Towards Dark Jargon Interpretation in Underground Forums

Dominic Seyler[1(✉)], Wei Liu[1], XiaoFeng Wang[2], and ChengXiang Zhai[1]

[1] University of Illinois at Urbana-Champaign, Champaign, IL, USA
{dseyler2,weil8,czhai}@illinois.edu
[2] Indiana University Bloomington, Bloomington, USA
xw7@indiana.edu

Abstract. Dark jargons are benign-looking words that have hidden, sinister meanings and are used by participants of underground forums for illicit behavior. For example, the dark term "rat" is often used in lieu of "Remote Access Trojan". In this work we present a novel method towards automatically identifying and interpreting dark jargons. We formalize the problem as a mapping from dark words to "clean" words with no hidden meaning. Our method makes use of interpretable representations of dark and clean words in the form of probability distributions over a shared vocabulary. In our experiments we show our method to be effective in terms of dark jargon identification, as it outperforms another baseline on simulated data. Using manual evaluation, we show that our method is able to detect dark jargons in a real-world underground forum dataset.

Keywords: Dark jargon · Hidden meaning interpretation · NLP

1 Introduction

When bad actors communicate in underground forums (e.g., Silk Road [5]), they often use jargons to obfuscate their true intentions. They make use of dark jargons, which are benign-looking words that have hidden, sinister meanings, especially among communities in underground forums. For example, when a user posts a thread wanting a "rat", what he/she might really want is malware, i.e., "Remote Access Trojan". As those jargons facilitate an enormous underground economy [18], identifying the real meaning of dark words is essential for understanding cybercrime activities and is an important step in order to measure, monitor and mitigate illicit activity.

Recently, there has been substantial research interest in the intersection of Cybersecurity, Information Retrieval [7–9,11,12,15,16,19], and Natural Language Processing [13,17,22–24]. However, dark jargon detection and interpretation has not been well studied since only two works are directly related: Yang et al. [20] proposes to detect dark jargon by utilizing a search engine. The authors scrape data from pages that tend to contain dark terms, filter out key words and use the search engine's similar search function to discover new dark words. Yuan

D. Hiemstra et al. (Eds.): ECIR 2021, LNCS 12657, pp. 393–400, 2021.
https://doi.org/10.1007/978-3-030-72240-1_40

et al. [21] leverages the context of a word as a representation for the word's meaning. The intuition is that dark words in dark forums appear in drastically different contexts compared to reputable online corpora (e.g., Wikipedia). Dark words are categorized into five general classes. For example, "blueberry" is categorized as "drug" and not as "marijuana", which would be more beneficial for interpretation. We address this limitation as our method provides more interpretable meaning representations by utilizing probability distributions over context words. Another shortcoming of previous approaches is that the actual meaning of the identified dark jargon is mostly unknown. We alleviate this problem by making our framework more expressive and allow dark terms to be mapped to any word/category where the meaning is known. Furthermore, our framework is completely general as it does not require external resources, such as Wikipedia or a search engine.

We formalize the problem of finding underground jargon into a general framework of finding a probabilistic mapping function of dark words to word meanings. We investigate a specific case of this general framework where we find binary mappings of dark words to "clean" words, which are words that have no hidden meaning. Further, we develop novel methodology to find dark jargon words in underground forums automatically using the difference in word distributions. This methodology enables us to create interpretable representations of jargon words that can be used to further explain their hidden meanings. In our experiments we make use of a dark corpus of underground forums and evaluate our methodology. We find that our method successfully identifies dark words in a simulated and a real-world setting.

2 Approach

2.1 General Framework

In our general framework we use words with no hidden meanings as an direct explanation for the hidden meaning of dark jargon words. Thus, in the most general sense we are interested in a mapping function $hidden\_meaning(V_{dark})$ that takes as input a vocabulary of dark words V_{dark} and outputs a mapping to a vocabulary of "clean" words V_{clean}, with no hidden meaning. This mapping can be a probability distribution, which expresses the probability of relatedness of a dark word in V_{dark} to all clean words in V_{clean}.

In this work, we investigate the specific case where the probability distribution is forced to have only a single element with probability 1.0. Thus, we are interested in a binary mapping from V_{dark} to V_{clean}. However, it is possible to retrieve a more fine-grained distribution, which we leave for future work.

2.2 Problem Setup

Our problem setup is as follows: given two text corpora, a dark corpus C_{dark} and a clean corpus C_{clean}, the goal is to find the words that are likely to have

hidden meanings in the dark corpus and identify their true meaning. We further build a joint vocabulary V, which is the most frequent N words from the union of C_{dark} and C_{clean}. Then, for each word $w_d \in C_{dark}$ we want to find a word $w_c \in C_{clean}$, such that w_c expresses the hidden meaning of w_d.

