $C$  and  $\gamma$ , and
3. to identify the active level of crime-relevant topics, and all their associated active authors over a given time interval  $[\tau_i^s, \tau_i^f]$  for each identified crime-relevant document.

## 4 Background information

Language modeling [18], a probability distribution over word sequences, provides a sound theoretical foundation to our research problem. The procedure first builds a probabilistic

**Table 1** Notations used in this paper

| Symbol         | Description                                                                          |
|----------------|--------------------------------------------------------------------------------------|
| $\alpha$       | Dirichlet parameters for topics (Dirichlet prior)                                    |
| $\bar{\alpha}$ | Dirichlet parameters for authors (Dirichlet prior)                                   |
| $\beta$        | Topic-dependent Dirichlet parameters for word index (Dirichlet prior)                |
| $\lambda$      | Topic-dependent Dirichlet parameters for time slots (Dirichlet prior)                |
| $\theta$       | Multinomial distribution of topics given the documents in the corpus                 |
| $\vartheta$    | Multinomial distribution of topics given the authors for the documents in the corpus |
| $\varphi$      | Multinomial distribution of words to topics                                          |
| $\eta$         | Multinomial distribution of time intervals to topics for the documents in the corpus |
| $D$            | Set of chat log documents                                                            |
| $d_c$          | Crime-relevant document (chat log)                                                   |
| $T$            | Universe of topics                                                                   |
| $c$            | A criminal topic                                                                     |
| $A$            | Number of authors                                                                    |
| $V$            | Set of distinct words in the vocabulary                                              |
| $W_c$          | Set of distinct crime-related words for $c$                                          |
| $N$            | Number of word tokens                                                                |
| $\Lambda_d$    | Set of authors in document $d$                                                       |

language model for both the chat log  $d$  and the crime-relevant topic  $c$ , then searches for the document  $d$  based on the probability of the model generating the  $c$ :  $P(M_c|M_d)$ , where  $M$  is a language model. Based on this knowledge, we propose a framework that searches the crime-relevant logs in collections of data. In this section, we briefly describe the statistical models LDA and AT and define the notations in Table 1.

#### 4.1 Latent Dirichlet Allocation (LDA)

*Latent Dirichlet Allocation (LDA)* [1] is an unsupervised generative probabilistic model that discovers latent semantic topics in a corpus with large collections of discrete data, such as the words in a set of documents. It is based on a “bag of words” assumption, which treats each document as a frequency of word counts, ignoring the order of appearance. In the language of probability theory, this is an assumption of “exchangeability”: words are *independent* and *identically* distributed over the topics, and the topics are *infinitely* exchangeable throughout the document, based on some conditional parameters [1, 16].

In LDA, a document can be viewed as a random mixture of hidden variables (i.e., topics) and observed data (i.e., words). Words in a document are generated from the hidden topics and are not linked to the documents directly, but are linked via latent variables (topics) that are responsible for using a particular word in the document drawn from a specific topic distribution that the document focuses on. The generative process can be described as follows:

1. For each document  $d$ , choose  $|D|$  multinomials  $\theta_d \sim \text{Dirichlet prior } \alpha$ ;
2. For each topic  $t$ , choose  $|T|$  multinomials  $\varphi_t \sim \text{Dirichlet prior } \beta$ ;
3. For each word  $w_{di}$  per document  $d$ , in the corpus:
  - choose a topic  $z_i \sim \text{multinomial } \theta_d; (P(z_i | \alpha))$
  - choose a word  $w_i \sim \text{multinomial } \varphi_{z_i}; (P(w_i | z_i, \beta))$

Since estimating  $\theta$  and  $\varphi$ , which provides the proportions of topics in each document and the proportions of words to these topics, respectively, is intractable, different complex algorithms have been proposed, including *variational inference* [1], *expectation propagation* [19], and *Gibbs sampling* [20]. Gibbs sampling is a form of *Markov Chain Monte Carlo*, used for obtaining an approximate inference about parameters. In this model, the posterior distribution of topics over words are calculated as follows:

$$P(z_i = t \mid w_i = w, z_{-i}, w_{-i}, \alpha, \beta) \propto \frac{n_{w_{-i}} + \beta}{\sum_{v \in V} n_{w_{-i},v} + |V|\beta} \times \frac{n_{d_{-i},t} + \alpha}{\sum_{t \in T} n_{d_{-i},t} + |T|\alpha} \quad (1)$$

where  $n_{w_{-i},t}$  is the vector count of the word  $w$  being assigned to the topic  $t$ , not including current word  $i$ .  $n_{d_{-i},t}$  is the vector count of topic  $t$  being assigned to some words, not including the current word  $i$ , in a document  $d$ . After several iterations, specified by the user, the multinomial distribution of documents over topics  $\theta$  and the multinomial distribution of topics over words  $\varphi$  are obtained from the posterior distribution of topics. The details for the Gibbs sampling and LDA can be found in [1,21], respectively.

#### 4.2 Author-Topic (AT)

LDA discloses the underlying topics in the documents in a corpus. However, LDA does not identify authors of a document nor the association of authors to each topic in the topics of a document. As a result, Rosen-Zvi [2] proposed the *Author-Topic (AT)* model, an extension of LDA, that models the content of a document and the interests of authors. Each word in this model consists of two latent variables: an author and a topic. The generative process for this model follows:

1. For each author  $a$ , choose  $A$  multinomials  $\vartheta_a \sim \text{Dirichlet prior } \bar{\alpha}$ ;
2. For each topic  $t$ , choose  $|T|$  multinomials  $\varphi_t \sim \text{Dirichlet prior } \beta$ ;
3. For each word  $w_{di}$  in each document  $d$ , in the corpus:
  - choose an author  $x_i \sim \text{uniform } \Lambda_d; (P(x_i \mid \Lambda_d))$
  - choose a topic  $z_i \sim \text{multinomial } \vartheta_a; (P(z_i \mid x_i, \bar{\alpha}))$
  - choose a word  $w_i \sim \text{multinomial } \varphi_z; (P(w_i \mid z_i, \beta))$

Formally, the procedure for generating a document starts by choosing an author  $x$ , uniformly at random, from the set of authors  $\Lambda_d$  for each word  $w_i$  specific to the document  $d$ , and then, a topic is sampled from the distribution of topics specific to that author  $x$ . Finally, the words are sampled from the distribution of topics over words [2]. This process is continued for all words in the document. However, it is important to note that there is no topic mixture for an individual document [5]. In other words, the multinomial distribution  $\theta_d$  of topics, given documents, is not sampled in the AT model, unlike the LDA model.

An analogy to LDA, the Gibbs sampler for the posterior distribution of topics is as follows:

$$P(z_i = t, x_i = a \mid w_i = w, z_{-i}, w_{-i}, x_{-i}, A_d, \bar{\alpha}, \beta) \propto \frac{n_{w_{-i}} + \beta}{\sum_{v \in V} n_{w_{-i},v} + |V|\beta} \times \frac{n_{x_{-i},t} + \bar{\alpha}}{\sum_{t \in T} n_{x_{-i},t} + |T|\bar{\alpha}} \quad (2)$$

where  $n_{w_{-i},t}$  is the vector counts of the word  $w$  being assigned to the topic  $t$ , not including current word  $i$ , and  $n_{x_{-i},t}$  is the vector count of words being assigned to topic  $t$  for author  $a$  to some words, not including the current word  $i$ . More details on the AT model are found in [2].

## 5 Proposed approach

Probabilistic topic models, such as LDA and AT, model the hidden semantic structure of a document collection without prespecifying whether a document contains a specific topic or not. Therefore, we employ language modeling for both chat log  $d$  and criminal topic  $c$  and then ask how different these two language models are from each other. The key point is to compute the probability of a language model  $M_c$  generating the document  $d$  and to determine the topics of interest in a given chat log. After that, we use two extended topic models to extract topics and discover the topics related to crime in  $d$ . The insight behind proposing the two models is to capture the topics' progressive information and to extract the authors' information as it evolves over time. We begin this section by describing some of the measurements used to process the search, and then, we present our approach in detail.

### 5.1 Kullback–Leibler divergence

The general representation of the two language models,  $M_d$  and  $M_c$ , is in the form of distribution of words. In order to measure the dissimilarity between them, we employ the *Kullback–Leibler (KL) divergence*. In language models, KL is often used in clustering as a measure of (dis)similarity of some given language models. KL divergence is calculated as follows:

$$\text{KL}(M_c \parallel M_d) = \sum_{w \in V} P(w \mid M_c) \log \frac{P(w \mid M_c)}{P(w \mid M_d)} \quad (3)$$

where  $d$  and  $c$  are two probability distributions representing a chat log and a criminal topic, respectively. When using a code based on  $d$ , KL measures the expected number of additional bits required to code samples from  $c$  [16]. In other words, it measures how bad the probability distribution  $M_d$  is at modeling  $M_c$ . In this paper, we compute the average of  $\text{KL}(M_c \parallel M_d)$  and  $\text{KL}(M_d \parallel M_c)$ .

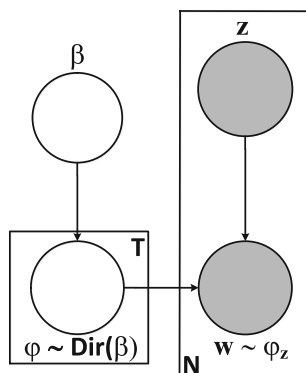
### 5.2 Criminal topic model

$n$ -grams are the most commonly used natural language model. It is a probabilistic model that takes the assumption that only the previous  $n - 1$  words, in a sequence, have any effect on the probabilities for the next word. In other word, the probability of a current word depends on the previous  $n$ -words. An  $n$ -gram model of size 1 is called a *unigram* model. In this model, the words for each document are drawn from a single multinomial distribution, independently. If we extend the unigram model by adding a discrete random topic  $c$ , the *mixture of unigrams* model is obtained [1]. We apply the mixture of unigrams model to explore a chat log and its relation to criminal activities. Throughout this paper, we use *Criminal Topic (CT)* to refer to the mixture of unigrams model. Under this model, a single topic  $c$  generates  $N$  words. We assume that the topic  $c$  and the words  $w$  are observable in the CT model. The key point of developing this model is the assumption of exhibiting several criminal topics for any detained chat logs, and each of these topics is composed of its own distribution of words. Therefore, comparing the topics distributions in  $d$  with  $c$  indicates the relevance of  $d$  to crime.

