Leveraging a natural language processing approach towards a more informed vulnerability documentation process

by

BreAnn Marie Anshutz

B.S., Kansas State University, 2019

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2024

Approved by:

Major Professor
Dr. Doina Caragea

Copyright

Abstract

Cybersecurity vulnerabilities are an ever-increasing threat to the current cybersecurity landscape. It has been previously suggested that Twitter is a robust data source for gathering Cyber Threat Intelligence data. This includes cyber vulnerabilities which can be retrieved via their Common Vulnerabilities and Exposures (CVE) identifier. However, the culture of post-disclosure vulnerability discussion is changing to sometimes include a "nickname", or a short name utilized instead of the CVE identifier. This trend poses a significant challenge to the retrieval of CVE-relevant information as not all text includes the CVE identifier.

To address this challenge, a system was designed by utilizing an off-the-shelf machine learning model to link tweets that do not explicitly mention a CVE Identifier to their corresponding CVE. The system was tested utilizing several datasets and metrics to determine parameters required to obtain satisfactory performance with regards to retrieved information. The results show that machine learning makes it possible to retrieve relevant information corresponding to a specific CVE in the absence of the CVE identifier.

Table of Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to thank my advisor Dr. Doina Caragea for her advice and support throughout this process and pushing me forward to be able to complete this degree. I would also like to thank Dr. Brad Potteiger for his support, patience, and guidance throughout the project. Alongside that, I would also like to thank Dr. Eugene Vasserman for his support and guidance especially throughout the editing process. I would also like to thank my friends and mentors who pushed me to continue forward down this path and the technical and non-technical guidance along the way. Finally I would like to thank my family for their continued support and motivation.

Chapter 1

Introduction

Cybersecurity threats are ever-increasing and in-the-wild zero-days, or vulnerabilities that are exploited before they are known to the vendor, continue to be a threat [1, 2]. However, alongside the threat of unknown vulnerabilities, known vulnerabilities continue to be a persistent issue within the cybersecurity community as suggested by the existence and volume of vulnerabilities listed within the Cybersecurity Infrastructure Security Agency's Known Exploited Vulnerability catalog [3]. These are vulnerabilities that are known to be either currently or previously under active exploitation alongside a federally-mandated remediation time frame [4]. Although in-the-wild zero-day vulnerabilities are interesting to researchers, post-disclosure is actually when a majority of the exploitation occurs [5].

Twitter[1] has historically been the de facto "town square" for the cybersecurity community with offensive security researchers, defensive security practitioners and many more interacting with one another in real-time via short text format [6]. These conversations can be gathered as Cyber Threat Intelligence [7]. As shown by Horawalavithana et al. [8], many vulnerabilities that are discussed on Twitter utilize the common identifier from the National Institute of Standards and Technology's Common Vulnerability and Exposures database [9]. The format of this identifier is CVE-YYYY-IdNum.

Despite the value of this data source and the discussions around it, the literature to date

[1] Although rebranded as X, https://www.cbsnews.com/news/twitter-rebrand-x-name-change-elon-musk-what-it-means/, this paper shall refer to posts as Tweets and the platform as Twitter

has been limited regarding systems that correlate tweets or other short pieces of text to a specific relevant CVE in the absence of the identifier being mentioned in the text. Highlighted in Chapter 2 are efforts by many to design systems to leverage Twitter for vulnerability discussion, however, these systems either generalize the categorization of tweets (instead of categorizing down to specific vulnerabilities) or only leverage tweets that mention the CVE identifier. To fill this gap we evaluated whether text embeddings built with an off-the-shelf transformer-based model could be used to correctly correlate tweets to specific vulnerabilities. Our overall contribution is the proof that off-the shelf models can be used to retrieve relevant information corresponding to a specific CVE in the absence of the CVE identifier.

1.1 Motivation

Despite the standardization of vulnerability identification, some vulnerabilities can become so well known that the identifier number is no longer used and a short name, or nickname, is used when discussing the vulnerability, mitigations, and mitigation bypasses. An example is the December 2021 security event originating with CVE-2021-44228, a remote code execution vulnerability in a Java logging library called Log4j[10]. As it became popular and was discussed on Twitter it was often referred to as Log4Shell[11]. There are several other examples of this nicknaming behavior [12, 13]. Within this work specifically the ProxyNotShell attack chain consisting of CVE-2022-41040 and CVE-2022-41082 which results in an authenticated remote code execution attack on Microsoft Exchange servers [14], is highlighted and included as the primary focus of one of the datasets within this paper.

With the trend of utilizing non-standard identifiers to discuss particularly impactful vulnerabilities, it becomes difficult to collect data surrounding discussions of particular vulnerabilities and correlate the posts to the correct vulnerabilities. Alongside this, there is little literature showing whether these vulnerability discussions are unique enough to be able to correlate a specific CVE to the correct Tweet utilizing Natural Language Processing (NLP) if the CVE identifier is not directly mentioned in the post.

1.2 Background

Within this section we provide a general overview of CVEs and their associated metadata as well as cover certain high-level aspects of NLP and text embeddings.

1.2.1 CVE Record and Associated Metadata

The CVE program, which is a partnership across multiple CVE Numbering Authorities, identifies, defines, and catalogs publicly disclosed vulnerabilities. These partners help publish CVE records to assist in creating a common identifier for discussions of specific vulnerabilities [15]. NIST utilizes these records and further enriches them with a variety of additional sources to aid defenders in understanding the impact and scope of a vulnerability. This collection of the CVE records and enrichments is called the National Vulnerability Database or NVD. Some highlighted enrichments are the Common Vulnerability Scoring System (CVSS) and Common Platform Enumeration (CPE) [16]. The CPE identifier is "a standardized method of describing and identifying classes of applications, operating systems, and hardware devices" [17]. The NVD maps these CPEs within the CVE records in order to provide a way to programmatically determine what software is impacted by a specific vulnerability [9].

1.2.2 Text Representations in Natural Language Processing

The field of Natural Language Processing (NLP) is a sub-area of artificial intelligence that focuses on enabling computers to understand and process human language. To enable this understanding, words and documents need to be represented as vectors in high-dimensional spaces. By using vector representations, the task of determining the similarity of two words or pieces of text is reduced to determining the cosine of the angle between the representation's vectors. This cosine calculation is often referred to as **cosine similarity** Vajjala et al. [18, chap. 3]. We begin with a high-level explanation of how text can be represented as vectors.

Traditional word and document representations

An initial approach for text representation for the purposes of NLP is to treat text as a "bag-of-words" with the unrealistic assumption that the order of words in a text does not matter. Using the bag-of-words assumption, a word, or term, is represented using the one-hot encoding in a high-dimensional vector space, which is often referred to as an embedding or encoding space. Specifically, each term is represented as a vector with as many dimensions as the number of terms in the vocabulary (i.e., the set of all words in the collection). In this vector all dimensions are zero, except the one corresponding to the position of the represented term in the vocabulary without any reference to the order it occurred. It should be noted that with one-hot encoding every two terms are independent (as the corresponding vectors are orthogonal). Furthermore, with the "bag-of-words" representation a document can also be represented as a vector with as many dimensions as the number of terms in the vocabulary. Each position is either zero or one depending on the corresponding term appearing or not appearing in that particular document, which is referred to as the binary representation of a document Vajjala et al. [18, chap. 3]. A better approach is to utilize the count of each term within a document to account for more frequent terms.

This representation of a document with the count of terms, however, over-emphasizes common terms (e.g., and, the, that, etc.) that occur often within the language when computing the similarity score. As an attempt to resolve this over-emphasis of frequent terms, Term Frequency-Inverse Document Frequency or TF-IDF can be used. This approach takes the initial vector representation utilized in the count approach and divides that by the number of documents containing the term, thereby increasing the weighting of rarer terms. This is much more flexible than the count approach and allows for the cosine similarity score to be higher for pairs of documents that share more unique terms. One major disadvantage of the previous document representations relates to the sparsity of the representations as only a small number of terms will appear in a document as compared to the size of the whole collection vocabulary Vajjala et al. [18, chap. 3].

Modern word and document representations

As the NLP field continued to develop, some key approaches arose to build more advanced distributed representations (i.e., more compact, dense representations where most of the dimensions are non-zero real numbers). Word2vec [19] is a term representation approach which utilized machine learning (specifically, neural networks) to train a model that could efficiently encompass millions of terms into distributed representations referred as embeddings. The word2vec representation allowed for better mapping of the relationships between terms (as compared to the one-hot term representations that cannot capture any term relationships). Text representations such as word2vec were further extended to document representations. For example, doc2vec [20], took the improvements of word2vec and built functionality to where vector representations could be built for documents of any length.

