Analyzing Expert Cybersecurity Twitter Accounts by Using Thesaurus Methods for Text Analytics

Qianwen Chen, Shruti Bakare, Aditee Verma, Christopher Casasanta, Christian White,
Andreea Cotoranu, and Avery Leider
Seidenberg School of CSIS, Pace University, Pleasantville, New York

{qc26641n, sb85060n, av11813n, cc54700n, cw02322p, acotoranu, aleider} @pace.edu

***Abstract*— This study performed text analytics on a large dataset of Twitter messages to create a thesaurus of cybersecurity terms. The dataset consisted of 70,115 tweets generated by 25 cybersecurity experts. This study built a thesaurus of cybersecurity terms indicative of cybersecurity expertise by experimenting with two text-processing tools: Python NLP libraries and RapidMiner, and then compared the results for accuracy. The analysis was extended by splitting the thesaurus into two groups, and applying the TF-IDF method to observe similarities between these groups; the highest similarity percentage between the two groups is approximately 50%. The final product from this study is a data driven thesaurus that can be used to describe a Twitter user as an expert or non-expert based upon the content of their Tweets. This thesaurus can be used for providing the building blocks for future work in modeling cybersecurity expert users.**

***Key terms*— Data processing, RapidMiner, Python, TF-IDF, Thesaurus, Twitter.**

I. INTRODUCTION

A. Background

Social media has been very popular for over a decade. One of the most popular social media applications is Twitter. Twitter is an online social networking application that allows users to post brief messages called "tweets." Twitter data can become miscellaneous due to the fact that the tweets can contain any character input from its users. In other words, Twitter data is free text and therefore unstructured as there are no tweet writing guidelines such as following grammar or spelling rules. There are about 6000 tweets posted every single second on Twitter, which amounts to a "big" data collection in a relatively short amount of time [16]. This data may yield meaningful information. Therefore, text analysis systems are designed to support processing of large data sets. Indexing and searching through a big data set can be done by organizing the information into a thesaurus. A thesaurus, also known as the data set, represents a precompiled list of words in a given domain of knowledge that provides a standard vocabulary for indexing and searching. For this study, tweets related to the field of cybersecurity were collected and processed. The focus was on Twitter accounts of well-known corporations, government agencies, and research groups that work in the cybersecurity space. In this study, these users will be referred to as "cybersecurity experts."

A previous study [7] focused on the analysis of tweets collected from 20 twitter accounts believed to be administered by cybersecurity experts. In that study, words related to cybersecurity such as malware or ransomware, were selected as key words. The data set was first pre-processed by filtering out all the tweets that didn't contain the key words. The data set was filtered further through standard text pre-processing techniques including tokenization, "stop-word" removal, and lemmatization. The processed data was then visualized with the use of word maps to highlight the most frequently used words. This study expends on the thesaurus building methodology described above in an effort to analyze cybersecurity expert models further.

B. Objectives

The purpose of this work is to create a cybersecurity thesaurus from tweets posted by a group of 25 cybersecurity experts. The methodology includes the following key components:

1) Identify 25 frequent twitter users that can be classified as "cybersecurity experts".
2) Compile the tweets of each user into a .csv file; each account will have its own file. Combine these files into one "data set."
3) Run text analysis on each user's data set using two preprocessing methods:
    a. Python Module
    b. RapidMiner
4) Split the Python data set into two groups.
5) Run frequency analysis on each group using TF-IDF method.
6) Compare TF-IDF results to discover text similarities among experts.

II. LITERATURE REVIEW

Although text analytics is one of the latest research fields, it has been used in real world industries to help people solve different problems more efficiently. For example, a software

company named, Expert Systems, developed various applications by applying text analytics technique to help: finacial operators to demonstrate risk management more sufficiently, healthcare industries to manage knowledge documents clear and reliablely, companies to make cybercrime preventaion easier, improve user experience in customer service and help insurance companies detect fraud with structured data [6].

In text analytics, it is common to use a dictionary or thesaurus to guide one's work. As Tavish Srivastava points out dictionaries help with converting unstructured text into structured data [13]. When analyzing unstructured text, important items may be overlooked due to slang spellings and improper grammar. Additionally, terminology can change rapidly over time. Dictionaries can be used to quickly analyze the general content of two or more data sources and to compare and contrast these sources to determine similarity.