The words are drawn from a single topic distribution:

$$P(w|c, \varphi) = \prod_{n=1}^N \varphi_{w_n, c} \quad (4)$$

**Fig. 2** The graphical model representation (plate notation) of Criminal Topic (CT) model



where  $\varphi$  is the distribution of words under  $c$ . It describes the probability of each word  $w$  conditioned on  $c$ .  $\varphi_c$  is calculated as follows:

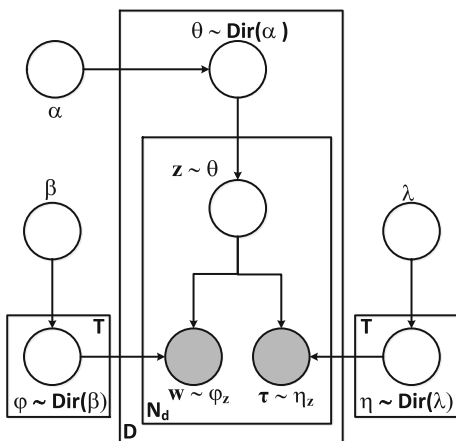
$$\varphi_c = \frac{n_{w_i,c} + \beta}{\sum_{n \in W_c} n_{w_i,c} + |W_c| \beta} \quad (5)$$

Using the CT model, in Fig. 2, KL is applied to estimate the distance between  $d$  and  $c$  in order to distinguish crime-relevant logs from others. In addition, it is also applied to compute the distance between discovered criminal topics and  $c$  after the two extended models described in the next subsections have generated topics from  $d$ .

### 5.3 Mining for crime-relevant chat logs, topics, and topics over time using LDA-TOT

Topic discovery is influenced not only by the occurrence of words and their frequencies, but also by the time stamp associated with each word in a chat log. The transition of topics over time in a given chat log can be estimated by introducing an observable variable  $t$  into the standard LDA model. Various models have been proposed to illustrate the transition of topics over time, such as the *TOT* model [13]. Nonetheless, we depict *LDA-Topics over Time (LDA-TOT)* model, in Fig. 3, that identifies topics and their evolution over time.

**Fig. 3** The graphical model representation (plate notation) of LDA-Topics over Time (LDA-TOT) model



The primary difference between this model and TOT is the use of discrete intervals of time instead of continuous time, as in TOT. Time intervals in a chat log are relatively short, ranging from a few minutes to a few hours. Therefore, we employ discrete time intervals in this model. Moreover, it is easy to include discretization of time, for learning and computation purposes, in order to generate the topics' distribution over time ( $\eta$ ), rather than using continuous beta distribution. The generative process for LDA-TOT is described as follows:

1. For each document  $d$ , choose  $|D|$  multinomials  $\theta_d \sim \text{Dirichlet prior } \alpha$  ;
2. For each topic  $t$ , choose  $|T|$  multinomials  $\varphi_t \sim \text{Dirichlet prior } \beta$  ;
3. For each word  $w_{di}$  in each document  $d$ , in the corpus:
  - choose a topic  $z_i \sim \text{multinomial } \theta_d; (P(z_i | x_i, \alpha))$
  - choose a word  $w_i \sim \text{multinomial } \varphi_z; (P(w_i | z_i, \beta))$
  - choose a time interval  $\tau_i \sim \text{multinomial } \eta_z; (P(\tau_i | z_i, \lambda))$

In the above procedure, the posterior distribution of topics depends on both word and time. We use Gibbs sampling to estimate the approximate inference, as done for both LDA and AT. We begin deriving with the joint distribution  $P(w, \tau, z | \lambda, \alpha, \beta)$ .

$$\begin{aligned}
 P(w, \tau, z | \lambda, \alpha, \beta) &= P(w | z, \beta) P(\tau | z, \lambda) P(z | \alpha) \\
 &= \prod_{t \in T} \frac{\Gamma(\sum_{v \in V} \beta_v)}{\prod_{v \in V} \Gamma(\beta_v)} \frac{\prod_{v \in V} \Gamma(\beta_v + n_{v,t})}{\Gamma(\sum_{v \in V} \beta_v + n_{v,t})} \\
 &\quad \times \prod_{t \in T} \frac{\Gamma(\sum_{v \in V} \lambda_v)}{\prod_{v \in V} \Gamma(\lambda_v)} \frac{\prod_{v \in V} \Gamma(\lambda_v + n_{v,t})}{\Gamma(\sum_{v \in V} \lambda_v + n_{v,t})} \\
 &\quad \times \prod_{d \in D} \frac{\Gamma(\sum_{t \in T} \alpha_t)}{\prod_{t \in T} \Gamma(\alpha_t)} \frac{\prod_{t \in T} \Gamma(\alpha_t + n_{d,t})}{\Gamma(\sum_{t \in T} \alpha_t + n_{d,t})}
 \end{aligned} \tag{6}$$

Using the chain rule, we obtain the conditional distribution  $P(z_i = t | w_i = w, z_{-i}, w_{-i}, \tau_{-i}, \lambda, \alpha, \beta)$ :

$$\begin{aligned}
 P(z_i = t | w_i = w, z_{-i}, w_{-i}, \tau_{-i}, \lambda, \alpha, \beta) &\propto \\
 &\frac{n_{w_{-i}} + \beta}{\sum_{v \in V} n_{w_{-i},t} + |V|\beta} \times \frac{n_{d_{-i},t} + \alpha}{\sum_{t \in T} n_{d_{-i},t} + |T|\alpha} \times \frac{n_{\tau_{-i}} + \lambda}{\sum_{v \in V} n_{\tau_{-i},t} + |V|\lambda}
 \end{aligned} \tag{7}$$

where  $n_{w_{-i},t}$  and  $n_{d_{-i},t}$  are the same as in the LDA model.  $n_{\tau_{-i},t}$  is the vector counts of the word  $w$  being assigned to the topic  $t$  under time interval  $\tau$ , not including current word  $i$ .

Now, we provide an algorithm to classify crime-relevant chat logs and to extract the underlying crime-relevant topics in these logs. We emphasize that this algorithm searches for a particular criminal topic in a chat log. An overview of the algorithm is shown in Algorithm 1. The process starts by employing the CT model to estimate  $\varphi$  for a single topic  $c$  (Line 2). This is the learning process for the CT model. It is common for a criminal topic  $c$  to contain a set of words  $W_c$  that might not be included in the pre-existing vocabulary set  $V_D$ . Therefore, we combine the words  $W_c$  in a criminal topic with the existing  $V_D$  words (Line 3).

It is important to emphasize that CT model is employed to collect the distribution of certain topics, such as sex and drugs. Although the Twitter-LDA model [22] integrates background topic into the model, we apply the CT model separately from the overall LDA-TOT and A-TOT models for computation purposes.

Next, the distance between a chat log  $d$  and the criminal topic  $c$  is calculated using KL divergence (Line 4), under the same vocabulary used for both  $c$  and  $d$ . At this point, the

distribution of words  $\varphi$  for  $d$  is computed using the similar formula to the Eq. 5. The results obtained from KL might or might not pass the user-specified threshold  $\epsilon$ . In case the distance measurement KL is lower than or equal to  $\epsilon$  (Line 5), the algorithm proceeds to the subsequent steps (Line 6–13); otherwise, it terminates (Line 14). Then, LDA-TOT is applied to extract crime-relevant topics in a chat log, where all words in  $d$  are randomly assigned to topics (Line 6).

The iteration process starts by employing collapsed Gibbs sampling parameter estimation with number of loops to estimate the hidden topic structure of unseen chat logs (Line 8).

Next, the KL distance between each topic  $t$  and the provided  $c$  is computed (Line 9–11). Finally, when a topic  $t$  passes the threshold  $\gamma$  (Line 12) and the user's termination criteria  $\zeta$  supports (Line 13), the algorithm terminates. The outputs are the three distributions  $(\theta, \varphi, \eta)$ ; these are further analyzed in the experimental section (Sect. 6), using other evaluation measures to evaluate the performance of the proposed procedure.

---

**Algorithm 1** Mining for crime-relevant chat logs, topics, and topics over time using LDA-TOT

---

```

1: Input:  $\alpha, \beta, \lambda, \epsilon, \gamma$ 
2:  $\varphi$  = Calculate criminal topic-word distribution( $|D|, \alpha, \beta, \lambda$ )
3:  $V = V_D \cup W_c$ 
4:  $\Delta = \text{KL}(d_i, c)$ 
5: if  $\Delta \leq \epsilon$  then
6:   Initialize randomly for all words  $w_i^N$  in a chat log  $d$  to topics  $z_i^{|T|}$ 
7:   repeat
8:      $\theta_d, \varphi, \eta, z_t = \text{GibbsSampling}(d, \tau_d, \alpha, \beta, \lambda)$ 
9:     for  $t = 1$  to  $|T|$  do
10:       $\sigma_t^{|T|} = \text{KL}(\theta_{d,t}^{|T|}, c)$ 
11:    end for
12:     $L = \text{GetLowest}(\sigma_t^{|T|})$ 
13:  until  $\zeta$ 
14: end if

```

---

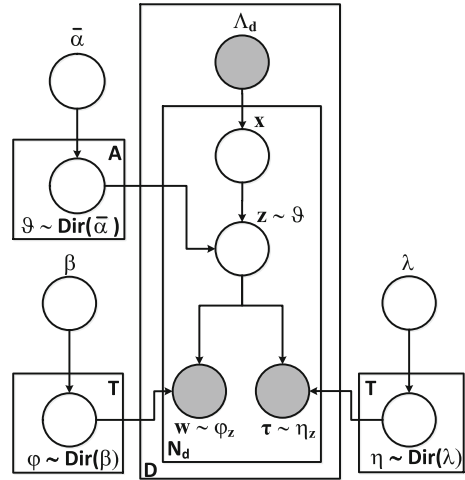
#### 5.4 Mining for crime-relevant chat logs, topics, authors, and authors-topics over time using A-TOT

To answer the second research question, we introduce the *Author-Topics over Time (A-TOT)* model as in Fig. 4. This model is an extension combined from both models, AT and TOT. The aim of this unsupervised learning model is to achieve topics extraction, authors-topics distribution, and authors-topics distribution over time.

The generative process for A-TOT, which corresponds to the Gibbs sampling for estimating the parameters, is as follows:

1. For each author  $a$ , choose  $A$  multinomials  $\vartheta_a \sim \text{Dirichlet prior } \bar{\alpha}$  ;
2. For each topic  $t$ , choose  $|T|$  multinomials  $\varphi_t \sim \text{Dirichlet prior } \beta$  ;
3. For each word  $w_{di}$  in each document  $d$ , in the corpus:
  - choose an author  $x_i \sim \text{uniform } \Delta_d; (P(x_i | \Delta_d))$
  - choose a topic  $z_i \sim \text{multinomial } \vartheta_a; (P(z_i | x_i, \bar{\alpha}))$
  - choose a word  $w_i \sim \text{multinomial } \varphi_z; (P(w_i | z_i, \beta))$
  - choose a time interval  $\tau_i \sim \text{multinomial } \eta_z; (P(\tau_i | z_i, \lambda))$

**Fig. 4** The graphical model representation (plate notation) of Author-Topics over Time (A-TOT) model



Formally, the set of authors  $\Lambda_d$  in a chat log  $d$  is observed. The procedure begins by choosing an author  $x$ , randomly at uniform, from the set of authors  $\Lambda_d$ . Afterward, the multinomial distribution  $\vartheta_d$ , from the Dirichlet distribution  $\bar{\alpha}$ , is picked, and this distribution determines which topics are most likely to be assigned to the author  $x$  in a chat log  $d$ . Next, a single topic  $z_i = t$  is sampled for each  $i$ th word ( $w_i$ ) in  $d$ , from the multinomial distribution  $\vartheta_d$  associated with the author  $x$  for that word. In general, we assume the  $i$ th word ( $w_i$ ) in  $d$  is written by  $x$  for the topic  $z_i = t$ . Finally, in order to generate a word, the model chooses a word  $w_i$ , from the vocabulary of  $V$  words, based on the multinomial distribution  $\varphi_z$ , and assigns a single time stamp  $\tau_i$  from  $\eta_z$  to  $w_i$ .  $\varphi_z$  is generated from the Dirichlet distribution  $\beta$  for each topic  $t$ .