A more recent and better document representation can be obtained with deep learning transformer-based models. An example of such models is BERT, or Bidirectional Encoder Representations from Transformers [21]. The improvements of BERT were made possible through advancements in deep learning, specifically the Transformers[22] architecture. Transformers improved the speed of self-supervised training for representation learning. This architecture was utilized by BERT to efficiently learn robust representations that can be used for a variety of NLP tasks [21].

More specifically, the initial BERT model was pre-trained on a large corpus of unannotated text, one of which being Wikipedia, using two self-supervised learning tasks: masked word prediction, or where random words in a text were masked and the model was trained to predict them from the context, and next sentence prediction, or where the model was trained to predict if two given sentences are related or not, [21]. BERT also was novel due to its bidirectional mapping, where words both before and after a masked word (i.e., its whole context) are used to predict the masked word during training. This was novel at the time [23], as other leading algorithms were either unidirectional, such as the initial GPT model by OpenAI, or only shallow bidirectional recurrent neural networks, such as ELMo .

BERT was further improved by SBERT, or Sentence-BERT [24]. This improvement

was built to enable large-scale semantic similarity comparison, clustering, and information retrieval. To conduct the similarity comparison they specifically used the cosine similarity metric to enable the comparison to happen at scale. Within this thesis, an SBERT model all-MiniLM-L6-v2 [25] was utilized in an off-the-shelf capacity. It was originally pre-trained on multiple question and answering pairs, Reddit comments, several Stack Exchange datasets, as well as the CodeSearch dataset [26].

Chapter 2

Related Works

Within this section we review several approaches to leverage Twitter data for Cyber Threat Intelligence or CTI.

The work of Alves et al. suggests that Twitter can be a rich source of information on known vulnerabilities in real time [6]. They also highlighted a small subset of CVEs within their collection that were published on Twitter before other sources including the NVD. This discussion of vulnerabilities can be considered CTI. While Rahman et al. listed multiple potential definitions of CTI within their survey, the definition we will be using is "any cybersecurity-relevant information ... that aids in predicting, preventing, or defending an attack, shortening the window between compromise and detection, and helping to clarify the risk landscape" [7].

When looking initially at papers utilizing Twitter as a source for vulnerability CTI discussion there were a few overarching themes. Several attempted to predict the point at which a vulnerability is publicly exploited [27–30]. However, these papers only utilized Twitter data that directly mentioned a CVE identifier. Additional papers attempted to start with a larger aperture in their collection procedures for Twitter data, however, these approaches simply categorized the conversations into generic categories (for example DDoS, Vulnerability, etc.) [31–33] or generically whether or not a Tweet is indicative of a cybersecurity event [34]. Trabelsi et al. clustered based on keywords [35], however, their focus was on zero-day

vulnerabilities, and not whether a Tweet was related to a CVE.

Rahman et al. conducted a survey surrounding various approaches to extract CTI from text [7]. They highlight how others have built various systems to collect, analyze, and aggregate CTI data. By delving deeper into the papers referenced that utilize Twitter from their categories of *CTI Relevance Classification, Cyberthreat Event Identification, Cyberthreat Relevant Word Topic Identification, and Vulnerability Information Extraction* it was determined that while much work has gone into Tweet classification for cyber threats, the works relevant to this paper can be categorized into several approaches. Some papers only determined whether a Tweet is relevant to the topic at hand of cyber threat intelligence [36–38]. Others used specific user-created keywords or generic topics to categorize [29, 39]. Some started to attempt to cluster Tweets in some capacity via automated means[40]. However, these automated clustering systems required training and the categorizations were based purely on the clustering algorithms. Some went a step beyond categorization and started enriching the data. Some simply added Named Entity Recognition [41] to extract named entities, while others extracted Indicators of Compromise from the cluster [42]. Within this survey there were other papers that did not fit in one of the aforementioned categories. One expanded the keywords collected [43] from Twitter. Another utilized a novelty classifier for alerting [44]. Yet another alerted on a new unknown word [45] with the presumption that an unknown word was some sort of new threat discussion.

To the best of our knowledge there is very little literature reflecting the use of BERT or SBERT with cybersecurity Twitter data. One of the few papers found is by Iorga et al. where they outlined how BERT could be used to determine if a piece of text was relevant to cybersecurity in general [46]. This work used a different data collection approach compared to others, and mainly scraped Tweets from specific cybersecurity-relevant Twitter accounts and utilized those Tweets to find and scrape blog posts and articles related to CTI. This work evaluated both the scraped long-form text and the Tweets from the selected Twitter accounts using a variety of methods, showing that BERT [21] can be leveraged to successfully identify relevant text in both the short text form of Tweets, alongside longer text forms of articles and papers. They only classified whether or not a Tweet or article was relevant

to cybersecurity, however, and used a custom-trained BERT model. They did note across all approaches they noted the dataset was small enough they were encountering over-fitting concerns. Another paper by Sumoto et al. showed how BERT could be used to label various elements of descriptions of CVEs within the NVD [47].

Chapter 3

Experimental Setup

Within this chapter we outline the research questions this paper is trying to answer as well as the datasets utilized to answer those questions.

3.1 Research Questions

The overall hypothesis within this paper is that modern "Off the shelf" embeddings can be used to represent cybersecurity Tweets and can be leveraged to correlate CVEs to relevant Tweets that do not directly contain a CVE identifier. Given the outstanding results obtained with BERT-like models we will use the pre-trained SBERT model all-MiniLM-L6-v2 to build embeddings to represent the tweets. Furthermore, we will use cosine similarity to compare tweets represented these embeddings as cosine is the go-to similarity metric in information retrieval research. The following research questions narrow some aspects of experimentation:

RQ1 Are CVE-related discussions unique enough that Tweets not mentioning a CVE identifier can be linked to the corresponding relevant CVE?

RQ2 Are CVE-related discussions specific enough to ensure the system removes cybersecurity-relevant data that is irrelevant to a given vulnerability?

RQ3 What cosine similarity score best balances both the potential for false positives and false negatives when ran against a real-world dataset?

3.2 Datasets

To answer these research questions, three datasets are utilized. Two were collected for this thesis by scraping Twitter utilizing specific terms and Twitter's search functionality which allows the specification of start and end dates and one was an existing research dataset.

3.2.1 Multi-Month "CVE-" dataset

This dataset was retroactively collected utilizing the term "CVE-" for a time span covering approximately May through November 2022 in November of 2022. This allowed us to build a historical dataset but any Tweets that were deleted before our data collection are not present in this dataset. The dataset was then filtered to remove Tweets which did not contain a CVE-ID within the body of the Tweet. The Tweets were then tagged with that CVE-ID. This overall process resulted in a dataset of 18,982 Tweets from 1893 unique usernames.

Another characteristic of the Twitter CVE discussions is the "burstiness" of the activity, or how quickly a CVE becomes relevant and activity dies off. Figure 3.1 shows cumulative Tweet activity per day for each CVE. The first few days are commonly the most active and then after about a week the activity typically decreases.

