Partition-then-Overlap Method for Labeling Cyber Threat Intelligence Reports by Topics over Time

Nagasawa, Ryusei ; Furumoto, Keisuke ; Takita, Makoto ; Shiraishi, Yoshiaki ; Takahashi, Takeshi ; Mohri, Masami ; Takano, Yasuhiro ;…

(Citation)
IEICE Transactions on Information and Systems,E104.D(5):556-561

(Issue Date)
2021-05

(Resource Type)
journal article

(Version)
Version of Record

IEICE TRANS. INF. & SYST., VOL.E104–D, NO.5 MAY 2021

LETTER *Special Section on Data Engineering and Information Management*

Partition-then-Overlap Method for Labeling Cyber Threat Intelligence Reports by Topics over Time

Ryusei NAGASAWA[†], ***Nonmember***, **Keisuke FURUMOTO**[††], **Makoto TAKITA**[†††], ***Members***, **Yoshiaki SHIRAISHI**[†,††††a)], ***Senior Member***, **Takeshi TAKAHASHI**[††], ***Member***, **Masami MOHRI**[†††††], ***Senior Member***, **Yasuhiro TAKANO**[†], ***Member***, ***and*** **Masakatu MORII**[†], ***Fellow***

SUMMARY The Topics over Time (TOT) model allows users to be aware of changes in certain topics over time. The proposed method inputs the divided dataset of security blog posts based on a fixed period using an overlap period to the TOT. The results suggest the extraction of topics that include malware and attack campaign names that are appropriate for the multi-labeling of cyber threat intelligence reports.

key words: *topic model, cyber threat intelligence, text mining, multi-labeling, security blog posts*

1. Introduction

Security blog posts published by security vendors include an analysis of threat information and alerts. Security blog posts are useful because they suggest methods to prevent and respond to cyberattacks.

However, the number of posts continues to increase day by day, and their contents change over time. It is not easy to find security blog posts that contain the desired content under such circumstances.

Security blog posts are occasionally labeled to aid in information retrieval, although the criteria for labeling are not standardized and vary from publisher to publisher. In addition, some posts do not have labels. In summary, there are no unified methods to search for desired information from a wide range of security blog posts.

It is necessary to assign appropriate multi-labels depending on their content to make it easy for security operators to obtain information from the security blog posts, which are increasing in number every day.

The goal of our study is to assign appropriate labels to security blog posts. This labeling is intended to allow security operators to collect relevant information associated with the subject of a search. Therefore, it is necessary to assign key phrases contained in a document to other documents as labels. Typical keyword extraction methods [1]–[3] cannot be used when attempting to achieve this goal because they extract keywords from the words in a document; thus, it is impossible to assign keywords that are not included in the document as labels. Named entity extraction [4], [5] cannot be used because it extracts key phrases from a document, whether it is supervised or unsupervised. In addition, when we extract named entities using supervised machine learning, labeled training data are required. However, publicly available trained models do not support security domain-specific key phrases (e.g., malware and attack campaign names). Thus, we cannot extract named entities appropriate for the labels.

Topic models are proposed as a statistical modeling method to obtain information from a large and heterogeneous set of documents. In addition to the LDA [6], which is a representative topic model, keyword extraction models based on LDA [7] and entity topic models (ETM) [8] have been developed. The multiple labels that meet the purpose of our study can be assigned to a document by understanding its topic.

The tendency of certain topics to occur in numerous document sets, including security blog posts, changes over time. However, the general topic model may result in unclear and suboptimal topics because it does not consider topic estimation.

The topics over time (TOT) model [9], which is a topic model that explicitly models time, is proposed to grasp topics in dynamic documents. The topics occurring in each document can be mapped in a time series using the TOT model.

The goal of this study is to assign appropriate labels to security blog posts. Our approach to meet this goal is as follows. First, we apply the security blog posts to the TOT model and generate time-sensitive topics. Next, we extract words with high compositional proportions from the generated topics and determine appropriate labels to be assigned to the posts.

However, it is unlikely that key phrases suitable as labels can be extracted when a dataset is entered as a batch into the above topic model, which includes TOT. Although multi labeling for security blog posts expects labels to include malware and attack campaign names, many of these named entities are not included in the documents. Therefore, to achieve the goal of this study, the extraction of a

Manuscript received June 26, 2020.
Manuscript revised November 16, 2020.
Manuscript publicized February 24, 2021.