From the procedure, A-TOT depends on both word and time for generating topics. A topic in this model is sampled from the distribution of topics specific to author  $x$ , and the words are sampled from the distribution of words over topics. The distribution of words over topics  $\sum_{v \in V} \varphi_{v,t} = 1$  is the same for both models, LDA-TOT and A-TOT. As for A-TOT, the distribution of topics over authors  $\sum_{t \in T} \vartheta_{a,t} = 1$ . Like LDA-TOT,  $\eta_z$  is a multinomial distribution for each word token  $w_i$  over time stamp  $\tau_i$ , under a topic  $z$ .

The joint distribution  $P(w, \tau, x, z | A, \lambda, \alpha, \beta)$  for A-TOT model is as follows:

$$\begin{aligned}
 P(w, \tau, x, z | A, \lambda, \alpha, \beta) &= P(w | z, \beta) P(\tau | z, \lambda) P(z | \alpha) P(x | A) \\
 &= \prod_{t \in T} \frac{\Gamma(\sum_{v \in V} \beta_v)}{\prod_{v \in V} \Gamma(\beta_v)} \frac{\prod_{v \in V} \Gamma(\beta_v + n_{v,t})}{\Gamma(\sum_{v \in V} \beta_v + n_{v,t})} \\
 &\quad \times \prod_{t \in T} \frac{\Gamma(\sum_{v \in V} \lambda_v)}{\prod_{v \in V} \Gamma(\lambda_v)} \frac{\prod_{v \in V} \Gamma(\lambda_v + n_{v,t})}{\Gamma(\sum_{v \in V} \lambda_v + n_{v,t})} \\
 &\quad \times \prod_{d \in D} \frac{\Gamma(\sum_{t \in T} \alpha_t)}{\prod_{t \in T} \Gamma(\alpha_t)} \frac{\prod_{t \in T} \Gamma(\alpha_t + n_{d,t})}{\Gamma(\sum_{t \in T} \alpha_t + n_{d,t})} \times \prod_{d \in D} \frac{1}{A_d^{N_d}}
 \end{aligned} \tag{8}$$

The conditional distribution  $P(z_i = t, x_i = a | w_i = w, z_{-i}, w_{-i}, x_{-i}, \tau_{-i}, A, \lambda, \bar{\alpha}, \beta)$  uses the Gibbs sampling and is obtained using chain rule:

$$P(z_i = t, x_i = a \mid w_i = w, z_{-i}, w_{-i}, x_{-i}, \tau_{-i}, A_d, \lambda, \bar{\alpha}, \beta) \propto \frac{n_{w_{-i}} + \beta}{\sum_{v \in V} n_{w_{-i}, t} + |V|\beta} \times \frac{n_{x_{-i}, t} + \bar{\alpha}}{\sum_{t \in T} n_{x_{-i}, t} + |T|\bar{\alpha}} \times \frac{n_{\tau_{-i}} + \lambda}{\sum_{v \in V} n_{\tau_{-i}, t} + |V|\lambda} \quad (9)$$

where  $n_{\tau_{-i}, t}$  is number of the word  $w$  being assigned to the topic  $t$  under time interval  $\tau$ , not including current word  $i$ .

The algorithm is similar to the previous one, and it searches for the crime-relevant logs, extracts topics with authors, and collects the authors' distributions over time intervals within the discovered topics using A-TOT instead of LDA-TOT. The outputs are in the form of three distributions  $(\vartheta, \varphi, \eta)$ .

## 6 Empirical study

In this section, we perform an empirical study on the two research questions addressed in Sect. 1 and provide the results with extensive details.

### 6.1 Data preparation

The chat logs used in the experiment are obtained from a website called *pervverted-justice.com* and *IRC* logs.

**Pervverted-Justice.** This dataset consists of chat logs from various instant messages, e.g., Yahoo! and AOL, containing information about adults who seek online sexual conversations with others who are posing as children or underage teenagers (pseudo-victims). The dataset contains 250 log files and 500 authors [23]. We use only the time intervals associated with messages in these chat logs, without considering the exact date and time.

**IRC.** This dataset is collected from various IRC channels by running a mIRC application for about 10 days. The dataset contains 170 authors and 50 log files with a total of 4,086 word tokens. There are 5 categories classified in multiple topics. Each message in the chat logs has a time stamp that is determined by the date and time intervals. As in the previous dataset, we use only the time intervals and ignore the date.

For both datasets, we first remove all links from the messages, stop words, numbers, and non-English letters. The words are downcased and stemmed to their root source, using porter stemmer. However, words that rarely appear in a chat log are not removed, because the chat log differs from the structured documents and the words might be of value to the results. The results from the preprocessing step for both datasets consist of 620 authors and a total of 31,952 word tokens. Some statistics of the two datasets after preprocessing are summarized in Table 2. The sizes of the two datasets are comparable to the size of datasets in real-life cases.

**Table 2** Summary of the datasets

| Dataset            | Documents | Words   | Unique words |
|--------------------|-----------|---------|--------------|
| Pervverted-justice | 250       | 27, 866 | 1, 455       |
| IRC                | 50        | 4, 086  | 276          |

## 6.2 Evaluation measures and parameters setting

To compute the correctness of the retrieved chat logs ( $d_c$ ) by algorithms using LDA-TOT and A-TOT models, we calculate *Precision*, *Recall*, and *F-Measure*. The *Precision* of a model describes the number of the discovered chat logs  $d_c$  that are correct from overall retrieved logs that seem to be relevant;  $d_c$  is the crime-relevant chat log.

$$\text{Precision} = \frac{\text{Number of true positives (truth } d_c)}{\text{Retrieved Documents}} \quad (10)$$

The *Recall* of a model describes the number of the relevant (truth) chat logs  $d_c$  are successfully discovered by the model.

$$\text{Recall} = \frac{\text{Number of true positives (truth } d_c)}{\text{Number of } d_c \text{ are correct}} \quad (11)$$

The *F-Measure* computes the weighted harmonic mean of precision and recall for a model.

$$F_\pi = (\pi^2 + 1) \cdot \frac{\text{Precision} \cdot \text{Recall}}{\pi^2 \cdot \text{Precision} + \text{Recall}} \quad (12)$$

where  $\pi \in [0, \infty]$ . In this paper, we use  $\pi = 2$ , which weighs recall higher than precision, and  $\pi = 1$ , which gives an equal weight for measures, recall, and precision.

In addition to KL divergence, *Normalized Mutual Information (NMI)* is applied as a distance function. The NMI measures the contribution of the presence/absence of a term for making the correct classification decision on  $c$ . In our application, it measures the mutual dependence of  $t$  and the given  $c$ .

$$\text{NMI}(\Omega_t, C_c) = \frac{I(\Omega_t, C_c)}{(H(\Omega_t) + H(C_c))/2} \quad (13)$$

where  $I(\Omega_t, C_c)$  refers to mutual information between the relevant topic  $\Omega_t$  and a given criminal topic  $c$ .  $H$  stands for entropy [18]. NMI is always a number between 0 and 1, implying the two topics are independent or a complete match, respectively.

As for the other settings, we do not estimate the hyper parameters  $\alpha$ ,  $\beta$ , and  $\lambda$ ; instead, they are fixed at  $\alpha = 1$ ,  $\beta = 0.01$ , and  $\lambda = 0.01$ , respectively. The number of topics  $T$  is also fixed at  $|T| = 5$  for both models.

The dataset *perverted-justice* contains explicit sex-related dialog, which is considered to be a crime-relevant topic. Table 3 shows the top 20 extracted Crime-related words using CT. To compute the  $\varphi$  distribution of topic  $c$ , where  $c$  is only sex related, we train the CT model with 200 chat logs from *perverted-justice*, and we keep 50 logs from *perverted-justice* and the other 50 from IRC for testing the outcomes from LDA-TOT and A-TOT. The objective of the test is to evaluate the effectiveness of using the proposed models to identify the crime-relevant, specifically the sex-related, chat logs. The chat logs are renamed  $d_1, d_2, d_3, \dots$  and authors are renamed  $a_1, a_2, a_3, \dots$  instead of using their true names due to privacy concerns.

We note that  $c$  could be any criminal topic and it is not fixed in its size. For the purpose of computing the likelihood of a language model  $M_c$  in generating the document  $d$  and to evaluate how different two probability distributions,  $c$  and  $d$ , are under different sizes of  $c$ , we use two sets of  $c$ , one containing 30 words and the other 50 words. In the experiments, we highlight different characteristics of the extracted topics and their transitions regarding the two sets of  $c$ . We also notice that some chat logs can be fully labeled as crime-relevant by providing only small set of terms in  $c$ .

**Table 3** Top 20 extracted crime-related words using CT

| Word   | Prob   | Word   | Prob   |
|--------|--------|--------|--------|
| luv    | 0.0126 | penis  | 0.0025 |
| girl   | 0.0057 | porn   | 0.0024 |
| babi   | 0.0042 | cum    | 0.0023 |
| sweeti | 0.0038 | playin | 0.0023 |
| sex    | 0.0032 | pretti | 0.0021 |
| suck   | 0.0031 | cam    | 0.002  |
| kiss   | 0.0031 | pussi  | 0.0017 |
| fuck   | 0.0029 | naked  | 0.0015 |
| watcha | 0.0027 | lick   | 0.0015 |
| bed    | 0.0026 | cock   | 0.0013 |

The experiments are executed on a PC running Windows 7 (32-bit) with Intel 2.13 GHz (2 CPUs) and 2 GB memory. We run the application several times at a fixed number of 2,000 iterations and record the outcomes each time in terms of  $KL(t_{i,d}, c)$ ,  $NMI(t_{i,d}, c)$ ,  $\vartheta_a$ , and  $\theta_d$ .

### 6.3 Scenario

In this subsection, we evaluate the effectiveness of our algorithms by performing an in-depth case study on the two aforementioned datasets to answer the two research questions. These two research questions elaborate on Definition 7, in Sect. 3. The resulting distributions  $(\theta, \vartheta, \varphi, \eta)$  from LDA-TOT and A-TOT models are further analyzed to capture various characteristics of topics, authors, topics evolutions over time, and authors-topics over time intervals.

**Q 1** How can an investigator determine which logs are crime-relevant? In identifying a crime-relevant log, what are the contributed topics in the log file? How have they evolved over time? Moreover, how can an investigator extract the crime-relevant topics from the identified crime-relevant log files?

To answer this question, we apply the mining algorithm, using LDA-TOT to extract the crime-relevant topics, one chat log at a time. We select several logs randomly and record the similarities among these logs. Next, we adopt two expected cases, based on the results from  $KL(d, c) : 1 - KL(d, c) \leq \epsilon$  when  $d$  is crime-relevant.  $2 - KL(d, c) > \epsilon$  when  $d$  is not crime-relevant.

