# DNRTI: A Large-scale Dataset for Named Entity Recognition in Threat Intelligence

Xuren Wang†, Xinpei Liu†
Information Engineering College
Capital Normal University
Beijing, China
wangxuren@cnu.edu.cn,
1161002076@cnu.edu.cn

Shengqin Ao, Ning Li,
Zhengwei Jiang†
Institute of Information Engineering
Chinese Academy of Sciences
School of Cyber Security
University of Chinese Academy of Sciences
Beijing, China
aoshengqin, lining6,
jiangzhengwei@iie.ac.cn

Zongyi Xu, Zihan Xiong,
Mengbo Xiong, Xiaoqing Zhang
Information Engineering College
Capital Normal University
Beijing, China
1161002095, 2181002064,
2181002015, 2191002072@
cnu.edu.cn

***Abstract*—Named entity recognition is an important and challenging problem in Natural language processing. Although the past decade has witnessed major advances in entity recognition in many fields, such successes have been slow to network security field, not only because of the data in the network security field is very professional, but also due to the sensitive information in the data. To advance named entity recognition research in network security field, we introduce a large-scale Dataset for Named Entity Recognition in Threat Intelligence (DNRTI). To this end, we collect more than 300 pieces of threat intelligence. The data in DNRTI is all annotated by experts in threat intelligence interpretation using 13 object categories. The fully annotated DNRTI contains 175220 words. To build a baseline for named entity recognition in the threat intelligence field, we evaluate some deep learning model on DNRTI. Experiments demonstrate that DNRTI well represents the key information in threat intelligence and are quite challenging.**

***Keywords- threat intelligence; dataset; BiLSTM; named entity recognition and classification***

## I. INTRODUCTION

Named entity recognition in threat intelligence field refers to identify objects of interest and predicting their categories in the threat intelligence. In contrast to conventional named entity recognition datasets, where entities are about time, place, person, etc. The entities in DNRTI are closely related to the content of threat intelligence.

As countries around the world attach importance to the network environment and defend against attacks by hacker organizations, extensive studies have been devoted to named entity recognition in network security. As early as in the KDD CUP competition held in 1999, the KDD99 dataset was used in the field of network security. This data set records the network connection, marking the network connection as normal or attack, and the attacks are subdivided into 4 categories, with a total of 39 attack types [1]. The disadvantage of the KDD99 data set is that it has a lot of redundant data [2]. The CSIC2010 data set is about web requests and is mainly used to test network attack protection systems. It contains 36,000 normal requests and more than 25,000 abnormal requests. HTTP requests are marked as normal or abnormal. The data set includes various attacks, such as SQL injection, buffer overflow, information collection, file disclosure, CRLF injection, cross-site scripting, server-side inclusion, parameter tampering, etc. The Honeynet data set is a hacker attack data set collected by the Honeynet organization. It can well reflect the hacker's attack pattern, including the time of the attack, the triggered Snort rule number, alarm content, source IP, destination IP, etc [3]. The malicious-URLs data set contains a large number of URLs. These URLs are marked as normal URLs or malicious URLs. This is a data set for machine learning [4]. MalwareTextDB is a new database for annotated malware texts. It is based on the MAEC vocabulary and 39 annotated APT reports with a total of 6,819 sentences [5].

Some of these data sets focus on traffic detection or a certain network attack method. Most of the data in the data set cannot be connected with other content such as specific hacker organizations, attack activities, and countermeasures. It is very unfavorable for those who do not have professional knowledge to use and the scalability of the data set is not high.