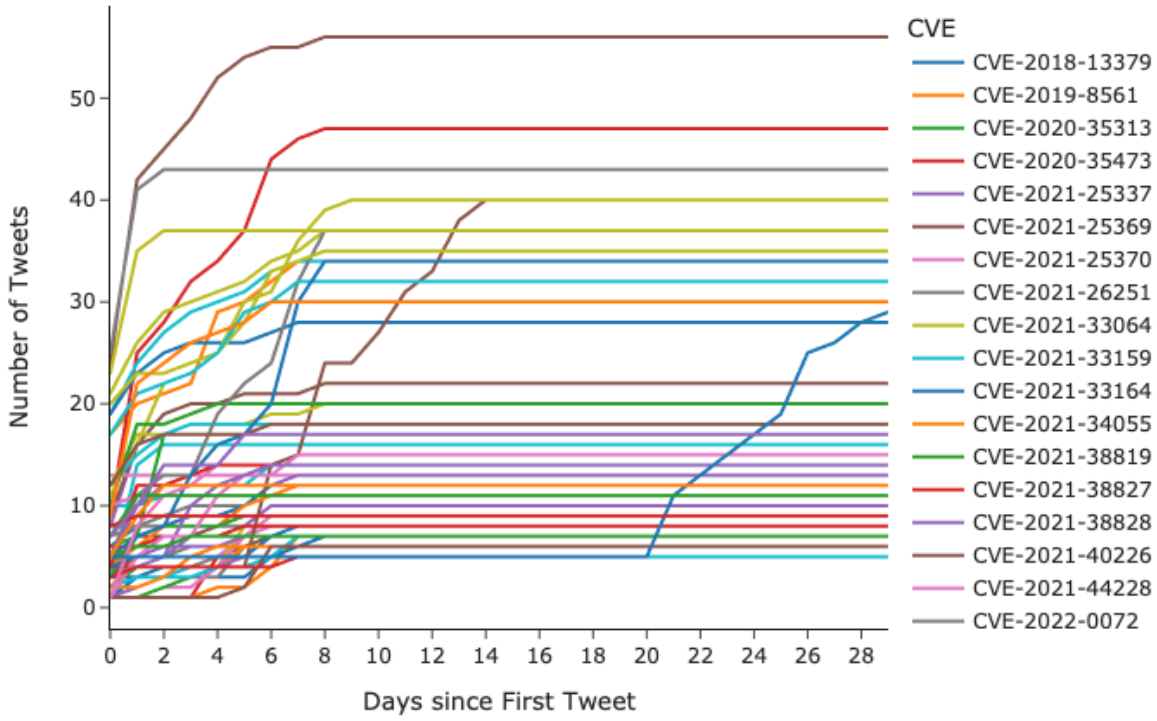

Figure 3.1: *Timeline of Cumulative Tweets per CVE*

3.2.2 ProxyNotShell dataset

One goal of this study was to conduct a near-real world test with a popular CVE which had a short name or nickname. During the early course of this research a set of two CVEs, CVE-2022–41040 and CVE-2022–41082, which when combined result in authenticated remote code execution on a Microsoft Exchange server. These two vulnerabilities were nicknamed "ProxyNotShell". The initial blog post using the nickname alongside a discussion of the vulnerability was published on September 29th, 2022 [48]. Microsoft's Security Response Center published the two CVEs alongside mitigations one day later, on September 30th, 2022 [49, 50].

The first part of the dataset was collected on October 3rd, 2022 and retroactively covered all partial Tweets from the previous week. This initial collection utilized the following terms: "microsoft", "poc", "cve-" , "rce", "exchange" "proxyshell", "proxynotshell". We later enhanced the dataset, collecting additional discussion using the subset of terms "ProxyNotShell" and "CVE-" on October 12th, 2022 to cover all public Tweets from October 3rd through the 12th.

Table 3.1 shows the total number of Tweets collected for each term. After deduplication using Twitter's unique identifiers, the dataset includes 120,815 unique Tweets.

Table 3.1: *Number of Tweets Collected Per Term*

Ingestion Term	Count
microsoft	61,092
poc	31,803
cve-	24,080
rce	2,429
exchange	1,531
proxynotshell	661
proxyshell	245
Unique Total	120,815

3.2.3 CyberTweets Dataset

We also use the CyberTweets dataset – a Twitter dataset containing a variety of cybersecurity relevant Tweets by Behzadan et al. [51]. This dataset was selected because it was already annotated and contained the body of the Tweet text itself. The dataset also contained annotated irrelevant data as well as cybersecurity marketing text that contains technical cybersecurity terms that are not relevant to vulnerabilities. We used these features to test our system for precision as well as a small-scale real world test. Tables 3.2 and 3.3 illustrate the annotation in the CyberTweets corpus.

Table 3.2: *CyberTweets Relevance Statistics*

	Count	Percentage
Business	2411	11.28%
Irrelevant	6708	31.39%
Threat	8347	39.06%
Unknown	70	0.33%
Unlabeled	3832	17.93%
Total	21368	100.00

Table 3.3: *CyberTweets Annotation Statistics*

	Count	Percentage
0day	733	3.43%
all	7	0.03%
botnet	702	3.29%
ddos	2248	10.52%
general	6946	32.51 %
leak	139	0.65 %
ransomware	3196	14.96%
vulnerability	7397	34.62%

Chapter 4

System Design

The overall system design is as follows: Tweets are separated into a collection of those that conyain a CVE Identifier (tagged) and those that do not (untagged). Both the tagged and untagged subsets are then pre-processed as outlined in Section 4.1. The tagged subset of processed Tweets is then stored as a collection of embeddings and associated metadata, which icludes the original Tweet(s) and any CVEs mentioned in the Tweet(s). The untagged subset of processed Tweets then is iterated through and an embedding is created for each individual Tweet. Semantic search, using cosine similarity to determine the similarity score, is then utilized to determine the highest matching Tweet. The found CVE-tagged Tweet and associated metadata, original Tweet and associated metadata, and similarity score provided by the semantic search are then stored for further evaluation. This process is illustrated in Figure 4.1.

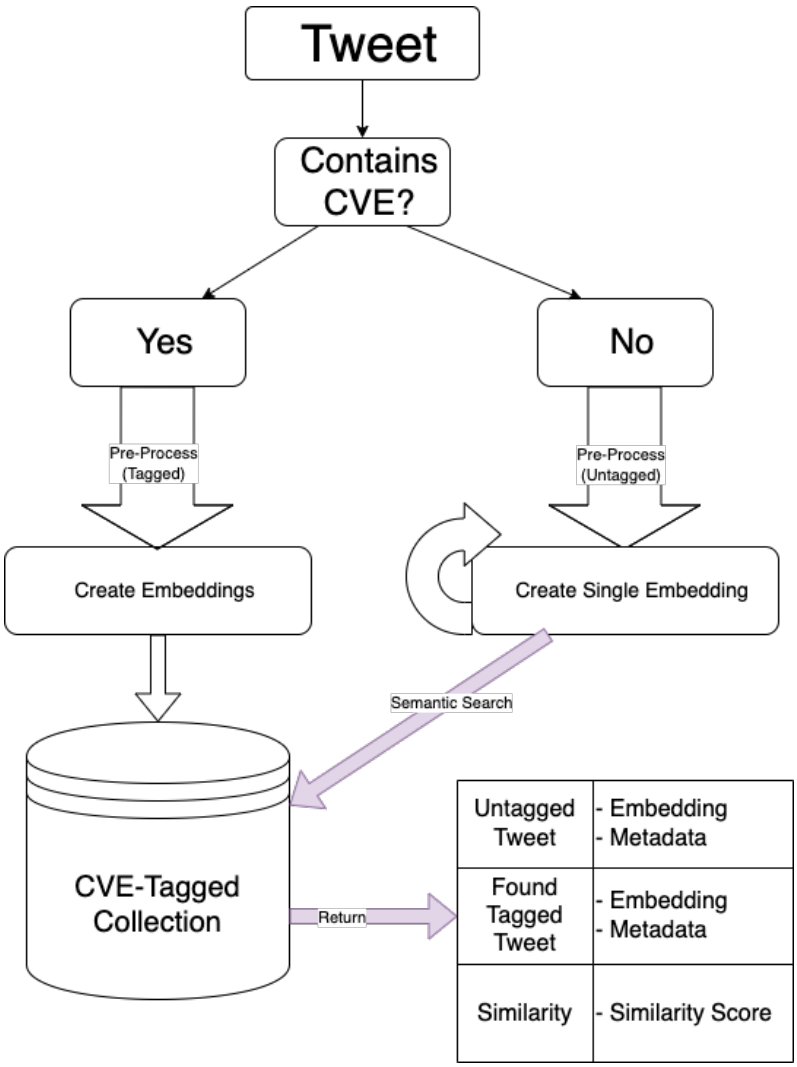

Figure 4.1: *System Diagram*

4.1 Pre-Processing

The data from Twitter needs to be pre-processed. The Tweets included things such as URLs and hashtags which are not reflected well in embedding spaces. It was also common to see in the dataset a Tweet of simply a URL, a CVE, and a few basic hashtags which do not add a lot of text to build an understanding from. Alongside this, we did not want duplicate contents or scripted automated accounts to bias the training data, therefore, the following pre-processing steps were taken for all Tweets:

1. Duplicate Tweets are dropped by ensuring uniqueness between the Twitter-provided

ID and the Tweet contents.

2. any CVE identifiers are extracted via regular expression

3. All Emoji are stripped from the text.

4. Non-English Tweets are naively removed by ensuring Tweets contents are all ASCII characters.

5. Hashtags (single strings containing one or multiple words preceded by a "#" used to tag Tweets by posters) and URLs are detected via regular expressions and removed.

For Tweets containing CVEs, additional pre-processing was completed prior to storing the collection of CVE-tagged embeddings and associated metadata:

1. Tweets from high-volume accounts are removed

    - These 25 accounts were determined by removing any account with more than 0.5%, or 94 Tweets within the Multi-Month Dataset 3.2.1, and were removed as that volume of output indicated likely automated Tweets for whatever reason. These 25 accounts contained nearly 90% of the total activity within the Multi-Month Dataset.

