Tools for Automated Analysis of Cybercriminal Markets

Rebecca S. Portnoff
UC Berkeley
rsportnoff@cs.berkeley.edu

Sadia Afroz
ICSI,UC Berkeley
sadia@icsi.berkeley.edu

Greg Durrett
UC Berkeley
gdurrett@cs.berkeley.edu

Jonathan K. Kummerfeld
UC Berkeley
jkk@cs.berkeley.edu

Taylor Berg-Kirkpatrick
UC Berkeley
tberg@eecs.berkeley.edu

Damon McCoy
NYU
mccoy@nyu.edu

Kirill Levchenko
UC San Diego
klevchen@cs.ucsd.edu

Vern Paxson
UC Berkeley
vern@berkeley.edu

ABSTRACT

Underground forums are widely used by criminals to buy and sell a host of stolen items, datasets, resources, and criminal services. These forums contain important resources for understanding cybercrime. However, the number of forums, their size, and the domain expertise required to understand the markets makes manual exploration of these forums unscalable. In this work, we propose an automated, top-down approach for analyzing underground forums. Our approach uses natural language processing and machine learning to automatically generate high-level information about underground forums, first identifying posts related to transactions, and then extracting products and prices. We also demonstrate, via a pair of case studies, how an analyst can use these automated approaches to investigate other categories of products and transactions. We use eight distinct forums to assess our tools: Antichat, Blackhat World, Carders, Darkode, Hack Forums, Hell, L33tCrew and Nulled. Our automated approach is fast and accurate, achieving over 80% accuracy in detecting post category, product, and prices.

Keywords

Cybercrime; Machine Learning/NLP; Measurement

1 Introduction

As technology evolves, abuse and cybercrime evolve with it. Much of this evolution takes place on underground forums that serve as both marketplaces for illicit goods and as forums for the exchange of ideas. Underground forums play a crucial role in increasing efficiency and promoting innovation in the cybercrime ecosystem. Cybercriminals rely on forums to establish trade relationships and to facilitate the exchange of illicit goods and services, such as the sale of stolen credit card numbers, compromised hosts, and online credentials.

Because of their central role in the cybercriminal ecosystem, analysis of these forums can provide valuable insight into cybercrime. Indeed, security practitioners routinely monitor forums to stay current of the latest developments in the underground [Krebs 2013a, Krebs 2013b]. Journalist Brian Krebs, for example, relied on forum data when he alerted Target to an ongoing massive data breach based on an influx of stolen credit card numbers being advertised for sale on an online forum [Krebs 2013a]. Information gleaned from forums has also been used by researchers to study many elements of cybercrime [Franklin et al. 2007, Garg et al. 2015, Holt and Lampke 2010, Motoyama et al. 2011, Stone-Gross et al. 2011, Yip et al. 2012].

Unfortunately, monitoring these forums is a labor-intensive task. To unlock this trove of information, human analysts must spend considerable time each day to stay current of all threads and topics under discussion. Understanding forums also requires considerable domain expertise as well as knowledge of forum-specific jargon. Moreover, a forum may be in a foreign language, creating an additional barrier for the analyst. Often, what one wants from a forum is not a deep understanding of a particular topic, but an aggregate summary of forum activity. For example, one may want to monitor forums for an uptick in offers to sell stolen credit cards, a strong indicator of a major data breach. In this case, the goal is to extract certain structured information from a forum. Continuing the example, the task is first to identify offers to sell credit card numbers and then extract from the post information like quantity and price. We can then use this structured data to carry out analyses of market trends, like detecting a sudden increase in supply.

In this work, we aim to develop and demonstrate *automatic* techniques for extracting such structured data from forums. Although extracting structured data from unstructured text is a well-studied problem, the nature of forum text precludes using existing techniques that were developed for the well-written English text of the Wall Street Journal. In contrast, forum posts are written in their own specialized and rapidly evolving vocabulary that varies from forum to forum and ranges from ungrammatical to utterly incomprehensible. As a result, off-the-shelf Named-Entity Recognition (NER) models from Stanford NER perform poorly in this dataset. Another approach is to use regular expressions to identify occurrences of the words related to the type of a post, well-known products, and prices. This simplistic approach also fails because different users use different words for the same products.

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052600

Rather than aiming for complete automatic comprehension of a forum, we developed a set of natural language processing building blocks aimed at a set of precise tasks related to trade that a human analyst might require when working with forum data. As we show in this paper, this approach allows us to extract key elements of a post with high confidence, relying on a minimal set of human-labeled examples. By focusing on extracting specific facts from a post, our tools make automatic analysis possible for text inaccessible using conventional natural language processing tools. In this work, we develop automatic tools for the following tasks:

- **Post Type.** Determine the nature of the post, specifically, whether it is an offer to *buy*, offer to *sell*, offer to *exchange currency*, or a post not related to trade.
- **Product.** Determine the product being offered or requested (buy/sell posts) or the currencies being exchanged (currency exchange posts).
- **Price.** Extract the price offered in commerce (buy/sell) posts or the exchange rate (currency exchange posts).

We applied subsets of these tools to eight underground forums (5 English, 1 Russian, and 2 German) (Section 3). We demonstrate how our product extractor can be used to quickly build a picture of the products being traded on a forum, even when we train the product extractor using data from a different forum (Section 5). Using two case studies, we show how to use these building blocks to carry out specific forum analysis tasks (Section 6). Our first case study shows how to use the product extractor as the first stage to a more discriminating classifier that distinguishes between subclasses of products, namely between different types of online service accounts sold on forums. The second case study shows how to use our tools to extract currency exchange trends from forum posts.

In summary, our contributions are as follows:

- ❖ We develop new natural language processing tools for a set of precise data extraction tasks from forum text of poor grammatical quality. In comparison with a simple regular expression–based approach, our approach achieves over 9 F-point improvement for product detection (Table 10) and over 40 F-point improvement for price extraction (Table 7). Our free and open-sourced tools are available: http://evidencebasedsecurity.org/forums/.
- ❖ We evaluate our tools on a set of eight underground forums spanning three languages.
- ❖ We present two case studies showing how to use our tools to carry out specific forum analysis tasks automatically and accurately, despite the poor quality of the data.

2 Related Work

Underground forum analysis. Our study is the first to create automated extraction techniques for conducting large-scale analyses of the products and pricing of goods offered on underground forums. Previous work used structured information (e.g., social graph, timestamps, usernames) [Motoyama et al. 2011, Soska and Christin 2015, Garg et al. 2015, Yip et al. 2012], handcrafted regular expressions [Franklin et al. 2007] and manual annotations of a small set of posts to understand products and pricing [Holt and Lampke 2010]. Our tools can analyze unstructured texts in large scale with little manual effort.

Forum analysis with NLP tools. NLP techniques have proven useful for answering a range of scientific questions in various disciplines including the humanities [Bamman et al. 2013] and the social sciences [O'Connor et al. 2013]. However, there has been relatively little work in specifically applying NLP techniques to web forums [Kim et al. 2010, Kaljahi et al. 2015]. Because of the high degree of domain dependence of NLP techniques [Daume III 2007], most out-of-the-box tools (like part-of-speech taggers or parsers) have various deficiencies in this setting, and in any case do not directly provide the information about forum posts in which we have the most interest.