This study aims at applying thesaurus methods to create a cybersecurity expert profile. "Experts are individuals who know a great deal about a domain and understand how the discipline is organized. This includes an ability to comprehend and contribute to the language and methodology of the discipline. As an expert, performance becomes more intuitive and automatic [3]."

In previous studies conducted by Danushka, "a sentiment sensitive thesaurus was created for the cross-domain sentiment classification. The sentiment sensitive thesaurus aligns different words that express the same sentiment in different domains. Labeled data from multiple source domains and unlabeled data from source and target domains was used to represent the distribution of features. Lexical elements (unigrams and bigrams of word lemma) and sentiment elements (rating information) were used to represent a user review. For each lexical element they measured its relatedness to other lexical elements, and grouped related lexical elements to create a sentiment sensitive thesaurus. In summary, their thesaurus shows the relatedness of lexical elements that appear in source and target domains based on context [1]."

Tweepy is a popular Python-module for the retrieval of tweets. "Tweepy uses the Twitter API to collect and store twitter data for any user account [2]." Once the data is gathered in a useable format, the next step is to preprocess the data. "What the data set should show is the content of each tweet, this is embedded in the text, and is the first step of the analysis [8]." "The work of Dalal and Zaveri on general text classification describes text preprocessing. Their work includes the following steps: determine sentence boundaries, eliminate "stop-words" from the text, and "stem" the text. Stop-words are words that are common and serve no purpose in determining meaning, such as "an," "the," or "of." Stemming involves trimming a word down to its root, usually by removing all suffixes and plurals [4]."

Another common preprocessing technique is Lemmatization. Lemmatization, while similar to stemming, can produce better results based on the context of the tweets.

Another approach to narrow down the searches to only key words is term frequency-inverse document frequency, or "TF-IDF [17]." This method uses the frequency of words in each tweet to determine which words are valuable and which words are "noisy."

III. METHODOLOGY

A. Identification of Cybersecurity Experts on Twitter

This study created a cybersecurity expert data set based on Twitter data. For the purpose of this study, a cybersecurity expert can be any corporation, government entity, or research group that frequently posts on cybersecurity-related topics. In order to create the data set, twitter accounts were investigated, and 25 Twitter users were selected, as listed in Table 1. The selection criteria included number of tweets per week and number of followers. A manual yet thorough review was completed for each Twitter profile selected.

Based on the literature review mentioned previously about what is an expert, this study applied the same criterion to found expert cybersecurity users on Twitter.

First action was to scan Twitter accounts for well-known and highly respected cybersecurity corporations, government-related organizations and academic institutions or research groups. This narrowed down the analysis to 25 Twitter handles with a large 'following' base, verification checks, and high twitter activity. Recent tweets from each possible choice were analyzed to validate the ultimate selection criteria. For each account, this study included a "credentials" column highlighting the expertise criterion, and a "field" column to highlight the specific area within the cybersecurity domaine corresponding to the account.

Table I. 25 chosen Twitter experts, their credentials, and what field of Cyber Security they can be categorized as

Account	Credentials	Field
@Cyber	Government affiliation (department of homeland security); large following base; tweets about current events in CyberSec	General
@CLTCBerkeley	Affiliated with prestigious school; tweets about current CyberSec technologies and privacy topics	General
@NJCybersecurity	Affiliated with well-known school; many tweets about security and privacy awareness; current events	General
@NISTcyber	Government affiliation; NIST implements all of today's CyberSec standards and best practices in US	General
@NortonOnline	Most well-known anti-virus software; large following base; latest news and updates in Cyber news	Virus
@DarkReading	Large following base; high amount of tweets; News about latest CyberSec threats, operated by CyberSec professionals from corporations including Fidelis	General
@TheCyberSecHub	Latest news in attacks, privacy tips; vulnerabilities, malware	General
@threatintel	Affiliated with Symantec (one of the largest data protection companies); large following base; latest news in threat intelligence; encryption; hacks	Network