2. Tweets that after the pre-processing only contained whitespace and a CVE Identifier were removed

3. Content uniqueness was ensured by grouping based on uniqueness of the post-processed text and all Tweets mapping to the same text were retained as associated metadata to the embedding.

    - This was done to ensure there was not an over-representation of the same exact text within the tagged embeddings.

4.2 Text Embeddings and Semantic Search

After the data is pre-processed, the text then needs to be converted into text embeddings. This process utilizes the sentence_transformers library [52] which implements SBERT. The sentence_transformers library also has a utility which implements semantic search. This function compares a single document against an embedding space of many documents and retrieves the most similar embedding as determined by cosine similarity. The model used for the embeddings was the "all-MiniLM-L6-v2" model in the HuggingFace repository [25]. The model was selected as it was trained on both technical datasets such as open-source code and Stack Exchange and non-technical datasets such as Wikipedia and Reddit. There were others that were trained on just programming language related text to prioritize programming language identification and code generation [53, 54]. While "all-mpnet-base-v2" [55] could have been another option, it created much larger embeddings and results were promising enough from the model selected others were not evaluated.

Within this project, the CVE-Tagged collection of processed Tweets is then embedded and stored together. The processed text from untagged Tweets is then iteratively converted into embeddings and the most similar tagged embedding is found using semantic search. From this, a singular record is crafted consisting of three parts:

- Semantic Similarity value
- Un-Tagged Object $(processed\_text, original\_text, Tweet\_id)$
- Tagged Object $(processed\_text, List[original\_text, tagged\_cves, Tweet\_id])$

Chapter 5 outlines how this system was used.

Chapter 5

System Evaluation and Results

In this Chapter the results of the experiment are outlined.

5.1 RQ1 Experiment

To answer **RQ1**, (*Are CVE-related discussions unique enough that Tweets not mentioning a CVE identifier can be linked to the corresponding relevant CVEs*) the metric of Recall, or sensitivity, is utilized. This metric was chosen as it can help determine how much actually relevant data in the absolutely ideal case (everything has a known mapping) is removed at each similarity score.

The formula for this metric is:

$$recall = \frac{TruePositive}{FalseNegative + TruePositive}$$

. The experiment performed to answer RQ1 used the the Multi-Month CVE- dataset from Section 3.2.1 is used. The dataset was fed through the Pre-Processing steps for CVE-Tagged data outlined in Section 4.1.

After that initial pre-processing, each individually processed Tweet record is duplicated into a record for each CVE mentioned in the original text prior to pre-processing. Each CVE with 10 or more records is retained. The threshold of 10 was selected to ensure that each

CVE had enough coverage for that CVE to be detected by the system. These records are then split 60% Train, 40% test on a per-CVE basis into CVE-Tagged and Untagged data. The Untagged data is then processed additionally to remove all CVEs directly mentioned in the Tweet, but the mention is retained as additional metadata not provided to the part of the system outlined in Section 4.2. After all tagged Tweets have passed through the system the results are analyzed.

The following was used as the logic to analyze the match as we wanted to check if the system was potentially not mapping the exactly correct CVE but was in fact mapping a CVE that was related to the correct CVE. The following describes how these matches were determined.

- L0 Match: A Match at this level occurs if at least 1 of the CVEs originally in the "Untagged" record is mentioned in the list of CVEs corresponding to the Tagged record.
- L1 Match: A match at this level occurs if one of the following statements is true:
    1. At least one of the CVEs in the "Untagged" record is mentioned in the NVD description of one of the CVEs mentioned in the retrieved Tagged record. This often occurs if a CVE is extremely similar in description to another and the NVD description has to clarify that two CVEs are not identical. An example of this is CVE-2021-34473 [56].
    2. At least one of the CVEs in the "Untagged" record is mentioned alongside a URL that is also mentioned alongside one of the CVEs mentioned in the retrieved Tagged record.
    3. An L0 Match occurred
- L2 Match: A Match at this level occurs if one of the following statements is true:
    1. At least one of the CVEs in the "Untagged" record shares a partial CPE containing the vendor and product fields within the CPE with one of the CVEs mentioned in the retrieved Tagged record.

2. An L1 Match occurred.

The Recall rate analysis utilizing these categories is illustrated in Figure 5.1. This graph illustrates that this system can make the correct correlation between an untagged and CVE-tagged piece of text. This also illustrates how the similarity score is inversely related to the recall rate as well as how as the similarity score increases, the matching tiers converge.

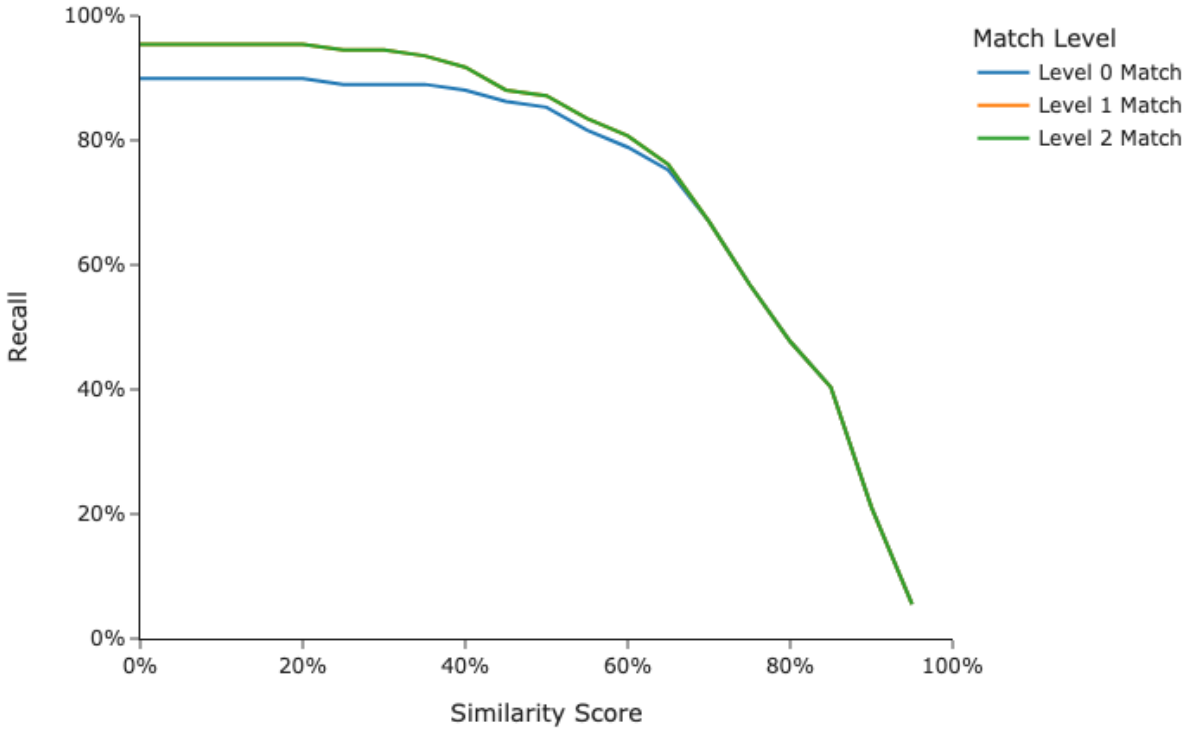

Figure 5.1: *Variation of Recall rate with similarity score using Multi-Month CVE dataset*

5.2 RQ2 Experiment

To answer **RQ2**, (*Are CVE-related discussions specific enough to ensure the system removes cybersecurity-relevant data that is irrelevant to a given vulnerability?*), the True Negative Rate, or specificity, is used. This metric was chosen as it can be used to check whether data

that should be removed is removed at each similarity score. The formula for this metric is:

$$TNR = \frac{TrueNegative}{FalsePositive + TrueNegative}$$

This experiment was performed using the CyberTweets dataset described in Section 3.2.3 is used. The Tweets from the dataset found to contain at least one CVE Identifier are pre-processed as part of the CVE-Tagged collection. Any Tweets marked in the pre-annotated dataset as "irrelevant" were fed into the system as the untagged subset. As this dataset was built to classify Tweets about Cyber Threat Indicators (of which vulnerabilities are a subset of) any Tweets tagged as "Irrelevant" were presumed to be not at all related to Cyber Threat Indicators but likely about cybersecurity-adjacent topics and thus a true negative for calculation purposes.