We first get a word vector for each word in both corpora, such that every word $w \in V$ has two word vectors w_d and w_c. Second, for each w_d, we rank all clean word vectors, such that we find the words in C_{clean} that are most similar to w_d, thereby assuming that the meaning of w_d is related to closeness of words in C_{clean} according to some similarity measure.

We propose to use two methodologies for achieving this mapping. We first introduce a novel method based on word distributions and Kullback-Leibler-divergence [10]. We then find another suitable method in cross-context lexical analysis [14]. In our experiments we compare both methods to understand which one is more performance for our task.

2.3 Word Distribution Modeling and KL-Divergence

We start by introducing the word distribution and KL-divergence method. The intuition is that a dark word, e.g., "rat", will appear in different contexts than the clean word "rat". It will therefore have a context more similar to a clean word like "malware", as it would have to "mouse". When we represent word contexts as probability distributions over words, we find that "rat" in the dark corpus and "malware" in the clean corpus have the most similar distributions.

For each word in our vocabulary V, we build a unigram probability distribution of all other words in V. In order to build this probability distribution we make use of a sliding window technique, where we look at k words before and after the occurrence of the word under consideration. We choose to employ this technique, since we are interested in a word's immediate context, as compared to the entire document, which is often used in unigram language modes.

More specifically, to build a word distribution for a word $w \in V$, we first get a length $|V|$ all zero word count vector, with each entry mapped to a word in V. We then go through the whole corpus C, and for each occurrence of w, we look at k words before and after it, increase the value of the counter vector at corresponding indices. To get a probability distribution over context words, we perform maximum-likelihood estimation and divide each element in the vector by the sum of all vector elements. We further employ smoothing to handle the zero-value probability problem, where we smooth the word distribution of w. We get two word distributions for each word $w \in V$: One distribution estimated from the dark text $P(w_d|C_{dark})$ and one from the clean text $P(w_c|C_{clean})$. To get two words' dissimilarity $dissim(w_d, w_c)$, we calculate the KL-Divergence between the two probability distributions as in Eq. 1. Finally, for each dark jargon we define it's hidden meaning as the clean word with the lowest dissimilarity to our target dark word w_d (Eq. 2).

$$dissim(w_d, w_c) = KL(P(w_d|C_{dark})||P(w_c|C_{clean})) \tag{1}$$

$$hidden\_meaning_{KL}(w_d) = \underset{w_c \in C_{clean}}{\arg\min}\ dissim(w_d, w_c) \tag{2}$$

2.4 Cross-context Lexical Analysis

Another suitable method for our problem setup is cross-context lexical analysis (CCLA) [14]. Here, the goal is to analyze differences and similarities of words across different contexts. Contexts are usually defined over document collections, which is very akin to our problem setting. Therefore, we can directly apply this methodology to our problem, where the two corpora under consideration are our dark and clean corpora C_{dark} and C_{clean}, respectively. Using CCLA as a framework, we can leverage it as yet another method to measure the difference of words in a clean and dark context.

Following Massung [14], we define a scoring function as in Eq. 3, where $cos(w_1, w_2, C)$ is the cosine similarity of the word vector of w_1 and w_2 computed over corpus C. $NN(w, C, k)$ is the corresponding length-k vector, where each entry has the value of the cosine similarity of w's word vector and the k closest word vectors. W_{common} is the intersection of the set of k words in corpus C with highest similarity to the word vectors w_d and w_c (Eq. 4). Note that our function is a slight variation of Massung [14], as we modify it to be suitable for two input words (w_d, w_c), rather than just a single input word. Essentially, ϕ measures the similarity of the usage of w_d and w_c across C_{dark} and C_{clean}. To generalize, for each word in $w_d \in C_{dark}$, we find a $w_c \in C_{clean}$ that maximizes ϕ, which is then used as the mapping for w_d (Eq. 5).

$$\phi(w_d, w_c, C_{dark}, C_{clean}, k) = \frac{\Sigma_{w \in W_{common}} cos(w, w_d, C_{dark}) * cos(w, w_c, C_{clean})}{||NN(w_d, C_{dark}, k)|| * ||NN(w_c, C_{clean}, k)||} \tag{3}$$

$$W_{common}(w_d, w_c, C_{dark}, C_{clean}, k) = W(w_d, C_{dark}, k) \bigcap W(w_c, C_{clean}, k) \tag{4}$$

$$hidden\_meaning_{CCLA}(w_d) = \underset{w_c \in C_{clean}}{\arg\max}\ \phi(w_d, w_c, C_{dark}, C_{clean}, k) \tag{5}$$

3 Experiments

3.1 Experimental Setup

We aim to answer three research questions: (1) What is the performance of the word distribution method? (2) What is the performance of CCLA compared to the word distribution method? (3) What are the qualitative results in terms of dark jargons identified?