To advance the named entity recognition research in network security field, this paper introduces a large-scale Dataset for Named entity Recognition in Threat Intelligence (DNRTI). We collect and analysis more than 300 pieces of threat intelligence reports from the open source threat intelligence websites. Each threat intelligence report is released by world-renowned security companies or government agencies and analyzed and annotated by experts in related fields. The fully annotated DNRTI dataset contains 175220 words. The main contributions of this work are:

To our knowledge, DNRTI is the largest named entity recognition dataset with a wide variety of categories in threat intelligence field. It can be used to develop and evaluate other artificial intelligence processing tools as named entity recognition model, knowledge Graph, etc. We will continue to update DNRTI, to grow in size and scope and to reflect real threat intelligence. You can find our DNRTI data set on the

978-1-6654-0392-4/20/$31.00 ©2020 IEEE
DOI 10.1109/TrustCom50675.2020.00252

GitHub, https://github.com/SCreaMxp/DNRTI-A-Large-scale-Dataset-for-Named-Entity-Recognition-in-Threat-Intelligence.

We also benchmark named entity recognition algorithms on DNRTI, which can be used as the baseline for future algorithm development.

## II. Related Work

In the past few years, threat intelligence is attracting social attention and more and more scientists are devoted to threat intelligence-related research. In 2014, in order to better realize cyber threat intelligence and information sharing, Sean Barnum et al. [6] proposed The Structured Threat Information Expression (STIX ™) for the first time. STIX is a structured language used to describe cyber threat information. Its main purpose is to allow cyber threat information to be shared, analyzed and stored in a unified way. It consists of 9 types of keywords, including Observable, Indicators, Incidents, TPP (Adversary Tactics, Techniques, and Procedures), Exploit Targets, Courses of Action, Campaigns, Threat Actors and Reports. It can be used in these tasks including: threat analysis, threat feature classification, emergency handling of threats and security incidents, and threat intelligence sharing. In 2018, an automated dataset generation system called CTIMiner [7] was proposed. The system collects threat data from publicly available security reports and malware repositories. There are 10 types of attributes stored in the dataset and they are the Hash, IP, URL, e-mail address, date and time, CVE, file name, PDB path, digital code sign serial number, and other string data, including the author and title of the document. According to cyber campaigns or threat actors, these data can be divided into 9 categories. By using these data, security experts can do correlation analysis based on some key information and do temporal analysis not only to defend against current incidents and presume the underlying intent, but also to draw the direction of adversarial activities from the big picture. Ba-Dung Le et al. [8] collected and analyzed data from Twitter. They developed a framework for automatically gathering Cyber threat intelligence from Twitter and collected over a period of twelve months from 50 influential Cyber security related accounts to evaluate the framework. The framework utilizes a novelty detection model that learns the features of Cyber threat intelligence from the CVE descriptions and classifies each input tweet as either normal or anomalous. By collecting these tweets and finding the relevant CVE identifier, security experts can get further information that are valuable for Cyber threat-related applications. Research shows that text-intensive and semi-structured data is of very little use for security experts due to its extent and lack of human-readability. In order to improve the efficiency of data utilization by security experts, Fabian Böhm et al. [9] proposed KAVAS, a knowledge-assisted visual analytics concept for the STIX. KAVAS consists of two main components. One is Cyber Threat Intelligence (CTI) Vault for storing and managing STIX and the other is a corresponding visual analytics component that enables users to understand and interact with complex threat intelligence information. KAVAS allows CTI Vault to persists STIX-based threat intelligence information in a graph database and it provides the possibilities to store externalized user knowledge in its knowledge base, while the integrity of the original information is preserved and ensured. In addition, KAVAS' visual component can display threat intelligence and enables security experts to interactively explore incidents and gain insight about what happened. The key content in the threat intelligence is complex and diverse. Malicious URL is one of them. Lu Zhi-gang et al. [10] designed a malicious URL detection system based on the threat intelligence platform, which combines URL detection with threat intelligence. The URL will be input into the malicious URL library in the threat intelligence database for matching. If the match is successful, this URL will be directly blocked. If the match fails, the system will extract the structural features, intelligence features, and sensitive word features of the URL and put it into the multi-classifier voting model for detection. The detection system can not only detect known malicious URLs efficiently, but also identify unknown malicious URLs. Malicious domain names are also an important part of threat intelligence. Daiki Chiba et al. [11] designed and implemented a unified analysis system combining current defense solutions to build actionable threat intelligence from malicious domain names. The basic concept underlying their system is malicious domain name chromatography. It can distinguish among mixtures of malicious domain names for websites. Lim K L A et al. [12] disclosed a system and method for consolidating threat intelligence data for a computer and its related networks. They collected raw threat intelligence data from a plurality of sources and used unsupervised machine learning algorithms to cluster the data. Then threat intelligence data subsequently undergoes a weighted asset-based threat severity level correlation process. Finally, threat intelligence data will be formatted into predefined formats. The knowledge graph construction technology of threat intelligence has become an important research direction in the field of network security. WANG Tong and others [13] proposed a supervised deep learning model, which can automatically extract the entity and entity relationship of threat intelligence, and visualizes the knowledge map through graph data. They used Deep Belief Network (DBN) as the classifier of threat intelligence entities and entity relationships and used Neo4j database to draw knowledge graph.