Account	Credentials	Field
@symantec	security/protection corporation; Info on products and security technologies and offerings for business/users; large following base	Virus
@McAfee	Large and well-known anti-virus/security corporation; large following base; info on attacks; malware; viruses; security technologies and offerings	Virus
@CheckPointSW	Largest data security provider; provides well-known suite of mobile, cloud, and network security products	Network
@PaloAltoNtwks	2nd Largest data security provider; large following base; tweets a lot about emerging security technologies and threats	Access Control
@TrendMicro	4th largest security company; large following base; well-known for protecting data transmission	Network
@Fortinet	5th largest pure CyberSec company; provides services to over 330K clients incl enterprises, ISPs, and government affiliations	Network
@Stanford_Cyber	Affiliated with Stanford and https://cyber.stanford.edu/. Involved in many publications and funded research projects in the field of CyberSec	General
@proofpoint	Next-generation cybersec company. Known for advanced threats and emerging tech	Network
@Incapsula_com	Well-known cloud-based security service. Has experience with data protection for websites	Network
@CyberArk	Provides solutions for cryptology and password management for many fortune 500 companies	Access
@avgaunz	Well-known antivirus company based out of Australia. Provides internet and antivirus security solutions	Virus
@Gemalto	Well-known digital security company that provides software applications, secure personal devices such as smart cards and tokens. Largest provider of SIM cards	Access
@LifeLock	Most well-known identity protection corporation. Large twitter following base. Vast knowledge on protecting user data	Access
@ManTech	Worldwide corporation that provides solutions for national security programs. Expertise in counter intel, analytics, forensics	Forensic
@splunk	Large provider of monitoring software for fortune 500 companies. Large follower base, high amount of tweets	Monitor
@VERISIGN	One of the largest providers of domain names and internet security, related to SSL and encryption. Large follower base and many tweets regarding web security	Network
@digicert	security through SSL certificates and IoT security	Network

Table I shows the criteria used to determine which Twitter accounts were chosen for the text analysis. In the credentials column, a brief reasoning is shown on why each account is suited to be classified as an expert in cybersecurity.

B. Extraction of Twitter Data Using Tweepy

Once the 25 expert Twitter accounts were collected, the next step was to extract all the tweets associated with these accounts using the Python module Tweepy, tweet_dumper.py. The module asks for the Twitter account handle and then pulls all the tweets associated with that account into a .csv file. A total of 70,115 tweets representing 25 cybersecurity expert users were extracted. In order to store the tweets into a .csv file for further processing, the data format had to be converted from a list to a 2D array. The .csv file had three columns: tweet id, tweet date and time, and tweet text.

C. Processing Data Using Python

Tweets are free text and therefore can contain any type of characters or text. Therefore, it is important to filter out noisy data such as misspellings, abbreviations symbols (most commonly the "#"), and URLs. Preprocessing techniques can assist with normalizing the data.

In this study, the NTLK library in Python was used to normalize the textual data.

The following normalization tasks were used: tokenization, part of speech tagging, case folding, removing Unicode, removing links, stemming, lemmatization, and removing stop words.

Tokenization divides sentences into parts called tokens. Tokens are then used to divide strings into lists of substrings. This task is performed by using word_tokenize functions from the NLTK tokenize library [8]." An example from this study is for the twitter account @NortonOnline. One of the tweets sentence is "How to secure your information in the cloud. From encryption to privacy settings." Tokenization is able to break this sentence into 13 individual words.

Part-of-speech (PoS) tagging "is used to classify words into their PoS and label them according the tag set. PoS tagging also called as word classes [14]." From this study, after tokenization, PoS tagging is done by classifying words according to the noun, verb, adjective, etc.

"Case folding is the procedure to convert words into the same pattern [8]." This study has transformed all the characters into lowercase by using the "string lower" function in Python.

"Lemmatization is fairly complex because it needs to understand the context and determine the part of speech of a word. This means that lemmatization can be a more accurate tool when compared to stemming [10]." In this study, the lemmatize function is used to find the lemma or root of the word from the .csv cybersecurity expert users' profiles.

The final task is stop-word removal which "eliminates the common and frequent words that may not add real value to the tweet itself. This step is conducted by importing the stop words list from nltk.corpus library [8]." An example for this study is shown with this tweet "Infosec: 5 novels with #hacking plot holes that need to be patched." Stop-word removal was able to clean this tweet by removing: "with," "that," "to," and "be."

After lemmatization is complete, all the words are stored in a basic frequency list. If the given word is not present in the frequency list, then add that word in the list. If the word is already present in the frequency list, then increase the

frequency counter. This means that the results will contain the search word and the frequency in the .csv file.