NLP methodology. The problems we consider in this work differ from those in past NLP efforts on forum analysis [Kim et al. 2010, Lui and Baldwin 2010, Wang et al. 2011]. Our tasks broadly fall into the category of slot-filling information extraction tasks [Freitag and McCallum 2000, Surdeanu 2013], where the goal is to populate a set of pre-specified fields based on the information in the text. However, much of the recent work on information extraction in the NLP literature has aimed to extract a very broad set of relations for open-domain text [Fader et al. 2011], as opposed to focusing on domain-specific or ontology-specific methods [Parikh et al. 2015]. The various kinds of information we consider (transaction type, products, prices) each necessitate different techniques: some tasks are formulated as classification problems with various structures, and our product extraction task is similar to named entity recognition [Tjong Kim Sang and De Meulder 2003] or entity detection [NIST 2005]. We use a variety of supervised machine learning methods in this work, drawing on well-established conventional wisdom about what features prove most effective for each of our tasks.

3 Forum Datasets

We consider eight underground forums (Table 1): Blackhat World, Darkode, Hack Forums, Hell, Nulled, Antichat, Carders and L33tCrew. We collected the forum data in two ways: partial scraping (Darkode, Hack Forums, Hell) and complete publicly leaked database dumps that contain all public posts and metadata prior to the leak (Blackhat World, Nulled, Antichat, Carders and L33tCrew).

Blackhat World. Blackhat World focuses on blackhat search engine optimization (SEO) techniques. The forum started in October, 2005 and is still active, although it has changed in character over the past decade.

Darkode. Darkode focused on cybercriminal wares, including exploit kits, spam services, ransomware programs, and stealthy botnets. We focused our attention on the four subforums that contained substantial amounts of commerce, ignoring twenty-eight other subforums unrelated to commerce. This forum was taken down in July of 2015 by a joint multinational law enforcement effort [of Justice 2015].

Hack Forums. Hack Forums covers a wide range of mostly cybersecurity-related blackhat (and non-cybercrime topics), such as crypters (software used to hide viruses), keyloggers, server "stress-testing" (denial-of-service flooding) and hacking tools. The forum started in 2007 and is still active. For our analysis, we focus on the subforums in Hack Forums related to buy, sell, and currency exchange.

Hell. Hell was an underground forum hosted as a Tor Hidden Service. It focused on credit card fraud, hacking, and data breaches. Hell made headlines when a hacker on the forum dumped the personal details of 4 million users of Adult Friend Finder, a dating website. The forum was shut down in July 2015 but relaunched in January 2016.

Nulled. Nulled advertises itself as a "cracking community" specializing in leaks and tools for data breach. The forum was hacked on May 2016 and the full database of the forum was released publicly.

Forum	Source	Primary Language	Date covered	Threads (Commerce)	Users
Blackhat World	Complete Dump	English	Oct 2005–Mar 2008	7270 (2.29%)	8,718
Darkode	Partial Scrape	English	Mar 2008–Mar 2013	7418 (27.94%)	231
Hack Forums	Partial Scrape	English	May 2008–Apr 2015	52,649 (97.34%)	12,011
Hell	Partial Scrape	English	Feb 2015–Jul 2015	1,120 (22.59%)	475
Nulled	Complete Dump	English	Nov 2012–May 2016	121,499 (32.81%)	599,085
Antichat	Complete Dump	Russian	May 2002–Jun 2010	201,390 (25.82%)	41,036
Carders	Complete Dump	German	Feb 2009–Dec 2010	52,188 (38.72%)	8,425
L33tCrew	Complete Dump	German	May 2007–Nov 2009	120,560 (30.83%)	18,834

Table 1: General properties of the forums considered.

Non-English Forums. We analyzed three non-English forums: Antichat, Carders and L33tCrew. Carders and L33tCrew were German-language forums that specialized in stolen credit cards and other financial accounts [Afroz et al. 2013]. Both of the forums were leaked and closed. Our data spans the entire lifetime of the forums. Antichat is a Russian-language forum. Unlike the other forums, Antichat does not specialize on a single topic but rather covers a broad array of underground cybercrime topics such as password cracking, stolen online credentials, email spam and SEO [Afroz et al. 2013].

4 Automated Processing

For each post appearing in a forum, we extract three properties — the type of transaction, the product, and its price—not explicitly marked. We take a supervised learning approach, labeling a small proportion of the data with ground truth and using those annotations to train a tool to label the rest. We divide the task of extracting all of this information into three sub-tasks. In every case, the input to the tool is a single post, while output structure varies by task. In this section we describe the development of our tools, and results of evaluations that assess their effectiveness.

4.1 Type-of-Post Classification

Different forums use different conventions to mark different types of posts. For example, Darkode and Hack Forums have dedicated subforums for buy, sell and trade posts; on Carders and L33tCrew, buy posts start with "[S]" ("suche" means "seeking", i.e., buying) and sell posts start with "[B]" ("biete" means offering). The rest of the forums do not have any explicit tagging to mark the type of a post. Identifying the commerce section of a forum will significantly reduce the workload of an analyst, because fewer than 40% of the posts are related to commerce on the majority of the forums.[1]

The type-of-post classifier detects whether a post concerns buying or selling a product, exchanging currency, or none of these (e.g., an admin post). Due to the lack of ground-truth data on the non-English forums, the classifier only detects buy and sell posts on those forums. We use a variety of token- and character-level features robust across languages and domains.

4.1.1 Labeling Ground Truth.

To build a ground-truth dataset for the type-of-post classifier, we strip out the information that explicitly indicates the posting type. For the non-English forums (Antichat, Carders and L33tCrew), we consulted one German and one Russian native speaker to confirm the accuracy of the labels. For Antichat, we look for the words related to trade, buy or sell. For example, "prodayu" is the first person singular present tense of *to sell*, meaning "I am selling," and is often used in posts to offer a product for sale. By identifying threads with these words we constructed a training set, with one of three confidence levels assigned to each thread based on the words present—a confidence level of 3 indicates 100% confidence in the labeling, a level of 2 indicates less than 75% confidence, and a level of 1 indicates less than 50% confidence. Table 2 shows the final dataset size for each forum.

4.1.2 Models.

We consider two models for type-of-post classification.

Most-Frequent Label (MFL). This model returns the most frequent label in the training set. This approach can appear to do much better than 50% in some cases because of natural imbalances in the number of "buy", "sell" , "currency exchange" and "other" posts in a forum (see Table 2).

Support Vector Machine (SVM). This model uses text-based features: word unigrams, word bigrams, and character n-grams of lengths 1 to 6. We train the SVM by coordinate descent on the primal form of the objective [Fan et al. 2008] with ℓ_2-regularization. We also considered a range of other features: part-of-speech labels, parse dependencies, text from replies to the initial post, the length of the post, and rank and reputation of the user who authored the initial post. None of these additions appreciably improved performance, and so we do not include them in the final classifier.

