DarkBERT: A Language Model for the Dark Side of the Internet

Youngjin Jin[1] **Eugene Jang**[2] **Jian Cui**[2]
Jin-Woo Chung[2] **Yongjae Lee**[2] **Seungwon Shin**[1]

[1]KAIST, Daejeon, South Korea
[2]S2W Inc., Seongnam, South Korea
[1]`{ijinjin,claude}@kaist.ac.kr`
[2]`{genesith,geeoon19,jwchung,lee}@s2w.inc`

Abstract

Recent research has suggested that there are clear differences in the language used in the Dark Web compared to that of the Surface Web. As studies on the Dark Web commonly require textual analysis of the domain, language models specific to the Dark Web may provide valuable insights to researchers. In this work, we introduce DarkBERT, a language model pretrained on Dark Web data. We describe the steps taken to filter and compile the text data used to train DarkBERT to combat the extreme lexical and structural diversity of the Dark Web that may be detrimental to building a proper representation of the domain. We evaluate DarkBERT and its vanilla counterpart along with other widely used language models to validate the benefits that a Dark Web domain specific model offers in various use cases. Our evaluations show that DarkBERT outperforms current language models and may serve as a valuable resource for future research on the Dark Web.

1 Introduction

The *Dark Web* is a subset of the Internet that is not indexed by web search engines such as Google *and* is inaccessible through a standard web browser. To access the Dark Web, specialized overlay network applications such as Tor (The Onion Router) (Dingledine et al., 2004) are required. Tor also hosts *hidden services* (*onion services*) — web services in which the client and the server IP addresses are hidden from each other (Biryukov et al., 2013).

This sense of identity obscurity provided to the Dark Web users comes with a catch; many of the underground activities prevalent in the Dark Web are immoral/illegal in nature, ranging from content hosting such as data leaks to drug sales (Al Nabki et al., 2017; Jin et al., 2022). As such, the popularity of the Dark Web as a platform of choice for malicious activities has garnered interest from researchers and security experts alike.

To handle the ever-changing landscape of modern cyber threats, cybersecurity experts and researchers have started to employ natural language processing (NLP) methods. Gaining evidence-based knowledge such as indicators of compromise (IOC) to mitigate emerging threats is an integral part of modern cybersecurity known as *cyber threat intelligence* (CTI) (Liao et al., 2016; Bromiley, 2016), and modern NLP tools have become an indispensable part of CTI research. As such, the use of NLP techniques has also been extended to the Dark Web (Jin et al., 2022; Yoon et al., 2019; Choshen et al., 2019; Al Nabki et al., 2017; Al-Nabki et al., 2019; Yuan et al., 2018). The continued exploitation of the Dark Web as a platform of cybercrime makes it a valuable and necessary domain for CTI research.

Recently, Jin et al. (2022) observed that using a BERT-based classification model achieves state-of-the-art performance among available NLP methods in the Dark Web. However, BERT is trained on *Surface Web*[1] content (i.e., Wikipedia and BookCorpus) (Devlin et al., 2019), which has different linguistic characteristics from that of the Dark Web (Choshen et al., 2019). In the context of CTI, this implies that popular pretrained language models such as BERT are not ideal for Dark Web research in terms of extracting useful information due to the differences in the language used in the two domains. Consequently, an NLP tool that is suitable for application in Dark Web domain tasks would prove to be valuable in the ongoing efforts of Dark Web cybersecurity.

In this paper, we propose **DarkBERT**, a new language model pretrained on a Dark Web corpus. To measure the usefulness of DarkBERT in handling cyber threats in the Dark Web, we evaluate DarkBERT in tasks related to detecting underground activities. We compare DarkBERT to other widely

[1]Web services and content that are readily available and indexed in common search engines such as Google

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7515–7533

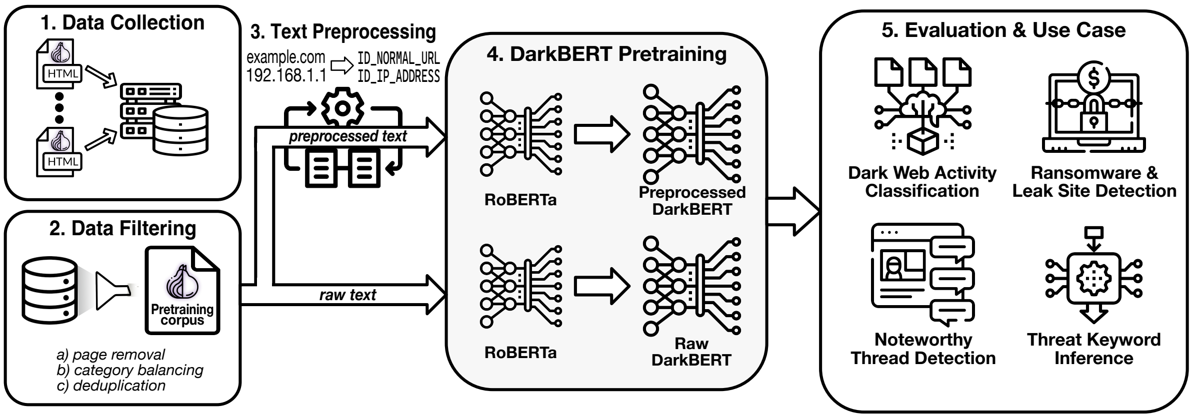


Figure 1: Illustration of the DarkBERT pretraining process and the various use case scenarios for evaluation.

used pretrained language models BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) that are trained on data found in the Surface Web to verify the efficacy of DarkBERT in Dark Web domain texts. Our evaluation results show that DarkBERT-based classification model outperforms that of known pretrained language models. Furthermore, we present potential use cases to illustrate the benefits of utilizing DarkBERT in cybersecurity-related tasks such as Dark Web forum thread detection and ransomware leak site detection.

Our contributions are summarized as follows:

- We introduce DarkBERT, a language model pretrained on the Dark Web which is capable of representing the language used in the domain compared to that of the Surface Web.
- We illustrate the effectiveness of DarkBERT in the Dark Web domain. Our evaluations show that DarkBERT is better suited for NLP tasks on Dark Web specific texts compared to other pretrained language models.
- We demonstrate potential use case scenarios for DarkBERT and show that it is better-suited for tasks related to cybersecurity compared to other pretrained language models.
- We provide new datasets used for our Dark Web domain use case evaluation.

2 Related Work

The recent availability of Dark Web resources (Jin et al., 2022; Al Nabki et al., 2017; Al-Nabki et al., 2019) has made it possible to explore the differences between the languages used in the Dark Web and the Surface Web. Choshen et al. (2019) explored the differences in the illegal and legal pages in the Dark Web and found a number of distinguishing features between the two domains such as named entity, vocabulary, and syntactic structure. Their analyses using standard NLP tools have also suggested that processing text in the Dark Web domain would require considerable domain adaptation. The linguistic differences between the Surface Web and the Dark Web were further examined by Jin et al. (2022) through linguistic features such as part-of-speech (POS) distribution and vocabulary usage between the texts in the two domains.

Recently, Ranaldi et al. (2022) explored the use of pretrained language models over Dark Web texts to examine the effectiveness of such models, and suggested that lexical and syntactic models such as GloVe (Pennington et al., 2014) outperform pretrained models in some specific Dark Web tasks. Meanwhile, Jin et al. (2022) demonstrated that pretrained language models in some Dark Web tasks such as Dark Web activity classification perform better than simple lexical models, suggesting that language models like BERT show promising results in the Dark Web. Either way, a domain-specific pretrained language model would be beneficial in that it would be able to represent the language used in the Dark Web, which may effectively reduce the performance issues faced in previous experiments.

3 DarkBERT Construction

In this section, we describe the process for building our Dark Web domain-specific pretrained language model, **DarkBERT**. We begin by collecting pages to build the text corpus used for pretraining DarkBERT (Section 3.1). Then, we filter the raw text corpus and employ text preprocessing methods for pretraining purposes (Section 3.2). Finally, we pretrain DarkBERT using the text corpus (Section 3.3).

