The impact of preprocessing in natural language for open source intelligence and criminal investigation

Jan William Johnsen
Dep. of Information Security and Communication Technology
Norwegian University of Science and Technology
Gjøvik, Norway
jan.w.johnsen@ieee.org

Katrin Franke
Dep. of Information Security and Communication Technology
Norwegian University of Science and Technology
Gjøvik, Norway
kyfranke@ieee.org

***Abstract*—Underground forums serves as gathering place for like-minded cyber criminals and are an continued threat to law and order. Law enforcement agencies can use Open-Source Intelligence (OSINT) to gather valuable information to proactively counter existing and new threats. For example, by shifting criminal investigation's focus onto certain cyber criminals with large impact in underground forums and related criminal business models. This paper presents our study on text preprocessing requirements and document construction for the topic model algorithm Latent Dirichlet Allocation (LDA). We identify a set of preprocessing requirements based on literature review and demonstrate them on a real-world forum, similar to those used by cyber criminals. Our result show that topic modelling processes needs to follow a very strict procedure to provide significant result that can be useful in OSINT. Additionally, more reliable results are produced by tuning the hyper-parameters and the number of topics for LDA. We demonstrate improved results by iterative preprocessing to continuously improve the model, which provide more coherent and focused topics.**

***Index Terms*—Digital forensics, Latent Dirichlet Allocation, reliability, document construction, underground marketplace, criminal investigation**

I. Introduction

OSINT exploits publicly available data such as pictures, video and text to piece together factual data – i.e. information – for an end goal. Two overlapping developments have particularly influenced the growth of OSINT: expansion of social media and big data [13]. Social media is a good example of big data in practice, as tons of user-produced videos and texts are uploaded onto the Internet every day. Information gathered from open sources can give insights into world events, however, piecing together relevant data from the vast sea of materials can be difficult. Furthermore, big data majorly consists of unstructured data, which current traditional analytical tools are not built to handle.

Researchers frequently repeat the '80 per cent rule', which refer to the quantification of open-source contribution to intelligence [10], [18]. It is difficult to put an estimate on how much OSINT contribute to an intelligence operation and the 80 per cent number is generally considered a mischievous red herring [10]; however, it provides an opportunity where OSINT can offer *significant value to proactive Cyber Threat Intelligence (CTI) to organisations about threats they were not previously aware of* [5], [17]. Consequently, data acquisition from OSINT are largely automated and can cause an increase in false positives [17]. In other words, the result of automated processes can have a negative effect on information reliability.

Law enforcement agencies has primarily used reactive approaches in criminal investigations for decades. New proactive approaches and utilising vast amount of unstructured data can assist law enforcement agencies to prevent crime and uphold the law. Information is key to any criminal investigation [2], where information is constructed from data. However, correctly structuring, analysing and extracting useful knowledge or facts from unstructured data is a challenge. The goal is to gather sufficient information to accurately and adequately explain circumstances of a situation or incident. Additionally, *the reliability and validity of data can change with attributes to the data source and the methods used to process the data* [2].

One goal of OSINT is to make sense of a lot of unstructured data, e.g. by automatically analyse various discussion forums to understand new trends or progression of malware development. Natural Language Processing (NLP) is used to process and analyse large amounts of natural language data, where LDA is one of the more popular algorithms. LDA is a generative statistical model, commonly used to categorise a set of observations (i.e. text) into unobserved groups that explain why some parts of the data are similar. LDA is described further in Section III.

Every algorithm, including LDA, is susceptible to the expression 'garbage in, garbage out'. In other words, results will be incorrect if the input is erroneous, regardless of the algorithm's accuracy. The way these LDA models are trained, and in particularly how their inputs are preprocessed (if at all) is something we find missing in previous research. Therefore, our research concentrates on improving our current understanding for how to best construct documents as input for the LDA algorithm. We first briefly explain how the Machine Learning (ML) and forensic process model can be linked, and then we define which requirements must apply for using LDA in a digital forensic context. With these requirements in mind, we will cross validate three different document

The research leading to these results has received funding from the Research Council of Norway programme IKTPLUSS, under the R&D project 'Ars Forensica - Computational Forensics for Large-scale Fraud Detection, Crime Investigation & Prevention', grant agreement 248094/O70.

construction methods for LDA and study it in detail on the Nulled dataset. We primarily focus on OSINT in the context of digital forensics, but it will hold the same for intelligence operations.