4.1.3 Validation Results.

We assessed our classifier both within a forum and across forums. The first gives a direct measure of performance on the task we trained the classifier to do. The second gives a measurement of how well the classifier would generalize to an unseen forum in the same language. For Antichat, in addition to doing the standard evaluation, we also considered performance when using only the threads with a high confidence annotation level (level 3). In within-forum evaluations, we split the data 80% / 20% to form training and test sets. In cross-forum evaluations, we used 100% of the data.

English. For Darkode, the buy vs. sell classifier is effective, achieving 98.2% accuracy overall, and 90.0% on the less common class. Our classifier is similarly effective on Hack Forums sell vs. currency (98.29% overall / 96.95% on the least common class) and Nulled buy vs. sell vs. other (95.27% overall / 85.34% on the least common class). When combining randomly sampled data from Darkode, Hack Forums and Nulled to get a dataset balanced between all four classes, we see uniformly high performance on all classes and 95.69% accuracy overall.

While Blackhat World and Hell are too small for within-forum evaluation, we can use the entire dataset as a test set to perform cross-forum evaluation. When training on Darkode and testing on Blackhat World, we see a performance drop relative to the within-forum evaluation on Darkode, but we achieve accuracy still well above the MFL baseline. The same holds when training on Nulled and testing on Blackhat World, Hell or Darkode. These results all indicate that the classifier generalizes well and analysts could use it in the future on other, completely unlabeled forums.

Non-English. On both German-language forums (Carders and L33tcrew) we see high performance both within-forum and across-

[1] In our dataset, one exception is Hack Forums where over 97% of the posts are commerce related because we only scraped the commerce section of the forum.

Forum	# Buy	# Sell	# Curr	# Other
Blackhat World	22	115	–	1
Darkode	1,150	205	14	1
Hack Forums	165	14,393	33,067	–
Hell	44	42	–	14
Nulled	2,746	8,644	49	1,025
Carders	8,137	5,476	–	–
L33tcrew	8,486	4,717	–	–
Antichat (all)	13,529	25,368	–	–
Antichat (confidence=3)	10,129	18,965	–	–

Table 2: Number of labeled posts by class for type-of-post classification.

	Accuracy (%)			
Train/Test Forum	**Buy**	**Sell**	**Overall**	**MFL**
Darkode	99.6	90.0	98.2	85.4
Darkode/BHW	85.7	90.6	90.5	84.8
Carders	95.65	89.52	93.20	60.0
L33tcrew	97.36	92.32	95.57	57.0
Carders/L33tcrew	95.75	88.21	93.05	64.27
L33tcrew/Carders	93.81	83.82	89.79	59.77
Antichat (all)	95.15	98.23	97.16	65.3
Antichat (confidence=3)	98.91	99.97	99.60	65.2

Table 3: Classification accuracy on the buy vs. sell task. MFL refers to a baseline that simply returns the most frequent label in the training set. Note that the test sets for within-forum evaluation comprise 20% of the labeled data, while the test sets for across-forum evaluation are 100% of the labeled data in the target forum.

forum. Performance when evaluating on Carders runs consistently lower, probably because of the more even distribution of buy and sell threads. For Antichat, we also see high performance within-forum, but are unable to evaluate across-forum performance because we do not have another Russian forum. Focusing on the high-confidence threads, we see even higher performance, as would be expected.

These results indicate the robustness of our feature-set to variation in language as well as forum, though to generalize to further languages would require additional labeled data.

4.1.4 Limitations.

We investigated why the additional features we considered did not improve the accuracy any further. We found two general issues. First, most core NLP systems like part-of-speech taggers and syntactic parsers target formal, well-formed, grammatical text. The data we consider strays far from that setting, and performance of those tools suffers accordingly, making them less informative. Second, as a thread continues, the topic will often drift and lose focus, becoming less related to the original transaction. This noise explains why using features of later posts in a thread did not improve performance.

	Accuracy (%)					
Train/Test Forum	**Buy**	**Sell**	**Curr**	**Other**	**Overall**	**MFL**
Hack Forums	—	96.95	98.89	—	98.29	69.67
Nulled	89.42	98.28	—	85.34	95.27	59.53
Darkode + Hack Forums + Nulled	92.86	95.72	98.35	96.26	95.69	27.42
Nulled/BHW	77.27	93.04	—	—	90.51	84.8
Nulled/Darkode	86.50	96.14	—	—	87.96	85.4
Nulled/Hell	86.36	85.71	—	—	86.1	51.2

Table 4: Classification accuracy on the buy vs. sell vs. currency exchange vs. other task. Omitted entries indicate categories with too little ground-truth data of that type to robustly evaluate.

0-initiator4856

TITLE: [buy] Backconnect bot
BODY: Looking for a solid backconnect bot .
If you know of anyone who codes them please let me know

0-initiator6830

TITLE: Coder
BODY: Need sombody too mod DCIBot for me add the following :
Update Cmd
Autorun Obfuscator (Each autorun diffrent and fud)
Startup Mod (needs too work on W7/VISTA)
Pm .

Figure 1: Example post and annotations from Darkode, with one sentence per line. We underline annotated product tokens. The second exhibits our annotations of both the core product (*mod DCIBot*) and the method for obtaining that product (*sombody*).

4.2 Product Extraction

Here we look at extracting the actual product being bought or sold in a thread. Our system outputs a set of spans in the text, each of which marks an explicit mention of the product. From this, we can extract a set of string representations of the product(s) being bought or sold. This task proves both very useful for analyzing criminal marketplace activity but also quite difficult. One general challenge is that a single post can mention many product-like items distinct from the item actually for sale, such as an account to contact for purchasing. We address this kind of ambiguity by building a machine-learned product extractor, which uses features that consider syntax and surface word context.

4.2.1 Labeling Ground Truth.

To start, while the task output is multi-word spans that describe products, we find manually annotating such spans a difficult and time-consuming process. To understand the challenge, consider this example from Darkode:

a keylogger coded completely in ASM

The correct span in this case could be keylogger, a keylogger, or the complete phrase. Linguistically, the first of these is a noun, the second is a base noun phrase (NP), and the third is a complex NP. We thus avoid defining rules on where to place the boundaries, and instead annotate the word common to all of the options—the head of the noun phrase, in this example, keylogger. Doing so provides a clearer definition of the annotations, enabling consistent labeling. Using automatic syntactic parsers we can subsequently recover the larger span (described further in Section 4.2.3), though we define our annotations over raw tokens to avoid tying ourselves to error-prone parser output.

Note that, when appropriate, we annotate both the outcome and the means of delivery. Figure 1 shows an example: *DCIBot* is closest to the core product, but *sombody* and *mod* are critical to the process and so we annotate them as well. However, we do not annotate features of products (*Update Cmd* in Figure 1), generic product references (*this*), product mentions inside "vouches" (reviews from other users), or product mentions outside of the first and last 10 non-whitespace lines of each post.[2] We make our full annotation guide available.[3]

[2] This reduces the annotation burden on the small number of posts (roughly 4% on Darkode) that are unusually long—these posts are also often outliers with few product references.