For estimating the value of  $\epsilon$ , we use the following equation:

$$\epsilon = \frac{\sum_{d \in D} KL(d, c)}{|D|} \times \varrho \quad (14)$$

The equation describes the average KL divergence between  $d$  and  $c$  of all the trained documents, and the result is multiplied by  $\varrho$ . Supplementing  $\varrho$ , which is a user's threshold, in the equation adds some flexibility in controlling the average values and it avoids some extreme scores that might occur during the average computation process. For the  $\gamma$  value, we adopt the similar equation:

$$\gamma = \frac{\sum_{t \in T} KL(t, c)}{|T|} \times \varrho \quad (15)$$

**Table 4** KL divergence between documents ( $d_1, d_2, d_3, d_4$ ) and  $c$  when  $|c| = 30$  and  $|c| = 50$  using LDA-TOT and A-TOT

| Documents | Size       | $KL(d_i, c)$ |
|-----------|------------|--------------|
| $d_1$     | $ c  = 30$ | 2.9806       |
|           | $ c  = 50$ | 3.0234       |
| $d_2$     | $ c  = 30$ | 3.4087       |
|           | $ c  = 50$ | 3.4684       |
| $d_3$     | $ c  = 30$ | 3.0113       |
|           | $ c  = 50$ | 3.0583       |
| $d_4$     | $ c  = 30$ | 2.9729       |
|           | $ c  = 50$ | 3.0220       |

**Table 5** KL divergence and NMI between crime-relevant topics from documents ( $d_1, d_2, d_3, d_4$ ) and  $c$  when  $|c| = 30$  and  $|c| = 50$  using LDA-TOT

| Documents | Size            | $KL(t_{i,d}, c)$ | $NMI(t_{i,d}, c)$ |
|-----------|-----------------|------------------|-------------------|
| $d_1$     | $t_3( c  = 30)$ | 1.9664           | 0.0167            |
|           | $t_4( c  = 50)$ | 1.9760           | 0.0196            |
| $d_2$     | $t_3( c  = 30)$ | 1.2750           | 0.0206            |
|           | $t_0( c  = 50)$ | 1.3439           | 0.1725            |
| $d_3$     | $t_2( c  = 30)$ | 1.9358           | 0.3007            |
|           | $t_4( c  = 50)$ | 1.9290           | 0.0555            |
| $d_4$     | $t_2( c  = 30)$ | 1.6693           | 0.0039            |
|           | $t_1( c  = 50)$ | 1.6688           | 0.1033            |

From the training samples, we obtain the values  $\epsilon = 3.3109$  and  $\gamma = 2.0977$ , as an average over all the trained documents, for  $\varrho = 0.5$  and we use these settings for describing the two cases on 4 selected chat logs as below:

**Case 1** ( $KL(d, c) \leq \epsilon$ ) From Definition 4, the document  $d$  is crime-relevant under this case. Based on the results from KL between  $d$  and  $c$ , as shown in Table 4, it is clear that 3 chat logs  $\{d_1, d_3, d_4\}$  follow this case, and they are related to crime. We remind the reader again that topic  $c$  is sex related. LDA-TOT generates 5 topics from each of these 3 chat logs, and the crime-relevant topics are shown in Table 6.

Not surprisingly, the top 10 relevant words, with high probabilities, provide sufficient information to classify these topics as crime-relevant, and the measurements from KL and NMI support our prospects as well. The  $\theta_d^t$  distributions (between the round brackets) for these topics are above 0.2, which represents about one-fifth of the logs. This computation is far more essential because it distinguishes the crime-relevant chat logs from others, and the importance of  $\theta_d$  is well demonstrated in case 2.

By observing  $KL(d, c)$  and  $KL(t_{i,d}, c)$  from Tables 4 and 5, we notice that the results are not always monotonic. For example,  $KL(d_1, c) = 2.981$  and  $KL(d_3, c) = 3.0113$  when the size of  $|c| = 30$ . However,  $KL(t_{2,d_3}, c) = 1.9358$  is more relevant to  $c$  than  $KL(t_{3,d_1}, c) = 1.9664$ . In addition, NMI seems to behave different for both of these topics  $t_{3,d_1} = 0.0167$  and  $t_{2,d_3} = 0.3007$ .

Probabilistic topic models, such as LDA-TOT, are based on the concept of generating topics randomly; each time it extracts topics with different probability distributions. Therefore, the results obtained from KL and NMI between discovered topics and  $c$  are

not necessarily monotonic. Nevertheless, the algorithm discovers crime-relevant chat logs if they exist in a collection of data texts and it terminates if  $\zeta$  is satisfied.

Furthermore, the NMI of topic  $t_{0,d_4}$  should obtain better results because  $KL(d_4, c) = 2.9729$  clearly indicates that  $d_4$  is more proximate to be classified as a crime-relevant log than the other chat logs shown in Table 4. After 4,000 iterations, we found  $KL(t_{3,d_4}, c) = 1.5944$  and NMI of this topic is  $t_{3,d_4} = 0.3564$ .

**Case 2** ( $KL(d, c) > \epsilon$ ) This case occurs whenever a chat log does not satisfy the user-specified threshold  $\epsilon$ . From Table 4, the only chat log that falls under this category is  $d_2$ . Obviously,  $t_{3,d_2}$  ( $|c| = 30$ ) for this chat log does not contain the expected words for it to be classified as crime-relevant.

We observe an interesting result from  $KL(t_{3,d_2}, c)$ , and it satisfies the user-specified threshold  $\gamma$ . In general, KL measures the distance between the two models ( $t$  and  $c$ ). This is achieved by comparing the probability of the shared words in both topics  $c$  and  $t_{3,d_2}$ . We do not consider fixed vocabulary in the comparison, rather we depend on the mutual words. Suppose the unique words for both  $|c| = 30$  and  $|t_1| = 500$ . If the two models have joint words, with similar probability, then the KL distance for both models is similar. Consequently, the result from  $KL(t_{3,d_2}, c)$  fits with the threshold  $\gamma$ .

On the other hand, the  $\theta_d^{t_1}$  distribution shows that approximately 0.1550 of  $d_2$  is about criminal subjects. From  $KL(d_2, c)$ , we conclude that the  $d_2$  is not crime-relevant.

One might ask whether the condition  $KL(t_{i,d}, c) = 0$  applies for both cases. This might occur, but it does not necessarily mean that a topic is crime-relevant, and case 2 sheds some light on it. A topic  $t$  is considered to be crime-relevant whenever the two conditions hold:  $KL(d, c) \leq \epsilon$  and  $KL(t_{i,d}, c) \leq \gamma$ .

When we alter the size of  $c$  by increasing the number of criminal terms to 50, the results from  $KL(t_{i,d}, c)$  and NMI are improved, as observed in Table 5. The top 10 words in Table 6 include new crime-relevant terms that were not observed when  $|c| = 30$ . This is not a coincidence, since the words used in  $c$  are drawn from the two datasets. In general, increasing the size of  $c$  gives better predictions about the distance between discovered topics and  $c$ .

In addition to topics extraction, the LDA-TOT is able to predict the time associated with each message in a chat log. Figure 5 includes the fluctuations of relevant topics from 4 chat logs when  $|c| = 30$ . The characteristics of the transitions can be classified through the transition function  $F(t)_{\tau_s}^{\tau_f}$  as *active* and *not active*. In many cases, the activity of topics is provided by investigators to assist them in analyzing different rise and falls of topics. Therefore, we define the transition function as follows:

$$F(t)_{\tau_s}^{\tau_f} = \begin{cases} \text{active} & \text{if } \sum_{\tau_s}^{\tau_f} p(t)^{\tau_s} \geq \text{user's threshold} \\ \text{not active} & \text{if } \sum_{\tau_s}^{\tau_f} p(t)^{\tau_s} < \text{user's threshold} \end{cases}$$

$\sum_{\tau_s}^{\tau_f} p(t)^{\tau_s}$  sums the probability of a topic  $t$  during interval  $[\tau_s, \tau_f]$ .  $F(t)_{\tau_s}^{\tau_f}$  indicates the activity of  $t$ . We found the best results are obtained when an average of  $\theta_d$  over the three highest topics is considered for estimating the user-specified threshold. For instance, when setting the user-specified threshold to 0.2143, as an average of  $\theta_{d_1}$  over 3 topics, the topic  $t_{3,d_1}$  ( $|c| = 30$ ) is active during [22:00, 1:00] and not active elsewhere.

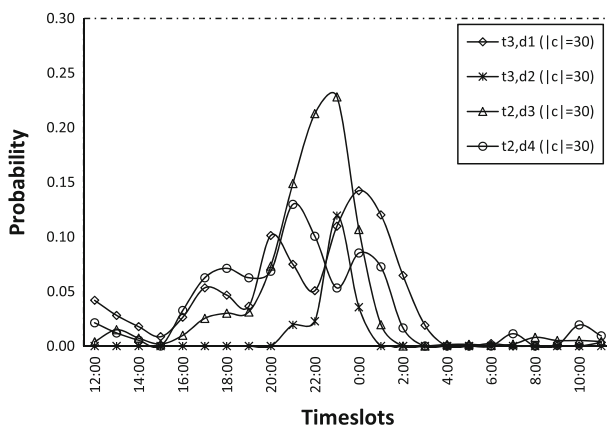
In general, the topics  $t_{3,d_1}$ ,  $t_{3,d_2}$ ,  $t_{2,d_3}$ ,  $t_{2,d_4}$  ( $|c| = 30$ ) are widely active during time intervals [15:00, 3:30] when  $p(t)_{\tau_s}^{\tau_f} \geq 0.2143$ , with a peak on [21:00, 1:00]. Investigators collect information, within certain intervals, that indicates the activity of crime-relevant topics, thus providing the start point for the investigation process.