As illustrated in Figure 5.2, as the similarity score increases, the number of true negatives removed also increase showing that with the correct similarity score, the system can properly remove cybersecurity-relevant but CVE-Irrelevant data.

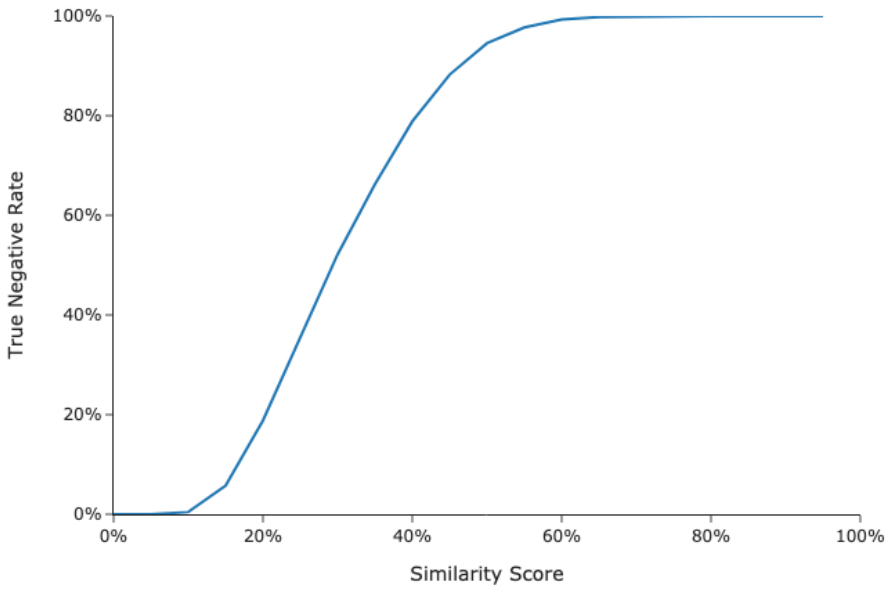

Figure 5.2: *Variation of True Negative Rate with similarity score using subset of CyberTweets dataset*

5.3 RQ3 Experiment

To answer **RQ3**, (*What cosine similarity score best balances both the potential for false positives and false negatives when ran against a real-world dataset?*), we need to determine a final similarity score.

5.3.1 Balanced Accuracy

To start with, we needed to determine the cutoff point for manual annotation for the next experiment to determine the precision. An easy metric to do so is called Balanced Accuracy, a metric that represents the average of the previous two metrics calculated. This allows for a basic weighting between the two other metrics and the formula is as follows:

$$BalancedAccuracy = \frac{TNR + Recall}{2}$$

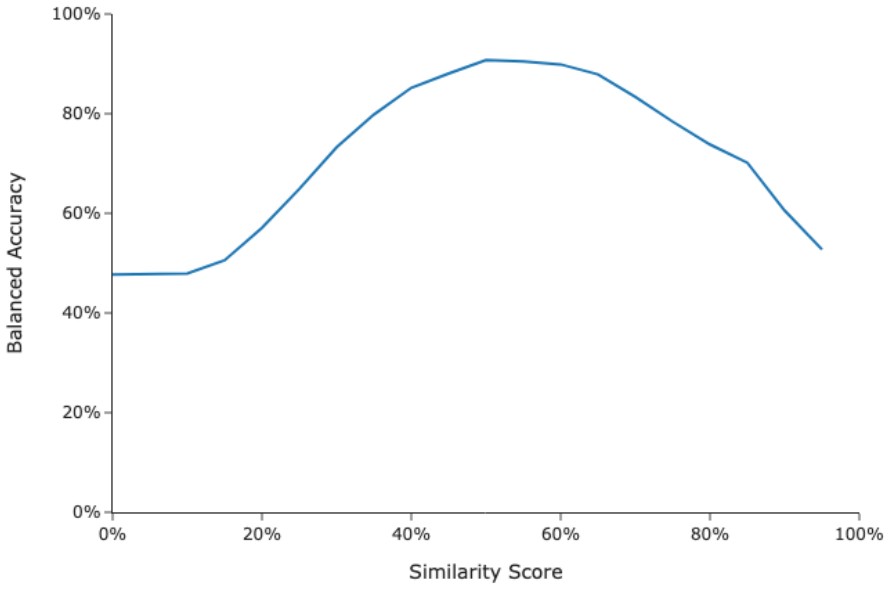

Figure 5.3: *Variation of Balanced Accuracy with similarity score using TNR and Recall data*

5.3.2 Precision

By selecting similarity scores based on the Balanced Accuracy metric, we have a cut off point for manual annotation when conducting a real-world test with the CyberTweets dataset as described in Section 3.2.3. As that dataset was designed to determine if a Tweet was relevant to cybersecurity, not whether not a Tweet was relevant to a specific vulnerability, this requires manual annotation after the system is ran. A cutoff point was established as the original dataset consisted of over 15,000 Tweets and a reduction in the amount of manual annotation to be completed would be greatly beneficial. A selected number of balanced accuracy values are shown in Table 5.1.

Table 5.1: *Balanced Accuracy*

Similarity	Balanced Accuracy
0.40	85.20%
0.45	88.05%
0.50	90.74%
0.55	90.47%
0.60	89.89%
0.65	87.87%
0.70	83.39%
0.75	78.40%

As can be shown in Table 5.1m the Balanced Accuracy using similarity scores of of 55% and 50% are within .03% of each other. With this close Balanced Accuracy score and the prioritization to reduce manual annotation, it was decided that the dataset would be annotated through the 55% similarity score, and the Tweets within the range of $50\% \leq x < 55\%$ would only be annotated if precision was trending upwards. A single annotator evaluated the full text prior to pre-processing and full URLs of any links in the Tweets for each match to determine the correctness of the match.

To conduct this experiment the Tweets containing a CVE Identifier were included as the CVE-Tagged subset and the remaining Tweets in the CyberTweets dataset were included as the Untagged subset.

After annotation, Precision was calculated using the following formula:

$$Precision = \frac{TruePositive}{FalsePositive + TruePositive}$$

The precision rate is outlined in Table 5.2 in 5% increments of the similarity score and the full graph of precision values is illustrated in Figure 5.4. This illustrates how quickly precision fell between the similarity score thresholds of 60% and 55% and as such, any matches with similarity scores below 55% were not annotated.

Table 5.2: *Precision using CyberTweets Dataset*

Similarity	Precision
0.55	35.75%
0.60	59.96%
0.65	74.78%
0.70	88.07%
0.75	96.62%
0.80	97.48%
0.85	100.00%
0.90	100.00%
0.95	100.00%

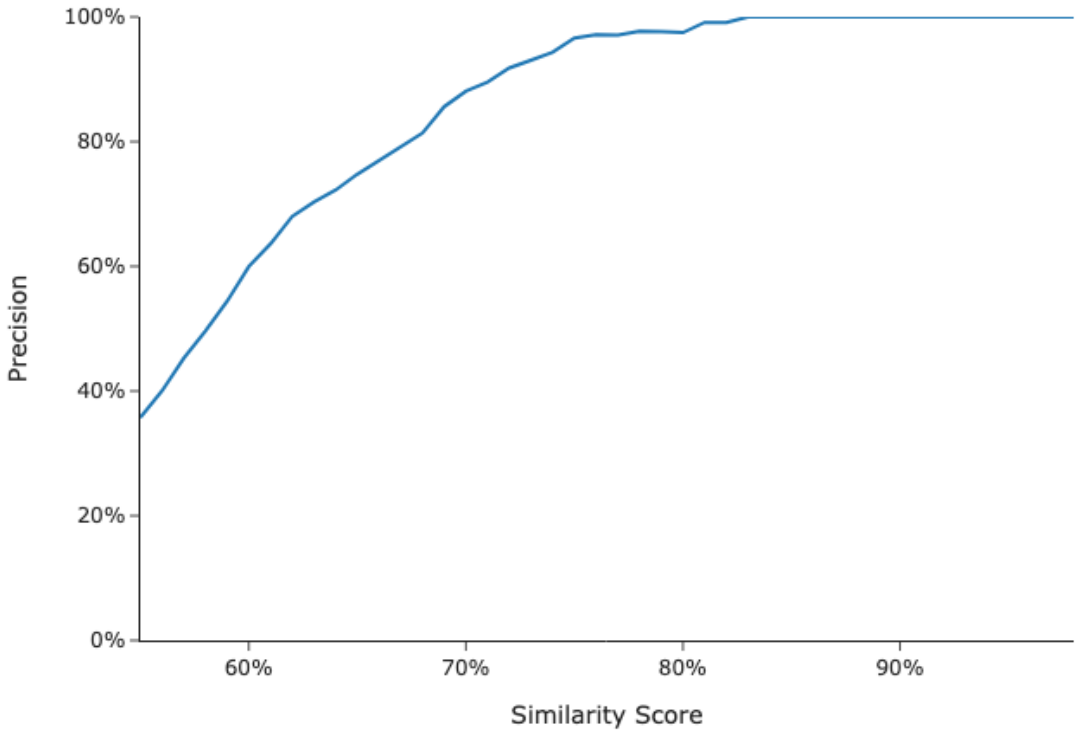

Figure 5.4: *Variation of Precision with similarity score using CyberTweets dataset*

5.3.3 Final Similarity Score

As this system now has a precision and a recall rate, an F-Score, or the Harmonic Mean of Precision and Recall, can be calculated to determine the best possible similarity score. The following formula was used:

$$F_1 = 2\frac{Recall \times Precision}{Recall + Precision}$$

and Figure 5.5 reflects the F-Scores of each similarity score when utilizing the L2 Recall scores from Section 5.1.