After the preprocessing was completed, "TF-IDF [17]" method was applied to identify the most important words in the data set. "TF stands for Term Frequency and measures how commonly a word or text appears in the data set. Since tweets can be very "noisy," it is possible that a word can appear more times in one tweet compared to another. Therefore, the term frequency needs to be separated by the total number of words shown in the tweet. IDF stands for Inverse Document Frequency, this uses the frequency of each word to determine how important the word actually is[15]." high level work flow of this data process is shown in Fig. 1.

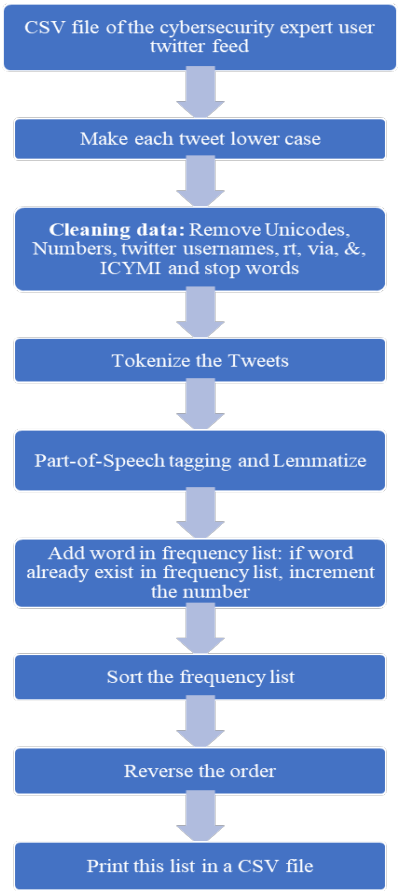

Fig. 1 Workflow of Data Processing using Python Libraries.

D. Processing Data Using RapidMiner

"Mining unstructured data with statistical modeling and machine learning techniques is a challenge, because the natural language text is inconsistent in syntax and semantics [18]." In order to ensure this study utilized the best tools, another preprocessing tool called "RapidMiner was tested to process the data [9]." "RapidMiner provides learning schemes, models and algorithms and can be extended using R and Python scripts [11]."

There are different extensions that can be used for text analytics in RapidMiner. This analysis used Text Analysis by AYLIEN, Information Extraction, and Text Processing. After several trials, it was discovered that Text Processing is the suitable extension tool for processing the data from the .csv files. Text Processing includes various operators such as Tokenization, Extraction, Filtering, Stemming and Transformation. This test case used Read Document, Documents to Data, Process Documents, Tokenize, Transform Cases, Filter Stopwords and Generate n-Grams (Terms).

There are various ways to load text into RapidMiner, from copy and paste, to HTML files, to database reads. Before processing, the first step was to load the text into RapidMiner. Since this study used Tweepy to extract the data from twitter into .csv files, then RapidMiner required the Read Document operator to make the .csv files readable. The second step was to use the "transfer the documents to data" operator. The steps were to drag the operator "Documents to Data" from operators' menu to the Process console, then link those two operators by connecting the out option from "Read Document" operator to the doc option in "Documents to Data" operator. This process takes each input collection document, and generates the data set for it. The text contained in the document is stored in a nominal attribute, which is a proper noun for one inside function of RapidMiner. If RapidMiner finds a label or metadata such as filename, then a label attribute is created within the program. This determines how the data is represented internally after the report is generated.

Once the data is successfully pre-processed into RapidMiner, run the console to obtain the results. This step generates a detailed table with row number, data type, metadata file, file type, path and date. The analysis can now continue with getting a word frequency table. In this step, the "Process Documents from Data" operator was used, which checked the tokens of a document, then used it to generate a vector numerically representing the document; the data type is Boolean. The next step is vector creation. RapidMiner selected the schema for creating the word vector, the range is: TF-IDF, Term Frequency, Term Occurrences and Binary Term Occurrences. In this case, the default is TF-IDF. The process flow for RapidMiner is shown in Figure 2. Within the Process Documents from Data operator, Tokenize, Transform Cases, Filter Stopwords(English), Filter Stopwords(Dictionary), Filter Tokens(by Length), and Generate n-Grams(Terms) operators were used. The Tokenize operator broke each tweet into phrases or words. The purpose for this tokenization is to identify the actual meaningful keywords for cybersecurity experts. A parser is needed for information retrieval because it is able to processes the tokenization from the file. It seems unnecessary because the data extracted from twitter are .csv files, which means during the tokenization, the incoming document was split into tokens on each of these characters, and the language for the used part of speech (POS) tagger is English.