**Table 6** Top 10 relevant words extracted for crime-relevant topics from documents ( $d_1, d_2, d_3, d_4$ ) and their distribution over documents using LDA-TOT

| $d_1$          |        |                |        | $d_2$          |        |                |        |
|----------------|--------|----------------|--------|----------------|--------|----------------|--------|
| $ c  = 30$     |        | $ c  = 50$     |        | $ c  = 30$     |        | $ c  = 50$     |        |
| $t_3$ (0.1666) | Prob   | $t_4$ (0.1725) | Prob   | $t_3$ (0.1550) | Prob   | $t_0$ (0.1944) | Prob   |
| girl           | 0.0463 | guess          | 0.0655 | dai            | 0.0194 | wed            | 0.0259 |
| feel           | 0.0301 | good           | 0.0512 | go             | 0.0194 | know           | 0.0156 |
| dad            | 0.0295 | sweet          | 0.0209 | talk           | 0.0161 | earlier        | 0.0130 |
| happi          | 0.0234 | sleep          | 0.0166 | sound          | 0.0065 | thing          | 0.0104 |
| stuff          | 0.0084 | sex            | 0.0130 | sad            | 0.0065 | skidoo         | 0.0104 |
| luv            | 0.0084 | hot            | 0.0076 | steal          | 0.0065 | phone          | 0.0052 |
| pantis         | 0.0080 | young          | 0.0074 | academi        | 0.0033 | week           | 0.0052 |
| plai           | 0.0071 | bodi           | 0.0059 | access         | 0.0033 | color          | 0.0052 |
| babi           | 0.0036 | big            | 0.0052 | channel        | 0.0033 | pick           | 0.0052 |
| dirty          | 0.0029 | leg            | 0.0026 | develop        | 0.0033 | trust          | 0.0026 |

| $d_3$          |        |                |        | $d_4$          |        |                |        |
|----------------|--------|----------------|--------|----------------|--------|----------------|--------|
| $ c  = 30$     |        | $ c  = 50$     |        | $ c  = 30$     |        | $ c  = 50$     |        |
| $t_2$ (0.1829) | Prob   | $t_4$ (0.1742) | Prob   | $t_2$ (0.1622) | Prob   | $t_3$ (0.1481) | Prob   |
| preciou        | 0.0513 | kiss           | 0.0203 | pic            | 0.0406 | sound          | 0.0223 |
| wear           | 0.0434 | babi           | 0.0168 | eat            | 0.0180 | eat            | 0.0194 |
| luv            | 0.0433 | time           | 0.0137 | phone          | 0.0173 | babi           | 0.0180 |
| penis          | 0.0252 | luv            | 0.0137 | babi           | 0.0166 | figur          | 0.0129 |
| wear           | 0.0138 | butt           | 0.0081 | cloth          | 0.0113 | sex            | 0.0115 |
| kiss           | 0.0098 | pic            | 0.0081 | sex            | 0.0107 | skirt          | 0.0115 |
| sweet          | 0.0077 | nite           | 0.0063 | hot            | 0.0100 | touch          | 0.0093 |
| excit          | 0.0070 | beauti         | 0.0059 | ass            | 0.0093 | hurt           | 0.0086 |
| finger         | 0.0056 | sweet          | 0.0044 | naked          | 0.0087 | shower         | 0.0072 |
| lick           | 0.0026 | lick           | 0.0029 | cam            | 0.0087 | masterb        | 0.0022 |

**Fig. 5** Evolution of crime-relevant topics using LDA-TOT when  $|c| = 30$

**Table 7** Cosine similarities between each of the documents ( $d_1, d_2, d_3, d_4$ ) and  $c$  for each words in  $V$ 

|                      | $d_1$  | $d_2$  | $d_3$  | $d_4$  |
|----------------------|--------|--------|--------|--------|
| $\text{sim}(d_i, c)$ | 0.8403 | 0.5257 | 0.8267 | 0.7999 |

**Table 8** Cosine similarities between pairs of the resulting  $V$  dimensional vectors for ( $d_1, d_2, d_3, d_4$ )

|                        | $d_1$  | $d_2$  | $d_3$  | $d_4$  |
|------------------------|--------|--------|--------|--------|
| $\text{sim}(d_i, d_1)$ | 1.0000 | 0.4358 | 0.6624 | 0.7586 |
| $\text{sim}(d_i, d_2)$ | 0.4358 | 1.0000 | 0.4342 | 0.3760 |
| $\text{sim}(d_i, d_3)$ | 0.6624 | 0.4342 | 1.0000 | 0.6373 |
| $\text{sim}(d_i, d_4)$ | 0.7586 | 0.3760 | 0.6373 | 1.0000 |

We conclude that a crime-relevant chat log  $d$  can be recognized through  $KL(d, c)$ , and the crime-relevant topics are determined by three factors:  $\theta_d^t$ ,  $KL(d, c)$ , and  $KL(t_{i,d}, c)$ . The characteristics of the relevant topics are studied through NMI, and the high probability of NMI means obtaining a better quality of discovered topics. In addition, the evolution of topics is demonstrated through the transition function  $F(t)_{\tau^s}^{\tau^f}$ , in terms of active or not active, in the given time intervals associated with each message in logs.

To further demonstrate our process for segregating crime-relevant documents, we use the cosine similarity to measure document similarity:

$$\text{sim}(d_1, d_2) = \frac{\vec{q}_{d_1} \cdot \vec{q}_{d_2}}{|\vec{q}_{d_1}| |\vec{q}_{d_2}|} \quad (16)$$

where  $\vec{q}_{d_1} \cdot \vec{q}_{d_2}$  represents the standard vector dot product, described as  $\sum_{v \in V} (q_{v,d_1} q_{v,d_2})$ , and the denominator represents the product of their *Euclidean lengths*, described as  $\sqrt{\sum_{v \in V} q_{v,d_1}^2} \cdot \sqrt{\sum_{v \in V} q_{v,d_2}^2}$ .

To compute vector  $q_c$ , we apply Eq. 5 to determine the weighted value for each criminal term during the learning phase over  $V$  instead of  $W_c$  on the training samples. For  $q_{d_i}$ , we use the same Eq. 5 to calculate the weighted value for each word in  $d_i$  over  $V$  dimensions. Table 7 quantifies the similarity scores between each of the 4 chat logs and  $c$  for each word in  $V$ . We observe that document  $d_2$  differs from  $d_1, d_3$ , and  $d_4$  in its similarity value. Furthermore, Table 8 shows the cosine similarities between pairs of the resulting  $V$  dimensional vectors for 4 chat logs. A cosine computation between documents  $d_1, d_3$ , and  $d_4$  shows that they are similar, but  $d_2$  is different from other chat logs. If 0.75 is chosen to be the splitting point in classifying crime-related documents, we obtain similar conclusions with  $KL(d_2, c)$  value from Table 4. Since this value does not satisfy the user-specified threshold  $\epsilon = 3.3109$ ,  $d_2$  is not crime-relevant.

**Q 2** Who are the contributors to a topic in a given chat log? How can an investigator track the activity of authors in a log file?

We divide this question into two parts. First, we determine the proportions of each author contributing in each of the extracted topics. Second, we explore the impacts of the authors throughout the time intervals on the extracted topics. This time, we employ the mining algorithm, using a A-TOT model to study the two parts of the question. The two users' threshold  $\epsilon$  and  $\gamma$  are empirically set to 3.3109 and 2.0977, respectively.

**Table 9** KL divergence and NMI between crime-relevant topics from documents ( $d_1, d_2, d_3, d_4$ ) and  $c$  when  $|c| = 30$  and  $|c| = 50$  using A-TOT

| Documents | Size            | KL( $t_{i,d}, c$ ) | NMI( $t_{i,d}, c$ ) |
|-----------|-----------------|--------------------|---------------------|
| $d_1$     | $t_2( c  = 30)$ | 1.9558             | 0.0557              |
|           | $t_4( c  = 50)$ | 1.8599             | 0.0118              |
| $d_2$     | $t_0( c  = 30)$ | 1.2967             | 0.1698              |
|           | $t_1( c  = 50)$ | 1.2960             | 0.0095              |
| $d_3$     | $t_1( c  = 30)$ | 1.9538             | 0.1597              |
|           | $t_1( c  = 50)$ | 1.9820             | 0.3810              |
| $d_4$     | $t_4( c  = 30)$ | 1.6872             | 0.0560              |
|           | $t_0( c  = 50)$ | 1.7164             | 0.1407              |

A-TOT implementation is slightly different from the proposed one because we are concerned with collecting information related to  $\theta_d$  and  $\vartheta_a$  distributions. We apply the same 4 chat logs that explore the first research question. From each of these chat logs, A-TOT generates 5 topics with authors associated with each. The  $\theta_d$  distribution for the crime-relevant topics, from the 4 chat logs, is displayed (between the round brackets) in Table 10.

**Table 10** Top 10 relevant words extracted for crime-relevant topics from documents ( $d_1, d_2, d_3, d_4$ ), their distribution over documents, and their distribution over top 3 authors using A-TOT

| $d_1$          |        |                |        | $d_2$          |        |                |        |
|----------------|--------|----------------|--------|----------------|--------|----------------|--------|
| $ c  = 30$     |        | $ c  = 50$     |        | $ c  = 30$     |        | $ c  = 50$     |        |
| $t_2$ (0.2382) | Prob   | $t_4$ (0.1648) | Prob   | $t_0$ (0.2096) | Prob   | $t_1$ (0.1790) | Prob   |
| make           | 0.0487 | talk           | 0.0280 | wed            | 0.0299 | wed            | 0.0312 |
| link           | 0.0403 | girl           | 0.0186 | dai            | 0.0179 | hour           | 0.0125 |
| gui            | 0.0340 | see            | 0.0107 | earlier        | 0.0149 | gnite          | 0.0094 |
| nice           | 0.0258 | nite           | 0.0092 | care           | 0.0120 | care           | 0.0063 |
| show           | 0.0187 | suck           | 0.0077 | feel           | 0.0120 | wonder         | 0.0063 |
| butt           | 0.0175 | cum            | 0.0069 | old            | 0.0060 | drive          | 0.0063 |
| babi           | 0.0155 | whatcha        | 0.0062 | quit           | 0.0060 | sound          | 0.0063 |
| luv            | 0.0119 | pic            | 0.0059 | pc             | 0.0060 | learn          | 0.0031 |
| cum            | 0.0110 | eat            | 0.0055 | week           | 0.0060 | convers        | 0.0031 |
| figur          | 0.0098 | bodi           | 0.0050 | sound          | 0.0060 | presum         | 0.0031 |
| Authors        | Prob   | Authors        | Prob   | Authors        | Prob   | Authors        | Prob   |
| $a_1$          | 0.3117 | $a_1$          | 0.1824 | $a_1$          | 0.2257 | $a_5$          | 0.2435 |
| $a_2$          | 0.2365 | $a_2$          | 0.1742 | $a_2$          | 0.2188 | $a_4$          | 0.2203 |
| $a_3$          | 0.2361 | $a_4$          | 0.1507 | $a_3$          | 0.2075 | $a_7$          | 0.2157 |
| preciou        | 0.0740 | think          | 0.1039 | feel           | 0.0238 | eat            | 0.0154 |
| look           | 0.0527 | luv            | 0.0558 | suck           | 0.0168 | phone          | 0.0149 |
| wear           | 0.0259 | tell           | 0.0543 | kiss           | 0.0116 | gf             | 0.0137 |
| babi           | 0.0175 | luv            | 0.0558 | show           | 0.0110 | show           | 0.0109 |
| touch          | 0.0170 | feel           | 0.0321 | fuck           | 0.0052 | luv            | 0.0109 |

**Table 10** continued

| $d_3$          |        |                |        | $d_4$          |        |                |        |
|----------------|--------|----------------|--------|----------------|--------|----------------|--------|
| $ c  = 30$     |        | $ c  = 50$     |        | $ c  = 30$     |        | $ c  = 50$     |        |
| $t_1$ (0.1777) | Prob   | $t_1$ (0.1930) | Prob   | $t_4$ (0.1907) | Prob   | $t_0$ (0.1941) | Prob   |
| hear           | 0.0103 | butt           | 0.0166 | watcha         | 0.0052 | figur          | 0.0103 |
| sweet          | 0.0101 | suck           | 0.0162 | porn           | 0.0052 | peopl          | 0.0097 |
| shower         | 0.0067 | nite           | 0.0126 | bra            | 0.0052 | tit            | 0.0074 |
| hand           | 0.0062 | big            | 0.0115 | stick          | 0.0046 | cam            | 0.0074 |
| eat            | 0.0062 | show           | 0.0075 | bike           | 0.0046 | rite           | 0.0074 |
| Authors        | Prob   | Authors        | Prob   | Authors        | Prob   | Authors        | Prob   |
| $a_1$          | 0.2000 | $a_3$          | 0.2258 | $a_1$          | 0.2355 | $a_1$          | 0.2210 |
| $a_2$          | 0.1998 | $a_4$          | 0.2227 | $a_2$          | 0.2020 | $a_2$          | 0.2008 |
| $a_3$          | 0.1613 | $a_1$          | 0.2000 | $a_3$          | 0.1835 | $a_3$          | 0.1992 |

We observe similar results when comparing the distribution  $\theta_d$ , from Tables 6 and 10, for both models, LDA-TOT and A-TOT. However, the comparison between A-TOT and LDA-TOT is not addressed in this paper.