[†]The authors are with the Department of Electrical and Electronic Engineering, Kobe University, Kobe-shi, 657–8501 Japan.
[††]The authors are with National Institute of Information and Communications Technology, Koganei-shi, 184–8795 Japan.
[†††]The author is with School of Social Information Science, University of Hyogo, Kobe-shi, 651–2197 Japan.
[††††]The author is with Center for Mathematical and Data Sciences, Kobe University, Kobe-shi, 657–8501 Japan.
[†††††]The author is with the Department of Electrical, Electronic and Computer Engineering, Gifu University, Gifu-shi, 501–1193 Japan.
a) E-mail: zenmei@port.kobe-u.ac.jp
DOI: 10.1587/transinf.2020DAL0002

security domain-specific vocabulary that appears locally is a problem to be solved.

Another difficulty with this problem is that common phrases in the security domain remain, even after preprocessing security blog posts with common natural language processing tools such as NLTK [10] to remove frequent words and stop words. These common phrases are widely distributed across the entire document set, whereas malware names and attack campaign names are distributed locally. It is therefore necessary to find phrases with a high frequency of occurrence locally, and not only phrases that are widely distributed throughout the document set. Partitioning of the dataset is expected to increase the likelihood that locally frequent phrases will be used in the organization of topics.

When a dataset is partitioned without an overlap period, if there is a concentration of important words near the partitioning points, these words will be divided. If we input the segmented data into the TOT, there is a possibility that important words will not be captured, and the topics related to these words will not be formed. The proposed method therefore divides the dataset by a fixed period with an overlap period and inputs it to TOT. It prevents malware names and attack campaign names from being buried by common phrases.

2. Proposed Method: Partition-then-Overlap for Labeling by TOT

2.1 Topics Over Time (TOT)

Before introducing the TOT model [9], our notations are summarized in Table 1; the graphical model representation of the TOT models is shown in Fig. 1. The TOT model is a topic model based on LDA [6]. The TOT model considers not only the co-occurrence information of words but also information on the document's published time in estimating topics. In other words, TOT prevents confusion with co-occurrence patterns and the occurrence of ambiguous and suboptimal topics by mapping the topics in a document to the time series.

Table 1 Symbol description

γ	number of topics
M	number of documents
N_d	number of word tokens in document d
θ_d	multinomial distribution of topics specific to document d
Φ_z	multinomial distribution of words specific to topic z
ψ_z	beta distribution of time specific to topic z
z_{di}	topic associated with ith token in document d
w_{di}	ith token in document d
t_{di}	timestamp associated with ith token in document d

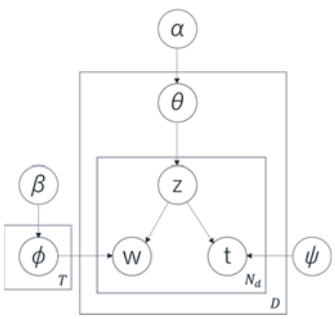

Fig. 1 TOT graphical model

TOT is a generative model of timestamps and words in timestamped documents. The TOT generative process is as follows. First, T multinomials Φ_z are drawn from a Dirichlet prior β for each topic z. For each document d, a multinomial θ_d is drawn from a Dirichlet prior α. Next, for each word w_{di} in document d, a topic z_{di} is drawn from a multinomial θ_d, a word w_{di} is drawn from a multinomial z_{di}, and a timestamp t_{di} is drawn from Beta $\psi_{z_{di}}$.

2.2 Partition-then-Overlap TOT

In the set of M documents to be analyzed, we define a dataset with an ordered time set, $T = \{t_1, t_2, \ldots, t_M\}$, and an ordered document set, $D = \{d_1, d_2, \ldots, d_M\}$, the lists of which are ordered by their publication dates. Here, $f: T \to D$ is an order-preserving bijection. With the proposed method, we divide the ordered set T, D into N pieces by specifying a fixed division period and overlap period.