Table 1. Clean-clean Evaluation of the Word Distribution (KL) and Cross-context Lexical Analysis (CCLA) Methods using the Mean Reciprocal Rank (MRR) metric.

Method	MRR all words	MRR dark words
KL	0.909	**0.892**
CCLA	**0.974**	0.479

Datasets. We make use of two datasets in our experiments, where each dataset has stowords and punctuation removed, words are lower-cased and stemmed: (1) *Dark Corpus.* Taken from Yuan et al. [21], our dark corpus contains user posts scraped from four major underground forums: Silk Road [5], Nulled [3], Hackforums [2] and Dark0de [1]. The combined corpus contains 376,989 posts. (2) *Clean Corpus.* The clean corpus contains a web scrape of 1.2 million reddit [4] threads from 1,697 top subreddits in terms the number of subscribers.

Evaluation Environments. In order to answer our research questions, we build two evaluation environments: The first environment aims to evaluate the quantitative performance of our method. Since no gold standard data is available for this task, we decided to simulate the dark jargons in the dataset. The second environment aims to measure the quality of the dark jargons identified on real data. Here, we manually check if the model can find real meanings of dark words on non-simulated data. The two environments are created as follows:

(1) *Clean-Clean*: We randomly split the documents in the clean corpus into two splits. In the first split, namely $clean_1$, we randomly select 500 words and prefix them with a dash ("_"). For example, if the word "strawberry" was selected, a sentence like "John loves **strawberry** milkshakes" would be turned into "John loves _**strawberry** milkshakes". The second split, namely $clean_2$, remains unmodified. Once we run the models on this corpus, for each word in the vocabulary in $clean_1$, we get its corresponding ranking list of nearest words in $clean_2$. We separately investigate the dashed words (words with "_"). For those words, the top-ranked word should be the word itself, i.e., the original word without the dash ("_"). We calculate the mean reciprocal rank (MRR) as a performance evaluation metric for the clean-clean dataset. We separately measure MRR for all words in the vocabulary and for our simulated dark words.

(2) *Dark-Clean*: For the real world dataset, we run our word distribution method and get a ranked list of nearest words in *clean* for each word in *dark*. We then do a manual evaluation of random dark words our method retrieves to find out their hidden meanings.

Hyperparameters. We use the following parameters for our methods, which we empirically found to perform best: We use a vocabulary size of 10,000. For the word distribution method, we use a sliding window size k of 10 and *Laplace*[1]

[1] We found that Dirichlet smoothing was less effective.

Table 2. Dark-clean Manual Evaluation based on our Word Distribution Method.

Dark Word	Clean word	Meaning
gdp	kush	Grand Daddy Purps (type of marijuana)
blueberry	kush	Type of marijuana
coke	cocaine	Nickname for cocaine
klonopin	xanax	Sedative medication
shrooms	lsd	Hallucinogenic drug similar to LSD
bubba	kush	Type of marijuana
ecstasy	mdma	Nickname for mdma
dilaudid	oxy, morphine	Strong painkiller (aka: hospital heroin)
pineapple	kush	Type of marijuana
zeus	botnet	Botnet malware
rat	malware	Remote Access Trojan (malware)

smoothing with $\alpha = 1$. For CCLA, we use an embedding size of 300 and a neighborhood size k of 100.

3.2 Experimental Results

We now move on to our experimental results and answer our three research questions. Table 1 shows the results of our proposed word distribution method (KL) and CCLA for all words in the vocabulary and our simulated dark words. To answer our first research question, we see that the KL method performs well, with an MRR around 0.9 for all words in the vocabulary and the simulated dark words. To answer research question two, we find that the CCLA method performs better for all words, however, it is performing much worse for the simulated dark words. Since finding dark words is the goal of our research, we can conclude that KL outperforms CCLA for our task.

To answer research question three, we perform a manual evaluation into the dark words that were identified by our method on a real-world corpus. In Table 2, we present a list of dark words identified by our word distribution method and the clean word that was mapped to the corresponding dark word. We also show the meaning that we manually identified using a slang dictionary or by searching for the highest ranked clean words online. As can be seen from the table, our method retrieves meaningful results since our analysis finds many drug-related and malware-related terms. We take these results as evidence for the potential of our method for finding dark term meanings in a real-world setting.