## III. Motivations

Datasets have played an important role in data-driven research in recent years. Large datasets like Enron Dataset [14] and CoNLL-2003 are instrumental in promoting named entity recognition research in different languages [15]. When it comes to the sentiment analysis tasks, the same is true for Multi-Domain Sentiment Dataset [16].

However, in the field of threat intelligence a dataset resembling CoNLL-2003 both in terms of entity number and detailed annotations has been missing, which becomes one of the main obstacles to the research in threat intelligence, especially for developing deep learning-based algorithms. Threat intelligence named entity recognition is extremely helpful for responding to cyber-attacks, maintaining the network environment, repairing and responding to software vulnerabilities. Therefore, a large-scale and challenging

dataset, being as close as possible to real threat intelligence, is imperative for promoting research in this field. We argue that a good Threat intelligence named entity recognition data set should possess four properties, namely,

- a large number of words.
- many instances per categories.
- properly annotation.
- many different categories, which make it approach to real threat intelligence.

## IV. ANNOTATION OF DNRTII

### A. Data Collection

In the field of network security, the quality and type of threat intelligence are factors that cause dataset biases. To eliminate the biases, threat intelligence in our dataset is collected from the websites of multiple security companies or government agencies around the world and GitHub. To increase the diversity of data, we have collected and analyzed a large amount of threat intelligence reports, and deleted the part that does not contain usable entities to ensure the high quality of the data set.

### B. Category Selection

13 categories are chosen and annotated in our DNRTI dataset, including hacker organization, attack, sample file, security team, tool, time, purpose, area, industry, organization, way, loophole, features. The corresponding labels of these 13 categories in the data set are HackOrg, OffAct, SamFile, SecTeam, Tool, Time, Purp, Area, Idus, Org, Way, Exp, Features respectively.

We compared DNRTI with the MalwareTextDB dataset, which is a relatively mature threat intelligence dataset.

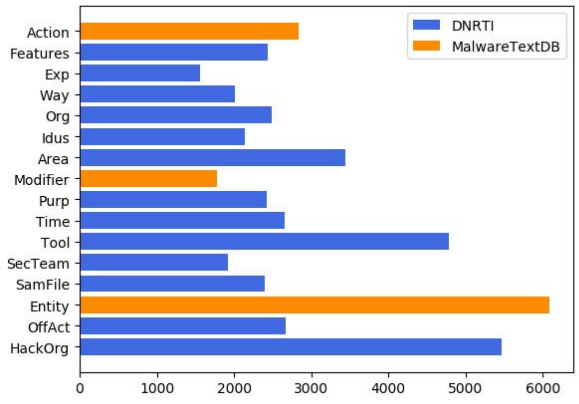


Figure 1. Comparison of data in DNRTI and MalwareTextDB.