[3] cs.berkeley.edu/~jkk/www2017-product-annotation-guide.pdf

We developed this approach through a series of rounds, first to investigate options for annotation methodologies, then to train annotators without security expertise. We annotated training, development, and test sets in several forums:

- Darkode training set (630 posts with 3 annotators per post, 30 with 7 annotators per post)
- Darkode development set and test set (both 100 posts with 8 annotators per post)
- Hack Forums training set (728 posts, 3 annotators per post, 30 posts with 8 annotators per post)
- Hack Forums test set (140 posts, 4 annotators per post)

We used the Fleiss Kappa measurement of inter-annotator agreement [Fleiss 1971] and found that our annotations had "substantial agreement" ($\kappa = 0.65$).

We derived the final annotations used by taking a majority vote among annotators; i.e., for each token in question, if at least half of annotators (rounded up) annotate it, then we treat it as ground truth. Roughly 95% of posts in Darkode and Hack Forums contained products according to this annotation scheme. We additionally pre-processed the data using the tokenizer and the sentence-splitter from the Stanford CoreNLP toolkit [Manning et al. 2014].

4.2.2 Models.

We consider two models for product extraction. In each case, our models deal with noun phrases as the fundamental units of products. We generalize ground-truth noun phrases from our headword annotation according to the output of an automatic parser [Chen and Manning 2014].

Noun-phrase classifier. We train an SVM to classify each noun phrase in the post as either a product or not. We structure our features around a set of key words that we consider, namely the first, last, and head words of the noun phrase, as well as the syntactic parent of the noun phrase's head, and up to three words of context on each side of these words. For each of these words, we fire features on its identity, character n-grams it contains, part of speech, and dependency relation to its parent. This gives us a rich set of contextual features examining both the surface and syntactic context of the noun phrase in question. For example, in Figure 1, when considering the noun phrase *a solid backconnect bot*, we fire features like *parent=for* and *parent-previous=looking*, the latter of which provides a strong indicator that our noun phrase corresponds to what the poster seeks. Finally we also use features targeting the noun phrase's position in the post (based on line and word indices), capturing the intuition that posters often mention products in a post's title or early in the post body.

We train the SVM by subgradient descent on the primal form of the objective [Ratliff et al. 2007, Kummerfeld et al. 2015]. We use AdaGrad [Duchi et al. 2011] to speed convergence in the presence of a large weight vector with heterogeneous feature types. We trained all product extractors in this section for 5 iterations with ℓ_1-regularization.

Post-level extractor. Rather than making decisions about every noun phrase in a post, we can support some kinds of analysis by a more conservative product identification scheme. If all we want to do is identify the general product composition of a forum, then we do not need to identify all references to that product in the body of the post, but might instead just identify an easy one in, say, the post title. Therefore, we also consider a post-level model, which tries to select one noun phrase out of a post as the most likely product being bought or sold. Structuring the prediction problem in this way naturally lets the model be more conservative in its extractions, and simplifies the task, since we can ignore highly ambiguous cases if the post includes a clear product mention. Put another way, doing so supplies a useful form of prior knowledge, namely that the post contains a single product as its focus.

We formulate this version of the model as a latent SVM, where the choice of which product noun phrase to extract is a latent variable at training time. We use the same datasets, features, and training setup as before.

4.2.3 Validation Results.

We considered three different metrics to validate the effectiveness of our product extractor:

- Performance on recovering individual product *noun phrases*. We compute precision (number of true positives divided by the number of system-predicted positives), recall (true positives over ground truth positives), and F-measure (F_1, the harmonic mean of precision and recall).
- Performance on recovering *product types* from a post: we compare the set of product head words extracted by our automated system with those annotated in the ground truth (after lowercasing and stemming), and evaluate with precision, recall, and F_1.[4]
- Evaluation of a single product chosen from the *post*, checking whether we accurately recovered a product from the post, or correctly decided that it contained no products.

The latter two of these better match the analysis we want to carry out in this work, so we will focus our discussion on them.

Table 5 shows these metrics for our noun phrase- and post-level classifiers. Throughout this table, we train on a combined set of annotated training data from both Darkode and Hack Forums. We compare against two baselines. Our *Frequency* baseline takes the most frequent noun or verb in a post and classifies it as a product. This method favors precision: it will tend to annotate one token per post. Our *Dictionary* baseline extracts a gazetteer of products from the training data and tags any word that appears in this gazetteer. This method favors recall: it will severely over-extract words like *account* and *website*, and will only fail to recover products when they have never been seen in the training set.

Our learned systems outperform the baselines substantially on each of the forums we consider. Overall, we find consistently lower results on Hack Forums across all metrics. One possible reason is that Hack Forums posts tend to be longer and more complex (10.4 lines per post on average as opposed to 6.0 for Darkode), as well as exhibiting a wider variety of products: when we ran the extractor on 1,000 posts from each forum, the Darkode sample contained 313 distinct products and the Hack Forums sample contained 393.

Our post-level system performs well on both post-level evaluation as well as on product type evaluation, indicating that posts generally have only one product type. We use this system for the analysis going forward. Overall, this system achieves 86.6% accuracy on posts from these two forums: high enough to enable interesting analysis.

4.2.4 Limitations.

Performance on other forums. Because we train our product extractor on data drawn from particular forums, we would expect it to perform better at prediction on those forums than on others. We

[4] Note that this is still an unnecessarily harsh metric in some cases: *mail list* and *emails* will not be considered the same product type, but getting both right may not be necessary for analysis.

System	Darkode							Hack Forums							Combined
	Noun phrases			*Product types*			*Posts*	*Noun phrases*			*Product types*			*Posts*	*Posts*
	Prec	Rec	F_1	Prec	Rec	F_1		Prec	Rec	F_1	Prec	Rec	F_1		
Frequency	61.8	27.9	38.4	61.8	50.0	55.2	61.8	41.9	16.1	23.3	41.9	35.9	38.7	41.9	50.3
Dictionary	57.0	60.0	58.5	67.6	55.8	61.1	65.9	38.3	47.8	42.5	50.3	43.1	46.4	45.4	54.1
NP-level	75.0	**79.4**	**77.2**	74.4	**86.7**	80.1	90.6	62.1	**60.6**	**61.4**	53.9	**74.3**	62.5	73.5	80.7
Post-level	**93.8**	37.0	53.1	**93.8**	70.3	80.4	93.8	**81.6**	23.7	36.8	**81.6**	54.7	65.5	**81.6**	**86.6**

Table 5: Results of the product extractor (trained on all training data) on the test sets of two forums. We report results for two baselines as well as for two variants of our system. Bolded results represent statistically significant improvements over all other values on that metric (in the same column) according to a bootstrap resampling test with $p < 0.05$. Our post-level system achieves 86.6% accuracy on product identification overall, making it robust enough to support many kinds of analysis.