Recently, LDA has been widely studied from a digital forensic perspective. Anwar et al. [3] analyse authorship attribution for Urdu text; Porter [14] splits his dataset into time intervals to find evolution of hacker tools and trends; Caines et al. [6] uses ML and rule-based classifiers to automatically label post type and intent from posts in underground forum; Samtani et al. [15] designed a novel CTI framework to analyse and understand threats present in hacker communities; L'huillier et al. [12] combine text mining and social network analysis to extract key members from darkweb forums.

Text preprocessing varies widely in these studies, e.g. grammatical mistakes and word preferences are relevant in authorship attribution [3] or hacker forums contain atypical language [14]. They have a few issues, such as using Google Translate to convert text into English [15] or not checking model fit [12]. Additionally, they frequently do not describe how they structure the LDA input.

This article is structured in the following way: Section II describes previous and relevant work for our research, linking the ML and forensic process model and defining LDA preprocessing requirements; Section III and IV report any preprocessing on the data, define the LDA document construction and provide results of our real-world scenario demonstration. We discuss the significance of our results and give a recapitulation of this article in Section V.

II. Previous work

Data preprocessing is an integral step from the perspective of the ML process model – as described by Kononenko and Kukar [11] – where data quality directly affects the ability of ML models to learn. Furthermore, a survey by CrowdFlower [8] found that 60 per cent of the professionals spend much of their time cleaning and organising data. The same emphasis of data quality also holds for digital forensics. Andersen [2] gives details of the digital forensic process, in relation to criminal cases. He points out that information is crucial, and it should be reliable to have any value in a court of law. It is beyond this article to have a complete comparison of both process models, but there is a mutual understanding in both domains that the preprocessing phase is the most crucial step. Data preprocessing is a time-consuming and crucial step, that consolidate and structure data to improve the accuracy of results.

Both the user of a system and the system itself have some requirements for it to be accurate and precise, i.e. reliable. We focus our requirements from the user's perspective: what they need to do to adeptly use the system, such as LDA in a digital forensic context. Text analysis typically begins with preprocessing the input data, but related literature varies widely with regards to which preprocessing method they utilise. Requirements should improve the algorithms' ability to identify interesting or important patterns in the data, instead of noise. The following list is composed of some common recommendations for cleaning the data [9], [14].

- **Word normalisation:** Inflected languages modifies words to express different grammatical categories. *Stemming* and *lemmatisation* are two methods to normalise text, as they help find the root form of words. Stemming removes suffixes or prefixes used with a word, without considering the resulting word belongs to a language. Lemmatisation reduces the inflected words properly while ensuring that the root word belongs to the language.
- **Stop word removal:** Words that are generally the most common words in a language, which tend to be over-represented in the result unless removed. They do not contain any important significance. However, removing stop words indiscriminately means you can accidentally filter out important data.
- **Uninformative word removal:** Similar to stop word removal, however, it is a domain specific list of uninformative words. It can be quite long and depend on the domain producing the text in question.
- **Word length removal:** Remove words that have fewer than x (e.g. three) characters.
- **Document de-duplication:** Eliminating duplicate copies of repeating data, i.e. removing identical documents that appear frequently.
- **Expanding/replacing acronyms:** Acronyms are used quite often and may need some subject matter expertise to understand.
- **Other:** Convert everything to lowercase and remove punctuation marks/special symbols. Finally, remove extra white-spaces.

Requirements which reduce the vocabulary size has clear advantages for the quality. For example, removing stop words leave remaining terms that convey clearly topic-specific semantic content. Schofield et al. [16] looked at some of the common practices we have listed and found that many have either no effect or a negative effect. For example: i) effects from document duplication were minimal until they had a substantial proportion of the corpus; ii) stop word removal (determiners, conjunctions and prepositions) can improve model fit and quality; and iii) stemming methods perform worse.

III. Methodology

There are several topic modelling algorithms [1], however, we selected LDA because it is typically more effective and generalises better than other algorithms. This is beneficial as our proposed method may generalise to more specific domains, such as those of underground forums. Furthermore, LDA can extract human-interpretative topics from a document corpus, where each topic is characterised by the words they are most associated with. LDA [4] is a way of 'soft clustering' using a set of documents and a pre-defined k number of topics. Each document has some probability to belonging to several topics, which allow for a nuanced way of categorising documents.