Among them, the category "Entity" in the MalwareTextDB dataset is subdivided into "HackOrg" and "OffAct" in DNRTI. The category "Modifier" is made of some prepositions or phrases that express purpose. The words in category "Action" are only some verbs, and we think these verbs are of little value. We believe that DNRTI surpass MalwareTextDB not only in the numbers of category, but also the number of instances per category. The type and quantity of each entity in the two data sets are shown in Fig. 1.

### C. Annotation Tools and Methods

We use Brat Rapid Annotation Tool to annotate threat intelligence, which is a web-based text annotation tool [17]. We choose the BIO labeling mode, which is the industry standard labeling form in the field of text labeling. We divide the entities in threat intelligence into entity start tags "B-X" and entity continuation tags "I-X", and no entity tags "O". The annotation result is shown in Fig. 2 and Fig. 3.


Figure 2. The Brat Rapid Annotation Tool.

```
The O
admin@338 B-HackOrg
started O
targeting O
Hong B-Area
Kong I-Area
media B-Org
companies I-Org
, O
probably O
in O
response O
to O
political B-Idus
and O
economic B-Idus
challenges O
in O
Hong B-Area
Kong I-Area
and O
China B-Area
. O
```

```
The O
admin@338 B-HackOrg
, O
active O
since O
2008 B-Time
, O
has O
been O
seen O
targeting O
organizations O
in O
the O
financial B-Idus
services I-Idus
, O
telecoms B-Idus
, O
government B-Idus
, O
and O
defense B-Org
sectors I-Org
. O
```

```
In O
August B-Time
2013 I-Time
, O
FireEye B-SecTeam
reported O
that O
admin@338 B-HackOrg
had O
been O
using O
the O
Poison B-Tool
Ivy I-Tool
RAT I-Tool
in O
its O
operations O
. O
```

Figure 3. The annotation result of DNRTI.

### D. Dataset Splits

In order to ensure that the training data and test data distributions approximately match, we randomly select 70% of the original text as the training set, 15% as validation set, and 15% as the testing set.

### E. Statistics of Dataset DNRTI

There are 175,220 vocabularies in this dataset, and all entities are divided into 13 categories, with a total of 27 tag types. Among them, there are 138808 entities with O-labels and 36,412 entities with non-O-labels. Entities with non-O tags account for 20.1% of the total vocabulary. The number and proportion of each label are shown in TABLE I.

TABLE I. THE NUMBER AND PROPORTION OF EACH LABEL

| DNRTI | Number and Proportion | |
|---|---|---|
| | *Number* | *Proportion* |
| HackOrg | 5465 | 15.01% |
| OffAct | 2669 | 7.33% |
| SamFile | 2400 | 6.59% |
| SecTeam | 1921 | 5.28% |
| Tool | 4784 | 13.13% |
| Time | 2659 | 7.30% |
| Purp | 2424 | 6.66% |
| Area | 3447 | 9.47% |
| Idus | 2136 | 5.87% |
| Org | 2489 | 6.84% |
| Way | 2018 | 5.54% |
| Exp | 1559 | 4.28% |
| Features | 2441 | 6.70% |

## V. EVALUATIONS

We evaluate the deep learning methods on DNRTI. For named entity recognition task in threat intelligence, we carefully select LSTM, BiLSTM methods as our benchmark testing algorithms for their excellent performance on general named entity recognition tasks [18] [19] [20]. We modify the original LSTM algorithm such that it can extract lexical-level features and character-level features as word features.

### A. Feature Extraction

Lexical-level features refer to the features that are related to words. We first use statistical tools to extract different words of the article, then input it into GLoVe model. After calculation in the GloVe model, the semantic similarity between two words can be calculated and we got the lexical-level features of the words [21].