Table 1: The two variations of Dark Web text corpus used to train DarkBERT.

Corpus	Data Size	Time Taken to Pretrain DarkBERT
Raw Text	5.83 GB	367.4 hours (15.31 days)
Preprocessed Text	5.20 GB	361.6 hours (15.07 days)

An overview of the DarkBERT construction process is illustrated in Figure 1.

3.1 Data Collection

A massive text corpus consisting of pages from the Dark Web is necessary for pretraining DarkBERT. We initially collect seed addresses from Ahmia[2] and public repositories containing lists of onion domains. We then crawl the Dark Web for pages from the initial seed addresses and expand our list of domains, parsing each newly collected page with the HTML title and body elements of each page saved as a text file. We also classify each page by its primary language using fastText (Joulin et al., 2016a,b) and select pages labeled as English. This allows DarkBERT to be trained on English texts as the vast majority of Dark Web content is in English (Jin et al., 2022; He et al., 2019). A total of around 6.1 million pages was collected. The full statistics of the crawled Dark Web data is shown in Table 8 of the Appendix.

3.2 Data Filtering and Text Processing

While the text data collected in Section 3.1 is of considerable size, a portion of the data contains no meaningful information such as error messages or duplicates of other pages. Therefore, we take three measures — *removal of pages with low information density*, *category balancing*, and *deduplication* — to retain useful page samples in the pretraining corpus and remove unnecessary pages. In addition, it is critical that the model does not learn representations from sensitive information. Although a previous study stated that language models pretrained with sensitive data are unable to extract sensitive information with simple methods, the possibility cannot be ruled out using more sophisticated attacks (Lehman et al., 2021). To this end, we preprocess the pretraining corpus to address ethical considerations using identifier masks or removing texts entirely, depending on the type of the target text. The details of filtering and text preprocessing are described in Sections B and C of the Appendix.

[2]https://ahmia.fi/

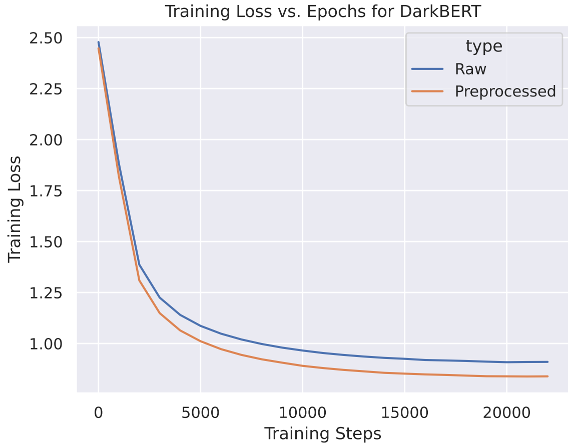


Figure 2: Training steps vs. training loss graph for raw and preprocessed versions of DarkBERT.

3.3 DarkBERT Pretraining

In order to observe the impact of text preprocessing on DarkBERT's performance, we build two versions of DarkBERT: one with raw text data (whitespace removal applied) and the other with preprocessed text following Section 3.2. The size of each pretraining corpus is shown in Table 1, and the training losses for the two models are shown in Figure 2.

We leverage an existing model architecture instead of starting from scratch for pretraining. This is done to reduce computational load and retain the general English representation learned by the existing model. We choose RoBERTa (Liu et al., 2019) as our base initialization model as it opts out of the Next Sentence Prediction (NSP) task during pretraining, which may serve as a benefit to training a domain-specific corpus like the Dark Web as sentence-like structures are not as prevalent compared to the Surface Web.

The Dark Web pretraining text corpus is fed to the `roberta-base` model in the Hugging Face [3] library as an initial base model. For compatibility between DarkBERT and RoBERTa, we use the same BPE (byte-pair encoding) tokenization vocabulary used in the original RoBERTa model, with each page in the pretraining corpus separated using RoBERTa's separator token `</s>`. The two versions of DarkBERT only differ in the corpus used for pretraining (*raw* vs. *preprocessed*); all other factors such as training hyperparameters are equally set. The models are pretrained using a script written in PyTorch (Paszke et al., 2019). Additional

[3]https://huggingface.co/

Table 2: Dataset statistics used for Dark Web activity categorization.

DUTA (DUTA-10K)		CoDA	
Category	**Page count**	**Category**	**Page count**
Hosting & Software	1949	Others	2131
Cryptocurrency	798	Pornography	1171
Down	714	Drugs	967
Locked	682	Financial	956
Personal	419	Gambling	756
Counterfeit Credit Cards	392	Crypto	745
Social Network	293	Hacking	630
Drugs	290	Arms	597
Services	284	Violence	482
Pornography	226	Electronics	420
Marketplace	189		
Hacking	182		
Forum	128		
Total	6524	Total	8855

details on pretraining including hyperparameters and training equipment are listed in Table 11 and Section D of the Appendix.

4 Evaluation: Dark Web Activity Classification

In this section, we describe the methods of evaluation and the datasets used to evaluate DarkBERT and other language models. Since page classification has often been performed in past works (Al Nabki et al., 2017; Choshen et al., 2019; Ranaldi et al., 2022), we also choose Dark Web activity classification as the main Dark Web domain benchmark experiment for evaluation. We additionally conduct experiments on multiple use case scenarios, which is described in detail in Section 5.

4.1 Datasets

The distribution of various activities has been studied at large, resulting in publicly available Dark Web text datasets known as DUTA (Al Nabki et al., 2017; Al-Nabki et al., 2019) and CoDA (Jin et al., 2022). We use english texts in the latest version of DUTA (also known as DUTA-10K) and CoDA in our experiments. Since DUTA and CoDA use different categorization methods, we train separate classifiers for each dataset. Since DUTA contains certain categories that are very small in size (for example, there are only 3 pages under the *Human Trafficking* category), we remove categories that have a low page count (under 1% of total page count). We also remove the *Empty* category in DUTA as these pages are mostly empty, which is not ideal for text classification. No modifications are made to the CoDA dataset. Finally, we preprocess texts in both DUTA and CoDA following Section 3.2. Per-category statistics for the two datasets used for our activity classification experiment are shown in Table 2.

4.2 Experimentation

The classification experiment is conducted on the two versions of DarkBERT and two widely used language models: BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). Although RoBERTa (and the two variants of DarkBERT which use RoBERTa as their base model) is a cased language model which distinguishes between capitalized words and uncapitalized words, BERT comes in two versions: a cased model and an uncased model. To observe if letter case has any effect on classification performance, we build a separate, *uncased* version of DUTA and CoDA in which every character is converted to lowercase. In summary, we evaluate the Dark Web activity classification task using DUTA and CoDA — each with two variants: cased and uncased corpus — on two versions of DarkBERT (*raw* and *preprocessed*), two versions of BERT (*cased* and *uncased*), and RoBERTa.

4.3 Results and Discussion

The result of Dark Web activity classification is shown in Table 3. We observe that DarkBERT outperforms other language models for both datasets and their variants. However, it is also worth noting that both BERT and RoBERTa exhibit relatively similar performances to DarkBERT. This is in line with previous classification experiments with CoDA, which have shown that BERT is able to adapt relatively well to other domains (Jin et al., 2022). RoBERTa also performs slightly better compared to BERT, which reflects the advantages in performance that RoBERTa has over BERT as mentioned in the original paper (Liu et al., 2019).