For $i = 1, 2, \ldots, N$, we denote p_{2i-1}, p_{2i} as the dates at both ends of the partition period. We then denote by $T_{P_i} = \{t_x \in T \mid p_{2i-1} < t_x \leq p_{2i}\}$ as the set of publication dates of documents that exist in the partitioning period (p_{2i-1}, p_{2i}). Here, T_{P_i} is a subset of T and is an ordered set. We denote $P = \{(p_1, p_2), (p_3, p_4), \ldots, (2_{2N-1}, p_{2N})\}$ the set of partition periods. We denote by D_{P_i} the set of documents $d_x \in D$ corresponding one-to-one to $t_x \in T_{P_i}$ during each partitioning period. If $p_{2i+1} < p_{2i}$, partitioning periods (p_{2i-1}, p_{2i}) and (p_{2i+1}, p_{2i+2}) have overlapping dates during the period (p_{2i}, p_{2i+1}). That is, partitioned datasets have fixed overlap periods. Figure 2 shows an idea of the partition-then-overlap method.

We denote the ordered set families of the partitioned time set T_{P_i} and partitioned document set D_{P_i} as $T_P = \{T_{P_1}, T_{P_2}, \ldots, T_{P_N}\}$, and $D_P = \{D_{p_1}, D_{P_2}, \ldots, D_{P_N}\}$, respectively.

We preprocess the elements of the ordered set family D_P as

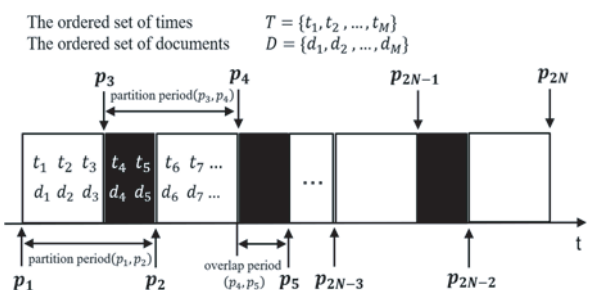

Fig. 2 Idea of the Partition-then-Overlap method

- Extract compound terms using TermExtract [11]
- Remove stop words
- Remove words with an occurrence rate of above 50%
- Remove words containing numbers, symbols, and quotation marks
- Remove words with less than four appearances

We denote by $W_i = \{w_1, w_2, \ldots, w_N\}$ the set of words made up of the remaining words and compound words after the preprocessing in a single document. We define the set of partitioned documents after preprocessing D_{P_i} as $\Delta_{P_i} = (W_1, W_2, \ldots, W_N)$, and then the ordered set family of Δ_{P_i} is $\Delta_P = (\Delta_{P_1}, \Delta_{P_2}, \ldots, \Delta_{P_N})$. Combining the ordered set families Δ_P and T_P, we construct the input set family as

$$(\{\Delta_{P_1}, T_{P_1}\}, \{\Delta_{P_2}, T_{P_2}\}, \ldots, \{\Delta_{P_N}, T_{P_N}\}).$$

The input set families are inputted into TOT by specifying the number of topics γ, the hyperparameters α and β, and the number of iterations δ. TOT outputs θ, ϕ, and ψ for $\{\Delta_{P_i}, T_{P_i}\}$. For an input set family, we obtain the output set family $(\{\theta_1, \phi_1, \psi_1\}, \{\theta_2, \phi_2, \psi_2\}, \ldots, \{\theta_N, \phi_N, \psi_N\})$.

From the topics obtained for each partitioning period, we estimate the characteristic topics containing key phrases that can be labeled. For topic estimation, we extract keywords with a high probability of belonging to topics using ϕ_i from the set of outputs of a partitioned period, $\{\theta_i, \phi_i, \psi_i\}$.

Defining K as the total number of words in the partitioned word set Δ_{P_i}, ϕ_i can be expressed as

$$\phi_i = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1\gamma} \\ h_{21} & h_{22} & \cdots & h_{2\gamma} \\ \vdots & \vdots & \ddots & \vdots \\ h_{K1} & h_{K2} & \cdots & h_{K\gamma} \end{bmatrix} \qquad (0 < i \leq N)$$

The columns of the two-dimensional array ϕ_i represent the multinomial distribution of topics specific to words, and the rows represent the multinomial distribution of words specific to a topic. Here, h_{kj} $(0 < k \leq K)$ represents the probability that word k belongs to a topic j. For each column vector of ϕ_i, Z words with the highest value of h_{kj} are extracted as the key phrases representing the topic. Appropriate labels were manually selected from the words that constitute the topics. In our study, we use malware names and attack campaign names as the leading labels.

3. Applying TOT to Security Blog Posts

The dataset consists of 2386 security blog posts from January 1, 2017 to December 31, 2019, collected from the blog pages of eight security vendors (Netscount, Barracuda, Cisco, Druva, FireEye, Paloalto, NortonLifeLock, and TrendMicro).