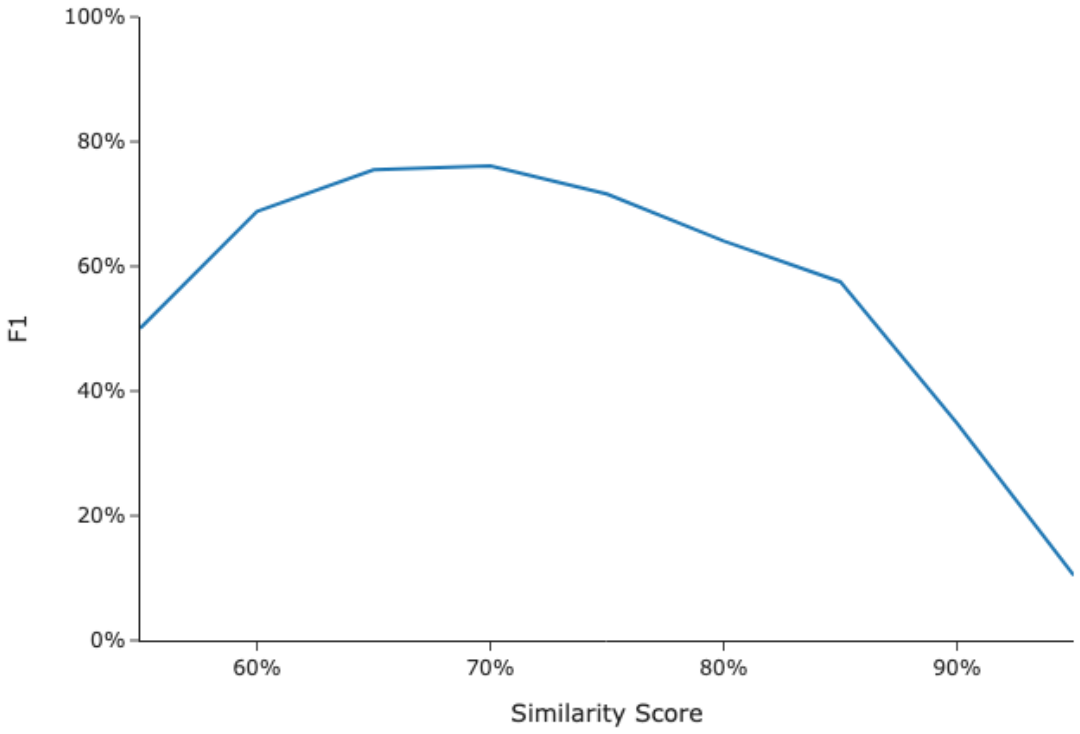

Figure 5.5: *Variation of F1 with similarity score using Recall and Precision metrics*

With all this testing, it is determined that the 70% similarity score is the best similarity score to use for this system moving forward as it had the best F-Score of 76.09%, which answers **RQ3**.

Table 5.3: *All Metrics for 70% Similarity Score*

Metric	Percentage
Recall (L0, L1,& L2)	66.97%
TNR	99.92%
Balanced Accuracy	83.39%
Precision	88.07%
F1	76.09%

5.4 Real-World Results

While each of the individual metrics in Chapter 5 were important to answer the research questions, they were also valuable in determining the final similarity score to run the system against the real-world test dataset as outlined in Section 3.2.2. This dataset was tailored specifically to collect the exemplar format of a CVE or set of CVEs, in this case CVE-2022-41040 and CVE-2022-41082, becoming so popular they are referred to by nickname and not CVE identifier, in this case "ProxyNotShell". This dataset started as as 120,815 Unique Tweets, with 20% containing a CVE Identifier. Of those Tweets containing a CVE, there were 4199 unique CVEs. After running the Tweets through the system, 998 new, unique CVE-relevant pieces of text were found that were not previously tagged with a CVE.

These 998 unique pieces of text originated from 1679 Tweets. Within these Tweets, 1056 unique URLs were referenced. Of these 1056 URLs, 40 overlapped with the original CVE-Tagged data. However, the remaining 1016 URLs appeared to have a high number of URL shorteners employed which can reduce the ability to determine the actual uniqueness of a URL. With this, the URLs were un-shortened which resulted in 604 unique URLs within the Tweets found to correlate to a CVE without being tagged, and 564 new URLs not mentioned within the CVE-Tagged dataset. The table in Appendix A illustrates the full results of this test and Table 5.4 illustrates some highlighted CVE results.

Table 5.4: *Highlighted Real-World Results*

CVE	New Processed Contents	New Twitter IDs	New Unique URLs	All Unique URLs	Tagged Tweets	Tagged URLs
CVE-2022-41040	688	1179	403	437	462	212
CVE-2022-41082	631	1084	378	410	428	209
CVE-2022-41033	200	381	136	145	48	27

As this data collection was specifically tailored towards CVE-2022-41040 and CVE-2022-

41082 it makes sense these CVEs would reflect a high level of enrichment in the Information Environment. CVE-2022-41033, however, was an unexpected increase, as such, some additional evaluation was completed. CVE-2022-41033 is a privilege escalation vulnerability in the Microsoft Windows COM+ Event System [57]. After further evaluation it was determined that this vulnerability received a high number of Tweet correlations due to a SecurityWeek Article which had the headline of "Microsoft Warns of New Zero-Day; No Fix Yet for Exploited Exchange Server Flaws" [58] and listed CVE-2022-41033 first while covering all the Microsoft patches released released on October 10th, 2022. Table 5.5 illustrates the breakdown of these Tweets.

Table 5.5: *CVE-2022-40133 Problematic Tweets*

Tweet	Twitter IDs Correlated
Microsoft Warns of New Zero-Day; No Fix Yet For Exploited Exchange Server Flaws - (CVE-2022-41033)	321
SecurityWeek: Microsoft Warns of New Zero-Day; No Fix Yet For Exploited Exchange Server Flaws - (CVE-2022-41033)	52

Chapter 6

Conclusion

Overall, this work has demonstrated a system that correlates vulnerability related discussion on Twitter to a specific vulnerability with a high degree of accuracy by leveraging pre-trained SBERT models. Within this research and results there were a few limitations, however. First of all, after the collection of this data the quality of Twitter as a CTI data source has fallen dramatically[59]. Second, the manually annotated data was only annotated by a single annotator which can result in mistakes in annotation and incorrect judgement calls. Another limitation, more within the system design as a whole, is the behavior of the system when a Tweet (whether by accident or malicious intent) is mentioned alongside the wrong CVE as illustrated by CVE-2022-40133. Finally, the uniqueness of both contents and URLs is handled rather naively. With these particular limitations in mind, there are some additional areas of future work to evaluate.

The first few areas of future work could potentially be evaluated is as follows. First of all, additional annotators can also be brought in within future work to reduce the single-annotator risk. Additional embedding models could also be tested to see if better results could be found via more advanced large language models. The uniqueness of contents could also be handled in a much more advanced way by leveraging a similarity score that could indicate a high degree of confidence that the text means the same thing. Alongside that, instead of retrieving the exact most similar Untagged tweet, multiple CVE-Tagged tweets

could be retrieved by similarity and an Untagged tweet could potentially be tagged only when there is an overlap of at least n tweets that mention the same CVE.