The generated  $\vartheta_a^t$  distribution using A-TOT is shown in Table 10. The top 3 authors with the highest probabilities for each of the crime-relevant topics in each of the 4 chat logs are displayed. For example, author  $a_1$  in  $d_3$  has a probability of 0.2000 for topic  $t_1$ , which outlines the contribution of  $a_1$  out of all authors to the crime-relevant topic  $t_1$  when  $|c| = 30$ .

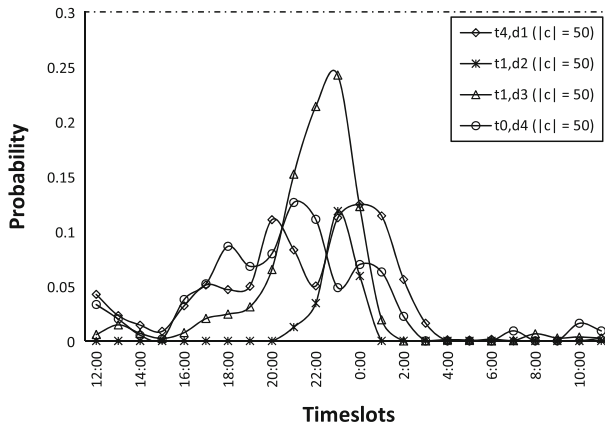
Though  $\vartheta_a^t$  distribution assists investigators to identify the plausible authors in the crime-relevant topics, it does not provide the contributions and activity of each author during specific time intervals within topics. From Definition 1, time  $\tau_d$  is associated with both message  $\mu_d$  and author  $a_d$ . Hence, for the second part of the question, we keep tracking the times since the messages were composed. Following up, we characterize the contributions of authors during time interval  $[\tau^s, \tau^f]$  by:

$$F(a_d^t)_{\tau^s}^{\tau^f} = \begin{cases} \text{active} & \text{if } p(a_d^t)_{\tau^s}^{\tau^f} \geq \text{user's threshold, } F(t)_{\tau^s}^{\tau^f} \text{ is active} \\ \text{not active} & \text{otherwise} \end{cases}$$

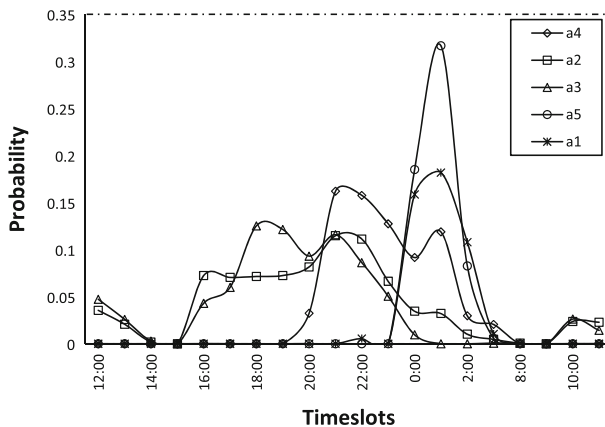
An author is said to be *active* during the interval  $[\tau^s, \tau^f]$  for topic  $t$  if the probability of an author participating in  $t$ , during that interval, exceeds the user-specified threshold, and  $F(t)_{\tau^s}^{\tau^f}$  is *active* within that period. The user-specified threshold is calculated by taking an average of  $\vartheta_a^t$  over authors for  $t$ . To compute  $p(a_{i,d}^t)_{\tau^s}^{\tau^f}$ , we first map the contribution of an author  $a_{i,d}^t$ , within  $[\tau^s, \tau^f]$ , using  $P(a^{\tau^s}|t) = \frac{p(a^{\tau^s}|d^{\tau^s}) \cdot p(t^{\tau^s}|d^{\tau^s})}{p(d^{\tau^s})}$  per time instance  $s$ . Next, we calculate  $\sum_{\tau^s}^{\tau^f} P(a^{\tau^s}|t)$ , as a total probability for author  $a^t$  during  $[\tau^s, \tau^f]$ .

The transitions of the crime-relevant topics when  $|c| = 50$  using A-TOT are shown in Fig. 6. From this figure and the mapping function, we determine the activity of authors over time. For example, let us analyze the activity of authors in topic  $t_4, d_4$  during [16:00, 19:00].

First, we determine the user-specified threshold, which is 0.1940 as an average of  $\vartheta^{t_4}$ . Next, the mapping function is calculated for all authors. For simplicity, let us pick an author  $a_3$  and time instance  $s = 16:00$ . Then, we compute the mapping function, which is  $P(a_{3,\tau^{16:00}}|t_4) = 0.0584$ . Afterward, the total probability of  $a_3$  is estimated by computing



**Fig. 6** Evolution of crime-relevant topics using A-TOT when  $|c| = 50$



**Fig. 7** Authors activity for crime-relevant topic  $t_{4,d_4}$  using A-TOT when  $|c| = 50$

$\sum_{\tau=16:00}^{19:00} P(a_3, \tau^s | t_4) = 0.2687$ . Consequently, we say the authors  $(a_2, a_3)$  for topic  $t_{4,d_1}$  are *active* for satisfying the two conditions when applying the transition function  $F(a_d^t)_{\tau^s}^f$ , while the authors  $(a_1, a_4, a_5)$  are not.

Figure 7 summarizes the activity of authors for the crime-relevant topic  $t_{4,d_1}$ . It can be observed that the most active time for authors occurred during  $[0:00, 7:00]$  and  $[15:00, 23:00]$ . This helps the investigators to determine the initiator of a topic and to capture the plausible authors within intervals. If the given time period  $[15:00, 19:00]$  is an important interval for an investigator, then the suspected authors are  $(a_2, a_3)$ , since they are active during that phase of time, while  $(a_1, a_4, a_5)$  are not active.

Similarly to LDA-TOT, when we increase the size of  $c$ , the probability of authors-topics is different in the context of crime-relevant topics. For example, from Table 10, the probability of author  $a_1$  in  $t_{2,d_1}$  when  $|c| = 30$  is 0.3117, unlike 0.1824 when  $|c| = 50$ . The NMI for the discovered crime-relevant topics, shown in Table 9, are improved and new words are obtained, as explored in Table 10. Hence, we determine that the NMI value of topics quantifies the best obtained results. Note, the criminal words used in  $c$  are collected from the two datasets.

**Table 11** Precision, Recall,  $F_1$ , and  $F_2$  using LDA-TOT and A-TOT

|            | Precision | Recall | $F_1$  | $F_2$  |
|------------|-----------|--------|--------|--------|
| $ c  = 30$ | 0.8235    | 1      | 0.9032 | 0.9211 |
| $ c  = 50$ | 0.8642    | 1      | 0.9272 | 0.9409 |

We conclude that  $\vartheta_a^t$ , which describes the authors-topics distribution, defines authors contributions in each topic. The characteristics of the authors during several intervals are studied through the transition function  $F(a_d^t)_{\tau^s}^{\tau^f}$ . In addition, integrating the two distributions,  $\vartheta_a$  and  $\eta_t$ , into the A-TOT model assists investigators in searching for authors-topics and topics over time, instead of relying on separate time-consuming computation.

For the purpose of estimating the precision and recall values regarding our mining algorithm, we compose randomly 100 chat logs as a combination of both aftermentioned datasets, perverted-justice and IRC, with minimum of 1,500 word tokens per log file. In Table 11, we list the mode values of precision, recall,  $F_1$ , and  $F_2$  measures for the two models previously described, LDA-TOT and A-TOT, after running the algorithm 10 times for 200 chat logs, 100 logs from the combination process and the other 100 are test logs that is described in Sect. 6.2. Both models found all the truth-relevant chat logs, achieving recall values of 1.0 for the two conditions ( $|c| = 30$  and  $|c| = 50$ ). For precision, there are 30 incorrect logs being retrieved for  $|c| = 30$  and 22 logs for  $|c| = 50$ ; therefore, the values are 0.8235 and 0.8642, respectively. The worst precision is 0.7910 and the best is 0.8485 for  $|c| = 30$  while running the algorithm 10 times. Similarly, for  $|c| = 50$ , the worst precision is 0.8434 and the best is 0.8861.

The different precision values with the two different sizes of  $c$  can be explained through  $KL(d, c)$ . Using fewer terms in  $c$  increases the  $KL(d, c)$  value and thus decreases the precision; vice versa is also true. The calculated results seem to be subjective. This is because the datasets are not large enough, and we expect precision to be low whenever the size of terms provided in  $c$  is small in a huge collection of data.

#### 6.4 Comparison with LDA and AT models

In this section, we compare our models with LDA and AT. We apply the same algorithm with LDA and AT and the same settings of parameters  $\epsilon = 3.3109$  and  $\gamma = 2.0977$  for  $\varrho = 0.5$ . From Tables 12 and 13, we determine that  $d_2$  is not crime-relevant, as with LDA-TOT and A-TOT, and  $d_1$ ,  $d_3$ , and  $d_4$  are crime-relevant. The topics' distributions,  $\theta$  and  $\vartheta$ , in  $d_1$ ,  $d_2$ ,  $d_3$ , and  $d_4$  for LDA and AT are illustrated in Tables 14 and 15. Both models, LDA and AT, are able to discover crime words in the extracted topics and somehow similar to the results obtained using LDA-TOT and A-TOT in Tables 6 and 10. By manual checking, we observe that LDA-TOT and A-TOT models extract  $\theta$  and  $\vartheta$  in which is better than the topics discovered using LDA and AT. Furthermore, the authors' distribution over topics in A-TOT is much better comparing with AT. The reason is that words in chatlogs are associated with time intervals and both models, LDA-TOT and A-TOT, embed this feature into the computation process unlike LDA and AT models. Although LDA and AT discovers similar topics distribution, the LDA-TOT and A-TOT models are still very useful for two factors: one, utilizing the topics over time and second, employing the authors-topics over time.