The implementation of the proposed method is based on Python3 with pandas, numpy, and scipy using a TOT code [12]. The experimental environment was Ubuntu 18.04 in Intel Core i7-7820X and 64 GB of memory.

The dataset was divided into a division period of 6 months and an overlap period of 3 months. As the initial setting, the number of topics $\gamma = 8$; the Dirichlet prior distribution hyperparameters α and β are $50/\gamma$ and 0.1, respectively; and the iteration number $\delta = 500$. The ordered set families Δ_P and T_P defined in Sect. 2.2 were entered into TOT. Table 1 shows the results.

In Table 2, we extracted the five words with a high proportion of topics for the number of topics specified for each partitioning period. Malware names and attack campaign names were underlined, and topics containing the words were enclosed in a bold frame.

From Table 2, we were able to grasp malware names and attack campaign names that can be labeled for each partition period. Malware names such as "wannacry" have long appeared, and malware names such as "bad rabbit" and "samsam" appeared only during a single partitioning period. We examined the relationship between the labels obtained in our experiments and the malware and attack campaigns that were prevalent at the same time. First, Meltdown and Spectre are both CPU vulnerabilities, and the article was published in November 2017 [13]. Since then, security updates for Meltdown and Spectre were made by Apple, Google, Microsoft, and other companies in March of 2018. The experimental results in Table 1 show that the Meltdown and Spectre labels appeared during the periods of "October 2017 to March 2018" and "January to June 2018." In addition, "mirai" is a malware that emerged around November 2016, and mirai variants have been active since December 2017 [14], [15]. The experimental results in Table 2 show that the mirai label appeared during the periods of "October 2017 to March 2018," "January to June 2018" and "April to September 2018." As shown in the above examples, the labels that appeared in the experimental results in Table 2 are thought to capture the malware and attack campaigns that were prevalent during the same period.

4. Evaluation of Partition-then-Overlap Method

We addressed the comparative results of the proposed method against the approach (batch method) in which the original dataset is entered into the TOT [9] and LDA [6] models in a batch. LDA was implemented using Gensim [16], an open-source library for unsupervised topic modeling and natural language processing. In the batch method, we set the number of topics to 46; the number of topics was derived using three ratings: KL divergence, pairwise cosine distance, and coherence. For hyperparameters α and β, we used the same values as in the proposed method applied to TOT and the default values of Gensim in LDA. The results when entering the datasets into TOT and LDA in batches are shown in Tables 3 and 4.

We compared the extracted malware names and attack campaign names from the results of the proposed method when applying the TOT (with the batch method) and LDA (with the batch method). Malware names and attack campaign names are useful as labels for security blog posts and significantly affect the accuracy of the labels. Therefore, we

Table 2 Result of entering partitioned datasets into TOT ($\gamma = 8$, $\alpha = 50/\gamma$, and $\beta = 0.1$)

