Automatically Labeling Cyber Threat Intelligence reports using Natural Language Processing

Hamza Abdi, Steven R. Bagley, Steven Furnell, and Jamie Twycross
School of Computer Science
University of Nottingham
Nottingham, NG8 1BB. UK.
{hamza.abdi, steven.furnell, jamie.twycross}@nottingham.ac.uk; srb@cs.nott.ac.uk

ABSTRACT

Attribution provides valuable intelligence in the face of Advanced Persistent Threat (APT) attacks. By accurately identifying the culprits and actors behind the attacks, we can gain more insights into their motivations, capabilities, and potential future targets. Cyber Threat Intelligence (CTI) reports are relied upon to attribute these attacks effectively. These reports are compiled by security experts and provide valuable information about threat actors and their attacks.

We are interested in building a fully automated APT attribution framework. An essential step in doing so is the automated processing and extraction of information from CTI reports. However, CTI reports are largely unstructured, making extraction and analysis of the information a difficult task.

To begin this work, we introduce a method for automatically highlighting a CTI report with the main threat actor attributed within the report. This is done using a custom Natural Language Processing (NLP) model based on the spaCy library. Also, the study showcases and highlights the performance and effectiveness of various pdf-to-text Python libraries that were used in this work. Additionally, to evaluate the effectiveness of our model, we experimented on a dataset consisting of 605 English documents, which were randomly collected from various sources on the internet and manually labeled. Our method achieved an accuracy of 97%. Finally, we discuss the challenges associated with processing these documents automatically and propose some methods for tackling them.

ACM Reference format:

Hamza Abdi, Steven Bagley, Steven Furnell, and Jamie Twycross 2023. Automatically Labeling Cyber Threat Intelligence reports using Natural Language Processing. In *Proceedings of ACM Symposium on Document Engineering 2023 (DocEng '23). ACM, Limerick Ireland, 4 pages.* https://doi.org/10.1145/3573128.3609348

DocEng '23, August 22–25, 2023, Limerick, Ireland

ACM ISBN 979-8-4007-0027-9/23/08.
https://doi.org/10.1145/3573128.3609348

CCS CONCEPTS

• Information systems → Information retrieval • Security and privacy → Intrusion/anomaly detection and malware mitigation • Computing methodologies → Artificial intelligence → Natural language processing → Information extraction

KEYWORDS

Advanced persistent threat, Cyber Threat Intelligence report, Attribution, Natural Language Processing, spaCy.

1 Introduction

The attribution of cyber-attacks is the process of identifying and analyzing the sources, actors, and motivations behind a cyber-attack or incident. Accurate attribution of Advanced Persistent Threat (APT) attacks is vital for developing targeted countermeasures, strengthening digital defenses, holding responsible parties accountable, and preventing future attacks [1].

In an increasingly interconnected world, understanding and addressing the threats posed by APT groups becomes even more significant since their attacks are considered highly sophisticated and can cause massive harm [2]. Several methods are used in the attribution of cyber-attacks, each with their own strengths and weaknesses. For example:

- Network events/artifacts analysis. [3]
- Dissecting and reverse engineering malicious software to identify unique traits. [4]
- Analyzing the attacker's behavior. [5]

Some of these methodologies hinge on extensive reports encompassing helpful information that aids in the attribution process. These reports may contain network artifacts that, when analyzed, can provide insights about the attacks and the attackers. Also, they provide more context about the nature of the attack: the attacker's methods, tactics, techniques, and procedures (TTPs), and a clearer picture of the APT landscape [6]. These reports are referred to as Cyber Threat Intelligence (CTI) reports. They are generated by cybersecurity experts who gather, analyze, and interpret data from various sources, such as network traffic, malware samples, and open-source intelligence. They are the foundation for effective collaboration within the cybersecurity community [7].

The main problem with these reports is that they are effectively unstructured documents due to their lack of standardization, which makes extracting information automatically from the reports a complex process. The current research aims to develop an automatic framework for APT attribution that harnesses the power of machine learning to analyze CTI reports and other helpful information to extract valuable features that can help in the attribution process.

