To cite this work: P. Evangelatos et al., ”Named Entity Recognition in Cyber Threat Intelligence Using Transformer-based Models,” *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, 2021, pp. 348-353, doi: 10.1109/CSR51186.2021.9527981.

Named Entity Recognition in Cyber Threat Intelligence Using Transformer- based Models

Pavlos Evangelatos*, Christos Iliou†, Thanassis Mavropoulos‡, Konstantinos Apostolou§, Theodora Tsikrika¶,
Stefanos Vrochidis‖, and Ioannis Kompatsiaris**
Information Technologies Institute
CERTH
Thessaloniki, Greece
Email: *pevangelatos@iti.gr, †iliouchristos@iti.gr, ‡mavrathan@iti.gr, §konapost@iti.gr, ¶theodora.tsikrika@iti.gr,
‖stefanos@iti.gr **ikom@iti.gr,

Abstract—**The continuous increase in sophistication of threat actors over the years has made the use of actionable threat intelligence a critical part of the defence against them. Such Cyber Threat intelligence (CTI) is published daily on several online sources, including vulnerability databases, CERT feeds, and social media, as well as on forums and web pages from the Surface and the Dark Web. Named Entity Recognition (NER) techniques can be used to extract the aforementioned information in an actionable form from such sources. In this paper we investigate how the latest advances in the NER domain, and in particular transformer-based models, can facilitate this process. To this end, the data set for NER in Threat Intelligence (DNRTI) containing more than 300 pieces of threat intelligence reports from the open source threat intelligence websites is used. Our experimental results demonstrate that such techniques are very effective in extracting cybersecurity- related named entities, by considerably outperforming the previous state-of-the-art approaches tested with DNRTI.**

Index Terms—**Cyber Threat Intelligence, Named Entity Recognition, CTI, NER, DNRTI, BERT, XLNet, RoBERTa, ELECTRA**

I. Introduction

Threat actors are becoming more advanced, introducing new, more sophisticated techniques over the years. Defending against such attacks can be challenging, but this process can be facilitated through the use of public information about known and new threat actors, as well as their Tactics, Techniques, and Procedures (TTPs). This information can be aggregated, further analysed, and enriched to generate intelligence about these threats, referred to as Cyber Threat Intelligence (CTI).

There are several public sources that can be leveraged to extract CTI, including CERT feeds, vulnerability databases, security reports, websites and forum posts from the Surface and Dark Web, social media posts, and more. Such sources contain a huge amount of information and intelligence about threats which is provided either in a semi- structured or in an unstructured manner. Extracting cybersecurity- related entities (e.g., malware names, hashes, tool names, purpose of attacks, the way the attacks were performed, etc.) from such sources can significantly assist towards extracting actionable CTI.

Named Entity Recognition (NER) is an information extraction task where named entities are first identified in a text and then classified into predefined categories. NER systems have shown very high performance in several domains including identifying named entities in legal documents [1], [2], social media posts [3], [4], documents related to chemistry [5], and biographical texts [6]. In recent years, deep network architectures have achieved significant improvements in the performance of various Natural Languages Processing (NLP) tasks including NER, while currently transformer-based models constitute the state of the art in NLP and NER tasks. However, such novel techniques have not been thoroughly tested in the CTI domain, where the focus in on named entities that are closely related to the content of threat intelligence.

In this work, we investigate how the latest advances in NER, and in particular transformer-based models, can facilitate the CTI extraction process. To this end, an experimental study is performed on DNRTI [7], the only data set with an extensive list of cybersecurity- related classes. DNRTI contains more than 300 pieces of threat intelligence obtained from open source threat intelligence websites; these reports contain $175,220$ annotated words in 13 classes. The number of available DNRTI classes, as well as their nature, play a critical role in the extraction of more “actionable” CTI. We show that the utilized architectures outperform the approaches previously tested on this data set.

II. Related Work

Named Entity Recognition is a sub-task of Information Extraction which aims to locate and classify named entities in unstructured text. NER is considered critical for several NLP tasks, including question answering procedures, information retrieval, co-reference resolution, and topic modelling [8].

Traditional machine learning models, while performing well in tasks regarding interpretation of sequential information in a text, have not managed to achieve human-like performance. They have been outperformed by deep network architectures, in particular Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks (with several variations). One such variation that has shown promising results is the BiLSTM- CRF model which combines a bidirectional LSTM network with a Conditional Random Fields layer [9].

BiLSTM- CRFs can encode dependencies on elements at both directions, via a sequential CRF layer, to jointly model tagging decisions and denote the correctly assigned labels.

Such approaches constituted until recently the most promising solutions to NLP problems. They have though also been outperformed by transformer-based models that contributed to a rise in the size of sets utilized for the pre-training, and also designated the proper environment for the development of the next generation of models, designed primarily for NLP. Transformer-based models are composed of layers of encoders and decoders which activate an attention mechanism, that adds weights to the linearly transformed inputs and accumulates some informative elements to generate the output. Their significant advantage that makes them effective is that they do not require vast amounts of labeled data; their initial training is performed in an unsupervised manner and, then, in the fine-tuning phase, a small data set is used for supervised learning.