2017/1–6	2017/4–9	2017/7–12	2017/10–2018/3	2018/1–6	2018/4–9	2018/7–12	2018/10–2019/3	2019/1–6	2019/4–9	2019/7–12
wannacry	**wannacry**	iot device	**wannacry**	ise	**wannacry**	**wannacry**	classifier	com	threat hunting	tco
exploit kit	tcp	packet	threat grid	smbs	istr	threat grid	**wannacry**	rdp	threat detection	directory
botnet	petya	nist csf	bitcoin	**wannacry**	healthcare organization	umbrella	smb	registry	medr	disaster recovery
cerber	smb	**wannacry**	**bad rabbit**	healthcare industry	medical device	threat hunting	certificate	scam	cybersecurity team	vms
com	steal data	worm	dashboard	transparent	healthcare industry	imperative	**samsam**	smb	active directory	ciso
shamoon	security vendor	ise	casb	federal agency	carrier	blockchain	federal agency	iot device	casb	xdr
shellcode	cloud ready	nonprofit	cloud apps	vms	apple	carrier	devops	federal agency	dlp	backup system
hitrust	msp	identity theft	equifax	salesforce	installation	transaction	endpoint detection	threat hunting	business unit	siem
middle east	msps	iot device	disclose	cloud environment	china	byte	cdm	threat response	workforce	magecart
python	hybrid environment	trustsec	dropbox	saas	ios device	middle east	macro	disclose	disclose	prevention
wannacry	linux	**petya**	shopper	threat grid	google play	federal agency	small business	red team	tweet	volatility
petya	exploit kit	bitcoin	uber	appdata	phone number	msps	ai model	accuracy	iran	dlp
fbi	**petya**	**wannacry**	cio	javascript	mobile security	prevention	cybersecurity team	istr	com	apis
smb	ukraine	**ddos attack**	holiday season	sep	social network	cio	flaw	taa	dataset	webinar
ukraine	kit	**notpetya**	eta	ciso	threat detection	attendee	germany	msps	align	workforce
certificate	macro	linux	sep	macro	appdata	atm	scam	mitre att	dns	shellcode
dlp	india	bot	federal agency	ios device	hash	small business	scammer	meantime	msps	binary
online safety	proxy	security vendor	javascript	vmware	javascript	installation	botnet	fraud	webinar	utility
proxy	align	webinar	ise	binary	macro	apple	china	medical device	backup system	byte
confidential data	legacy	likelihood	mobile security	rdp	dll	ciso	bot	ponemon institute	cloud workload	threat grid
medical device	nist csf	dmarc	iot device	**mirai**	blockchain	scam	atm	casb	email threat	threat hunting
apple	healthcare organization	google play	**mirai**	iot device	**mirai**	exploit kit	theft	rto	query	workplace
ciso	cerber	sender	dns	dns	transaction	taa	emea	saas	response team	ise
trump administratio n	congress	smbs	vms	**ddos attack**	cryptocurrenc y mining	japan	steal data	ciso	downtime	local government
wmi	hitrust	console	default password	connect device	taa	governance	transaction	security program	conjunction	duo
keen lab	bot	asaf	dmarc	scammer	threat grid	ise	ise	active directory	red team	msps
team sniper	android device	shopper	**meltdown**	**meltdown**	binary	vmware	taa	malware analysis	devops	pillar
tencent security	ddos attack	invoice	controller	google play	com	api	msps	destination	cio	meantime
uaf	dlp	impersonatio n	healthcare organization	**spectre**	ciso	cwp	apis	sensor	ttps	it team
bot	google play	redirect	**spectre**	shellcode	saas	saas	api	dlp	appliance	It professional
conference	medical device	controller	bot	blockchain	**bec**	**bec**	iot device	devops	scam	scam
india	shellcode	kit	directory	hash	msps	dlp	rdp	china	**bec**	scammer
threat grid	iot device	cloud apps	temp	bitcoin	scam	devops	com	bot	scammer	**bec**
amp	endpoint protection	it organization	folder	miner	smbs	iiot	registry	login credential	fraud	dns
api	worm	business leader	packet	transaction	mssp	entity	**bec**	guidance	gmail	potential victim
dpa	bitcoin	equifax	scammer	folder	federal agency	iot device	iiot	source code	source code	cybersecurity team
csf	authority	south korea	google play	cloud apps	dlp	com	umbrella	**carbanak**	xdr	threat detection
macro	cloudsoc	com	middle east	casb	vpnfilter	**samsam**	persistent threat	scammer	**carbanak**	linux
tcp	nonprofit	cloudsoc	transparency	apple	simulation	bot	threat hunting	scam	siem	google play
nist csf	threat grid	ncsam	transparent	mobile security	ise	binary	threat response	binary	ciso	medr

used them in our evaluation method.

As indicated in Tables 2, 3, and 4, we extracted 12 malware and attack campaign names when applying the proposed method, 10 names when applying TOT (with the batch method), and 6 names when applying LDA (with the batch method). The proposed method extracted the greatest number of malware and attack campaign names because it could extract the names of the malware and attack campaigns that appear locally in the document set. In particular, "bad rabbit," "samsam," "notpetya," and "meltdown" are only extracted by the proposed method. Documents containing them were published within a short period of time. Using the proposed method, we extracted the names that appeared locally in the document set. Thus, we could extract more key phrases as label candidates than either the TOT (with the batch method) or LDA (with the batch method) models. The proposed method is effective in appropriately labeling security blog posts, which is the objective of our study.