We also observe that all language models perform significantly better for the CoDA dataset compared to DUTA. Upon closer inspection on the DUTA dataset, we find that some of the included categories in DUTA may not be suitable for classification tasks. For example, many of the pages in the *Hosting & Software* category contain duplicate texts, which may overfit the model during fine-tuning (DUTA in general has duplicate texts as mentioned by Al-Nabki et al. (2019)). In addition, some of the pages seem to be ambiguous in terms of classification the DUTA dataset; for eaxmple, we observe pages classified as *Hosting & Software*

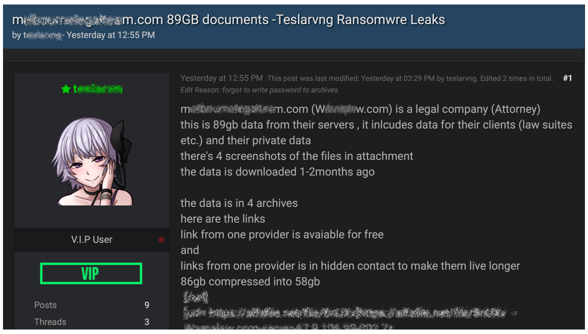
Table 3: Dark Web activity classification evaluation results. Boldface indicates best performance.

Dataset	Model	Precision	Recall	F1 score	Dataset	Model	Precision	Recall	F1 score
DUTA$_{cased}$	BERT$_{cased}$	77.31	76.91	77.09	CoDA$_{cased}$	BERT$_{cased}$	92.12	92.16	92.13
	BERT$_{uncased}$	78.21	78.20	78.19		BERT$_{uncased}$	92.83	92.67	92.75
	RoBERTa	78.54	78.79	78.63		RoBERTa	93.36	93.27	93.31
	DarkBERT$_{raw}$	**80.11**	79.94	**80.01**		DarkBERT$_{raw}$	94.15	**94.35**	94.25
	DarkBERT$_{preproc}$	79.90	**80.08**	79.98		DarkBERT$_{preproc}$	**94.26**	94.33	**94.29**
DUTA$_{uncased}$	BERT$_{cased}$	78.11	77.97	77.99	CoDA$_{uncased}$	BERT$_{cased}$	92.86	92.85	92.85
	BERT$_{uncased}$	78.21	78.20	78.19		BERT$_{uncased}$	92.83	92.67	92.75
	RoBERTa	78.42	78.36	78.37		RoBERTa	93.30	93.40	93.34
	DarkBERT$_{raw}$	79.47	79.49	79.47		DarkBERT$_{raw}$	**94.46**	94.45	**94.46**
	DarkBERT$_{preproc}$	**79.65**	**79.77**	**79.69**		DarkBERT$_{preproc}$	94.31	**94.53**	94.42

that do not contain any activities related to hosting or software related terms.

We take a deeper look at the activity classification results on the CoDA (cased) dataset by constructing confusion matrices (Figure 8 of the Appendix) to check for misclassifications. We find that in general, the two versions of DarkBERT show the best classification performance for most categories. The highest number of correct classifications for every category occurs in either one of the DarkBERT models. However, some categories such as *Drugs*, *Electronics*, and *Gambling* show very similar performances across all four models. This is likely due to the high similarity of pages in such categories, making classification easier even with the differences in the language used in the Dark Web. Finally, we inspect the language models using their predictions through error analysis, which is described in Section E.1 of the Appendix.

(a) A ransomware leak site sample



(b) A noteworthy thread sample

Figure 3: A ransomware leak site and noteworthy thread samples that DarkBERT correctly classified but are misclassified by other language models.

5 Use Cases in the Cybersecurity Domain

In this section, we introduce three Dark Web domain use cases for DarkBERT and demonstrate its effectiveness over existing language models in cybersecurity / CTI applications. We list details on the experimental setup for each use case in Sections E.2 and E.3 of the Appendix.

5.1 Ransomware Leak Site Detection

One type of cybercrime that occurs on the Dark Web is the selling or publishing of private, confidential data of organizations leaked by ransomware groups. This can occur in the form of leak sites that expose victims and threaten to release sensitive data (such as financial information, private assets, and personal identification) of uncooperative victims (Yuste and Pastrana, 2021). It would thus be beneficial for security researchers to automatically identify such websites. We formulate the task of leak site detection as a binary classification problem of predicting whether a given page is a leak site or not. We compare the effectiveness of this task using the pretrained language models used for evaluation in Section 4 (BERT, RoBERTa, and DarkBERT).

Datasets: We monitor leak sites of 54 popular ransomware groups for two years (from May 2020

Table 4: Ransomware leak site detection performance. Boldface indicates the best performance.

Input	Model	Precision	Recall	F1 score
Raw	BERT_{cased}	75.83	69.52	71.01
	$\text{BERT}_{uncased}$	77.18	73.90	72.77
	RoBERTa	39.83	36.00	36.27
	DarkBERT_{raw}	**78.81**	**83.62**	**79.98**
Preprocessed	BERT_{cased}	76.81	68.19	70.13
	$\text{BERT}_{uncased}$	71.97	71.62	70.77
	RoBERTa	48.36	45.14	44.31
	$\text{DarkBERT}_{preproc}$	**85.16**	**84.57**	**84.11**

Table 5: Noteworthy thread detection performance. Boldface indicates best performance.

Input	Model	Precision	Recall	F1 score
Raw	BERT_{cased}	55.09	19.91	26.90
	$\text{BERT}_{uncased}$	52.34	23.49	28.51
	RoBERTa	28.97	17.89	21.38
	DarkBERT_{raw}	**75.93**	**43.08**	**52.85**
Preprocessed	BERT_{cased}	61.43	20.48	28.81
	$\text{BERT}_{uncased}$	45.46	21.52	26.16
	RoBERTa	29.04	15.27	18.71
	$\text{DarkBERT}_{preproc}$	**72.44**	**45.13**	**54.17**

to April 2022), and periodically download HTML files from these sites especially when new victims are revealed[4]. Leak sites typically contain the victim organization name, descriptions of leaked data, and threat statements with sample data (refer to Figure 3a for an example leak site page).

We collect pages by randomly choosing a maximum of three pages with different page titles from each of the 54 leak sites, and label them as positive examples. To create negative data, rather than collecting random pages in the Dark Web, we consider pages with content similar to that of leak sites to make the task more challenging. To select such pages, we utilize the activity category classifier from Section B used for balancing the pretraining corpus. The intuition behind using the activity classifier to select negative data is that the text content of certain categories like *Hacking* are more similar to that of leak sites than other less relevant categories such as *Pornography* and *Gambling*. Our pilot study suggests leak sites are mostly classified by the activity classifier as *Hacking*, followed by *Cryptocurrency*, *Financial*, and *Others*. Thus, we only collect Dark Web pages that are classified into one of these four categories and treat them as negative examples. Our training text data consists of 105 positive and 679 negative examples (pages). Training is done using 5-fold cross validation.

Results and Discussion: As shown in Table 4, DarkBERT outperforms other language models, demonstrating the advantages of DarkBERT in understanding the language of underground hacking forums on the Dark Web. Figure 3a shows a leak site sample correctly classified by DarkBERT but misclassified by other models. We also observe that while DarkBERT uses RoBERTa as a base model, RoBERTa itself shows a sharp drop in performance compared to the other models.

In addition, DarkBERT with preprocessed input performs better than the one with raw input, which highlights the importance of the text preprocessing step in terms of reducing superfluous information. As lengthy words or cryptocurrency addresses have been replaced with mask identifier tokens in the preprocessed input, such words present in the raw input may cause the tokenizer to produce uninformative tokens and affect task performance.

5.2 Noteworthy Thread Detection

Dark Web forums are often used for exchanging illicit information, and security experts monitor for *noteworthy* threads to gain up-to-date information for timely mitigation. Since many new forum posts emerge daily, it takes massive human resources to manually review each thread. Therefore, automating the detection of potentially malicious threads can significantly reduce the workload of security experts. Identifying noteworthy threads, however, requires a basic understanding of Dark Web-specific language. Similar to the aforementioned leak site detection, we can formulate this task as a binary classification problem to predict whether a given forum thread is noteworthy. We compare the performance of noteworthy thread detection for DarkBERT and the baseline models: BERT and RoBERTa.