4 Conclusion and Future Work

We have shown that our approach based on word distributions derived from a word's context is effective for jargon detection and it outperformed a related

method based on cross-context lexical analysis. Furthermore, our method leverages word distributions and is therefore inherently interpretable, as individual word probabilities can be thought of as importance weights of a word's context. In the future, we plan to further improve interpretability of dark terms by leveraging external large-scale knowledge resources that define the meaning of slang words, such as Urban Dictionary [6].

Acknowledgment. This material is based upon work supported by the National Science Foundation under Grant No. 1801652.

References

1. Dark0de (forum). https://en.wikipedia.org/wiki/Dark0de
2. Hackforums.https://hackforums.net
3. Nulled (forum). https://www.nulled.to
4. reddit (forum). https://www.reddit.com
5. Silk Road (marketplace). https://en.wikipedia.org/wiki/Silk_Road_(marketplace)
6. Urban dictionary. https://urbandictionary.com
7. Husari, G., Al-Shaer, E., Ahmed, M., Chu, B., Niu, X.: TTPDrill: automatic and accurate extraction of threat actions from unstructured text of CTI sources. In: Proceedings of the 33rd Annual Computer Security Applications Conference, pp. 103–115 (2017)
8. Husari, G., Niu, X., Chu, B., Al-Shaer, E.: Using entropy and mutual information to extract threat actions from cyber threat intelligence. In: International Conference on Intelligence and Security Informatics (ISI), pp. 1–6 (2018)
9. Khandpur, R.P., Ji, T., Jan, S., Wang, G., Lu, C.T., Ramakrishnan, N.: Crowdsourcing cybersecurity: cyber attack detection using social media. In: Proceedings of the Conference on Information and Knowledge Management, pp. 1049–1057 (2017)
10. Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Stat. **22**(1), 79–86 (1951)
11. Liao, X., Yuan, K., Wang, X., Li, Z., Xing, L., Beyah, R.: Acing the IOC game: toward automatic discovery and analysis of open-source cyber threat intelligence. In: Proceedings of the Conference on Computer and Communications Security, pp. 755–766 (2016)
12. Liao, X., et al.: Seeking nonsense, looking for trouble: efficient promotional-infection detection through semantic inconsistency search. In: Symposium on Security and Privacy (SP) (2016)
13. Lim, S.K., Muis, A.O., Lu, W., Ong, C.H.: MalwaretextDB: a database for annotated malware articles. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 1557–1567 (2017)
14. Massung, S.A.: Beyond topic-based representations for text mining. Ph.D. thesis, University of Illinois at Urbana-Champaign (2017)
15. Mittal, S., Das, P.K., Mulwad, V., Joshi, A., Finin, T.: Cybertwitter: using twitter to generate alerts for cybersecurity threats and vulnerabilities. In: Proceedings of the International Conference on Advances in Social Networks Analysis and Mining, pp. 860–867 (2016)

16. Mulwad, V., Li, W., Joshi, A., Finin, T., Viswanathan, K.: Extracting information about security vulnerabilities from web text. In: Proceedings of the International Conferences on Web Intelligence and Intelligent Agent Technology, pp. 257–260 (2011)
17. Seyler, D., Li, L., Zhai, C.: Semantic text analysis for detection of compromised accounts on social networks. In: Proceedings of the International Conference on Advances in Social Network Analysis and Mining (2020)
18. Thomas, K., et al.: Framing dependencies introduced by underground commoditization. In: Workshop on the Economics of Information Security (2015)
19. Tsai, F.S., Chan, K.L.: Detecting cyber security threats in weblogs using probabilistic models. In: Pacific-Asia Workshop on Intelligence and Security Informatics, pp. 46–57 (2007)
20. Yang, H., et al.: How to learn klingon without a dictionary: Detection and measurement of black keywords used by the underground economy. In: Symposium on Security and Privacy (SP), pp. 751–769 (2017)
21. Yuan, K., Lu, H., Liao, X., Wang, X.: Reading thieves' cant: automatically identifying and understanding dark jargons from cybercrime marketplaces. In: USENIX Security Symposium (2018)
22. Zhou, S., Long, Z., Tan, L., Guo, H.: Automatic identification of indicators of compromise using neural-based sequence labelling. In: 32nd Pacific Asia Conference on Language, Information and Computation (2018)
23. Zhu, Z., Dumitras, T.: Featuresmith: automatically engineering features for malware detection by mining the security literature. In: Proceedings of the Conference on Computer and Communications Security, pp. 767–778 (2016)
24. Zhu, Z., Dumitras, T.: Chainsmith: Automatically learning the semantics of malicious campaigns by mining threat intelligence reports. In: European Symposium on Security and Privacy (EuroS&P), pp. 458–472 (2018)