However, with the batch method, we extracted the malware and attack campaign names that were not extracted using the proposed method, i.e., "emotet," "trick-

Table 3 Result of entering the dataset into TOT in a batch ($\gamma = 46$, $\alpha = 50/\gamma$, and $\beta = 0.1$)

salesforce	**petya**	persistence	small business	macro	byte	wordpress	vmware	dmarc	api
trump administration	tweet	binary	security patch	source code	smbs	authentication	dlp	pxgrid	apis
hitrust	cio	siem	workforce	misconfiguration	query	hxxp	ios device	outlook	redacted
headline	credential theft	scheduled task	sep	stack	netflow	**spectre**	hijack	splunk	ukraine
it team	submission	malicious script	resilient	nation state	packet	macro	full visibility	cache	fileless malware
atm	keen lab	redirect	dns	blockchain	registry	ise	iiot	**wannacry**	
cybersecurity team	saas	google play	**bec**	transaction	powershell script	taa	classifier	microsoft edge	
linux	following command	email account	bot	umbrella	wmi	disaster recovery	eta	bitcoin	
istr	security update	germany	encrypted traffic	data subject	msps	bulletin	usage	tencent security	
audit	certificate	name pipe	bec attack	dpo	equifax	threat hunting	hipaa	contestant	
threat grid	shellcode	**wannacry**	nist csf	**mirai**	bot	iot device	apple	com	
mobile security	prevention	devops	financial institution	iot device	webinar	attendee	flaw	china	
email threat	utility	certificate	headline	it organization	medical device	rsa	malicious document	com object	
security service	top priority	cyber risk	security strategy	installation	docker	noc	team sniper	cloudsoc	
coinminer	obtain	devops team	utilize	datacenter	sep	business process	sep mobile	retailer	
ascii	federal agency	**shamoon**	carrier	certification	medical device	deep web	scam	exploit kit	
emotet	**carbanak**	middle east	sensor	nonprofit	prediction	default password	likelihood	workforce	
parameter	casb	iran	unmanaged device	endpoint detection	android device	appliance	configuration file	asaf	
critical system	miner	saudi arabia	data backup	online safety	outage	inception	shadow it	hxxps	
fraud	amazon	security analyst	healthcare industry	leak	susceptible	pawn storm	iocs	security practitioner	
appdata	playbook	rdp	red team	**ddos attack**	threat grid	worm	scammer	tcp	
threat detection	**triton**	healthcare organization	hash	vpn	asert	vms	iot device	bec	
javascript	controller	magecart	folder	compliant	rto	emulation	ciso	baseline	
fake news	home network	guidance	dot	digital world	behavioral indicator	mitre att	tor	python	
airport	connect device	delete	mimikatz	entity	disclosure	classify	smb	dashboard	

Table 4 Result of entering the dataset into LDA in a batch ($\gamma = 46$, $\alpha = 1/\gamma$, and $\beta = 1/\gamma$)

apis	federal agency	certificate	**bec**	com object	red team	security analytics	**wannacry**	iran	msps
api	dhs	web application	**bec scam**	backup system	case study	corporate resource	malware analysis	volatility	template
encrypt traffic	federal government	sep	playbook	dns	ftc	india	salesforce	tweet	encrypted traffic
unmanaged device	malicious bot	dns server	potential victim	business executive	washington	threat hunting	stack	singapore	msp
nsa	tipp	guidance	**bec attack**	cybersecurity breach	blue team	endpoint protection	devops	accountability	trustworthy
pillar	pii	cybersecurity team	google play	sep mobile	splunk	appliance	registry key	guardrail	
shamoon	personal email	disclose	malicious apps	timestamp	csf	business unit	mitre att	android device	
five pillar	equifax	smart home	mobile security	user account	nist csf	workforce	outlook	headline	
credential theft	public internet	it organization	leak	atm	dpa	ciso	such attack	insurer	
prasanna malaiyandi	personal account	dump	malicious app	it professional	secops team	many customer	accuracy	suspicious email	
online account	prediction	magecart	**trickbot**	tweet	macro	ise	retailer	com	
password manager	vms	british airway	iiot	cwp	phone number	duo	data backup	persona	
workplace	disaster recovery	tor	engagement	severity	utility	mfa	smbs	investigator	
governance	tco	shortage	bos	docker	code execution	threat hunting	firepower	identical	
login	rpo	concise	business process	emea	dridex	threat hunt	it team	middle east	
registry	webinar	casb	metasploit	vmware	binary	threat grid	blockchain	workforce	
folder	siem	politic	entity	**emotet**	shellcode	persistence	dlp	educational institution	
window server	agile	cloud apps	documentation	shellcode	source code	excel	vpn	transparency	
active directory	silo	it leader	**petya**	enterprise network	byte	noc	exploit kit	cto	
prevention	netflow	branch office	ukraine	byod	response team	artifact	transaction	violation	
small business	iot device	certification	cdo	thrip	threat detection	medical device	scammer	classifier	
dmarc	dataset	workplace	airport	taa	endpoint detection	email attack	scam	home network	
downtime	botnet	saas	enforcement	mati	ediscovery	simulation	local government	mar	
devsecops	bot	log data	cloud workload	mimikatz	platform approach	verizon	relevance	home router	
bandwidth	ddos	traditional endpoint	security analytic	psexec	medr	gmail	bitcoin	secure access	