Before that research can be conducted, there is a more fundamental problem that needs solving, namely knowing which specific APT group a specific CTI report was discussing. This paper outlines our approach to automatically extract the APT group referred to by a CTI report by applying standard Document Engineering techniques to a corpus of CTI reports we have collated from disparate sources across the web. This work encompasses various components, including the methodologies employed, data collection, data processing techniques, and experiments. Finally, we will discuss the results obtained and evaluate the proposed method's effectiveness in accurately labeling CTI reports.

2 Methodology

This section elucidates the methodologies and techniques employed throughout this work, encompassing various stages.

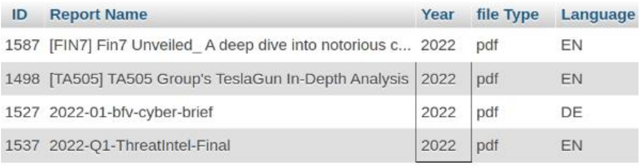
2.1 Data Gathering and Handling

The first step in our work involved collecting publicly available CTI reports from various sources on the Internet. A total of 1,000 random reports were downloaded from 207 different sources in various formats, predominantly PDF documents or web pages. All non-PDF reports were converted into PDF documents to maintain consistency and facilitate further analysis. Table 1 presents the top 5 sources.

Table 1: Top 5 CTI reports sources along with the number of downloaded reports per source.

Source	Number of documents
Trend Micro	62
Securelist by Kaspersky	56
WeLiveSecurity by ESET	43
Cisco Talos	35
Check Point	28

These reports were published between 2019 and 2022 and are diverse in their content, where some reports discuss specific APT groups and highlight their attacks, others discuss various APT groups or simply discuss threats in general. A more in-depth look into the reports and their contents will be done at a later point in this paper. Once the CTI reports were gathered and converted, they were stored in a database, ensuring that essential information about each document was recorded. This metadata included the source, document format, date of publication, and other relevant details. Organizing the data this way made it easier to manage and access the collected reports.



ID	Report Name	Year	file Type	Language
1587	[FIN7] Fin7 Unveiled_ A deep dive into notorious c...	2022	pdf	EN
1498	[TA505] TA505 Group's TeslaGun In-Depth Analysis	2022	pdf	EN
1527	2022-01-bfv-cyber-brief	2022	pdf	DE
1537	2022-Q1-ThreatIntel-Final	2022	pdf	EN

Figure 1: Sample metadata stored about each report in the database.

2.2 Data Processing

The first step involves the extraction of the report's text using three different Python libraries, *PyMuPDF*, *PyPDF2*, and *pdftotext*. The use of various libraries is intentional, enabling us to assess their efficacy relative to our work and identify the one that best suits our requirements and offers the best performance. Afterward, the extracted text is stored in the database and fed to a custom *spaCy PhraseMatcher*. *spaCy* is a powerful Natural Language Processing (NLP) library that offers a range of features such as tokenization, named entity recognition, and dependency parsing.[8] The PhraseMatcher method allows us to extract entities or find specific patterns within a document.

The PhraseMatcher is populated with a list of APT group names, their respective aliases, and any possible naming conventions. e.g., apt 1, apt_1, apt-1, apt1. The reason behind this is that the designation of an APT can vary across different reports. Also, the names and aliases of the APTs are gathered from various sources on the internet each time the PhraseMatcher is populated to ensure that the list is always up to date. Then, the PhraseMatcher is applied to the report's text, and a list of matches is returned. The order of these matches is based on the order they were found in the text, for example: [apt43, apt43, kimsuky, thallium]. Finally, the most common name from the matches is extracted using the Counter class from the collections library in Python.

To further enhance the accuracy of the APT group labeling method, an additional step was taken by extracting the title of each report. The assumption was that the name of the APT group in the report would be mentioned in the title thus by checking if the identified APT group from spaCy was mentioned in the title, accuracy could be better guaranteed.

The title extraction process involved two approaches: extracting the title from the document's metadata or extracting the title written in the document, which usually appears on the document's first page. The Python library '*PyMuPDF*' was used to extract the metadata and the document's written title from the first page. The paragraphs on the first page of the report are extracted along with their font sizes using the library and a custom function is ran against the text to extract the phrase with the largest font size. Once the title has been extracted, it is stored in the database. This approach was used based on the assumption that titles typically are set in larger fonts.