**Table 12** KL divergence and NMI between crime-relevant topics from documents ( $d_1, d_2, d_3, d_4$ ) and  $c$  when  $|c| = 30$  and  $|c| = 50$  using LDA

| Documents | Size            | $KL(t_{i,d}, c)$ | $NMI(t_{i,d}, c)$ |
|-----------|-----------------|------------------|-------------------|
| $d_1$     | $t_3( c  = 30)$ | 1.4080           | 0.0328            |
|           | $t_3( c  = 50)$ | 1.4214           | 0.0457            |
| $d_2$     | $t_2( c  = 30)$ | 1.2693           | 0.0589            |
|           | $t_0( c  = 50)$ | 1.2795           | 0.1037            |
| $d_3$     | $t_1( c  = 30)$ | 1.3101           | 0.0119            |
|           | $t_1( c  = 50)$ | 1.3257           | 0.0690            |
| $d_4$     | $t_1( c  = 30)$ | 1.3101           | 0.0119            |
|           | $t_2( c  = 50)$ | 1.3211           | 0.0249            |

**Table 13** KL divergence and NMI between crime-relevant topics from documents ( $d_1, d_2, d_3, d_4$ ) and  $c$  when  $|c| = 30$  and  $|c| = 50$  using AT

| Documents | Size            | $KL(t_{i,d}, c)$ | $NMI(t_{i,d}, c)$ |
|-----------|-----------------|------------------|-------------------|
| $d_1$     | $t_1( c  = 30)$ | 1.7662           | 0.0339            |
|           | $t_1( c  = 50)$ | 1.9558           | 0.0518            |
| $d_2$     | $t_1( c  = 30)$ | 1.2874           | 0.0075            |
|           | $t_2( c  = 50)$ | 1.3173           | 0.0206            |
| $d_3$     | $t_0( c  = 30)$ | 1.9482           | 0.3730            |
|           | $t_3( c  = 50)$ | 1.9533           | 0.1249            |
| $d_4$     | $t_3( c  = 30)$ | 1.5904           | 0.2105            |
|           | $t_1( c  = 50)$ | 1.5638           | 0.0900            |

In addition to the previous qualitative evaluating of our models, we estimate the power of LDA-TOT and A-TOT models by computing the perplexity of these models and comparing them with LDA and AT models. Perplexity is a standard measurement in document modeling that is used to measure the ability of a model to predict unseen data. Thus, the goal is to obtain high likelihood on a held-out test set. In other words, our objective is to evaluate the predictive power of the model in unseen data. A lower perplexity score indicates the higher the likelihood, and a better generalization performance can be achieved. For AT and A-TOT, we do not explore the range of perplexity scores that these models assign to test sets from specific authors [2]. Formally, the perplexity for a set of test documents  $D_{\text{test}}$  is defined as

$$\text{Perplexity}(w) = \exp \left\{ -\frac{\sum_{d \in D_{\text{test}}} \log p(w_d)}{\sum_{d \in D_{\text{test}}} N_d} \right\} \quad (17)$$

We train LDA, AT, LDA-TOT and A-TOT models on a set of 200 logs from *perverted-justice* while holding 100 logs, 50 logs from *perverted-justice*, and the other 50 from IRC, for computing and comparing the perplexity of the after-mentioned models. Figure 8 depicts the average runtime in minutes for LDA, AT, LDA-TOT, and A-TOT models on the training samples. It clearly shows that the LDA and LDA-TOT outperform the AT and A-TOT models. Figure 9 shows the average perplexity results for multiple numbers of the topics varying from 3 to 25.

As can be seen, the A-TOT model achieves a significant improvement on the generalization performance in test set with respect to LDA, AT, and LDA-TOT models. By utilizing authors and time intervals into the A-TOT model, A-TOT has the lowest perplexity among the other

**Table 14** Top 10 relevant words extracted for crime-relevant topics from documents ( $d_1, d_2, d_3, d_4$ ) and their distribution over documents using LDA

| $d_1$          |        |                |        | $d_2$          |        |                |        |
|----------------|--------|----------------|--------|----------------|--------|----------------|--------|
| $ c  = 30$     |        | $ c  = 50$     |        | $ c  = 30$     |        | $ c  = 50$     |        |
| $t_3$ (0.2038) | Prob   | $t_2$ (0.2977) | Prob   | $t_2$ (0.1703) | Prob   | $t_2$ (0.1526) | Prob   |
| rest           | 0.0178 | believ         | 0.0149 | think          | 0.0103 | wed            | 0.0186 |
| scream         | 0.0111 | gross          | 0.0148 | thank          | 0.0069 | good           | 0.0130 |
| wine           | 0.0089 | nice           | 0.0138 | wine           | 0.0069 | dai            | 0.0112 |
| butt           | 0.0089 | feel           | 0.0135 | grei           | 0.0069 | earlier        | 0.0093 |
| sweat          | 0.0067 | fuck           | 0.0126 | daughter       | 0.0035 | minut          | 0.0093 |
| nervou         | 0.0067 | butt           | 0.0122 | morn           | 0.0035 | talk           | 0.0093 |
| leg            | 0.0045 | sound          | 0.0122 | stomach        | 0.0035 | feel           | 0.0075 |
| pussi          | 0.0045 | believ         | 0.0122 | dinner         | 0.0035 | nice           | 0.0075 |
| crash          | 0.0045 | whatcha        | 0.0114 | internet       | 0.0035 | care           | 0.0075 |
| virgin         | 0.0045 | gross          | 0.0102 | appoint        | 0.0035 | hour           | 0.0075 |
| $d_3$          |        |                |        | $d_4$          |        |                |        |
| $ c  = 30$     |        | $ c  = 50$     |        | $ c  = 30$     |        | $ c  = 50$     |        |
| $t_1$ (0.2267) | Prob   | $t_1$ (0.1975) | Prob   | $t_2$ (0.1927) | Prob   | $t_2$ (0.1658) | Prob   |
| miss           | 0.0627 | see            | 0.0474 | cool           | 0.0169 | think          | 0.0222 |
| feel           | 0.0475 | time           | 0.0359 | see            | 0.0162 | cool           | 0.0176 |
| butt           | 0.0358 | luv            | 0.0310 | luv            | 0.0156 | see            | 0.0168 |
| wear           | 0.0214 | tell           | 0.0271 | look           | 0.0149 | luv            | 0.0162 |
| cam            | 0.0209 | look           | 0.0204 | pic            | 0.0148 | cum            | 0.0155 |
| kiss           | 0.0182 | miss           | 0.0196 | cum            | 0.0138 | girl           | 0.0153 |
| concentr       | 0.0157 | feel           | 0.0189 | tell           | 0.0129 | wear           | 0.0143 |
| scroll         | 0.0144 | butt           | 0.0176 | girl           | 0.0116 | nite           | 0.0134 |
| christma       | 0.0131 | kiss           | 0.0166 | wear           | 0.0116 | feel           | 0.0120 |
| friend         | 0.0131 | talk           | 0.0162 | nite           | 0.0116 | pussi          | 0.0120 |

models and over multiple numbers of topics. That is, A-TOT is a better model by assigning higher likelihood to held-out documents than LDA, AT, and LDA-TOT. Intuitively, all the four models have lower perplexity when  $|T| = 5$  and higher elsewhere. Furthermore, the perplexity escalates when  $|T| > 5$  for all of the models in Fig. 9. This can be justified by manually analyzing the test datasets, and we observe that the number of topics barely exceeds 5 to 6 topics, and therefore, the perplexity is high and this is the reason behind choosing  $|T| = 5$  in our experiments.

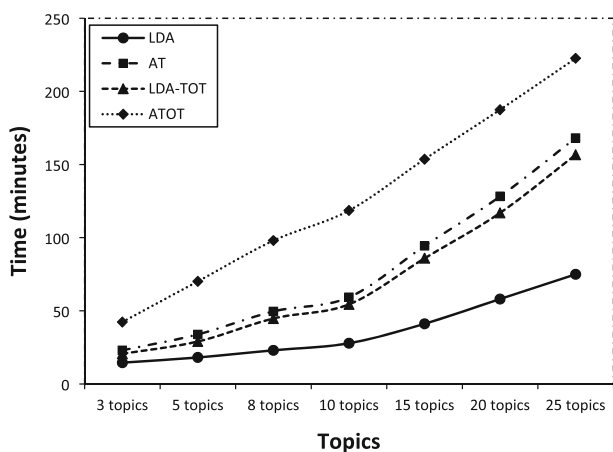
It is possible to further lower the perplexity of LDA-TOT and A-TOT by computing the time intervals in per word perplexity. But we do not conduct this direction because we focus on the illustrative part in our perplexity experiments and not necessarily conclusive. Finally, we note that the main focus for developing these two models is to reduce the computation and to capture temporal information, whether as topics over time, as LDA-TOT, or as authors-topics over time, as A-TOT, in addition to covering the first three differences in Sect. 1.

**Table 15** Top 10 relevant words extracted for crime-relevant topics from documents ( $d_1, d_2, d_3, d_4$ ), their distribution over documents, and their distribution over top 3 authors using AT