The three hyper-parameters k, α and η adjust the LDA learning. Where k is a predefined amount of topics and α

and η regulate two Dirichlet distributions. These Dirichlet distributions adjust the LDA model document-topic density and topic-word density, respectively. More specifically, LDA models assumes documents consists of fewer topics at low α values, while higher α values documents can consist of more than one topic. Higher values will likely produce a more uniform distribution, so a document will have an even mixture of all the topics. Hyper-parameter η works similarly, but adjust the word distribution per topic. Thus, topics consist of less words at low η values and more words at higher values. LDA is most commonly used to i) shrink a large corpus of text to some sequence of keywords, ii) reduce the task of clustering or searching a huge number of documents, iii) summarise a large collection of text or iv) automatically tag new incoming text by the learned topics.

We use the previously mentioned requirements and pre-processing recommendations from Schofield et al. [16], such as removing about 700 of the most common English stop words. Following their recommendations, we decided to not remove duplicated documents nor use stemming, as this was reported to have little effect. We removed additional text such as HTML tags (incl. their attributes), HTML entities (e.g.), symbols and extra spaces. Finally, we removed all rows with an empty text field and converted everything to lower case characters.

Users can write public *posts* to communicate with other forum users. These posts can have two distinctions: a *subject* is started by an initial post by a user, while other users are able to *reply* with their own posts to subjects. There is always zero or more replies associated with each subject. Figure 1 illustrate this type of interactions between users, where each user is depicted with different colours.

We focused our document construction method on the criteria to include all available posts found on the forum and ended up with identifying three distinct ways that we named: A, B and C. Figure 1 also portrays these document construction methods, where A is subject-centred, B is subject-user-centred and C is user-centred. Other construction approaches, than those shown in Figure 1, can be created and would yield different results. However, we decided to not consider them further as they would have too much information loss due to ignoring many posts.

Construction A keep the original subject-structure found on the forum. In other words, one document is the combination of the subject starter and all its replies. Construction B builds upon this idea of being subject-centred. However, this approach combines the posts from users in a subject into separate documents. Finally, construction C combines all posts for distinct users into a separate document; i.e. one document consists of all posts that has been written by a specific user.

The motivation for construction A is to capture the overall activity on the forum, to get a high-level overview of topics that users are talking about. However, combining all posts from various users per subject might obscure the result. Therefore, we designed construction B to be subject and user-centred, as this could produce a more accurate result. While construction C is user-centric and should capture more of the interests for forum users.