	Product Type		
Train/Test Forums	Prec	Rec	F_1
Darkode-Darkode	92.7	69.5	79.5
Darkode-Hack Forums	69.9	46.8	56.0
Hack Forums-Hack Forums	81.6	54.7	65.5
Hack Forums-Darkode	89.6	67.2	76.8
Both-Blackhat	82.2	64.5	72.3
Both-Darkode	93.8	70.3	80.4
Both-Hack Forums	81.6	54.7	65.5
Both-Hell	81.8	42.5	55.9
Both-Nulled	87.2	67.5	76.1

Table 6: Cross-forum evaluation of the post-level product extractor. We report product type F-measure on the test sets for three variants of the post-level system: one trained on Darkode, one trained on Hack Forums, and one trained on both (as in Table 5). When the system is missing data from a particular forum, its performance degrades; the combined system works well on a range of forums.

can evaluate this limitation by training and evaluating the system on distinct forums among those we annotated. Table 6 shows variants of our system trained on just Darkode, just Hack Forums, or on both training sets (the condition from Table 5). In both cross-forum evaluation settings, performance of the extractor significantly degrades due to the reliance of the system on fine-grained features. Hack Forums contains many more posts related to online gaming, which are virtually absent in Darkode, so a Darkode-trained extractor does not perform as well on these due to having never seen the relevant terms before. In experiments, we found that our extractor was roughly twice as likely to make an error identifying a product not seen in the training data. However, our extractor still works on several other forums with only small losses in performance.

Handling posts with multiple products. One potential problem with the post-level approach is that we can only partially capture posts selling more than one product, since the system only returns a single noun phrase. We find that this does not commonly occur: we analyzed a sample of 100 posts from Darkode and Hack Forums, and found that only 3 of them actually reflected selling multiple products.

4.3 Price Extraction

For each thread, we want to extract the price of the product bought or sold and the payment method (e.g., USD, PayPal or Liberty Reserve). Price extraction proves challenging because we need to distinguish the actual price from any other price-like phrases. For example, consider the following sentence:

$150 worth of vouchers for $5

Here, $5 is the actual price of the product and $150 is not. We need to be able to extract "$5" from the post, while ignoring "$150".

	Regex			SVM		
Train/Test Forums	Prec	Rec	F_1	Prec	Rec	F_1
Darkode	-	-	-	97.7	97.7	97.7
Hack Forums	-	-	-	84.3	91.3	87.6
Darkode/Hack Forums	-	-	-	76.0	69.7	72.7
Hack Forums/Darkode	-	-	-	83.7	81.8	82.7
Both/Darkode	33.7	37.8	35.6	97.8	100.0	98.8
Both/Hack Forums	21.9	45.5	29.6	84.7	91.7	88.1
Both/Hell	24.3	47.2	32.1	83.8	64.6	72.9
Both/Nulled	23.8	42.4	30.5	87.4	66.1	75.2

Table 7: Evaluation of the **Regex**- and **SVM**-based price extractors.

4.3.1 Labeling Ground Truth.

On every post, we annotate the price of the product, the payment method, and the currency, unless the post states the price in US Dollars (which we skip annotating for convenience). We do not annotate the payment method in the absence of prices. We annotate prices on the same dataset used for product extraction.

4.3.2 Models.

We consider one baseline model and one machine-learning based model for price extraction:

Regex extractor. Extracts all the numbers and known currencies from a post as the price(s) of the product mentioned in the post. Ignores any contextual information from the posts.

SVM based extractor. Labels each token as a price or a payment method. The classifier uses token counts, position of a token in a post, parts-of-speech of the token, and membership in the Brown clusters as features. Brown clustering is a hierarchical clustering approach that creates clusters of similar words [Brown et al. 1992]. It can help disambiguate words used to refer to similar concepts.

4.3.3 Validation Results.

We evaluated both models on four forums: Darkode, Hack Forums, Hell and Nulled. We excluded Blackhat World for this analysis because of its low number of threads with prices. In our annotation dataset, 11.02% of the posts on Darkode mention pricing information. The rest of the posts usually ask the prospective buyer to send a private message to the poster to negotiate price. We noticed the opposite on Hack Forums, where 49.45% of the posts mention price. Hell and Nulled also have a higher number of posts with prices than Darkode, 19% and 44% respectively.

The **Regex** extractor performs poorly compared to the **SVM** extractor (Table 7). In our dataset, over 40% of the numbers and currencies mentioned in a post are related to prices. Without contextual information, the **Regex** extractor cannot recognize various ways of mentioning a price, and cannot distinguish regular numbers from prices.

For the **SVM** extractor, we achive both higher precision and higher recall when we train and test the model on the same forum. The accuracy on Hack Forums exceeds that for the other forums, perhaps due to Hack Forums much larger size than Darkode, thus providing more data for training the classifier. The majority of the errors occur for words used for both pricing and non-pricing information. For example, in the following sentence "pm" means private message: "Contact me via xmpp or pm"; in other contexts, "pm" can also mean Perfect Money, as in "Only accept PM or BTC."

4.3.4 Limitations.

The accuracy of the price extractor decreases when we train and test on separate forums. The discrepancy in accuracy may reflect different forums using different payment methods and discussing pricing information differently. For example, Bitcoin is one of the most used currencies on Hack Forums, but we never find it mentioned with a price on Darkode.

For some product categories, a price is not meaningful without a unit, which our current classifiers do not extract. For example, the price of 1,000 accounts is likely to be higher than the price of one account. While knowing the unit is important (especially if we want to compare the price of one category of product across multiple forums), in our dataset unit pricing is relatively rare. Only 4.09% of the posts on Darkode and 12% of the posts on Hack Forums mention a unit.

4.4 Currency Exchange Extraction

Some of the forums we considered contain large sections focused on exchanging money between electronic currencies and payment systems such as Liberty Reserve, Bitcoin, and PayPal. We treated these posts entirely separately, since the "product" is not a single noun phrase, and the price is not a single number. Instead, we consider the task of extracting several pieces of information: currencies offered, currencies desired, amounts offered, amounts desired, and exchange rates. For each of these, we wish to extract either a value, or a decision that the post does not contain it (for example, no amount appears).

Labeling Ground Truth. Our annotators find this task much more clear-cut than product extraction or price extraction. Three annotators labeled 200 posts, 100 of which we used as a development set and 100 for validation. Two of the annotators also each labeled an additional 200 posts, producing a 400 post training set. In each case, we annotated tokens as either relating to what the post offers, what it requests, or the rate.

Models. We considered two baselines and two learned models:

Fixed Order. Uses regular expressions to identify tokens corresponding to numerical amounts or known currencies. We consider the first amount and currency mentioned as offered, and the second amount and currency mentioned as requested.

Pattern-Based. Extracts patterns of token sequences from the training data, with infrequent words discarded, and numerical values and currencies collapsed into special markers. When a pattern matches text in a post, we marked the tokens as currency or amount according to the pattern.

Token Classifier. A learned classifier using local context to label each token as one of the pieces of information to be extracted.

Global Extractor. An extension of the token classifier that makes decisions about all tokens in the post simultaneously. This means decisions can interact, with the label for one token depending on labels chosen for other tokens.

Validation Results. Table 8 shows validation results on Hack Forums. In the evaluation, we de-duplicate trades mentioned multiple times in a single post (i.e., when a single post describes the exchange more than once, the system only gets credit once for getting it right).

As expected, of the two baselines, the pattern-based approach has higher precision, but cannot raise recall. The learned models balance these two more effectively, leading to further overall improvements in F-measure, and reaching 50% exact match on posts. In the remaining cases, the errors relate to a mixture of the three types of data of interest.