Datasets: Identifying a thread as noteworthy is a highly subjective task. While there can be many different definitions for noteworthiness, we focus on activities in hacking forums that can potentially cause damage to a wide range of victims. To incorporate perspectives from the cybersecurity industry and ensure the quality of the dataset, we recruit

[4]URLs of such leak sites can be found in cybersecurity news, social media, open-source repositories, and so on. We used URLs taken from https://github.com/fastfire/deepdarkCTI/blob/main/ransomware_gang.md.

two researchers from a cyber threat intelligence company specializing in the analysis of hacking forums on the Dark Web to discuss types of noteworthy threads, and set annotation guidelines accordingly. We consider a thread of hacking forums to be *noteworthy* if it describes one of the following activities:

1. Sharing of confidential company assets such as admin access, employee or customer information, transactions, blueprints, source codes, and other confidential documents.
2. Sharing of sensitive or private information of individuals such as credit information, medical records, political engagement, passports, identifications, and citizenship.
3. Distribution of critical malware or vulnerabilities targeting popular software or organizations.

In particular, we place emphasis on activities targeting large private companies, public institutions, and industries. We choose RaidForums, one of the largest hacking forums, as our data source (together with its mirror and follow-up sites[5]). We collect 1,873 forum threads posted from July 2021 to March 2022 and work with the recruited annotators to select noteworthy threads. They first annotate the same 150 threads and achieve an inter-annotator agreement of 0.704 as measured by Cohen's Kappa, which indicates *substantial* agreement. All disagreements in the annotated dataset are then discussed and resolved by both annotators. The final dataset contains 249 positive (noteworthy) and 1,624 negative threads. We use the title and body text of each thread from the HTML source as input to the classifier and exclude any thread replies to simulate the practical scenario in which we categorize the noteworthiness of threads as soon as they are posted, and training is done using 5-fold cross validation.

Results and Discussion: As seen in Table 5, DarkBERT outperforms other language models in terms of precision, recall, and F1 score for both inputs. Similar to ransomware leak site detection, we see a noticeable performance drop for RoBERTa compared to the other models. Figure 3b shows a noteworthy thread sample that is correctly classified by DarkBERT but misclassified by other models. Due to the difficulty of the task itself, the overall performance of DarkBERT for real-world noteworthy thread detection is not as good compared to those of the previous evaluations and tasks. Nevertheless, the performance of DarkBERT over other language models shown here is significant and displays its potential in Dark Web domain tasks. By adding more training samples and incorporating additional features like author information, we believe that detection performance can be further improved.

It should also be noted that the performances for both raw and preprocessed inputs are similar for DarkBERT. Unlike data used for ransomware leak site detection, thread content is generally shorter than general webpage content, and sensitive information such as URLs and email addresses often influences the noteworthiness of threads (e.g., whether a victim is a leading global company or not). Since such information is masked for preprocessed inputs, contents of noteworthy threads and non-noteworthy threads may look similar from the viewpoint of the language models, which in turn deteriorates the performance of this task.

5.3 Threat Keyword Inference

In this section, we describe how we utilize the fill-mask function to derive a set of keywords that are semantically related to threats and drug sales in the Dark Web. Fill-mask is one of the main functionalities of BERT-family language models, which finds the most appropriate word that fits in the masked position of a sentence (masked language modeling). It is useful for capturing which keywords are used to indicate threats in the wild. In order to show that DarkBERT is robust in handling this task, we compare DarkBERT and $\text{BERT}_{\text{Reddit}}$, a BERT variant fine-tuned on a subreddit corpus whose topic is drugs (Zhu et al., 2021).

Figure 4 shows a sample drug sales page from the Dark Web in which a user advertises a Dutch MDMA pill with the *Philipp Plein* logo[6]. We then mask MDMA in the title phrase: `25 X XTC 230 MG DUTCH MDMA PHILIPP PLEIN`, and let DarkBERT and $\text{BERT}_{\text{Reddit}}$ suggest the most semantically related words. In Table 6, we list the suggested candidate words by the two language models, respectively. The result shows that DarkBERT suggests drug-related words (i.e., *Oxy* and *Champagne*) and a word closely related to drugs (i.e., *pills*). On the other hand, $\text{BERT}_{\text{Reddit}}$ mainly

[5] http://raidforums.com, http://rfmirror.com, http://breached.co

[6] While *Philipp Plein* normally refers to a German fashion brand, in this case, it indicates an MDMA pill on which the brand logo is imprinted. Well known car brands such as *Tesla*, *Rolls Royce*, and *Toyota* are also used in a similar manner.

Figure 4: An MDMA sales page excerpted from the Dark Web.

Table 6: Fill-mask task results. DarkBERT suggests specific words related to drugs while BERT suggests general words.

Language Model	Semantically Related Words
DarkBERT	pills, import, md, dot, translation, speed, up, oxy, script, champagne
$\text{BERT}_{\text{Reddit}}$	##man, champion, singer, rider, driver, sculptor, producer, manufacturer, ##er, citizen

suggests professions such as *singer*, *sculptor*, and *driver*, which are not relevant to drugs. This comes from the fact that the preceding word, *Dutch*, is usually followed by a vocational word in the Surface Web. We evaluate how each language model produces keyword sets semantically related to drugs in a quantitative fashion.

Datasets: To evaluate the language models, we use the sample dataset provided by Zhu et al. (2021). This dataset is composed of ground truth data (i.e., drug names and their euphemisms) and sentences containing the drug names[7].

Experimental Setting: We compare three language models: $\text{DarkBERT}_{\text{CoDA}}$, $\text{BERT}_{\text{CoDA}}$, and $\text{BERT}_{\text{Reddit}}$. The first two language models are fine-tuned on a subset of CoDA documents classified as drugs, whose base model is DarkBERT and BERT, respectively. $\text{BERT}_{\text{Reddit}}$ is a BERT variant fine-tuned on a subreddit corpus whose topic is drugs. To compare them quantitatively, we also use precision at k ($P@k$) following Zhu et al. (2021). Here, precision at k is the proportion of inferred keywords that are semantically related to a given drug name in the top-k set that are synonymous.

[7]The ground truth data are from the DEA Intelligence Report: https://www.dea.gov/sites/default/files/2018-07/DIR-022-18.pdf

Table 7: Quantitative performance metric of threat keyword inference. Precision at k ($P@k$) is measured with varying k in increments of 10.

	Top-10	Top-20	Top-30	Top-40	Top-50
$\text{DarkBERT}_{\text{CoDA}}$	**0.60**	**0.60**	0.50	0.42	0.42
$\text{BERT}_{\text{CoDA}}$	0.40	0.40	0.50	0.50	0.40
$\text{BERT}_{\text{Reddit}}$	0.40	0.45	**0.60**	**0.57**	**0.52**

Results and Discussion: The measured $P@k$ values are presented in Table 7. $\text{DarkBERT}_{\text{CoDA}}$ outperforms $\text{BERT}_{\text{Reddit}}$ for k ranging from 10 to 20, but is overtaken for higher values of k. Although $\text{DarkBERT}_{\text{CoDA}}$ shows better performance when k is small, the ground truth dataset contains euphemisms mainly derived from the Surface Web, and the words that $\text{DarkBERT}_{\text{CoDA}}$ infers as semantically related words are not contained in the dataset. For instance, *Tesla* and *Champagne* are drug names frequently seen in the Dark Web, but are not recognized as such in Zhu et al. (2021). On the other hand, *crystal* and *ice* are detected by both $\text{DarkBERT}_{\text{CoDA}}$ and $\text{BERT}_{\text{Reddit}}$ because they are used in both the Surface Web and the Dark Web.

6 Conclusion

In this study, we propose DarkBERT, a Dark Web domain-specific language model based on the RoBERTa architecture. To allow DarkBERT to adapt well to the language used in the Dark Web, we pretrain the model on a large-scale Dark Web corpus collected by crawling the Tor network. We also polish the pretraining corpus through data filtering and deduplication, along with data preprocessing to address the potential ethical concerns in Dark Web texts related to sensitive information. We show that DarkBERT outperforms existing language models with evaluations on Dark Web domain tasks, as well as introduce new datasets that can be used for such tasks. DarkBERT shows promise in its applicability on future research in the Dark Web domain and in the cyber threat industry. In the future, we also plan to improve the performance of Dark Web domain specific pretrained language models using more recent architectures and crawl additional data to allow the construction of a multilingual language model.