Second, "Transform Cases" operator was used to lowercase all the text from the data. Third, two operators were used to remove the stop words: "Filter Stopwords(English)" and "Filter Stopwords(Dictionary)." These operators filter English stopwords from a document by removing every token which equals a stopword from a built-in stopword list.

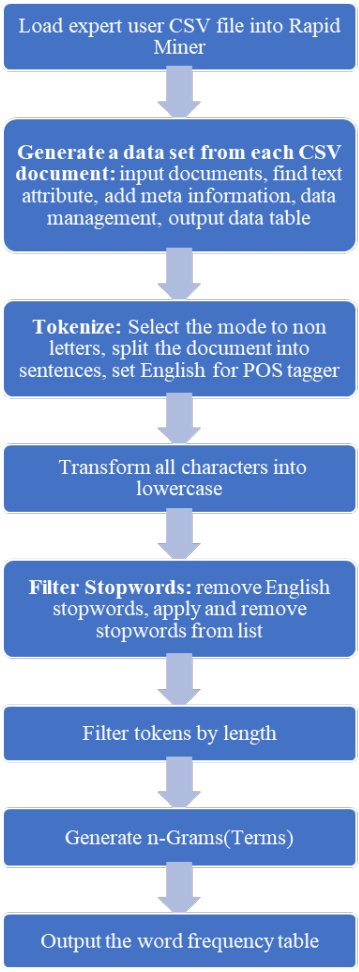

Fig. 2 By creating a workflow of Data Processing using RapidMiner, it will be able to compare the differences with Python method and how it influenced the results.

For RapidMiner to work properly, every token should represent a single English word only. That is why it was needed to tokenize the document by applying the Tokenize operator beforehand. This helped obtain a document with each token representing a single word. After the English stopwords were filtered out, some unnecessary characters/words still remained, such as "http." A dictionary file was created by adding stopwords that were not previously captured. This step is done manually. Therefore, this concluded that the use of Python method is better than RapidMiner because Python is almost fully automated.

To improve on the accuracy of the results, this study filtered the tokens further by setting the length between 3 and 25 characters. The last step was to generate the terms by creating a series of consecutive tokens of length. Using the table format discussed previously, the operator was linked to the result output and ran through the console to get the final results in a machine-readable format. However, some special characters remained including: punctuation, parentheses and hyphens.

E. Comparing the data set with TF-IDF method

In order to test the theory described in Section B, the use of TF-IDF to describe the relative weight of a word was implemented. In this study, a Python program was written to calculate the total number of words, total number of words that are similar, and percentage of similarity between two groups of twitter cybersecurity experts.

A specific word may appear in many tweets, which means that in those cases the IDF will be zero, resulting in a zero TF-IDF for all tweets for that specific word. In particular, words that do not help in deciding the correct words that relate to cybersecurity are automatically filtered out, as their TF-IDF value will be zero. Therefore, these words are eliminated from the word frequency list and not included in the thesaurus [15].

For this study, a program called Tfidf.py was used to calculate TF-IDF. This python code requires two input files of two different expert users. The program reads the two columns from the .csv file which contains words and its frequency from each .csv file. The words from each .csv file are stored in separate bag of words. Using union function, the program joins these two strings from the two different .csv files. Then it created a dictionary to keep the word count for each user. The words are counted from the bag of words for each user. After this, a matrix of words is created. Using the computeTF() function, the words are ranked according to the relevance. Next was to compute the IDF. To calculate IDF, the first step was to count the number of tweets that contains a particular word, divide it by total number of tweets and take log of that division. This produced the TF and IDF. Results were stored in the form of csv file. This csv file contains 3 columns: word, TF, IDF.

IV. RESULTS

A. Compare Python and Rapid Miner Results

Excel was used to create the rank table by taking the input of the top 25 frequency words from one expert sample. The title of the total three columns are the rank of the words, the name of the word, and frequency for the exact times the words appeared in this profile's total tweets. This table can show the top frequency words clearly and gives an easy to read format when comparing the two processing methods: Python and RapidMiner. After supplying all the values, it produces the graph shown in Table II. This table shows the user @threatintel as an example. The table represents the most frequently used words by this "cybersecurity expert."