	All Fields				Payment Methods Only				
Models	Prec	Rec	F_1	Ex.	Prec	Rec	F_1	Ex.	Rev
Fixed	69	58	63	29	80	64	71	61	14
Pattern	90	56	69	39	93	67	78	64	3
Classifier	81	79	80	46	81	83	82	61	6
Global	87	73	80	50	86	80	83	70	5

Table 8: Validation results for the currency exchange extractor. We report results for all four models, evaluating extraction of all fields (left), and for only the currencies being exchanged (right). We assess metrics of Precision, Recall, F-measure, percentage of fully matched posts (Ex.), and for currencies, the percentage of posts in which we find the transaction direction reversed.

5 Analysis

5.1 End-to-end error analysis

To compute the end-to-end error of the type-of-post, product, and price classifiers, we manually evaluate 50 posts from Hack Forums and Nulled. For this evaluation, we consider the product classifier output as correct if it extracts the correct product noun phrase. Overall, 14% of the posts on Nulled and 16% of the posts on Hack Forums had at least one misclassification. For both of the forums, the three classifiers never made an error in the same post—understandable given the differing nature of the three classification tasks.

5.2 Broadly Characterizing a Forum

To get a shallow picture of the activity going on in a forum, we can simply assess the most frequently bought and sold products. The first two columns of Table 9 show the 10 most frequently occurring products in Darkode and Hack Forums extracted according to two methods: take the most frequent nouns, or take the most frequent product headwords. We much more consistently extract actual products, as opposed to other features of e-commerce like currencies. Moreover, they highlight interesting differences between the forums that the word frequency method misses: Darkode has a higher amount of activity surrounding malware installs and exploits, whereas Hack Forums has a larger amount of activity related to online gaming (*cod*, *boost*). This rough picture could provide an analyst with a starting point for more in-depth investigation.

Our product extractor also supports finer-grained analysis, with its prediction of complete noun phrases. If we collect the most frequent noun phrases (last column of Table 9), as opposed to headwords, we have a new frequency distribution that surfaces terms like *steam account* for Hack Forums, a gaming-related concept. The category *account* disappears and others rearrange because they are fragmented into subtypes. Accurately characterizing activity surrounding accounts poses a challenging task that we address in more detail in the next section.

5.3 Performance

We focused our evaluation of our automated tools on accuracy rather than runtime, because our tools execute quickly enough to enable in-depth, real-time analysis. For the type-of-post classifier, training the classifier from scratch and running it on the complete forum took less than 5 minutes on the English language forums (using four threads on a quad-core Macbook Pro). For the German and Russian language forums, it took 10 minutes. Our product extractor can also process the forums in 5-to-15 minutes (15-to-30 posts

Word freq.		Products		Product NPs
Darkode	Hack Forums	Darkode	Hack Forums	Hack Forums
pm	pm	install	account	crypter
price	vouch	account	service	space
site	service	traffic	crypter	service
traffic	account	email	space	setup
bot	am	bot	setup	cod
email	view	root	cod	crypt
u	paypal	exploit	crypt	boost
server	price	service	bot	steam account
anyone	method	rdp	boost	server
lr	time	site	server	method

Table 9: Top frequently-occurring stemmed nouns in Darkode and Hack Forums from two methods: simple frequency counts, and looking only at nouns tagged as products by our product extractor. The product extractor filters out numerous frequent but uninteresting concepts related to commerce (*price*, *lr*) and allows analysts to more quickly see differences of greater interest.

per second on a single core of a Macbook Pro). The price extraction and currency exchange pipelines had similarly fast runtimes, analyzing a forum in a few minutes.

6 Case Studies

The methods developed in Section 4 provide tools that an analyst can use to answer specific questions of interest. To demonstrate this, in this section we present two case studies. In Section 5.2 we showed that our product extractor can provide useful high-level characterization of the activity in a forum. Section 6.1 then shows how to take this starting point and extend it to a more fine-grained analysis of particular products. This analysis requires only a few simple rules that an analyst might write down in an hour or two of study, and shows what our methodology can provide “out of the box.”

Then, in Section 6.2, we delve deeper into a subset of posts not handled well by our existing tools, namely those involving currency exchange. Tackling this part of the underground economy requires developing additional extraction machinery: we show that we can use a process similar to that for annotating our product extraction dataset to build a currency exchange detection system here as well.

6.1 Identifying Account Activity

Noun phrases produced by our product extractor may not immediately expose the types of cybercriminal activity of interest to an analyst. Table 9 shows that the head *accounts* is very common across two forums, but these posts might correspond to users selling hacked accounts or merely selling access to one-off accounts that they legally own, a distinction of potential interest to an analyst. Knowing the type of account (*steam account* versus *instagram account*) does not necessarily help us narrow this down either.

To analyze account activity in more depth, we can use our product extractor as a starting point. We gather all posts related to accounts according to the product extractor: these include posts with product headwords *email*, *account*, or names of popular services from a small whitelisted set (*hotmail*, *snapchat*, etc.).[5] After gathering these posts, we observed a simple rule: we find that plural headwords (*accounts*, *emails*) almost always reflect users trafficking in illegally acquired accounts, whereas singular headwords typically reflect users selling their own accounts.

	Prec	Rec	F_1
Grep Baseline	58.1	64.3	61.0
Product Extractor	69.0	71.4	70.2

Table 10: Accounts case study. We have a three-class classification task of posts: they deal in original accounts, in bulk/hacked accounts, or not in accounts at all. Compared to a grep baseline, a method based on our product extractor performs better at identifying relevant posts. Precision measures how frequently we obtain a correct extraction when we identify a post related to either type of account, and recall measures how many of the gold-standard account-type posts we identify and classify correctly.

We can evaluate the efficiency of this simple set of rules on top of our product extractor. To do so, one of the authors undertook a fine-grained labeling of a set of 294 forum posts distinct from those used to train the product extractor. This labeling distinguished original (“OG”) accounts (58 posts out of our 294) from bulk/hacked accounts (28 posts out of 294). We can then evaluate the accuracy of our product extractor and rules in surfacing account posts, and correctly distinguishing between the two account classes.

Table 10 shows the results from our method on this dataset. We compare against a simple heuristic: we grep for occurrences of *accounts*, declare those to be bulk/hacked accounts, and then grep for occurrences of *account* in what remains.[6] Our method outperforms this metric by roughly 9 F_1, with gains in both precision and recall. Note that the F-measure here captures both how often we can surface account posts as well as how often we correctly distinguish between the two classes. Our method saves the analyst time (by having higher precision) and finds a higher number of relevant posts (recall) compared to our baseline.

6.2 Currency Exchange Patterns

In Figure 2, we show the result of extracting transactions from the three forums using our tool. We label rows with the payment mechanism offered and columns with the one sought. Each cell of the table shows the number of posts of the designated type. The most popular three payment mechanisms are Liberty Reserve (now defunct), Bitcoin, and Paypal.