Ethical Considerations

Crawling the Dark Web

While crawling the Dark Web, we take caution not to expose ourselves to content that should not be accessed. For example, illicit pornographic content (such as child pornography) are easily found on the Dark Web. However, our automated web crawler takes the approach of removing any non-text media and only stores raw text data. By doing so, we do not expose ourselves to any sensitive media that is potentially illegal.

Sensitive Information Masking

Since the Dark Web harbors many activities considered to be malicious in nature, it is of utmost importance that sensitive data be left out of the text corpus used for pretraining. In particular, it is possible that some contents in the Dark Web may include private information such as e-mails, phone numbers, or IP addresses. To prevent DarkBERT from learning representations from sensitive texts as mentioned above, we mask our data before feeding it to our language model. While we have used both DarkBERT pretrained on preprocessed text and raw text for our experiments, we have used both of the models only for evaluation purposes. In addition, we only release the preprocessed version of DarkBERT in order to avoid any malpractices [8]. Through extensive testing on fill-mask and synonym inference tasks, we observe that it is infeasible to infer any characteristics or data that might be considered sensitive or private in nature using the preprocessed version of DarkBERT.

Annotator Ethics

For the task of noteworthy thread detection, we recruited two researchers from a cyber threat intelligence company as mentioned in Section 5.2, who agreed to assist us in our research methods. For a fair annotation process in the discussion of noteworthy threads, both recruited annotators handled the same set of thread data and were given equal compensations.

Use of Public Dark Web Datasets

Both DUTA and CoDA are available upon request by the respective authors, and due to the sensitive nature of the Dark Web domain, these datasets are only to be used for academic research purposes. We adhere to this guideline and only utilize the provided data in the context of research for this work. On the other hand, we do not plan to publicly release the Dark Web text corpus used for pretraining DarkBERT for similar reasons.

Limitations

Limited Usage for Non-English Tasks

As mentioned in Section 3, DarkBERT is pretrained using Dark Web texts in English. This is mainly our design choice as the vast majority (around 90%) of Dark Web texts is primarily in English (Jin et al., 2022). We believe that with the limited number of collected pages in non-English languages in our pretraining corpus, building a multilingual language model for the Dark Web domain would pose additional challenges, such as downstream task evaluations becoming more difficult to perform as they would require high-quality annotations of task-specific datasets in multiple languages. As such, while our language model is suitable for Dark Web tasks in English, further pretraining with language-specific data may be necessary to use DarkBERT for non-English tasks.

Dependence on Task-Specific Data

Although DarkBERT is a useful tool that can be directly applied to many existing Dark Web domain-specific tasks, some tasks may require further fine-tuning through task-specific data (as seen in Ransomware Leak Site Detection and Noteworthy Thread Detection use case scenarios in Section 5). However, there is a shortage of publicly available Dark Web task-specific data. While we provide the datasets used to fine-tune DarkBERT in this paper, additional research on tasks that do not have readily available datasets for use may require further manual annotation or handcrafting of necessary data to leverage DarkBERT to its maximum potential.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT). (No. 2022-0-00740, The Development of Darkweb Hidden Service Identification and Real IP Trace Technology)

[8] Researchers can request access to DarkBERT and related use case datasets by filling out the request form in the following url: https://s2w.inc/resources/darkbert/

References

Mhd Wesam Al Nabki, Eduardo Fidalgo, Enrique Alegre, and Ivan de Paz. 2017. Classifying illegal activities on tor network based on web textual contents. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 35–43, Valencia, Spain. Association for Computational Linguistics.

Mhd Wesam Al-Nabki, Eduardo Fidalgo, Enrique Alegre, and Laura Fernández-Robles. 2019. Torank: Identifying the most influential suspicious domains in the tor network. Expert Systems with Applications, 123:212–226.

Andrei Barysevich and Alexandr Solad. 2018. Litecoin emerges as the next dominant dark web currency. Recorded Future.

Alex Biryukov, Ivan Pustogarov, and Ralf-Philipp Weinmann. 2013. Trawling for tor hidden services: Detection, measurement, deanonymization. In 2013 IEEE Symposium on Security and Privacy, pages 80–94.

Andrei Z Broder, Moses Charikar, Alan M Frieze, and Michael Mitzenmacher. 2000. Min-wise independent permutations. Journal of Computer and System Sciences, 60(3):630–659.

Matt Bromiley. 2016. Threat intelligence: What it is, and how to use it effectively. SANS Institute InfoSec Reading Room, 15:172.

Leshem Choshen, Dan Eldad, Daniel Hershcovich, Elior Sulem, and Omri Abend. 2019. The language of legal and illegal activity on the Darknet. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4271–4279, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. Tor: The Second-Generation onion router. In 13th USENIX Security Symposium (USENIX Security 04), San Diego, CA. USENIX Association.

Siyu He, Yongzhong He, and Mingzhe Li. 2019. Classification of illegal activities on the dark web. In Proceedings of the 2019 2nd International Conference on Information Science and Systems, ICISS 2019, page 73–78, New York, NY, USA. Association for Computing Machinery.

Youngjin Jin, Eugene Jang, Yongjae Lee, Seungwon Shin, and Jin-Woo Chung. 2022. Shedding new light on the language of the dark web. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5621–5637, Seattle, United States. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomás Mikolov. 2016a. Fasttext.zip: Compressing text classification models. CoRR, abs/1612.03651.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. 2016b. Bag of tricks for efficient text classification. CoRR, abs/1607.01759.

Seunghyeon Lee, Changhoon Yoon, Heedo Kang, Yeonkeun Kim, Yongdae Kim, Dongsu Han, Sooel Son, and Shin Seungwon. 2019. Cybercriminal minds: An investigative study of cryptocurrency abuses in the dark web. In NDSS 2019.

Eric Lehman, Sarthak Jain, Karl Pichotta, Yoav Goldberg, and Byron Wallace. 2021. Does BERT pretrained on clinical notes reveal sensitive data? In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 946–959, Online. Association for Computational Linguistics.

Xiaojing Liao, Kan Yuan, XiaoFeng Wang, Zhou Li, Luyi Xing, and Raheem Beyah. 2016. Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16, page 755–766, New York, NY, USA. Association for Computing Machinery.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in

Python. Journal of Machine Learning Research, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Leonardo Ranaldi, Aria Nourbakhsh, Arianna Patrizi, Elena Sofia Ruzzetti, Dario Onorati, Francesca Fallucchi, and Fabio Massimo Zanzotto. 2022. The dark side of the language: Pre-trained transformers in the darknet.

Kyle Soska and Nicolas Christin. 2015. Measuring the longitudinal evolution of the online anonymous marketplace ecosystem. In 24th USENIX Security Symposium (USENIX Security 15), pages 33–48, Washington, D.C. USENIX Association.

Philipp Winter, Richard Köwer, Martin Mulazzani, Markus Huber, Sebastian Schrittwieser, Stefan Lindskog, and Edgar Weippl. 2014. Spoiled onions: Exposing malicious tor exit relays. In Privacy Enhancing Technologies, pages 304–331, Cham. Springer International Publishing.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Changhoon Yoon, Kwanwoo Kim, Yongdae Kim, Seungwon Shin, and Sooel Son. 2019. Doppelgängers on the dark web: A large-scale assessment on phishing hidden web services. In The World Wide Web Conference, WWW '19, page 2225–2235, New York, NY, USA. Association for Computing Machinery.

Kan Yuan, Haoran Lu, Xiaojing Liao, and XiaoFeng Wang. 2018. Reading thieves' cant: Automatically identifying and understanding dark jargons from cybercrime marketplaces. In 27th USENIX Security Symposium (USENIX Security 18), pages 1027–1041, Baltimore, MD. USENIX Association.

Javier Yuste and Sergio Pastrana. 2021. Avaddon ransomware: An in-depth analysis and decryption of infected systems. Computers & Security, 109:102388.