The data shown in Table II is related to an expert because of the frequency and meaning of each term. The analysis

suggests that a novice or individual with little cybersecurity knowledge would not use these terms as frequent as @threatintel does.

Table II. Selected top 25 words from the frequency table generated by the two methods for cybersecurity expert twitter account @threatintel

Rank	Python Results		Rapid Miner Results	
	Word	Frequency	Frequency	Frequency
1	infosec	520	infosec	522
2	cybersecurity	400	cybersecurity	402
3	security	235	security	256
4	attack	233	symantec	240
5	malware	219	malware	239
6	ransomware	201	ransomware	215
7	data	152	attacks	175
8	device	149	data	159
9	vulnerability	124	iot	140
10	privacy	118	android	132
11	tech	118	tech	131
12	threat	116	devices	121
13	scam	108	privacy	118
14	patch	106	attackers	112
15	wannacry	101	cyber	112
16	email	100	hacking	110
17	update	95	read	110
18	android	93	wannacry	106
19	user	90	bug	102
20	history	88	icymi	95
21	symantec	84	million	91
22	tip	83	google	88
23	attacker	81	history	88
24	cyber	81	users	87
25	read	77	online	86

In order to process a Twitter expert's tweets, two methods described above; RapidMiner and Python were tested. The similarities between the tools lay within the preprocessing techniques. Both methods used tokenization, stemming, stop word removal, and case folding. However, where the Python module took it one step further was with the lemmatization. This was able to give the data the extra set of cleaning based on the context of the text and produce a slightly more narrowed down version of the cybersecurity terms. The results are slightly different, which proves that both methods of preprocessing are not the exact same. One disadvantage with using RapidMiner is that it cannot output the results as an Excel file or other files with our source file, users can only view within the application. Another disadvantage is it is not flexible as Python in terms of input different types of source files, as well as further analytics. For example, there are some TF-IDF operators that is for analyzing text files, but no operators are suitable for analyzing the .csv source file in this study. In addition, RapidMiner cannot analyze larger data sets unless the user pays for the additional functions. An example was it would not analyze the merged .csv file because it contained 70,115 tweets for all the 25 cybersecurity accounts.

B. Thesaurus for All 25 Profiles

Based on the disadvantages of RapidMiner, it was decided to continue the analysis using Python. Therefore, all 25 profiles were merged into one .csv file, to create a data set. Table III shows the top 50 frequency words in this 25-user data set.

Table III. Top 50 frequency words from the thesaurus for all 25 profiles

Results for Top 1-25 Words			Results for Top 26-50 Words		
Rank	Word	Frequency	Rank	Word	Frequency
1	security	9122	26	secure	1461
2	cybersecurity	7318	27	time	1444
3	infosec	4823	28	service	1439
4	threat	3462	29	year	1401
5	attack	3313	30	symantec	1397
6	learn	3239	31	repo	1361
7	data	3065	32	read	1306
8	ransomware	2638	33	tip	1293
9	malware	2462	34	cyberaware	1275
10	cloud	2060	35	vulnerability	1269
11	cyber	2051	36	ddos	1262
12	business	1966	37	hacker	1252
13	email	1917	38	check	1247
14	thanks	1900	39	website	1239
15	help	1894	40	splunk	1234
16	today	1874	41	breach	1231
17	online	1854	42	webinar	1217
18	network	1766	43	risk	1171
19	mobile	1725	44	great	1163
20	device	1697	45	account	1161
21	join	1641	46	customer	1099
22	team	1624	47	domain	1061
23	chatstc	1529	48	identity	1054
24	week	1482	49	user	1052
25	blog	1463	50	suppo	1041

C. TF-IDF Method Results

The TF-IDF applied in Python adds a level of sophistication to the frequency analysis that will increase the number of actual expert related terms received and minimize the common and typical cybersecurity terms.

To prove the study was successful and able to create the accurate expert dictionary, several tests were performed. First, the users were randomly split into two groups; Group A which consisted of 13 users and totaled 37,807 tweets and then Group B which contained 12 users and totaled 32,308 tweets. Once all the .csv files for each group were merged, the data was processed with Python using the TF-IDF method. After comparing all the words for Group A and Group B, a total of 20866 words was found. This analysis obtained a similarity word count of 6567. The percentage of the similarity between the two groups is 31.47%. In order to identify patterns in the data, this study experimented with different numbers of words as shown in Table IV. This study achieved the highest similarity percentage of 57.52% with the top 750 words.