The character-level features of texts have been widely used as another direction for natural language processing and have achieved good results. For example, Kuru O [22] expressed sentences as character sequences for named entity recognition. Lee [23] performed text translation by mapping character sequences to target character sequences. Lecun [24] used character-level text as the original signal for text classification. In our paper, character-level features are also included in the word embedding layer. We first use statistical tools to extract different characters of the article, then use the Embedding layer to generate vectors for each character as its preliminary character-level features. Obviously, these preliminary character-level features cannot represent the language attributes of the text. After that, we use the stacked bidirectional LSTM network to process the preliminary character-level features. Through forward calculation and reverse calculation, we put them into Concatenate layer to get character-level features. Now we have the lexical-level feature vectors extracted by the GLoVe model, and the character-level feature vectors extracted by the stacked bidirectional LSTM. Then use the Concatenate layer to connect them as word features.

### B. LSTM and BiLSTM

Input word features into LSTM and BiLSTM models for classification and processing. We use BiLSTM or LSTM to get the contextual features of each word in a long distance. Despite the distance between the words, it can learn and capture the complex dependencies and key information between multiple words in sequential data. For the BiLSTM model, there are two layers of hidden nodes from two separate LSTMs. These two LSTMs can capture dependencies in two directions. For the input objects $x^t$, $x^{t+1}$, $x^{t+2}$, they are sequentially input into the forward LSTM to obtain three vectors $\{\vec{h}_{L0}, \vec{h}_{L1}, \vec{h}_{L2}\}$, and then sequentially input into the backward LSTM to obtain three vectors $\{\overleftarrow{h}_{R0}, \overleftarrow{h}_{R1}, \overleftarrow{h}_{R2}\}$, and finally concatenate the forward and backward hidden vectors, namely $h_0$, $h_1$, $h_2$. The two states capture semantic information in both directions of the word sequence [25] [26]. Dropout is also used to avoid overfitting [27].

### C. Output Layer

In the output layer there are two Dense layers. The output of the first Dense layer is fed into the second Dense layer with n hidden neurons, where n is the number of label categories. Then, we feed the vectors into the Softmax function for prediction. w and b are the parameters to be learned. p is a probability that indicates which type of entity the word belongs to. The Softmax function is shown by formula (1). The flow chart of the model is shown in Fig. 4.

$$\mathrm{p} = \mathrm{Softmax}([wh + b]) \tag{1}$$

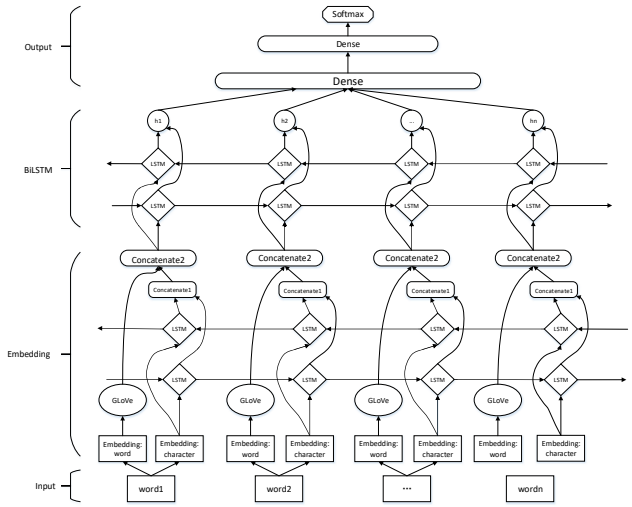


Figure 4. The flow chart of the model.

### D. Hyper-parameters

- The dimension of the vocabulary-level embedding vector was set to 100 in the embedding layer

- The dimension of the character-level embedding vector was set to 200 in the embedding layer
- The dimension of the word embedding vector was set to 300 in the embedding layer
- The neural models were trained using a batch size of 5 on the training set
- The number of hidden layer nodes in BiLSTM model was set to 200
- The number of hidden layer nodes in LSTM model was set to 200
- Adam was employed as an optimization algorithm
- The learning rate was set to 0.001
- The dropout rate was set to 0.2

## VI. EXPERIMENTAL ANALYSIS

After training, we compare the prediction results of the two models on the same test set, including the prediction accuracy, precision, recall, and F1-score of each type of label. The comparison results are shown in TABLE II and TABLE III.