Yet another area of improvement is working towards potentially leveraging additional pieces of data from the NVD's CVE record. With the acknowledgement that some additional NVD enrichments are delayed from when a CVE is initially published, fields as they do appear could be utilized to build additional confidence metrics. One particular idea could utilize the vendor and product fields from the CPE field in a CVE record. By using Named Entity Recognition, software Vendors and Product names could be detected in the text. If these detected names from the text are listed as impacted by the NVD, the confidence in the correlation could increase. This approach could also help reduce the incorrect correlation of discussion surrounding generic attack vectors such as "Cross Site Scripting" and "SQL Injection" to specific CVEs.

A larger area of future work could be evaluating where the Cybersecurity Twitter community has moved to, and of those platforms which allow for data retrieval for research purposes. The system as a whole is dynamic enough to not be constrained to just Tweet-sized contents and as such platforms such as Reddit and others could be used for test datasets.

In conclusion, we have demonstrated that vulnerability related discussion, specifically that on the platform of Twitter, is able to correlate with a high degree of accuracy discussion surrounding a *specific* vulnerability without the vulnerability being mentioned in every post by utilizing the embeddings from a generically trained SBERT model without any fine tuning. This suggests that pre-trained transformer-based models could be used to correlate many other text-based cyber threat intelligence datasets. Overall, this research illustrates the ability to leverage transformer-based models to analyze larger corpora of discussions of particular vulnerabilities.

Bibliography

[1] Maddie Stone. The more you know, the more you know you don't know: A year in review of 0-days used in-the-wild in 2021, April 2022. URL https://googleprojectzero.blogspot.com/2022/04/the-more-you-know-more-you-know-you.html.

[2] James Sadowski. Zero tolerance: More zero-days exploited in 2021 than ever before, April 2022. URL https://www.mandiant.com/resources/blog/zero-days-exploited-2021.

[3] Cybersecurity and Infrastructure Security Agency. Known exploited vulnerabilities. URL https://www.cisa.gov/known-exploited-vulnerabilities.

[4] Cybersecurity and Infrastructure Security Agency. Reducing the significant risk of known exploited vulnerabilities. *Binding Operational Directive 22-01*, 2021. URL https://www.cisa.gov/news-events/directives/bod-22-01-reducing-significant-risk-known-exploited-vulnerabilities.

[5] Leyla Bilge and Tudor Dumitraş. Before we knew it: an empirical study of zero-day attacks in the real world. *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 833–844, 2012.

[6] Fernando Alves, Ambrose Andongabo, Ilir Gashi, Pedro M Ferreira, and Alysson Bessani. Follow the blue bird: A study on threat data published on Twitter. *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25*, pages 217–236, 2020.

[7] Md Rayhanur Rahman, Rezvan Mahdavi Hezaveh, and Laurie Williams. What are the attackers doing now? automating cyberthreat intelligence extraction from text on pace

with the changing threat landscape: A survey. *ACM Comput. Surv.*, 55(12), March 2023. ISSN 0360-0300. doi: 10.1145/3571726. URL https://doi.org/10.1145/3571726.

[8] Sameera Horawalavithana, Abhishek Bhattacharjee, Renhao Liu, Nazim Choudhury, Lawrence O. Hall, and Adriana Iamnitchi. Mentions of security vulnerabilities on Reddit, Twitter and GitHub. *Proceedings - 2019 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2019*, 2019. doi: 10.1145/3350546.3352519.

[9] National Institute of Standards and Technology. National vulnerability database. URL https://nvd.nist.gov.

[10] Cyber Safety Review Board. *Review of the December 2021 Log4j Event*, July 2022. URL https://www.cisa.gov/sites/default/files/publications/CSRB-Report-on-Log4-July-11-2022_508.pdf.

[11] Cybersecurity and Infrastructure Security Agency, Federal Bureau of Investigation, National Security Agency, Australian Cyber Security Centre, Canadian Centre for Cyber Security and Computer Emergency Response Team, Computer Emergency Response Team New Zealand, New Zealand National Cyber Security Centre, and United Kingdom's National Cyber Security Centre. *Joint Cybersecurity Advisory (CSA) AA21-356A - Mitigating Log4Shell and Other Log4j-Related Vulnerabilities*, December 2021. URL https://www.cisa.gov/news-events/cybersecurity-advisories/aa21-356a.

[12] Cybersecurity and Infrastructure Security Agency. *PrintNightmare, Critical Windows Print Spooler Vulnerability*, June 2021. URL https://www.cisa.gov/news-events/alerts/2021/06/30/printnightmare-critical-windows-print-spooler-vulnerability.

[13] 'Adrian Sanchez Hernandez, Govand Sinjari, Joshua Goddard, Brendan Mckeague, and John Wolfram. Pst, want a shell? ProxyShell exploiting Microsoft Exchange servers, September 2021. URL https://www.mandiant.com/resources/blog/pst-want-shell-proxyshell-exploiting-microsoft-exchange-servers.

[14] The Hacker News. ProxyNotShell – the new proxy hell? *The Hacker News*, October 2022. URL https://thehackernews.com/2022/10/proxynotshell-new-proxy-hell.html.

[15] CVE® Program. Cve® program overview, . URL https://www.cve.org/About/Overview.

[16] CVE® Program. Cve® program related efforts, . URL https://www.cve.org/About/RelatedEfforts.

[17] Brant A Cheikes, Brant A Cheikes, Karen Ann Kent, and David Waltermire. *Common platform enumeration: Naming specification version 2.3*. US Department of Commerce, National Institute of Standards and Technology, 2011.

[18] Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, and Harshit Surana. *Practical natural language processing: a comprehensive guide to building real-world NLP systems*. O'Reilly Media, 2020.

[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[20] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. *International conference on machine learning*, pages 1188–1196, 2014.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, June 2019. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[23] Jacob Devlin and Ming-Wei Chang. Open sourcing BERT: State-of-the-art pre-training for natural language processing, November 2018. URL https://blog.research.google/2018/11/open-sourcing-bert-state-of-art-pre.html.

[24] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, November 2019. URL http://arxiv.org/abs/1908.10084.

[25] Sentence Transformers. sentence-transformers/all-minilm-l6-v2, . URL https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.

[26] Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. CodeSearchNet challenge: Evaluating the state of semantic code search. *arXiv preprint arXiv:1909.09436*, 2019.

[27] Rizal Tjut Adek, Bustami Bustami, and Munirul Ula. Systematics review on detecting cyberattack threat by social network analysis and machine learning. *Proceedings of Seventh International Congress on Information and Communication Technology: ICICT 2022, London, Volume 2*, pages 567–577, 2022.

[28] Haipeng Chen, Rui Liu, Noseong Park, and V.S. Subrahmanian. Using Twitter to predict when vulnerabilities will be exploited. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 3143–3152, 2019. doi: 10.1145/3292500.3330742. URL https://doi.org/10.1145/3292500.3330742.

[29] Carl Sabottke, Octavian Suciu, and Tudor Dumitras. Vulnerability disclosure in the age of social media: Exploiting Twitter for predicting Real-World exploits. *24th USENIX Security Symposium (USENIX Security 15)*, pages 1041–1056, August 2015. URL https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/sabottke.

[30] Benjamin L Bullough, Anna K Yanchenko, Christopher L Smith, and Joseph R Zipkin. Predicting exploitation of disclosed software vulnerabilities using open-source data. *Pro-*

ceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics, pages 45–53, 2017.

[31] Sudip Mittal, Prajit Kumar Das, Varish Mulwad, Anupam Joshi, and Tim Finin. Cybertwitter: Using Twitter to generate alerts for cybersecurity threats and vulnerabilities. *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 860–867, 2016.

[32] Vahid Behzadan, Carlos Aguirre, Avishek Bose, and William Hsu. Corpus and deep learning classifier for collection of cyber threat indicators in Twitter stream. *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, 2019. doi: 10.1109/BigData.2018.8622506.

[33] K Simran, Prathiksha Balakrishna, R Vinayakumar, and KP Soman. Deep learning approach for enhanced cyber threat indicators in Twitter stream. *International Symposium on Security in Computing and Communication*, pages 135–145, 2019.

[34] Thea Riebe, Tristan Wirth, Markus Bayer, Philipp K
hn, Marc-André Kaufhold, Volker Knauthe, Stefan Guthe, and Christian Reuter. Cysecalert: An alert generation system for cyber security events using open source intelligence data. *International Conference on Information and Communications Security*, pages 429–446, 2021.

[35] Slim Trabelsi, Henrik Plate, Amine Abida, M. Marouane Ben Aoun, Anis Zouaoui, Chedy Missaoui, Sofien Gharbi, and Alaeddine Ayari. Mining social networks for software vulnerabilities monitoring. *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–7, 2015. doi: 10.1109/NTMS.2015.7266506.