2.3 Report Labeling

Now that we have both the APT name provided by *spaCy* and the title, we begin by verifying if the APT name is present in the title. However, because APT's name could be misspelled or incomplete in the title, we leverage the Levenshtein string distance metric algorithm to compare and correlate effectively. String distance metrics are computational methods that measure the similarity or differences between two text strings, such as names or phrases.

If the name appears in the title, it is designated as the label for the CTI report. Alternatively, if it is not present in the title, the name provided by spaCy is used as the label. Figure 2 showcases the whole process.

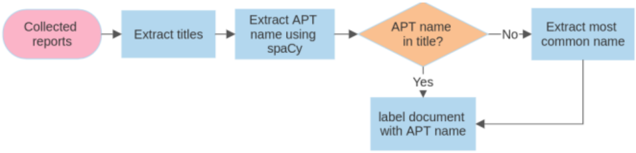


Figure 2: Flowchart of proposed work

3 Experimental Results and Evaluation

In order to assess the accuracy of the developed approach, each document was manually labeled. This process involved reading and reviewing each document, highlighting which APT group the document was discussing, extracting the title, and finally, categorizing the whole document. Manually labeling the documents allowed us to gain more insights into the nature of these documents. From the 1000 reports gathered, 947 (94%) were written in English, while the others were written in Chinese, Japanese, Ukrainian, German, and Russian. From the English reports, 605 discussed a single APT group, 28 discussed multiple APT groups, 81 discussed unknown groups, and the rest discussed various other topics. Out of the 190 unique groups mentioned in the 605 reports, the most mentioned APT groups were Lazarus Group, APT 41, APT 29, Magic Hound and Gamaredon Group.

Since the aim of this work is to label reports that discuss single APT groups, all the experiments and later insights will be based on the 605 English reports.

An assessment of the accuracy of the title extraction method was conducted by manually comparing the exported title with the actual report's title during the manual labeling process and highlighting which titles were accurate and calculating the accuracy. The assessment revealed an accuracy of 99%.

Subsequently, several evaluations of the APT name extraction using *spaCy* method were conducted, given that we used three libraries for text extraction. Accuracy was measured by checking if the APT name provided by *spaCy* is the same as the manually verified APT name or one of the APT's aliases. Table 2 presents the different accuracy achieved per library in report labeling by using the most common APT name method, or by checking if the APT name is in the title or by applying both together.

Table 2: Accuracy achieved per library with different techniques.

Library	APT mentioned in title	Most referenced APT	Both methods
PyMuPDF	70.5%	94.3%	97%
PyPDF2	62.7%	81.6%	91.7%
pdftotext	71.3%	94.5%	97.1%

The results indicate that *PyMuPDF* and *pdftotext* performed similarly in any of the methods applied, with *pdftotext* giving the highest accuracy, whereas *PyPDF2* performed poorly. The results also indicate that the highest accuracy can be achieved by using both methods.

A thorough examination of the performance difference was conducted. Based on the analysis, *PyPDF2* fails to extract the complete text from the document, while both other libraries almost exported the exact text. The cosine similarity between the exported text per document was calculated using Sentence Transformers framework, and the averages per library were calculated. The results indicate that *PyMuPDF* and *pdftotext* exported similar text, whereas the text exported by *PyPDF2* is less similar. Also, by checking the similarity of every English report included (947 reports) in the dataset and not just the 605 reports this work focuses on, we can notice that the performance of the libraries remains the same even if the content of the documents differs. Table 3 presents the similarity averages achieved per library on different reports.

Table 3: Cosine similarity average per library on different number of reports.

Library	Similarity AVG on 605 reports	Similarity AVG on 947 reports
PyMuPDF	82.4%	82.9%
PyPDF2	67.2%	68.7%
pdftotext	82.3%	82.8%

Additionally, the average length of the text extracted by each library was calculated, assuming that the longest length returned by any library corresponds to the maximum obtainable length. It should be noted here that since 947 of the documents are in English, the tests were conducted on the main 605 reports as well as the rest of the documents. Our analysis showed that *PyMuPDF* and *pdftotext* extracted almost the same amount of text with an average length of 99% of the whole document, while *PyPDF2* had an average length of 85%. Figure 3 showcases the length of extracted text per library on five different CTI reports.