TABLE II. COMPARISON OF EACH LABEL

| Entity | LSTM | | | BiLSTM | | |
|---|---|---|---|---|---|---|
| | ***Precision*** | ***Recall*** | ***F1-score*** | ***Precision*** | ***Recall*** | ***F1-score*** |
| B-HackOrg | 0.55 | 0.87 | 0.67 | 0.67 | 0.82 | 0.74 |
| I-HackOrg | 0.6 | 0.87 | 0.71 | 0.64 | 0.9 | 0.75 |
| B-OffAct | 0.8 | 0.59 | 0.68 | 0.75 | 0.77 | 0.76 |
| I-OffAct | 0.84 | 0.7 | 0.77 | 0.78 | 0.68 | 0.73 |
| B-SamFile | 0.68 | 0.56 | 0.61 | 0.69 | 0.65 | 0.67 |
| I-SamFile | 0.71 | 0.74 | 0.73 | 0.77 | 0.77 | 0.77 |
| B-SecTeam | 0.85 | 0.72 | 0.78 | 0.94 | 0.8 | 0.86 |
| I-SecTeam | 0.66 | 0.7 | 0.68 | 0.71 | 0.64 | 0.67 |
| B-Time | 0.84 | 0.91 | 0.87 | 0.91 | 0.89 | 0.9 |
| I-Time | 0.86 | 0.83 | 0.84 | 0.84 | 0.82 | 0.83 |
| B-Way | 0.6 | 0.77 | 0.67 | 0.77 | 0.77 | 0.77 |
| I-Way | 0.47 | 0.8 | 0.59 | 0.53 | 0.75 | 0.62 |
| B-Tool | 0.58 | 0.29 | 0.39 | 0.69 | 0.53 | 0.6 |
| I-Tool | 0.76 | 0.47 | 0.58 | 0.72 | 0.49 | 0.58 |
| B-Idus | 0.76 | 0.54 | 0.63 | 0.90 | 0.65 | 0.76 |
| I-Idus | 0.75 | 0.47 | 0.58 | 0.71 | 0.45 | 0.55 |
| B-Org | 0.43 | 0.41 | 0.42 | 0.55 | 0.57 | 0.56 |
| I-Org | 0.62 | 0.72 | 0.67 | 0.66 | 0.76 | 0.71 |
| B-Area | 0.81 | 0.94 | 0.87 | 0.82 | 0.92 | 0.87 |
| I-Area | 0.81 | 0.83 | 0.82 | 0.81 | 0.83 | 0.82 |
| B-Purp | 0.62 | 0.13 | 0.22 | 0.66 | 0.25 | 0.36 |
| I-Purp | 0.59 | 0.32 | 0.41 | 0.66 | 0.33 | 0.44 |
| B-Exp | 0.9 | 0.91 | 0.9 | 0.95 | 0.95 | 0.95 |
| I-Exp | 0.86 | 0.88 | 0.87 | 0.92 | 0.98 | 0.95 |
| B-Features | 0.74 | 0.45 | 0.56 | 0.78 | 0.49 | 0.6 |
| I-Features | 0.8 | 0.54 | 0.64 | 0.89 | 0.31 | 0.46 |

TABLE III. COMPARISON OF ACCURACY AND F1-SCORE

| Overall | F1-score | Accuracy |
|---|---|---|
| LSTM | 67.09 | 89.53 |
| BiLSTM | 71.29 | 90.85 |

From Table II and Table III, we can find that the BiLSTM model is stronger than the LSTM model for named entity recognition in DNRTI. For each category of entities, we found that the entity categories "HackOrg", "OffAct", "SamFile", "SecTeam", "Time", "Way", "Area", "Exp" have better recognition results, and their F1-score are around 75, which is a satisfactory result.