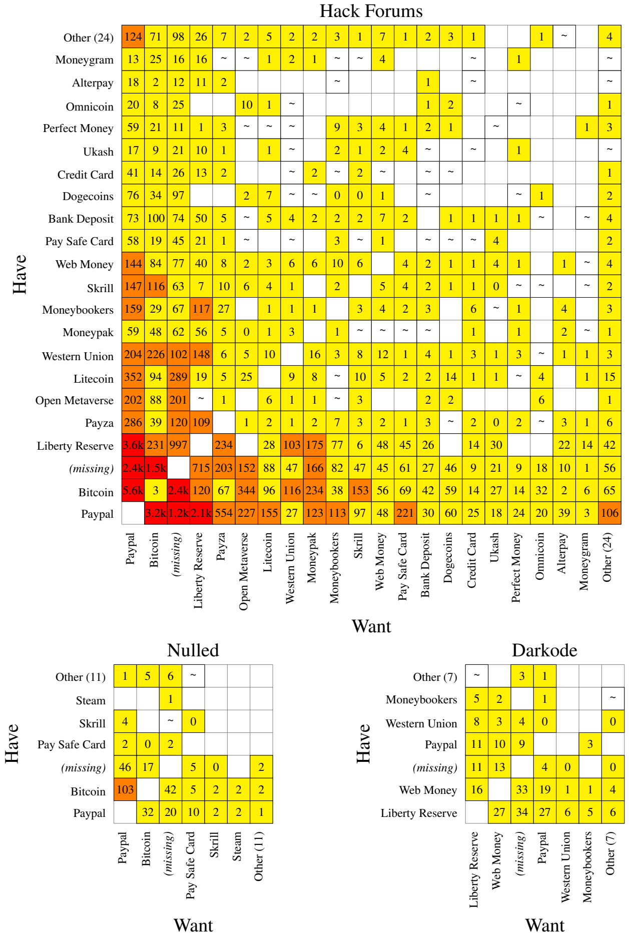
By far the most popular exchange offered is Bitcoin for PayPal, both on Hack Forums and Nulled. We suspect the reason for the demand is that exchangers can profit by charging on average a 15% fee to exchange Bitcoin and other difficult to obtain currencies for PayPal (calculated using extracted amounts and rates).

One unusual value is the square for Hack Forums showing Bitcoin–Bitcoin transactions. These indicate mistakes in our analysis of the extractor’s output, where we treated “coins” as referring to Bitcoin, when in some cases they mean other types of cryptocurrencies. Fortunately, this issue rarely occurs.

We also found surprising to observe demand for moving money from Paypal, Bitcoin, and other payment mechanisms to credit cards. Further investigation of thirty posts showed that half of these reflect requests for someone to make purcahses using with a credit card, using some other means to repay them. The other half arose from a combination of errors in our extraction, mostly related to statements regarding “CC verified” paypal accounts. These errors contrast sharply with the high accuracy observed when spot-checking one hundred of the Bitcoin to Paypal transactions (97% correct), indicating that our accuracy depends significantly on currency.

[5] We further exclude a few common noun phrases that correspond to spamming services instead, namely *bulk email*, *mass email*, or *[number] email*.

[6] Expanding what we grep for improves recall but harms precision; for example, including *email* as well decreases F-score to 54.3.



Hack Forums

Have \ Want	Paypal	Bitcoin	*(missing)*	Liberty Reserve	Payza	Open Metaverse	Litecoin	Western Union	Moneypak	Moneybookers	Skrill	Web Money	Pay Safe Card	Bank Deposit	Dogecoins	Credit Card	Ukash	Perfect Money	Omnicoin	Alterpay	Moneygram	Other (24)
Other (24)	124	71	98	26	7	2	5	2	2	3	1	7	1	2	3	1			1	~		4
Moneygram	13	25	16	16	~	~	1	2	1	~	~	4				~		1				~
Alterpay	18	2	12	11	2					~				1		~						~
Omnicoin	20	8	25			10	1	~						1	2			~				1
Perfect Money	59	21	11	1	3	~	~	~		9	3	4	1	2	1		~				1	3
Ukash	17	9	21	10	1		1	~		2	1	2	4	~		~		1				~
Credit Card	41	14	26	13	2			~	2	~	2	~		~	~							1
Dogecoins	76	34	97			2	7	~	~	0	0	1		~				~	1			2
Bank Deposit	73	100	74	50	5	~	5	4	2	2	2	7	2		1	1	1	1	~		~	4
Pay Safe Card	58	19	45	21	1	~		~		3	~	1		~	~	~	4					2
Web Money	144	84	77	40	8	2	3	6	6	10	6		4	2	1	1	4	1		1	~	4
Skrill	147	116	63	7	10	6	4	1		2		5	4	2	1	1	0	~	~		~	2
Moneybookers	159	29	67	117	27		1	1	1		3	4	2	3		6	~	1		4		3
Moneypak	59	48	62	56	5	0	1	3		1	~	~	~	~		1		1		2	~	1
Western Union	204	226	102	148	6	5	10		16	3	8	12	1	4	1	3	1	3	~	1	1	3
Litecoin	352	94	289	19	5	25		9	8	~	10	5	2	2	14	1	1	~	4		1	15
Open Metaverse	202	88	201	~	1		6	1	1	~	3			2	2				6			1
Payza	286	39	120	109		1	2	1	2	7	3	2	1	3	~	2	0	2	~	3	1	6
Liberty Reserve	3.6k	231	997		234		28	103	175	77	6	48	45	26		14	30			22	14	42
(missing)	2.4k	1.5k		715	203	152	88	47	166	82	47	45	61	27	46	9	21	9	18	10	1	56
Bitcoin	5.6k	3	2.4k	120	67	344	96	116	234	38	153	56	69	42	59	14	27	14	32	2	6	65
Paypal		3.2k	1.2k	2.1k	554	227	155	27	123	113	97	48	221	30	60	25	18	24	20	39	3	106

Nulled

Have \ Want	Paypal	Bitcoin	*(missing)*	Pay Safe Card	Skrill	Steam	Other (11)
Other (11)	1	5	6	~			
Steam			1				
Skrill	4		~	0			
Pay Safe Card	2	0	2				
(missing)	46	17		5	0		2
Bitcoin	103		42	5	2	2	2
Paypal		32	20	10	2	2	1

Darkode

Have \ Want	Liberty Reserve	Web Money	*(missing)*	Paypal	Western Union	Moneybookers	Other (7)
Other (7)	~		3	1			
Moneybookers	5	2		1			~
Western Union	8	3	4	0			0
Paypal	11	10	9			3	
(missing)	11	13		4	0		0
Web Money	16		33	19	1	1	4
Liberty Reserve		27	34	27	6	5	6

Figure 2: Number of transactions of each type observed in each forum. Numbers indicate how many posts had each type of transaction. If multiple currencies appear for either side of the transaction, then we apportion a fraction to each option. Colors indicate less than 100 (yellow), 100–1,000 (orange), and 1,000+ (red). Values are rounded, with values between 0 and 0.5 indicated by a ~ . Values for *(missing)* show when that side of the transaction was either not mentioned or not extracted (antepenultimate row in the tables). Values for "Other (#)" are the sum over other payment methods (combined to save space).