Wanzheng Zhu, Hongyu Gong, Rohan Bansal, Zachary Weinberg, Nicolas Christin, Giulia Fanti, and Suma Bhat. 2021. Self-supervised euphemism detection and identification for content moderation. In 42nd IEEE Symposium on Security and Privacy.

A Appendix

We list some additional details such as example figures from the DarkBERT evaluation and cybersecurity use case experiments mentioned in Sections 4 and 5. Select portions of figures have been blurred out to comply with the ethical guidelines to hide sensitive information.

B Data Filtering Details

Removal of pages with low information density: Initially, we decide to leave out pages that have an abnormally high or low character count. This is done to exclude content that is not seemingly useful in the representation of the Dark Web. For example, most of the pages containing an abnormally low character count are error messages such as “404 not found” or “Captcha error” and log-in messages such as “Sign In” or “Already have an account?”. On the other hand, the pages that contain an abnormally high character count are mostly large lists of keywords or continuous repetitions of certain strings. These texts are not very useful as they contain low information density of Dark Web content, and are therefore removed from the pretraining corpus.

To decide on the minimum and the maximum threshold of character counts to remove from the crawled data, we measure the per-page character count statistics as shown in Table 8, and use approximately half the character count value from the 25th quartile (500 characters) and double the character count value from the 75th quartile (10,000 characters). This is done so that the majority of the pages are still included in the pretraining corpus while also serving as a generous threshold for pages containing unwanted data as shown above. By filtering out pages below the minimum and above the maximum threshold for their character count, we are left with 5.43 million pages out of the initial 6.1 million.

Table 8: Dark Web data collection statistics

Statistics	Value
Total number of collected pages	6.1 M
Average number of characters per page	7,980
Minimum number of characters in a page	7
Maximum number of characters in a page	17,786,986
Per-page character count statistics	Character count
Q_1 (25th quartile)	1,318
Q_2 (50th quartile)	2,581
Q_3 (75th quartile)	5,753

Category balancing: Previous studies (Al Nabki et al., 2017; Jin et al., 2022) have found through their web crawling that pornographic content is one of the most common activities found in the Dark Web. One of the challenges in pretraining DarkBERT is to use text data that consists of various content found in the Dark Web while avoiding skewness in which certain activities constitute a significant fraction of the entire dataset. If these activities (that take up a large portion of the corpus) exist, then the learned representation of the language model would be more biased towards such activities through pretraining.

To address the issue of balancing content in the pretraining corpus, we attempt an automated categorization of every page. A general categorization of various activities and the guidelines for each activity were addressed by Jin et al. (2022), where each page in the Dark Web was sorted into a total of 10 categories. Following this classification methodology, we train a simple page classification model using BERT (Devlin et al., 2019). Although the use of vanilla BERT may seem contradicting due to the domain differences between the Surface Web (the domain of origin for the texts that BERT was pretrained with) and the Dark Web, it is not necessary for this classification model to achieve high performance since our goal is to obtain a general grasp of the pretraining corpus category distribution.

We implement the model by finetuning the `bert-base-uncased` model from the Hugging Face library (Wolf et al., 2020) with the CoDA Dark Web text corpus (Jin et al., 2022). This model is then run through the entire pretraining corpus to output a specific category for each page. We use 9 of the 10 predefined categories from CoDA and exclude the *Others* category, because most of the pages that fit in this category (log-in pages, error pages, etc.) have already been filtered out from the pretraining corpus through character count filtering. In addition, we found that pages are more likely to be misclassified as *Others* category compared to other categories, meaning that the exclusion of *Others* category would yield a more accurate category distribution.

The page category statistics resulting from classification is shown in Table 9. We observe from our data that *pornography* accounts for the highest fraction of all categories in the Dark Web, making up 41.7% of all pages. Meanwhile, categories such as *gambling* and *arms / weapons* make up less than 1% of all pages each. Even with the use

Table 9: Dark Web page classification and pretraining data statistics. The statistics marked as *(full)* represent the original data collection, and *(pretraining)* represents the data after deduplication and category balancing are applied.

Category	Page Count (full)	Total Size (full)	Average Size per Page (full)	Page Count (pretraining)	Total Size (pretraining)	Deduplication Rate	Total Reduction Rate
Pornography	2,267,628	9.70 GB	4.28 KB	224,781	971.0 MB	2.91%	89.98%
Drugs	503,433	1.75 GB	3.47 KB	228,965	766.7 MB	23.31%	56.19%
Financial	637,917	2.10 GB	3.29 KB	253,171	874.1 MB	12.45%	58.38%
Gambling	43,041	0.15 GB	3.38 KB	40,584	137.5 MB	5.37%	5.37%
Cryptocurrency	412,349	1.36 GB	3.29 KB	249,811	897.6 MB	10.28%	34.00%
Hacking	801,330	3.51 GB	4.38 KB	57,183	242.7 MB	75.73%	93.09%
Arms / Weapons	46,616	0.14 GB	2.70 KB	43,250	129.9 MB	6.15%	6.15%
Violence	323,738	1.21 GB	3.74 KB	253,566	959.8 MB	4.02%	20.68%
Electronics	401,196	0.89 GB	2.21 KB	381,218	850.4 MB	4.17%	4.45%
Total	5,437,248	20.79 GB	-	1,732,529	5.83 GB	18.69%	71.96%

of vanilla BERT and the exclusion of the *Others* category taken into consideration, it is evident that the variation of content in the pretraining corpus is unbalanced. To this end, we take a rather simple approach of random removal of pages from over-represented categories until all categories have similar amounts of content.

Deduplication: A significant portion of the Dark Web is duplicate content. Since pretraining language models requires considerable resource and time, reducing the pretraining corpus size through deduplication is beneficial. This process is handled by minhashing (Broder et al., 2000) each page in the corpus and removing duplicate pages until all remaining minhash values are unique.

The pretraining corpus statistics after applying random removal of over-represented pages and deduplication is shown in Table 9. The deduplication rate represents the reduction in data size as a result of deduplication only, while the total reduction rate represents the reduction in data size as a result of both deduplication and random removal for category balancing. Both are based on the ratio between the initial data size (Table 9) and the final data size of each category. We observe that most categories have deduplication rates of less than 10%. However, categories such as *drugs* and *hacking* exhibit high deduplication rates. In addition, the deduplication rate and the total reduction rate of *gambling* and *arms / weapons* categories are equal, since we did not perform random removal of pages as these categories were already initially small in terms of data size. Finally, the size difference between the smallest category (*arms / weapons*) and the largest category (*pornography*) is 7-fold in the final pretraining corpus, compared to the 70-fold difference in size observed from the initial data.

C Identifier Mask Details

Here, we give an extended discussion on each of the identifier masks used for text processing mentioned in Section 3.2. The types of identifier masks used for preprocessing the pretraining corpus is illustrated in Table 10.

Implementation: Some identifier types such as URLs and IP addresses always contain distinct patterns. These identifiers are searched and undergo substitution using regular expressions. Other identifier types such as emails and phone numbers are masked using the text preprocessing API provided by textacy [9].

Email Addresses: Email addresses are often seen in the Dark Web as a means of communication. Unlike the contacts commonly seen in the Surface Web, many of the email addresses listed in the Dark Web are those that provide end-to-end encryption services such as ProtonMail [10] to prioritize privacy. However, some email addresses can include strings that can be traced to a single individual, so all email addresses are masked.

URLs: There are two identifier types for URLs: onion domain addresses and non-onion domain addresses. While URLs do not necessarily expose personal information themselves, it is possible that links to some URLs may be contain harmful information or data. To eliminate the possibility of such URLs from being learned as a representation of the Dark Web, we mask all URLs.

IP Addresses: Although the Dark Web is used to hide IP addresses, some pages contain IP addresses in their texts. Many of the pages that contain IP addresses are Tor relay sites, which show information such as Tor exit relay node addresses (the IP addresses listed in the Tor relay sites can also easily be found on the Surface Web at `https://metrics.torproject.org/rs.html`). Given the frequent illegal activities occurring in the Dark Web,

[9] `https://textacy.readthedocs.io/en/latest/`

[10] `https://proton.me/mail`

Table 10: The types of identifier masks and the list of preprocessed texts.