For entity categories "Tool", "Idus", "Org", the lower recall rate is the reason why their F1-score are unsatisfactory. These entities are composed of the tools, industries, organizations in each threat intelligence. Due to the large number of these entities and most of them are different, the low repetition rate of each entity is very unfavorable for the deep learning.

The F1-score of entities "Purp" and "Features" are worrying. It is mainly due to its excessively low recall rate. After analysis, we not only find that these entities are too complex and there is no specific lexical form, but also know that they are small in number and have low repetition rate. In future work, we can consider changing the definition of these types of entities to determine the types of these entities more clearly.

In order to compare the performance of DNRTI and MalwareTextDB on deep learning algorithms. We used the BiLSTM model to do another experiment on MalwareTextDB. We still use the ratio of 7:1.5:1.5 to divide the entire data set into training set, test set and validation set and we keep all hyperparameters unchanged. The comparison of DNRTI and MalwareTextDB 's accuracy rate and F1-score are shown in Table IV. The precision, recall and F1-score of each type of entity are shown in Table V.

TABLE IV. COMPARISON OF ACCURACY AND F1-SCORE

| Overall | F1-score | Accuracy |
|---|---|---|
| DNRTI | 71.29 | 90.85 |
| MalwareTextDB | 47.52 | 87.53 |

TABLE V. EXPERIMENTAL RESULTS OF MALWARETEXTDB

| Entity | Precision | Recall | F1-score |
|---|---|---|---|

| | | | |
|---|---|---|---|
| B-Modifier | 0.39 | 0.37 | 0.38 |
| I-Modifier | 0.67 | 0.12 | 0.20 |
| B-Entity | 0.52 | 0.46 | 0.49 |
| I-Entity | 0.44 | 0.57 | 0.50 |
| B-Action | 0.50 | 0.46 | 0.48 |
| I-Action | 0.39 | 0.33 | 0.35 |

From Table IV and Table V, we can find that our dataset DNRTI is not only higher than the data set MalwareTextDB in terms of entity type and number of entities, but also have better performance in deep learning algorithms. Besides, the accuracy is lower than our dataset DNRTI, which is 87.53.

## VII. CROSS-DATASET VALIDATIONS

The cross-dataset generalization is an evaluation for the generalization ability of a dataset [28]. There are almost no data sets related to threat intelligence marked by BIO. We choose the universal data set CoNLL-2003 to do cross-dataset generalization, because some categories in its data set are similar to the data in DNRTI, such as area and organization names. The results are shown in Table Ⅴ. For irrelevant data sets with repeated content, the model can better identify categories. It suggests that DNRTI data set is professional, readable and comprehensive in the field of threat intelligence.

TABLE VI. RECOGNITION RESULTS OF SIMILAR DATA

| Entity | Precision | Recall | F1-score |
|---|---|---|---|
| B-Area | 0.69 | 0.48 | 0.56 |
| I-Area | 0.45 | 0.3 | 0.36 |
| B-HackOrg | 0.36 | 0.49 | 0.42 |
| I-HackOrg | 0.56 | 0.57 | 0.56 |

## VIII. CONCLUSION

We build a large-scale dataset for named entity recognition in threat intelligence which is much larger than any existing datasets in this field. We use the industry standard BIO annotation method, which makes this data set very scalable and easy to use. We assume this dataset is challenging but very similar to the real threat intelligence, which are more appropriate for practical applications. We also establish a benchmark for named entity recognition in threat intelligence. We believe DNRTI will not only promote the development of named entity recognition algorithms in threat intelligence, but also promote the establishment of relationships between threat intelligence entities and even apply it to the construction of knowledge graphs in the field of threat intelligence or the rapid response of network security defense.

## ACKNOWLEDGMENT

We thank Nan Ma, Wengen Xie, Zongyi Xu, Mengjie Guo, Xue Zhao, Ziyang Chen, Jing Zeng, Jinyu Hou, Jie Yang, Zihan Xiong, Mengbo Xiong and all the others who involved in the annotations of DNRTI. This work is supported by the National Key Research and Development Program of China (No.2019QY1302, No.2016YFB0801004).