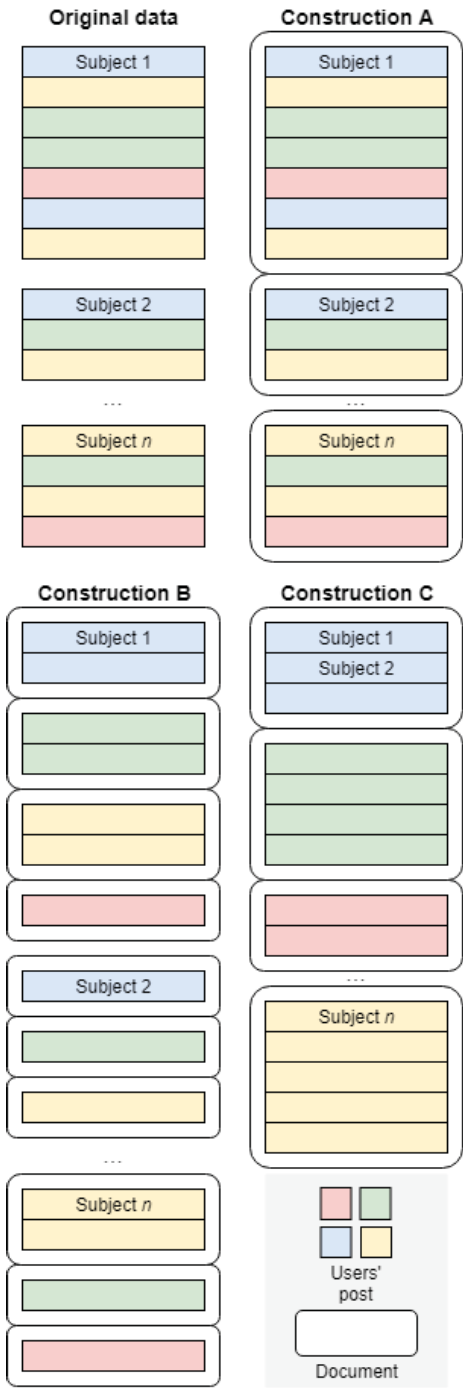

Fig. 1. Document construction approaches analysed in this article. Construction A is subject-centred; construction B is subject-user-centred; while construction C is user-centred. Unique users are marked with different colours.

The number of latent topics, k, is a parameter we have to set in LDA models; we explore $k = 10$, 20, 30, 40, 50 and 60 in our experiment. The other parameters α and η are either inferred from the data ($\frac{1}{k}$ when they are set to *None*) or set to

the values 0.05, 0.1, 0.5, 1, 5 and 10.

Finally, we have to evaluate the model quality after the unsupervised learning process. We use k-fold cross-validation to assess how well the LDA models will generalise to an independent data set. For each analysis, we split the data into five folds: each fold is used for training the LDA model four times and testing the model one time. We use *perplexity* to objectively measure how well our model predicts the testing fold, where a low perplexity score indicates a better model. Furthermore, we use mean perplexity (i.e. the arithmetic mean for each fold) to compare all 882 ($k \times \alpha \times \eta$ combinations) models between each other. We select models with the lowest perplexity for further manual inspection.

IV. Experiment and results

We explicitly concentrate our attention on data preprocessing in this research article, where LDA document construction is centre. It is, therefore, out of our scope to focus on the data gathering process, such as running web scraping tools to extract OSINT from real-world underground forums. Instead, we will use a dataset of 'Nulled' that was leaked in May 2016. It is a hacker forum on the deep web, that facilitate the brokering of compromised passwords, stolen bitcoins and other sensitive data. Nulled's Structured Query Language (SQL) database was leaked in its original form, without any filtration or preprocessing. Their database contained details about 599 085 user accounts, 800 593 private messages and 3 495 596 public messages. We imported it to a MySQL server and exported the necessary information from tables and fields with a Python script, using the Pandas package. More specifically, we stored information found in database tables 'topics' and 'posts' (columns: 'author_id', 'post', 'topic_id') in a file for further analysis.

We used Pandas to group the three construction methods following the design described in Section III and depicted in Figure 1. The text column 'post' was further processed (described in Section III) to make it suitable for LDA and document generation. We fit the LDA algorithm from the Scikit-learn package, for all the possible parameter combinations. All three document construction approaches was analysed using 294 distinct combinations of LDA hyper-parameters. We ran a total of 882 (294×3) LDA analyses to find the optimal combination of parameters. Table I shows the best ten models with the lowest perplexity.

Interestingly, our best result had very low hyper-parameters and 10 topics. While Samtani et al. [15] found an optimal topic number ranging from between 80 and 100. More importantly, Chang et al. [7] found that perplexity is not strongly correlated to human interpretation, as they found that the most frequent words in topics usually do not describe a coherent idea for those topics. A human forensic analyst would at least manage to interpret and understand fewer topics than something like 80 and 100 topics. However, fewer topics with a low perplexity score are not guaranteed to be easier interpreted by a human analyst. An important note is that low hyper-parameters also result in a slower convergence rate. While this solution might not be suitable for any time-critical criminal investigation, it could be applied to proactive OSINT gathering.