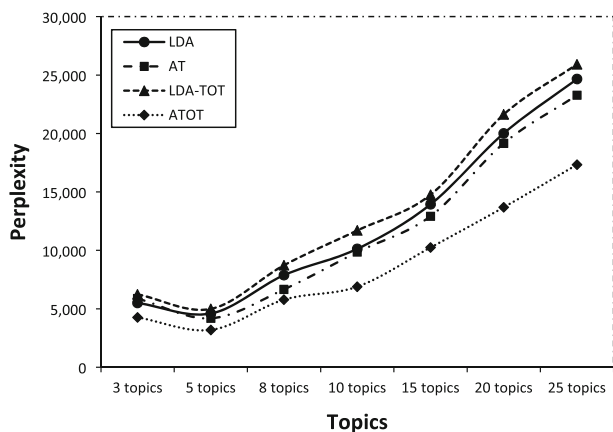
| $d_1$          |        |                |        | $d_2$          |        |                |        |
|----------------|--------|----------------|--------|----------------|--------|----------------|--------|
| $ c  = 30$     |        | $ c  = 50$     |        | $ c  = 30$     |        | $ c  = 50$     |        |
| $t_1$ (0.2182) | Prob   | $t_2$ (0.2380) | Prob   | $t_1$ (0.1856) | Prob   | $t_2$ (0.1550) | Prob   |
| think          | 0.0195 | friend         | 0.0262 | feel           | 0.0124 | wed            | 0.0226 |
| eat            | 0.0176 | nice           | 0.0258 | night          | 0.0124 | part           | 0.0129 |
| wonderin       | 0.0176 | fun            | 0.0210 | physic         | 0.0062 | feel           | 0.0129 |
| place          | 0.0113 | sweet          | 0.0114 | miss           | 0.0052 | morn           | 0.0097 |
| badli          | 0.0113 | watcha         | 0.0114 | work           | 0.0045 | pick           | 0.0065 |
| tire           | 0.0113 | skirt          | 0.0112 | monei          | 0.0031 | wine           | 0.0065 |
| front          | 0.0094 | luv            | 0.0098 | start          | 0.0031 | drive          | 0.0065 |
| lookin         | 0.0076 | sex            | 0.0096 | final          | 0.0031 | sunni          | 0.0033 |
| rite           | 0.0069 | kiss           | 0.0086 | job            | 0.0031 | place          | 0.0033 |
| friend         | 0.0063 | miss           | 0.0082 | blow           | 0.0031 | academi        | 0.0033 |
| Authors        | Prob   | Authors        | Prob   | Authors        | Prob   | Authors        | Prob   |
| $a_1$          | 0.2887 | $a_3$          | 0.2986 | $a_{10}$       | 0.2838 | $a_3$          | 0.2075 |
| $a_5$          | 0.2738 | $a_5$          | 0.2953 | $a_7$          | 0.2157 | $a_6$          | 0.2000 |
| $a_3$          | 0.2694 | $a_4$          | 0.2704 | $a_3$          | 0.2075 | $a_7$          | 0.1961 |
| $d_3$          |        |                |        | $d_4$          |        |                |        |
| $ c  = 30$     |        | $ c  = 50$     |        | $ c  = 30$     |        | $ c  = 50$     |        |
| $t_0$ (0.1598) | Prob   | $t_3$ (0.1745) | Prob   | $t_3$ (0.1959) | Prob   | $t_1$ (0.1969) | Prob   |
| luv            | 0.1305 | luv            | 0.0614 | nite           | 0.0436 | phone          | 0.0328 |
| look           | 0.0582 | miss           | 0.0580 | peopl          | 0.0409 | rub            | 0.0278 |
| peni           | 0.0524 | pretti         | 0.0443 | sex            | 0.0334 | whatcha        | 0.0265 |
| wish           | 0.0233 | sound          | 0.0390 | keep           | 0.0306 | brave          | 0.0215 |
| cam            | 0.0214 | stick          | 0.0363 | bad            | 0.0204 | boob           | 0.0152 |
| babi           | 0.0193 | night          | 0.0185 | fingerin       | 0.0158 | sex            | 0.0152 |
| bad            | 0.0111 | eat            | 0.0155 | leg            | 0.0130 | acoust         | 0.0126 |
| kiss           | 0.0111 | kiss           | 0.0127 | shave          | 0.0130 | masterb        | 0.0114 |
| excit          | 0.0111 | cum            | 0.0111 | wrong          | 0.0130 | chicken        | 0.0114 |
| dream          | 0.0111 | figur          | 0.0102 | imag           | 0.0121 | penis          | 0.0101 |
| Authors        | Prob   | Authors        | Prob   | Authors        | Prob   | Authors        | Prob   |
| $a_1$          | 0.2000 | $a_1$          | 0.2000 | $a_5$          | 0.1445 | $a_5$          | 0.1043 |
| $a_2$          | 0.1873 | $a_3$          | 0.1935 | $a_3$          | 0.1132 | $a_3$          | 0.0760 |
| $a_3$          | 0.1613 | $a_2$          | 0.1834 | $a_1$          | 0.1123 | $a_1$          | 0.0978 |

## 7 Conclusions and lessons learned

We propose an effective method to extract information from collections of documents. The collected information includes authors, topics, topics time-trends, and authors-topics over



**Fig. 8** Average time versions varying the number of topics for LDA, AT, LDA-TOT and A-TOT models on the training samples



**Fig. 9** Average perplexity for LDA, AT, LDA-TOT, and A-TOT models

time. We sought to study chat logs, in different granularity, to identify and segregate crime-relevant logs and topics associated with these logs. Next, we studied the concept of evolution of topics over time in order to explore the temporal information in these topics. We went a step further by exploring the activity of authors within these topics, which represents the evolution of authors-topics over time. In an attempt to build our proposed method, we developed two models, LDA-TOT and A-TOT, with multiple modality attributes influenced by three past models, LDA, AT, and TOT. As for evolution, we used discretization of time to capture different fluctuations of topics over discrete time stamps, instead of using continuous time as does the TOT model.

We conducted extensive empirical study of the proposed models by applying results on two real-life datasets, and we demonstrated that our approach can identify crime-relevant topics. Furthermore, based on topics expressed in a log and the activity of the authors, the system is capable of determining the most plausible authors.

Despite the advantages, probabilistic models, ours and in general, suffer from several shortcomings. We list the major limitations when applied to chat logs:

- **Document size.** Due to the short size of chat logs in general, it was hard to obtain the best mixture of topics  $\theta_d$  and the authors-topics distribution  $\vartheta_a$  when conducting experiments, and we applied the two algorithms for 10 times and captured the outcomes each times, and the displayed results are randomly selected. Therefore, we deduce that the accuracy of the extracted topics depends on the size of the chat logs. Although several works, such as [5], deal with short text environments (microblogging), such as Twitter, none of them define a proper method for dealing with texts in chat logs.
- **Input processing.** In many cases, we observe that depending on the “bag of words” assumption might not infer a true topic in a chat log. For example, if a chat log is related to a drug topic and drug-related terms occur a few times, the model might generate topics not related to drugs. Additionally, none of these models care much about the words processing. These words might contain a lot of noise, ambiguity, and even imprecision.
- **Users’ thresholds.** We used several thresholds in our experiments, such as the number of topics ( $|T|$ ). Though Teh et al. [24] proposed a *Hierarchical Dirichlet Processes model* that automatically infers the number of topics among the documents, other thresholds as  $F(a_d^t)_{\tau_s}^{\tau_f}$ , which lies outside the A-TOT model, are not automatically defined. Nonetheless, our choices are somewhat subjective, as there is no standard way to determine the optimal values.

These limitations motivate us to consider additional future research directions to address the limitations.

Finally, we would like to share our collaborative experience with the law enforcement sector. Criminal data are complex, often a combination of relational data, transaction data, and textual data. So far, our project focuses only on the textual data, but we notice that there is a pressing need for more crime data mining methods for heterogeneous types of data. Besides the technical issue, it is equally important to educate law enforcement management and frontline officers about the latest data mining technology. When management encounters a case that involves a large volume of digital data, the initial response is to allocate more team members to the case. In fact, alternative techniques are available that can significantly speed up the investigation process, such as the topic modeling technique presented in this paper.

**Acknowledgments** The authors would like to thank the anonymous reviewers for their constructive comments that greatly helped improve this paper. The research is supported in part by research grants from Le Fonds québécois de la recherche sur la nature et les technologies (FQRNT) new researchers start-up program, Concordia ENCS seed funding program, and the National Cyber-Forensics and Training Alliance Canada (NCFTA Canada).

## References

1. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. *J Mach Learn Res* 3:993–1022
2. Rosen-Zvi M, Griffiths T, Steyvers M, Smyth P (2004) The author-topic model for authors and documents. In: *Proceedings of the 20th UAI*, pp 487–494
3. Wang X, Mohanty N, McCallum A (2005) Group and topic discovery from relations and text. In: *Proceedings of the 3rd ACM LinkKDD*, pp 28–35
4. Ramage D, Hall D, Nallapati R, Manning CD (2009) Labeled lda: a supervised topic model for credit attribution in multi-labeled corpora. In: *Proceedings of the 2009 EMNLP*, vol 1, pp 248–256
5. Hong L, Davison BD (2010) Empirical study of topic modeling in twitter. In: *Proceedings of the 1st SOMA*, pp 80–88
6. Banerjee S, Agarwal N (2012) Analyzing collective behavior from blogs using swarm intelligence. *KAIS*, pp 1–25

7. Blei D, McAuliffe J (2008) Supervised topic models. *Adv Neural Inf Process Syst* 20:121–128
8. Lacoste-julien S, Sha F, Jordan MI (2008) DiscLDA: discriminative learning for dimensionality reduction and classification. In: *Proceedings of the 22nd NIPS*, pp 897–904
9. Ramage D, Heymann P, Manning CD, Garcia-Molina H (2009) Clustering the tagged web. In: *Proceedings of the 2nd ACM WSDM*, pp 54–63
10. Rubin T, Chambers A, Smyth P, Steyvers M (2012) Statistical topic models for multi-label document classification. *Mach Learn* 88:157–208
11. Chang J, Boyd-Graber J, Blei DM (2009) Connections between the lines: augmenting social networks with text. In: *Proceedings of the 15th ACM SIGKDD*, pp 169–178
12. Song X, Lin CY, Tseng BL, Sun MT (2005) Modeling and predicting personal information dissemination behavior. In: *Proceedings of the 11th ACM SIGKDD*, pp 479–488
13. Wang X, McCallum A (2006) Topics over time: a non-markov continuous-time model of topical trends. In: *Proceedings of the 12th ACM SIGKDD*, pp 424–433
14. Wang C, Blei DM, Heckerman D (2008) Continuous time dynamic topic models. In: *UAI'08*, pp 579–586
15. Blei DM, Lafferty JD (2006) Dynamic topic models. In: *Proceedings of the 23rd ICML*, pp 113–120
16. AlSumait L, Barab   D, Domeniconi C (2008) On-line lda: adaptive topic models for mining text streams with applications to topic detection and tracking. In: *Proceedings of the 8th IEEE ICDM*, pp 3–12
17. Du L, Buntine W, Jin H, Chen C (2012) Sequential latent dirichlet allocation. *KAIS* 31:475–503
18. Manning CD, Raghavan P, S  tze H (2008) *Introduction to information retrieval*. Cambridge University Press, Cambridge
19. Minka T, Lafferty J (2002) Expectation-propagation for the generative aspect model. In: *Proceedings of the 18th UAI*, pp 352–359
20. Griffiths TL, Steyvers M (2004) Finding scientific topics. *Proc Natl Acad Sci* 101:5228–5235
21. Heinrich G (2004) Parameter estimation for text analysis. Technical Report
22. Zhao WX, Jiang J, Weng J, He J, Lim EP, Yan H, Li X (2011) Comparing twitter and traditional media using topic models. In: *Proceedings of the 33rd ECIR*. Springer, Berlin, pp 338–349
23. PJF Inc. Chat log conviction numbers. Available: <http://www.ciise.concordia.ca/~fung/pub/convictions.txt>
24. Teh YW, Jordan MI, Beal MJ, Blei DM (2004) Sharing clusters among related groups: hierarchical dirichlet processes. In: *Proceedings of the 19th NIPS*, pp 1385–1392

## Author Biographies



**Abdur Rahman M. A. Basher** received a B.Sc. degree from King AbdulAziz University, Jeddah, Saudi Arabia, in 2008 and a M.A.Sc degree from Concordia University, Montreal, Canada, in 2011. Before pursuing his masters degree, he worked in the Center of Scientific Publishing as a professional software developer, Jeddah, Saudi Arabia from 2008 to 2009. His research interests include machine learning and pattern recognition, data mining, and bioinformatics.



**Benjamin C. M. Fung** is an Associate Professor at Concordia University in Montreal, and a research scientist of the National Cyber-Forensics and Training Alliance Canada. He received a Ph.D. degree in Computing Science from Simon Fraser University in 2007. He has over 60 refereed publications that span across the prestigious research forums of data mining, privacy protection, cyber forensics, web services, and building engineering. His data mining work in authorship analysis has been widely reported by media worldwide. Before pursuing his academic career, he worked at SAP Business Objects as a system software developer for 4 years. Dr. Fung is a licensed professional engineer in software engineering and is currently affiliated with the Computer Security Lab in CIISE.