[36] Sofia Alevizopoulou, Paris Koloveas, Christos Tryfonopoulos, and Paraskevi Raftopoulou. Social media monitoring for iot cyber-threats. *2021 IEEE Interna-*

tional Conference on Cyber Security and Resilience (CSR), pages 436–441, 2021. doi: 10.1109/CSR51186.2021.9527964.

[37] Amirreza Niakanlahiji, Lida Safarnejad, Reginald Harper, and Bei-Tseng Chu. Iocminer: Automatic extraction of indicators of compromise from Twitter. *2019 IEEE International Conference on Big Data (Big Data)*, pages 4747–4754, 2019. doi: 10.1109/BigData47090.2019.9006562.

[38] Hyejin Shin, WooChul Shim, Saebom Kim, Sol Lee, Yong Goo Kang, and Yong Ho Hwang. twiti: Social listening for threat intelligence. *Proceedings of the Web Conference 2021*, page 92–104, 2021. doi: 10.1145/3442381.3449797. URL https://doi.org/10.1145/3442381.3449797.

[39] Uğur Tekin and Ercan Nurcan Yilmaz. Obtaining cyber threat intelligence data from Twitter with deep learning methods. *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 82–86, 2021. doi: 10.1109/ISMSIT52890.2021.9604715.

[40] Fernando Alves, Pedro Miguel Ferreira, and Alysson Bessani. Design of a classification model for a Twitter-based streaming threat monitor. *2019 49th annual IEEE/IFIP international conference on dependable systems and networks workshops (DSN-W)*, pages 9–14, 2019.

[41] Nuno Dionísio, Fernando Alves, Pedro M Ferreira, and Alysson Bessani. Cyberthreat detection from Twitter using deep neural networks. *2019 international joint conference on neural networks (IJCNN)*, pages 1–8, 2019.

[42] Fernando Alves, Aurélien Bettini, Pedro M Ferreira, and Alysson Bessani. Processing tweets for cybersecurity threat awareness. *Information Systems*, 95:101586, 2021.

[43] Quentin Le Sceller, ElMouatez Billah Karbab, Mourad Debbabi, and Farkhund Iqbal. Sonar: Automatic detection of cyber security events over the Twitter stream. *Proceed-*

ings of the 12th International Conference on Availability, Reliability and Security, pages 1–11, 2017.

[44] Ba-Dung Le, Guanhua Wang, Mehwish Nasim, and Muhammad Ali Babar. Gathering cyber threat intelligence from Twitter using novelty classification. *2019 International Conference on Cyberworlds (CW)*, pages 316–323, 2019. doi: 10.1109/CW.2019.00058.

[45] Anna Sapienza, Sindhu Kiranmai Ernala, Alessandro Bessi, Kristina Lerman, and Emilio Ferrara. Discover: Mining online chatter for emerging cyber threats. *Companion Proceedings of the The Web Conference 2018*, page 983–990, 2018. doi: 10.1145/3184558.3191528. URL https://doi.org/10.1145/3184558.3191528.

[46] Denis Iorga, Dragos-Georgian Corlatescu, Octavian Grigorescu, Cristian Sandescu, Mihai Dascalu, and Razvan Rughinis. Yggdrasil — early detection of cybernetic vulnerabilities from Twitter. *2021 23rd International Conference on Control Systems and Computer Science (CSCS)*, pages 463–468, 2021. doi: 10.1109/CSCS52396.2021.00082.

[47] Kensuke Sumoto, Kenta Kanakogi, Hironori Washizaki, Naohiko Tsuda, Nobukazu Yoshioka, Yoshiaki Fukazawa, and Hideyuki Kanuka. Automatic labeling of the elements of a vulnerability report CVE with NLP. *2022 IEEE 23rd International Conference on Information Reuse and Integration for Data Science (IRI)*, pages 164–165, 2022. doi: 10.1109/IRI54793.2022.00045.

[48] Kevin Beaumont. Proxynotshell— the story of the claimed zero days in microsoft exchange, September 2022. URL https://doublepulsar.com/proxynotshell-the-story-of-the-claimed-zero-day-in-microsoft-exchange-5c63d963a9e9.

[49] Microsoft Security Response Center. Microsoft exchange server remote code execution vulnerability: Cve-2022-41082, September 2022. URL https://msrc.microsoft.com/update-guide/vulnerability/CVE-2022-41082.

[50] Microsoft Security Response Center. Microsoft exchange server elevation of privilege

vulnerability: Cve-2022-41040, September 2022. URL https://msrc.microsoft.com/update-guide/vulnerability/CVE-2022-41040.

[51] Vahid Behzadan, Carlos Aguirre, Avishek Bose, and William Hsu. Cybertweets repository, February 2019. URL https://github.com/behzadanksu/cybertweets.

[52] Nils Reimers. Sentence-transformers. URL https://www.sbert.net/index.html.

[53] Shafiq Joty Steven C.H. Hoi Yue Wang, Weishi Wang. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. 2021. URL https://huggingface.co/Salesforce/codet5-base-multi-sum.

[54] Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. CodeSearchNet Challenge: Evaluating the State of Semantic Code Search. *arXiv:1909.09436 [cs, stat]*, September 2019. URL http://arxiv.org/abs/1909.09436. arXiv: 1909.09436.

[55] Sentence Transformers. sentence-transformers/all-mpnet-base-v2, . URL https://huggingface.co/sentence-transformers/all-mpnet-base-v2.

[56] National Vulnerability Database. Cve-2021-34473, 7 2021. URL https://nvd.nist.gov/vuln/detail/CVE-2021-34473.

[57] National Vulnerability Database. Cve-2022-40133, September 2022. URL https://nvd.nist.gov/vuln/detail/CVE-2022-41033.

[58] Ryan Naraine. Microsoft Warns of new zero-day; no fix yet for exploited Exchange Server flaws, October 2022. URL https://www.securityweek.com/microsoft-warns-new-zero-day-no-fix-yet-exploited-exchange-server-flaws/.

[59] Jay Jacobs. Death of infosec Twitter, July 2023. URL https://www.cyentia.com/the-death-of-infosec-twitter/.

Appendix A

Real-World Test Results

CVE	New Unique Contents	New Twitter IDs	New Unique URLs	All Unique URLs
CVE-2022-41040	688	1179	403	437
CVE-2022-41082	631	1084	378	410
CVE-2022-41033	200	381	136	145
CVE-2022-41352	23	23	3	3
CVE-2022-40684	16	16	1	2
CVE-2022-3236	13	13	5	8
CVE-2022-36934	12	15	8	8
CVE-2022-27492	8	11	7	7
CVE-2019-15107	6	7	4	4
CVE-2022-35405	5	6	4	4
CVE-2022-36804	5	6	4	5
CVE-2007-4559	3	3	0	0

Continued on next page

CVE	New Unique Contents	New Twitter IDs	New Unique URLs	All Unique URLs
CVE-2022-39197	3	3	1	1
CVE-2022-37975	3	3	0	1
CVE-2022-37994	3	3	0	1
CVE-2022-37993	3	3	0	1
CVE-2021-31207	2	2	0	0
CVE-2021-34473	2	2	0	0
CVE-2021-34523	2	2	0	0
CVE-2022-3332	2	2	0	0
CVE-2022-4108	2	2	1	1
CVE-2022-23960	2	2	1	1
CVE-2021-44228	2	2	1	1
CVE-2022-2294	1	1	0	0
CVE-2021-42847	1	1	0	0
CVE-2021-40444	1	1	0	0
CVE-2022-24086	1	1	0	0
CVE-2021-26857	1	1	0	0
CVE-2022-26134	1	1	0	0
CVE-2022-28219	1	1	0	0
CVE-2021-40164	1	1	0	0
CVE-2021-32471	1	1	0	0
CVE-2021-27065	1	1	0	0
CVE-2021-26858	1	1	0	0

Continued on next page

CVE	New Unique Contents	New Twitter IDs	New Unique URLs	All Unique URLs
CVE-2022-40140	1	1	0	0
CVE-2021-26855	1	1	0	0
CVE-2020-5902	1	1	0	0
CVE-2019-19781	1	1	0	0
CVE-2019-11510	1	1	0	0
CVE-2016-1000027	1	1	0	0
CVE-2014-0160	1	1	0	0
CVE-2022-3307	1	1	0	0
CVE-2022-30190	1	5	0	0
CVE-2022-3309	1	1	0	0
CVE-2022-37969	1	1	0	0
CVE-2022-3349	1	1	0	0
CVE-2022-3311	1	1	0	0
CVE-2022-3318	1	1	0	0
CVE-2022-3314	1	1	0	0
CVE-2022-42724	1	1	0	0