TABLE I
Ten best models with hyper-parameter combinations

Construction A				
#	α	η	k	Perplexity
1	0.05	0.05	10	5855.00
2	0.10	0.05	10	5886.47
3	None	0.05	10	5960.00
4	0.50	0.05	10	6035.86
5	0.05	0.10	10	6279.13
6	1.00	0.05	10	6299.63
7	None	None	10	6325.16
8	0.10	None	10	6354.32
9	0.10	0.10	10	6354.98
10	0.50	0.10	10	6476.63
Construction B				
#	α	η	k	Perplexity
1	None	0.05	10	7088.24
2	0.10	0.05	10	7133.69
3	0.50	0.05	10	7133.89
4	0.05	0.05	10	7268.40
5	1.00	0.05	10	7484.53
6	0.05	None	10	7763.43
7	None	None	10	7768.09
8	0.05	0.10	10	7870.99
9	None	0.10	10	7877.45
10	0.50	None	10	7937.83
Construction C				
#	α	η	k	Perplexity
1	None	0.05	10	8111.34
2	0.05	0.05	10	8276.60
3	0.10	0.05	10	8344.80
4	0.50	0.05	10	8492.75
5	0.10	0.10	10	8687.27
6	None	None	10	8785.00
7	0.50	None	10	8865.91
8	0.05	None	10	8889.34
9	None	0.10	10	8930.34
10	1.00	0.05	10	8947.48

Table II show the five most frequent words from each topic, from the three best models which was manually inspected. These topics are not sorted in any particular order. Some words appear in multiple topics, such as *hide*, *color*, *http/https* and numbers, which does not provide any meaningful interpretation of topic. For example, 'hide' is a tag in the BBcode lightweight markup language, commonly used to format posts in many message boards. It is frequently used to withhold information until a visitor creates a user account on the forum and gains privileges to view the hidden content.

The various document construction methods (as seen in Table II) does not show much variance in the identified keywords. The main difference was the number of documents that the LDA could learn from. Document construction A have 120 875 documents, B contain 2 794 304, and C have 272 023. Although document construction B had 2 212 per cent greater number of documents to learn from than method A, it didn't produce any significant differently result. Thus, it can be recommended to go with the two other document

TABLE II
FIVE MOST FREQUENT WORDS FOR TOPICS

#	Construction A
1	account, good, help, time, accounts
2	80, 8080, 120, 195, 3128
3	ty, thx, nice, man, hide
4	gmail, hotmail, yahoo, net, aol
5	ty, fixed, version, download, bot
6	bol, scripts, script, https, legends
7	php, inurl, site, v1, 123456789a
8	color, ru, size, http, hide
9	http, https, youtube, watch, members
10	game, origin, sims, email, games
#	Construction B
1	80, 8080, 120, 195, 3128
2	account, http, hide, accounts, kappa
3	download, hide, bot, https, bol
4	http, site, de, php, net
5	sharing, testing, script, best, scripts
6	ty, http, members, 123456a, tx
7	gmail, hotmail, yahoo, check, thx
8	man, php, bro, mate, yahoo
9	test, works, lol, hope, game
10	thx, nice, good, work, share
#	Construction C
1	account, hide, http, https, accounts
2	80, 8080, 195, 120, 3128
3	download, bot, version, file, script
4	thx, nice, man, good, bro
5	ty, test, thx, nice, bro
6	gmail, hotmail, yahoo, php, http
7	site, php, color, hide, http
8	game, gmail, hotmail, games, captured
9	tks, unknown, 5900, password, null
10	55336, 123456789a, 123, ruddy, asdf3425j3d

constructions (A and C) as they produce a similar and faster result using fewer documents.

We need to further improve our result found in Table II to make the topics more clear for human analysts. We repeat the previous preprocessing steps and adding some new steps to enhance the result. We begin by iteratively identify and remove BBcode tags and additional uninformative words[1] from topics. We also removed numbers during the preprocessing, as numbers had very little meaning other than being related to network ports or passwords. Finally, we used lemmatisation due to the frequent similar words such as 'account', 'accounts', 'member', 'members' and so forth.

After conducting the iterative preprocessing, we use the previous gained knowledge to adjust the hyper-parameters in our experiment. We re-run the experiment for all document construction approaches using low hyper-parameters: where α and η are set to values None, 0.05 and 0.1 and k set to values 10 and 20. Resulting in running 54 (18×3) additional analyses. Table III show that the perplexity increase for the iterative preprocessing steps.