Identifier Type	Example Text or Description	Preprocess Action Type	Identifier Mask Token
Email Addresses	example@email.com	Replace with token	`ID_EMAIL`
URLs (non-onion domain)	www.example.com https://www.example.com/home	Replace with token	`ID_NORMAL_URL`
URLs (onion domain)	facebookwkhpilnemxj7asaniu7vnjjbiltxjqhye3mhbshg7kx5tfyd.onion	Replace with token	`ID_ONION_URL`
IP Addresses (IPv4 & IPv6)	`192.168.1.1` `fe80::1ff:fe23:4567:890a%eth2`	Replace with token	`ID_IP_ADDRESS`

Cryptocurrency Addresses	BTC, ETH, LTC addresses	Replace with token	`ID_BTC_ADDRESS`
			`ID_ETH_ADDRESS`
			`ID_LTC_ADDRESS`

Lengthy "Words"	Any group of non-whitespace characters that are 38 or more letters long	Replace with token	`ID_LONGWORD`
Uncommon Characters	Any characters out of Unicode range from `U+0000` to `U+00FF`	Remove from text	-
Whitespaces	Newline characters, tabs, spaces, etc.	Truncate to a single space	-

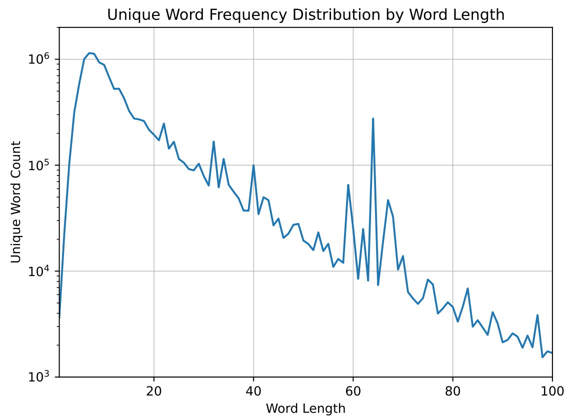


Figure 5: Unique word length distribution for the pretraining corpus before preprocessing is applied. Word lengths greater than 100 are omitted for brevity.

it is possible that some IP addresses listed in these pages may exist for malicious purposes. For example, Winter et al. (2014) has shown that some malicious exit relays have been engaging in HTTPS man-in-the-middle attacks. Therefore, we found it necessary to mask all IP addresses (both IPv4 and IPv6 addresses are masked).

Lengthy Words: While exploring some of the unpreprocessed text in the pretraining corpus, we found that certain pages contain words (string of characters separated by whitespace) that are extremely long in length. On closer inspection, most of these lengthy words are URLs, code snippets, hash values, file names, cryptocurrency addresses, and even binaries. While URLs and cryptocurrency addresses can be removed through the preprocessing mask identifiers, other types such as hashes and file names are not separately processed in advance. Hashes in particular would incur overhead in building meaningful vocabulary through tokenization as they do not have specific lexical patterns. In addition, since we do not want executable content such as binaries or detailed file names to be learned by our language model, we decide to mask all lengthy words. To this end, we define *lengthy words* by studying the word length distribution, as well as manual inspection of example words for some notable word lengths.

The unique word length distribution for the pretraining corpus is shown in Figure 5. The word length distribution shows a steep upward trend at shorter word lengths (peaking at length of 7) similar to the English language word distribution, and gradually decreases with longer word lengths. As observed in the figure, some specific word lengths appear in much greater frequencies at higher levels. Upon inspection, we find that this is due to some of the commonly used string formats that happen to have specific lengths. For example, many words of length 59 found in our corpus are content identifier (CIDv1) hashes commonly used in IPFS [11], which is a decentralized, hypermedia distribution protocol. Similarly, words of length 64 are mostly SHA-256 hashes.

Our manual inspection of some of the vocabularies present for each word length shows that at around length of 38 to 40, the majority of words take the form of hash-like values and meaningless noisy strings. Therefore, we classify words with lengths of 38 or more characters as *lengthy words*, and mask them from the pretraining corpus. Note that the masking process of *lengthy words* is performed after masking all other identifiers mentioned previously such as email addresses, URLs, and cryptocurrency addresses. Since texts belonging to such identifiers are lengthy (ex. onion V3 addresses are 56 characters long, and Ethereum addresses consist of 40 digit hexadecimal strings), masking these texts with their associated mask identifiers beforehand prevents them from being misclassified as *lengthy words*.

[11] https://ipfs.io/

Table 11: The hyperparameters used for pretraining the two versions of DarkBERT.

Hyperparameter	Value
Number of Layers	12
Hidden Size	768
Feedforward NN Inner Hidden Size	3072
Attention Heads	12
Attention Head Size	64
Dropout	0.1
Attention Dropout	0.1
Max Sequence Length	512
Warmup Steps	24000
Peak Learning Rate	6e-4
Batch Size	8192
Weight Decay	0.01
Max Steps	20K
Learning Rate Decay	Linear
Adam ϵ	1e-6
Adam β_1	0.9
Adam β_2	0.98
Gradient Clipping	0.0

Uncommon Characters: As mentioned in Section 3.1, we collect pages that are classified as "English". However, some of these collected pages contain multilingual characters that are not standard English. The inclusion of such nonstandard characters results in noisy tokens during the tokenization process and produces unnecessary token vocabularies, so we remove all the characters that are "uncommon" in contemporary English. Specifically, we remove all Unicode characters that are not one of the 256 characters in the `Basic Latin` (ASCII characters) and the `Latin-1 Supplement` (accented alphabets that are often seen in English) category.

Cryptocurrency Addresses: Decentralized digital assets like cryptocurrencies are used to make unidentifiable transactions. As many cryptocurrencies are secure by design and provide pseudonymity, the synergy with the anonymous nature of the Dark Web makes them the preferred method of choice for transactions. Studies show that cryptocurrencies have been involved in illegal underground operations (Lee et al., 2019) in the Dark Web and underground marketplaces in general (Soska and Christin, 2015). While cryptocurrencies are known for their pseudonymous properties, many of the transactions are traceable as the entire blockchain is public (for some cryptocurrencies). In particular, we mask Bitcoin, Ethereum, and Litecoin addresses as these three cryptocurrencies are among the most popular in the Dark Web with transparent transaction details (Monero and Dash are also popular in the Dark Web, but they incorporate added layers of anonymity to further conceal their transactions). (Barysevich and Solad, 2018).

D DarkBERT Pretraining Details

Both versions of DarkBERT are pretrained on a machine with Intel Xeon Gold 6348 CPU @ 2.60GHz and 4 NVIDIA A100 80GB GPUs. All 4 GPUs were used to run the pretraining process, and each version of DarkBERT took about 15 days to run (up to 20K training steps — we stopped the pretraining process at training loss convergence). Both versions of DarkBERT share relatively similar training losses over the 20K training steps. Since training loss for both versions of DarkBERT stopped decreasing at around 20K steps, we use the models saved at 20K steps for evaluation.

E Evaluation Details

E.1 Dark Web Activity Classification

We implement a classification pipeline using the language models available in the Hugging Face library (`bert-base-cased`, `bert-base-uncased`, and `roberta-base`) and add a fully-connected classification layer on top of the `[CLS]` token with PyTorch. Evaluation is performed for each model using k-fold cross validation ($k = 10$), which is implemented using scikit-learn's `StratifiedKFold` module (Pedregosa et al., 2011). Each fold is run up to 10 epochs with a learning rate of `2e-5`.