Table IV. Similarities with different words amount

Top Frequent Words Count	Similar Words Count	Similarity Percentage
50	31	46.27%
100	63	46.67%
500	364	57.41%
750	547	57.52%
1000	726	57.08%
2000	1425	55.38%

Moreover, the 25 profiles were split into group A and group B by evenly distributing the experts based on the sub-fields represented by each profile as shown in Table V.

Table V. Set the group by fields

Group A		Group B	
Account	Field	Account	Field
@Cyber	General	@Stanford_Cyber	General
@CLTCBerkeley	General	@DarkReading	General
@NJCybersecurity	General	@TheCyberSecHub	General
@NISTcyber	General	@avgaunz	Virus
@NortonOnline	Virus	@McAfee	Virus
@symantec	Virus	@Incapsula_com	Network
@threatintel	Network	@TrendMicro	Network
@VERISIGN	Network	@Fortinet	Network
@digicert	Network	@CheckPointSW	Network
@proofpoint	Network	@CyberArk	Access
@Gemalto	Access	@PaloAltoNtwks	Access Control
@LifeLock	Access	@splunk	Monitor
@ManTech	Forensic		

These new groups were compared using the same method, providing the results shown in Table VI. The results are similar when comparing the percentage between Tables IV and VI.

Table VI. Similarities based on Table V

Top Frequent Words Selected	Similarity Count	Similarity
50	28	40%
100	63	46.67%
500	354	54.97%
750	530	54.75%
1000	712	55.37%
2000	2599	53.83%

Another approach to determine which terms belongs to the thesaurus is to take those terms that are in the top frequencies of multiple lists. For example, Table VII shows the 31 words that occur in both Group A and Group B from Table IV. These words are deemed relevant due to their appearance in both groups.

Table VII. Top 31 frequency words occurring from Table IV in both Group A and Group B of TF-IDF study

Similar words between Group A and Group B in top 50 frequent words from table IV			
Rank	Similar Word	Frequency in Group A	Frequency in Group B
1	cybersecurity	4761	2557
2	infosec	3618	1205
3	learn	1989	1250
4	attack	1912	1401
5	data	1855	1210
6	threat	1724	1738
7	ransomware	1584	1054
8	malware	1580	882
9	thanks	1359	541
10	cyber	1295	756
11	help	1215	679
12	mobile	1123	602
13	device	1095	602
14	online	1032	822
15	cloud	980	1080
16	today	856	1018
17	service	826	613
18	business	824	1142
19	join	806	835
20	time	806	638
21	tip	773	520
22	email	766	1151
23	repo	755	606
24	year	729	672
25	secure	721	740
26	breach	705	526
27	webinar	700	517
28	website	628	611
29	week	624	858
30	great	612	551
31	team	605	1019

Another method of obtaining a thesaurus using TF-IDF was also researched. In this study, the top 20 frequent words of each of the 25 cybersecurity experts were gathered, making the total word count 500 words. A TF-IDF 'score' for these 25 documents was then computed. In order to calculate this 'score', a python module called "tfidfdocuments.py" was developed. The higher the 'score', then the more important the word. After running the 25 documents through the Python program, the highest TF-IDF score came out to "0.12629". 40 total words with the high score were chosen. The words are shown in Table VIII below.

Table VIII. Highest scored TF-IDF words from top 500 words

Word
Scam
password
Spam
User
Safe
microsoft
mobilesecurity
Host
master
cyberark

privileged
credential
access
privilege
crash
breach
contact
authentication
Site
Bot
botnet
webperf
lifelock
system
task
order
framework
incident
advisory
crime
trap
proofpoint
fraud
compliance
visibility
malicious
splunk
share
analytics
devops

V. CONCLUSION

Text analytics is a powerful way to examine data. The analysis above demonstrated that preprocessing can be used on a corpus of Tweets to build a thesaurus on a specific topic. The text analysis and frequency analysis that was performed on a data set of over 70,000 Tweets was able to show a high percentage of similarities among all 25 "expert" users. Even though each user was different and had their own record with thousands of Tweets, the analysis was able to pick out the cybersecurity related terms that they tweeted most about and then use that information to determine which of these words were important to this analysis, and which ones were "noise." As a result, this study generated a highly filtered thesaurus of expert cybersecurity related words. The final product is a data driven thesaurus that can be used to identify a Twitter user as an expert or non-expert based upon the content of their Tweets.