[1]http, https, www, gmail, hotmail, yahoo, inurl, ty, font, color, youtube, asp, well, post, myfonts, otf, abc, qwerty, ru, qwe, rar, add, true, beta, day, ip, net, aol, uk, function, live, fr, msn, var, de, br, nulled, menu, wa, time, people, ha, window, thing, start, year, de, site, php, zip, uk, pl, web, edition, lol, work, aspx, xmlrpc, html, view, content, xd

TABLE III
ITERATIVE TEN BEST MODELS WITH HYPER-PARAMETER COMBINATIONS

Construction A				
#	α	η	k	Perplexity
1	0.05	0.05	10	18249.81
2	None	0.05	10	18332.84
3	0.10	0.05	10	18343.05
4	0.10	0.10	10	19380.88
5	None	0.10	10	19434.33
6	0.05	None	10	19525.33
7	0.10	None	10	19546.14
8	0.05	0.10	10	19576.82
9	None	None	10	19793.34
10	None	None	20	22685.33
Construction B				
#	α	η	k	Perplexity
1	None	0.05	10	17582.44
2	0.10	0.05	10	17825.29
3	0.05	0.05	10	17827.16
4	None	None	10	18732.37
5	0.05	None	10	18788.12
6	None	0.10	10	18822.09
7	0.05	0.10	10	18998.47
8	0.10	None	10	19077.16
9	0.10	0.10	10	19131.95
10	None	0.05	20	22562.58
Construction C				
#	α	η	k	Perplexity
1	0.10	0.05	10	29255.27
2	0.05	0.05	10	29595.43
3	None	0.05	10	29608.72
4	0.10	None	10	29721.30
5	0.05	None	10	29834.72
6	0.10	0.10	10	30328.28
7	None	None	10	30579.64
8	0.05	0.10	10	30974.97
9	None	0.10	10	31947.12
10	0.05	None	20	41342.10

The more frequent words per topic, as seen in Table IV, also show greater coherent ideas per topics after additional iterative preprocessing. For example, there exist topics that: i) express gratitude or appreciation (work, thx, nice, share, good), ii) about popular games (lol, battlefield, fifa, sims, origin), iii) leaking of credentials (username, password), iv) various malicious tools (stealer, crypter, phisher, rat) and v) administrative purposes (member, ban, pm).

Document construction A can be suitable to get an overview of what the underground forum is about, as it shows a relation to accounts, leaks of credentials and games. Document construction B show less diverse topics as many of them can be categorised as expressing some gratitude. Thus, making this construction approach less suitable for a digital forensic investigation. Construction C can be suitable for understanding the different users within a forum, including their interest or possibly role on the forum. For example, people with a high proportion of expression of gratitude (thanks, thx, nice, etc.) in their messages might belong to the majority group

TABLE IV
ITERATIVE FIVE MOST FREQUENT WORDS FOR TOPICS

#	Construction A
1	account, file, bot, download, link
2	comcast, music, song, sbcglobal, rr
3	game, origin, sims, email, github
4	capture, type, key, unit, local
5	mail, password, username, unknown, user
6	member, wp, pro, stealer, clean
7	game, play, watch, best, good
8	script, update, enemy, auto, download
9	account, bol, legend, help, crack
10	thx, nice, share, test, man
#	Construction B
1	help, crack, link, guy, bol
2	share, check, skin, gg, account
3	download, dude, bot, update, version
4	bro, great, watch, rep, hello
5	thx, nice, test, hope, wow
6	file, tnx, download, gonna, password
7	wub, member, god, omg, gj
8	tks, cool, awesome, wp, tyty
9	good, script, mate, love, best
10	account, man, kappa, ban, lot
#	Construction C
1	nice, bro, tnx, tyy, gg
2	tks, ea, member, mail, info
3	account, game, link, crack, free
4	script, bol, update, download, game
5	thx, man, share, nice, good
6	file, download, bot, version, update
7	clean, stealer, rat, crypter, password
8	capture, account, member, gmx, key
9	wp, thnx, pro, unit, local
10	unknown, user, creed, assassin, unite

of less technical skilled cyber criminals. Additional steps for removing unnecessary and less informative words may result in highlighting more skilled cyber criminals.

V. CONCLUSION

Cybercrime continue to be a treat to our economy and the general sense of justice. Law enforcement agencies can exploit OSINT to gather proactive CTI, which might make them more effective to combat cybercriminals. The challenge of OSINT comes from a lot of unstructured data which may result in unreliable information from automated processes. Our research shows that automated algorithms such as LDA must follow a set of requirements to reduce the vocabulary size and improve the quality. We recommend repetitive preprocessing steps, e.g. continuously remove common words, until the result contains coherent and clear topics. Data cleaning is invariably an iterative process as there are always problems that are overlooked the first time around.