## REFERENCES

[1] Stolfo S J. KDD cup 1999 dataset[J]. UCI KDD repository. http://kdd. ics. uci. edu, 1999.

[2] John M. The 1998 lincoln laboratory ids evaluation: Acritique[C]//Proceedings of International Symposium on Recent Advances in Intrusion Detection RAID). 2000: 145-161.

[3] Holt T J, Kilger M. Know your enemy: The social dynamics of hacking[J]. The Honeynet Project, 2012: 1-17.

[4] Cheng A. Using Machine Learning to Detect Malicious URLs[J]. 2017.

[5] Lim S K, Muis A O, Lu W, et al. Malwaretextdb: A database for annotated malware articles[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017: 1557-1567.

[6] Barnum S. Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX)[J]. Mitre Corporation, 2012, 11: 1-22.

[7] Kim D, Kim H K. Automated Dataset Generation System for Collaborative Research of Cyber Threat Analysis[J]. Security and Communication Networks, 2019, 2019.

[8] Le B D, Wang G, Nasim M, et al. Gathering cyber threat intelligence from Twitter using novelty classification[C]//2019 International Conference on Cyberworlds (CW). IEEE, 2019: 316-323.

[9] Böhm F, Menges F, Pernul G. Graph-based visual analytics for cyber threat intelligence[J]. Cybersecurity, 2018, 1(1): 16.

[10] Xin W , Yang W U , Zhi-Gang L U . Study on Malicious URL Detection Based on Threat Intelligence Platform[J]. Computer ence, 2018.

[11] Chiba D, Akiyama M, Yagi T, et al. DomainChroma: Building actionable threat intelligence from malicious domain names[J]. Computers & Security, 2018, 77: 138-161.

[12] Lim K L A. System and method for high speed threat intelligence management using unsupervised machine learning and prioritization algorithms: U.S. Patent Application 14/891,621[P]. 2017-8-10.

[13] Tong W, Zhong-liang A I, Xian-guo Z. Knowledge Graph Construction of Threat Intelligence Based on Deep Learning[J]. Computer and Modernization, 2019 (12): 21.

[14] Shetty J, Adibi J. The Enron email dataset database schema and brief statistical report[J]. Information sciences institute technical report, University of Southern California, 2004, 4(1): 120-128.

[15] Sang E F, De Meulder F. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition[J]. arXiv preprint cs/0306050, 2003.

[16] Dragoni M, Tettamanzi A, da Costa Pereira C. Using Fuzzy Logic For Multi-Domain Sentiment Analysis[C]//International Semantic Web Conference (Posters & Demos). 2014: 305-308.

[17] Stenetorp P, Pyysalo S, Topić G, et al. BRAT: a web-based tool for NLP-assisted text annotation[C]//Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics. 2012: 102-107.

[18] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[19] Gers F A, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM[J]. 1999.

[20] Limsopatham N, Collier N. Bidirectional lstm for named entity recognition in twitter messages[J]. 2016.

[21] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation[C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.

[22] Kuru O, Can O A, Yuret D. Charner: Character-level named entity recognition[C]//Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. 2016: 911-921.

[23] Lee J, Cho K, Hofmann T. Fully character-level neural machine translation without explicit segmentation[J]. Transactions of the Association for Computational Linguistics, 2017, 5: 365-378.

[24] Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification[C]//Advances in neural information processing systems. 2015: 649-657.

[25] Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations[J]. arXiv preprint arXiv:1802.05365, 2018.

[26] Limsopatham N, Collier N. Bidirectional lstm for named entity recognition in twitter messages[J]. 2016.

[27] Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors[J]. arXiv preprint arXiv:1207.0580, 2012.

[28] Torralba A, Efros A A. Unbiased look at dataset bias[C]//CVPR 2011. IEEE, 2011: 1521-1528.