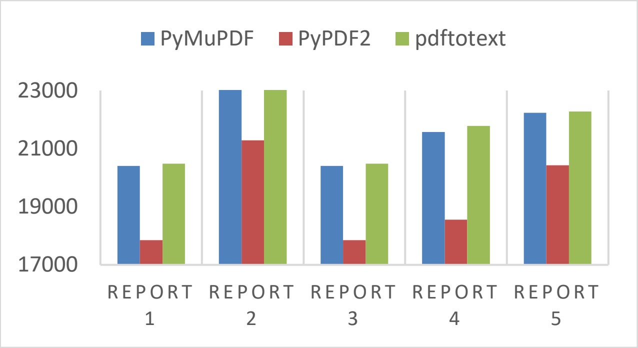


Figure 3: Text's length of extracted documents per library

It is imperative to acknowledge that achieved accuracy was attained through a series of iterative analyses of mislabeled reports over an extended period of time. This process enabled us to identify and address several challenges that emerged. These challenges include:

- The APT in the document might be unknown and yet to be named.
- The APT name or alias used in the document might be new and not included in the database or any public source.
- The APT name can be a common or a generic term which can confuse the NLP model and lead to mislabeled reports.
- The document is composed of scanned text (images).

4 Conclusion

In conclusion, this research has presented a robust and effective methodology for extracting and analyzing information from Cyber Threat Intelligence (CTI) reports. By collecting a comprehensive dataset of 1,000 documents and employing advanced natural language processing techniques with the *spaCy* library, the study successfully identified and extracted relevant entities and titles with an accuracy of 97%.

As a result of this research, future work can focus on refining the methodology to address the encountered challenges and further improve the extraction process. Integrating Optical Character Recognition (OCR) technology and expanding the database to include unknown or emerging APT groups contribute to developing an even more accurate and comprehensive model. Furthermore, employing advanced Large Language Models (LLMs) like GPT4 presents an opportunity to address complex challenges, such as labeling documents discussing multiple APT actors or those focused solely on malware, as these tasks require a deep understanding of the report's content's context.

REFERENCES

[1] Thomas Rid and Ben Buchanan. 2014. Attributing Cyber Attacks. Journal of Strategic Studies 38, 1 (2014), 4–37. DOI: https://doi.org/10.1080/01402390.2014.977382

[2] Pedro Ramos Brandao. 2021. Advanced Persistent Threats (APT)-Attribution-MICTIC Framework Extension. Journal of Computer Science 17, 5 (2021), 470–479. DOI: https://doi.org/10.3844/jcssp.2021.470.479

[3] Yangyang Mei, Weihong Han, Shudong Li, Xiaobo Wu, KaiHan Lin, and Yulu Qi. 2022. A Review of Attribution Technical for APT Attacks. 2022 7th IEEE International Conference on Data Science in Cyberspace (DSC) (2022). DOI: https://doi.org/10.1109/dsc55868.2022.00077

[4] Aylin Caliskan, Fabian Yamaguchi, Edwin Dauber, Richard Harang, Konrad Rieck, Rachel Greenstadt, and Arvind Narayanan. 2018. When Coding Style Survives Compilation: De-anonymizing Programmers from Executable Binaries. Proceedings 2018 Network and Distributed System Security Symposium (2018). DOI: https://doi.org/10.14722/ndss.2018.23304

[5] Lior Perry, Bracha Shapira, and Rami Puzis. 2019. NO-DOUBT: Attack Attribution Based On Threat Intelligence Reports. 2019 IEEE International Conference on Intelligence and Security Informatics (ISI) (2019). DOI :https://doi.org/10.1109/isi.2019.8823152

[6] Ghaith Husari, Ehab Al-Shaer, Mohiuddin Ahmed, Bill Chu, and Xi Niu. 2017. TTPDrill. Proceedings of the 33rd Annual Computer Security Applications Conference (2017). DOI :https://doi.org/10.1145/3134600.3134646

[7] Umara Noor, Zahid Anwar, Tehmina Amjad, and Kim-Kwang Raymond Choo. 2019. A machine learning-based FinTech cyber threat attribution framework using high-level indicators of compromise. Future Generation Computer Systems 96, (2019), 227-242. DOI: https://doi.org/10.1016/j.future.2019.02.013

[8] spaCy 101: Everything you need to know · spaCy Usage . Retrieved June 14, 2023 from https://spacy.io/usage/spacy-101