Contemporary related research mostly focuses on using topic modelling to get a quick overview of a lot of documents. This article tries to reduce the gap between reliability of automated processes to make them applicable in digital forensic contexts. We identified three distinct ways user's posts could be constructed into documents, each approach focused on different aspects: subject-centred, subject-user-centred and user-centred. While they did not produce any significant different result in keywords between topics; our result shows that more documents do not necessary improve the quality of topics.

Data is key to piece together any criminal investigation and more research are needed to further improve the reliability of automated processes/algorithms. Small changes in the input can produce an unreliable output, which in turn forensic analysts can misinterpret. Thus, we need to move further than contemporary research's focus on using LDA to produce a general overview of a large corpus of text. For example by applying techniques described in this article on real-world dark web underground forums. Furthermore, we need to design reliable and automated processes suitable in a digital forensic context. For example, to distinguish between individuals that produce advance tools for cybercrime and from those who simply are consumers of such tools. Finally, similar research as Chang et al. [7] should be conducted to analyse human understandable topics and evaluation metrics (e.g. perplexity) in a digital forensic context.

REFERENCES

[1] Rubayyi Alghamdi and Khalid Alfalqi. A Survey of Topic Modeling in Text Mining. *International Journal of Advanced Computer Science and Applications*, 6(1), 2015.

[2] Stig Andersen. Technical Report: A preliminary Process Model for Investigation. preprint, SocArXiv, May 2019.

[3] Waheed Anwar, Imran Sarwar Bajwa, M. Abbas Choudhary, and Shabana Ramzan. An Empirical Study on Forensic Analysis of Urdu Text Using LDA-Based Authorship Attribution. *IEEE Access*, 7:3224–3234, 2019.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[5] Matt Bromiley. Threat Intelligence: What It Is, and How to Use It Effectively, 2016.

[6] Andrew Caines, Sergio Pastrana, Alice Hutchings, and Paula J. Buttery. Automatically identifying the function and intent of posts in underground forums. *Crime Science*, 7(1):19, December 2018.

[7] Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David M Blei. Reading Tea Leaves: How Humans Interpret Topic Models. page 10, 2009.

[8] CrowdFlower. Data Science Report. URL: https://visit.figure-eight.com/rs/416-ZBE-142/images/CrowdFlower_DataScienceReport_2016.pdf, 2016.

[9] Isuf Deliu, Carl Leichter, and Katrin Franke. Extracting cyber threat intelligence from hacker forums: Support vector machines versus convolutional neural networks. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 3648–3656, Boston, MA, December 2017. IEEE.

[10] R. Dover, M.S. Goodman, and C. Hillebrand. *Routledge Companion to Intelligence Studies*. Routledge Companions. Taylor & Francis, 2013.

[11] I. Kononenko and M. Kukar. *Machine Learning and Data Mining*. Elsevier Science, 2007.

[12] Gaston L'Huillier, Hector Alvarez, Sebastián A. Ríos, and Felipe Aguilera. Topic-based social network analysis for virtual communities of interests in the Dark Web. page 9, 2011.

[13] Matthew Moran. Big data brings new power to open-source intelligence, 2014.

[14] Kyle Porter. Analyzing the DarkNetMarkets subreddit for evolutions of tools and trends using LDA topic modeling. *Digital Investigation*, 26:S87–S97, July 2018.

[15] Sagar Samtani, Ryan Chinn, Hsinchun Chen, and Jay F. Nunamaker. Exploring Emerging Hacker Assets and Key Hackers for Proactive Cyber Threat Intelligence. *Journal of Management Information Systems*, 34(4):1023–1053, 2017.

[16] Alexandra Schofield, Måns Magnusson, Laure Thompson, and David Mimno. Understanding Text Pre-Processing for Latent Dirichlet Allocation. page 4, 2017.

[17] Ryan Williams, Sagar Samtani, Mark Patton, and Hsinchun Chen. Incremental Hacker Forum Exploit Collection and Classification for Proactive Cyber Threat Intelligence: An Exploratory Study. In *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 94–99, Miami, FL, November 2018. IEEE.

[18] Hamid Akın Ünver. Digital Open Source Intelligence and International Security: A Primer. 2018.