For future work, researchers can also try to compare all 25 profiles at the same time, to see what similarities are found among the experts. For example, select top 50 to 100 for each profile, then run TF-IDF on all 25 documents, to see what data frame it will get, and what they can do with those data. This approach may able to find to correct similar top frequent words, and those words can be a more accurate for building a thesaurus for cybersecurity expert. In addition, researchers can also set up a certain number for the similarity percentage, for example if the similarity percentage is around or above 50%, then assume it is a cybersecurity expert. However, to evaluate whether the percentage is suitable more supportive evidence is needed. Researchers can also look back to all the processes in this study, compare with similar studies that have been done, to see whether other approaches could provide improved results.

VI. REFERENCES

[1] Bollegala, D. Member, IEEE, Weir. D, and Carroll, J. "Cross-Domain Sentiment Classification using a Sentiment Sensitive Thesaurus",2013

[2] Bonzanini, M. (2017, July 19). Mining Twitter Data with Python (Part 1: Collecting data). Retrieved February 13, 2018, from https://marcobonzanini.com/2015/03/02/mining-twitter-data-with-python-part-1/

[3] Cognitive Performance Group, July 24, 2013. "4 Differences Between Experts and Novices" http://cognitiveperformancegroup.com/2013/07/24/4-differences-experts-novices/#respond accessed Feb 10, 2018.

[4] Dalal M and Zaveri, M, "Automatic Text Classification: A Technical Review," Semantics Scholar, 2011.

[5] Divya, F, "Real Time Sentiment Classification Using Unsupervised Reviews", IJIRCCE, March 2014.

[6] Expert System, Apr 18, 2016. "10 Text Mining Examples" http://www.expertsystem.com/10-text-mining-examples/ accessed

[7] Fiore, V. Nahal, S. Matyunin, D. Padilla, E. Satpal, J. and Cotoranu, A. "Generating a Cybersecurity Thesaurus Based On Tweets", http://vinnyfiore.com/Group%206%20Paper.pdf . Accessed Feb 08, 2018.

[8] Hidayatullah. A and Ma'arif. M 2017 "Pre-processing Tasks in Indonesian Twitter Messages"J. Phys.: Conf. Ser. 801 012072

[9] Hofmann. M, Klinkenberg. R, "RapidMiner: Data Mining Use Cases and Business Analytics Applications (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series)," CRC Press, October 25, 2013.

[10] Jivani, A. Nov, 2011. "A Comparative Study of Stemming Algorithms", https://pdfs.semanticscholar.org/1c0c/0fa35d4ff8a2f925eb955e48d655494bd167.pdf . Accessed Feb 20, 2018.

[11] Norris. D, "RapidMiner - a potential game changer," Bloor Research, November 13, 2013.

[12] Ross, K. G., Phillips, J. K., Klein, G., & Cohn, J. (2005). "Creating expertise: A framework to guide technology-based training", accessed Feb 10, 2018.

[13] Srivastava. T, September 4, 2014. "Framework to build a niche dictionary for text mining" https://www.analyticsvidhya.com/blog/2014/09/creating-dictionary-text-mining/ accessed Feb 10, 2018.

[14] TextMiner. July 09, 2015. "Dive Into NLTK, Part III: Part-Of-Speech Tagging and POS Tagger", http://textminingonline.com/dive-into-nltk-part-iii-part-of-speech-tagging-and-pos-tagger . Accessed Feb 19, 2018

[15] TFIDF, http://www.tfidf.com/. Accessed Mar 24, 2018

[16] Twitter Usage Statistics. http://www.internetlivestats.com/twitter-statistics/. Accessed April 16, 2018.

[17] Wu. H and Luk. R and Wong. K and Kwok. K. "Interpreting TF-IDF term weights as making relevance decisions". ACM Transactions on Information Systems, 26 (3). 2008.

[18] Verma. T, Renu, Gaur D. April 2014, "Tokenization and Filtering Process in RapidMiner", http://research.ijais.org/volume7/number2/ijais14-451139.pdf accessed Feb 20, 2018.