Error Analysis: We further scrutinize model performance by looking at specific pages in the CoDA dataset that are correctly classified by DarkBERT but are misclassified by the other models. We find that most pages that have been misclassified by BERT and RoBERTa but correctly classified by DarkBERT contain many domain-specific jargons or key phrases seen in that particular activity in the Dark Web. For example, one of the pages under the *Financial* category that is misclassified by both BERT and RoBERTa as *Others* contains the name of a credit card seller service (we choose not to reveal the service name for ethical considerations). Another page under the *Pornography* category contains the phrase *red room* which is highly correlated to this category of pages in the Dark Web, but is misclassified by both BERT and RoBERTa as *Others*. Finally, a page under the *Crypto* category contains blockchain and cryptocurrency terms, but is misclassified by BERT and RoBERTa as *Others*. As shown in the above examples, DarkBERT is able to correctly classify pages that contain phrases mostly seen in the Dark Web but are not commonly used in the Surface Web, whereas BERT and RoBERTa tend to misclassify such pages in the *Others* category as these models consider such words and phrases as generic attributes rather than activity-specific terms.

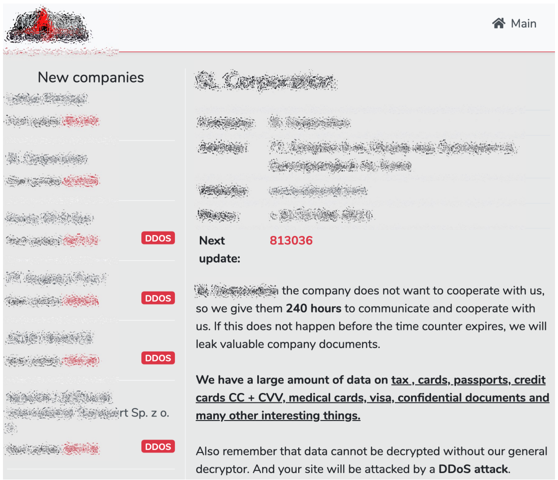


Figure 6: A leak site page sample in the dataset.

E.2 Ransomware Leak Site Detection

We use the same classification pipeline as activity classification in Section 4 with k-fold cross-validation ($k = 5$) and connect fully-connected classification layers on top of the `[CLS]` token. Similarly, the evaluation is performed on both raw and preprocessed inputs. An early stopping strategy using validation loss is utilized to avoid overfitting. Due to the limited size of the dataset, we choose to repeat k-fold validation 5 times to mitigate the variations in performance per run and average the results. An example data sample used for this task can be seen in Figure 6 and additional details on used hyperparameters can be found in Table 12.

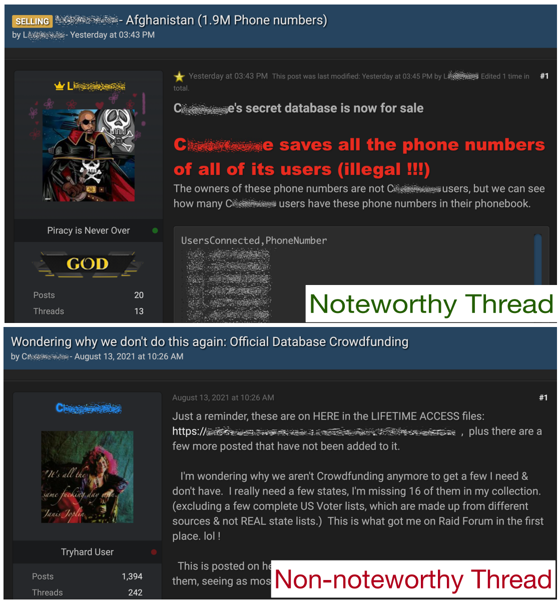


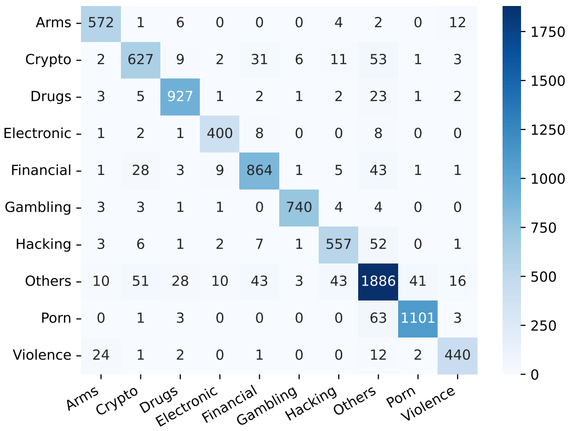
Figure 7: Noteworthy and non-noteworthy thread samples in the dataset.

E.3 Noteworthy Thread Detection

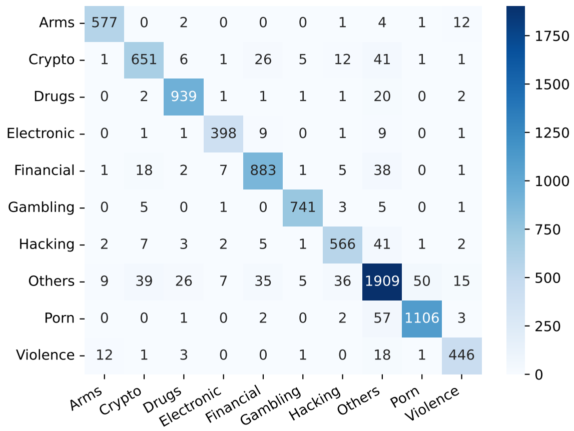
Similar to ransomware leak site detection, we adopt k-fold cross validation ($k = 5$) for each model and employ early stopping strategy. Due to the limited size of the dataset, we again use repeated k-fold validation, where the number of repetitions is set to 5. An example data sample used for this task can be seen in Figure 7 and additional details on used parameters can be found in Table 12.

Table 12: The hyperparameters used in ransomware leak site detection and noteworthy thread detection.

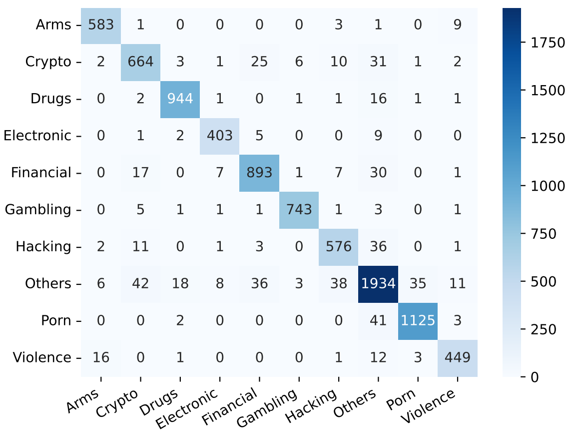
Hyperparameter	Ransomware leak site detection	Noteworthy thread detection
Epochs	100	100
Batch Size	32	32
Learning Rate	1e-4	1e-5
Number of Layers	2	2
Hidden Size	64	64
Dropout	None	0.5



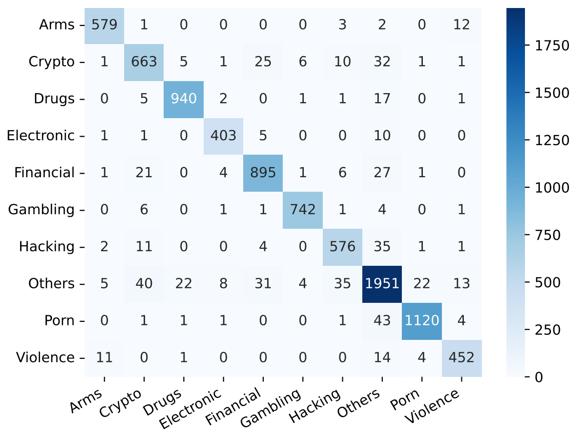
(a) BERT$_{cased}$



(b) RoBERTa



(c) DarkBERT$_{raw}$



(d) DarkBERT$_{preproc}$

Figure 8: Confusion matrices for selected language models evaluated on the CoDA$_{cased}$ dataset

ACL 2023 Responsible NLP Checklist

A For every submission:

☐ A1. Did you describe the limitations of your work?
Left blank.

☐ A2. Did you discuss any potential risks of your work?
Left blank.

☐ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☐ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☐ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☐ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D ☐ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.