bot," and "triton" because the documentation on these malware names was widely distributed over a five-part period (1.5 years). Moreover, unlike the documents on "wannacry" and "bec," which are also widely distributed, there are few documents on these types of malware. Therefore, the proposed method, which is a partitioning approach for extracting local key phrases, could not capture "emotet," "trickbot," or "triton."

5. Conclusions

In this paper, we proposed a method for constructing data into TOT that allows us to extract distinctive labels from security blog posts. The key idea is to not enter the datasets into TOT in batches (using a batch method), but to instead enter the partitioned dataset into TOT with a fixed period of overlap. Our proposed method captures malware and attack campaign names that appear locally and extracts key phrases that can be more useful labels than when applying the batch method. In addition, by adding overlaps, we could extract malware names such as "bad rabbit," which had been buried when using the partitioning method. Therefore, we can state that the partition-then-overlap method is useful for extracting key phrases that can be used as labels. By using the results of both the batch method and the partition-then-overlap method, we obtain more appropriate search results. It is a future task to confirm the usability of a search system of the security blog post with these labels through user experiments.

Acknowledgments

This research was conducted under a contract of "Research and development on IoT malware removal/make it non-functional technologies for effective use of the radio spectrum" among "Research and Development for Expansion of Radio Wave Resources (JPJ000254)", which was supported by the Ministry of Internal Affairs and Communications, Japan.

References

[1] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic keyword extraction from individual documents," in Text Mining: Applications and Theory, eds. M.W. Berry and J. Kogan, Wiley Online, pp.1–20, 2010.
[2] R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts," Proc. 2004 Conference on Empirical Methods in Natural Language Processing, pp.404–411, 2004.
[3] I.H. Witten, G.W. Paynter, E. Frank, C. Gutwin, and C.G. Nevill-Manning, "KEA: Practical automatic keyphrase extraction," Proc. Fourth ACM Conference on Digital Libraries, pp.254–255, 1999.
[4] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D.S. Weld, and A. Yates, "Unsupervised named-entity extraction from the web: An experimental study," Artificial Intelligence, vol.165, no.1, pp.91–134, 2005.
[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," Proc. NAACL-HLT 2019, pp.4171–4186, 2019.
[6] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent dirichlet allocation," Journal of Machine Learning Research, vol.3, pp.993–1022, 2003.
[7] J. Park, J. Kim, and J.-H. Lee, "Keyword extraction for blogs based on content richness," Journal of Information Science, vol.40, no.1 pp.38–49, 2014.
[8] H. Kim, Y. Sun, J. Hockenmaier, and J. Han, "Etm: Entity topic models for mining documents associated with entities," Proc. 2012 IEEE 12th International Conference on Data Mining, pp.349–358, 2012.
[9] X. Wang and A. McCallum, "Topics over Time: A non-Markov continuous-time model of topical trends," Proc. 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.424–433, 2006.
[10] Natural Language Toolkit, https://www.nltk.org/.
[11] pytermextract, http://gensen.dl.itc.u-tokyo.ac.jp/pytermextract/. (in Japanese).
[12] Topics Over Time, https://github.com/ahmaurya/topics_over_time
[13] Kernel index [LWN.net] - Meltdown and Spectre, https://lwn.net/Kernel/Index/#Security-Meltdown_and_Spectre.
[14] Rise of One More Mirai Worm Variant, https://www.fortinet.com/blog/threat-research/rise-of-one-more-mirai-worm-variant.
[15] Warning: Satori, a Mirai Branch is Spreading in Worm Style on Port 37215 and 52869, http://blog.netlab.360.com/warning-satori-a-new-mirai-variant-is-spreading-in-worm-style-on-port-37215-and-52869-en/.
[16] Gensim · PyPI, https://pypi.org/project/gensim/.