7 Conclusion and Future Work

In this work, we built several tools to enable the automatic classification and extraction of information from underground forums. We can apply our tools across a variety of forums, accommodating differences in language and forum specialization. We tested our tools on 8 different underground forums, achieving high performance both within-forum and across-forum. We also performed two case studies to show how to analysts can use these tools to investigate underground forums to discern insights such as the popularity of original vs. bulk/hacked accounts, or what kind of currencies have high demand. Our tools allow for future researchers to continue this type of large-scale automated exploration to extract a holistic view of a single or several underground forums, as well as potentially provide support to law enforcement investigating cybercrime.

In the future, we seek to explore how private messages (where the actual transactions occur) affect price. Analysis relying on both private and public data vs. just public may reach different conclusions about the revenue of a market. This work could assess the soundness of market analysis on public data, perhaps even allowing the prediction of private information (like finalized price) from public. We also want to expore the social networks of users. Finding key players central in criminal networks could provide insight into a market's organization and structure.

Acknowledgments

This work was supported in part by the National Science Foundation under grants CNS-1237265 and CNS-1619620, by the Office of Naval Research under MURI grant N000140911081, by the Center for Long-Term Cybersecurity and by gifts from Google. We thank all the people that provided us with forum data for our analysis; in particular Scraping Hub and SRI for their assistance in collecting data for this study. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

8 References

[Afroz et al. 2013] Sadia Afroz, Vaibhav Garg, Damon McCoy, and Rachel Greenstadt. 2013. Honor among thieves: A common's analysis of cybercrime economies. (2013), 1–11.

[Bamman et al. 2013] David Bamman, Brendan O'Connor, and Noah A. Smith. 2013. Learning Latent Personas of Film Characters. In *Proceedings of ACL*.

[Brown et al. 1992] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics* 18, 4 (1992), 467–479.

[Chen and Manning 2014] Danqi Chen and Christopher D Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of EMNLP*.

[Daume III 2007] Hal Daume III. 2007. Frustratingly Easy Domain Adaptation. In *Proceedings of ACL*.

[Duchi et al. 2011] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *JMLR* 12 (2011), 2121–2159.

[Fader et al. 2011] Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying Relations for Open Information Extraction. In *Proceedings of EMNLP*.

[Fan et al. 2008] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. (2008), 1871–874.

[Fleiss 1971] J. L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76, 5 (1971), 378–382.

[Franklin et al. 2007] Jason Franklin, Vern Paxson, Adrian Perrig, and Stefan Savage. 2007. An Inquiry into the Nature and Causes of the Wealth of Internet Miscreants. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*. ACM, 375–388.

[Freitag and McCallum 2000] Dayne Freitag and Andrew McCallum. 2000. Information Extraction with HMM Structures Learned by Stochastic Optimization. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*.

[Garg et al. 2015] Vaibhav Garg, Sadia Afroz, Rebekah Overdorf, and Rachel Greenstadt. 2015. Computer-Supported Cooperative Crime. In *Financial Cryptography and Data Security*. 32–43.

[Holt and Lampke 2010] T. J. Holt and E. Lampke. 2010. Exploring Stolen Data Markets Online: Products and Market Forces. *Criminal Justice Studies* 23, 1 (2010), 33–50.

[Kaljahi et al. 2015] Rasoul Kaljahi, Jennifer Foster, Johann Roturier, Corentin Ribeyre, Teresa Lynn, and Joseph Le Roux. 2015. Foreebank: Syntactic Analysis of Customer Support Forums. In *Proceedings of EMNLP*.

[Kim et al. 2010] Su Nam Kim, Li Wang, and Timothy Baldwin. 2010. Tagging and Linking Web Forum Posts. In *Proceedings of CoNLL*.

[Krebs 2013a] Brian Krebs. 2013a. Cards Stolen in Target Breach Flood Underground Markets. http://krebsonsecurity.com/2013/12/cards-stolen-in-target-breach-flood-underground-markets. (2013).

[Krebs 2013b] Brian Krebs. 2013b. Who's Selling Credit Cards from Target? http://krebsonsecurity.com/2013/12/whos-selling-credit-cards-from-target. (2013).

[Kummerfeld et al. 2015] Jonathan K. Kummerfeld, Taylor Berg-Kirkpatrick, and Dan Klein. 2015. An Empirical Analysis of Optimization for Max-Margin NLP. In *Proceedings of EMNLP*.

[Lui and Baldwin 2010] Marco Lui and Timothy Baldwin. 2010. Classifying User Forum Participants: Separating the Gurus from the Hacks, and Other Tales of the Internet. In *Proceedings of the Australasian Language Technology Association Workshop (ALTA)*.

[Manning et al. 2014] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of ACL: System Demonstrations*.

[Motoyama et al. 2011] Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage, and Geoffrey M. Voelker. 2011. An Analysis of Underground Forums. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*. ACM, 71–80.

[NIST 2005] NIST. 2005. The ACE 2005 Evaluation Plan. In *NIST*.

[O'Connor et al. 2013] Brendan O'Connor, Brandon M. Stewart, and Noah A. Smith. 2013. Learning to Extract International Relations from Political Context. In *Proceedings of ACL*.

[of Justice 2015] Department of Justice. 2015. Major Computer Hacking Forum Dismantled. https://www.justice.gov/opa/pr/major-computer-hacking-forum-dismantled. (2015).

[Parikh et al. 2015] Ankur P. Parikh, Hoifung Poon, and Kristina Toutanova. 2015. Grounded Semantic Parsing for Complex Knowledge Extraction. In *Proceedings of NAACL*.

[Ratliff et al. 2007] Nathan J. Ratliff, Andrew Bagnell, and Martin Zinkevich. 2007. (Online) Subgradient Methods for Structured Prediction. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*.

[Soska and Christin 2015] Kyle Soska and Nicolas Christin. 2015. Measuring the longitudinal evolution of the online anonymous marketplace ecosystem. In *24th USENIX Security Symposium (USENIX Security 15)*. 33–48.

[Stone-Gross et al. 2011] Brett Stone-Gross, Thorsten Holz, Gianluca Stringhini, and Giovanni Vigna. 2011. The Underground Economy of Spam: A Botmaster's Perspective of Coordinating Large-scale Spam Campaigns. In *Proceedings of the 4th USENIX Conference on Large-scale Exploits and Emergent Threats (LEET'11)*.

[Surdeanu 2013] Mihai Surdeanu. 2013. Overview of the TAC2013 Knowledge Base Population Evaluation: English Slot Filling and Temporal Slot Filling,. In *Proceedings of the TAC-KBP 2013 Workshop*.

[Tjong Kim Sang and De Meulder 2003] Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of CoNLL*.

[Wang et al. 2011] Li Wang, Marco Lui, Su Nam Kim, Joakim Nivre, and Timothy Baldwin. 2011. Predicting Thread Discourse Structure over Technical Web Forums. In *Proceedings of EMNLP*.

[Yip et al. 2012] M. Yip, N. Shadbolt, and C. Webber. 2012. Structural analysis of online criminal social networks. In *Intelligence and Security Informatics (ISI), 2012 IEEE International Conference on*. 60–65.