BERT [10] a 12 layer deep network (12 transformer blocks and 12 attention heads), adopts a multi-layer bidirectional transformer logic, instead of the legacy left-to-right, predicting randomly masked tokens and successive sentences. As a first step BERT randomly masks out 10% to 15% of the words in the training data attempting to predict the masked words [10]. Except for the masked language modeling, BERT optimizes next classification objective [11]. The lower layers encode local syntax which is useful for part-of-speech tagging and higher layers can extract complex semantics like aspects of word meaning useful for word sense disambiguation tasks [12]. BERT, released from Google AI Language, is pre-trained on the Wikipedia corpus (2500 million words) and the Book corpus (800 million words). A NER model is trained by feeding the output vector of each token into a classification layer which predicts the label.

RoBERTa, released from Facebook [13], is a replication research on Google's BERT that executed multiple comparisons and presented some performance assets, highlighting the importance of some key hyper-parameters and the size of the training data that can have a great impact on the final result.

XLNet, released from Google/CMU, [14] was developed to address some of BERT's negative aspects and revives the Recurrent Network logic (segment recurrence mechanism) and integrates a policy which is notated as ”relative positional encoding”, borrowed from the predecessor model named Transformer-XL. This is associated with the model's ability to learn how to estimate some weights for the previous and the following words to the temporary central one.

Electra [15] aims to reduce high training computational cost of models like BERT. Instead of masking the input it replaces some tokens with alternatives sampled from a small generator network. Then, instead of training a model that predicts the original identities of the corrupted tokens a discriminative model is used to predict whether each token in the altered input was replaced by a generator sample or not [15].

In the cybersecurity and, more specifically, in the CTI field, Ma et al. [16] proposed a XBiLSTM- CRF model which combines a BiLSTM network with a CRF layer in a way that the input is also concatenated with the BiLSTM output. For the experiments, a public dataset was used with nearly 5000 entities. Joshi et al. [17] used a Stanford NER model based on CRF and a hand annotated dataset containing 3800 entities. Gasmi et al. [18] used a corpus with entities from the National Vulnerability Database (NVD)[1], MS Bulletins[2], and Metasploit[3] to compare an LSTM-CRF and a CRF model, concluding that the first approach is significantly better. Bridges et al. [19] created their own dataset which was initially auto-labeled (using database matching, heuristics, and dictionary stored terms), and then fed to a history-based Maximum Entropy Model trained with an averaged perceptron classifier.

In terms of annotated datasets for NER in the CTi domain, MalwareTextDB is a corpus of annotated malware texts that was constructed in 2017 and consists of 4 entities [20]. There are limitations regarding the confined size and the small number of entities that can lead to vague results if applied to unseen sentences coming from CTI reports and other sources.

III. Methodology

The Huggingface Transformers [21] framework is a library that provides variations of state-of-the-art NLP pre-trained, transformer-based models that can be fine-tuned on particular data sets and tasks. Token and Sequence classification can be used for NER and Sentiment Analysis respectively, while models with a span classification heads on top can be used for Question Answering. It was selected as the framework to accomplish the goal of this work by using the BERT base, XL-Net base, RoBERTa base and ELECTRA base discriminators which are considered the most sophisticated models available.

A. Data Set

Data preparation and pre-processing are critical parts of a thorough analysis. To accomplish this goal, a solid gold set containing annotation for the particular sort of technical entities is crucial. Hand-annotated corpora are hard to create because they require domain expertise and a lot of time.

For the purposes of this work, the DNRTI data set [7] was selected as it is the most comprehensive, detailed, and coherent cybersecurity- related data set currently available, and thus, can lead to solid, strong and consistent insights regarding Cyber Threat Intelligence mentions on unseen text. In particular, DNRTI is a large data set released in 2020 that contains $175,220$ words, annotated in 13 different entity categories using the IOB/BIO annotation scheme [22]. According to this scheme, every token of a sentence is labeled as (i) **B-label** (e.g. 'B-HackOrg') if the token constitutes the beginning of a named entity, (ii) **I-label** (e.g. 'I-HackOrg') if it is inside a named entity, but not positioned first, or (iii) **O-label** ('O') if it is not part of a named entity, i.e. it is outside of it.

The released version of the data set includes pre-fixed training/validation and test sets. Following some minor cor-

[1] https://nvd.nist.gov/

[2] https://us-cert.cisa.gov/ncas/bulletins

[3] https://www.metasploit.com/

rections[4], the final format of the provided training/validation sets consists of $157,945$ tokens overall ($9,180$ unique), with $140,526$ tokens in the training set and $17,602$ tokens in the validation set. The 'O' class tokens are the majority and number $124,739$ tokens, while the 'B' class and the 'I' class number $20,143$ and $11,254$ tokens respectively.

Consider for instance the 'Hacking Organisations' entity category. The pre-fixed training and validation parts of the data set contain $4,963$ appearances of tokens referring to 'Hacking Organisations', such as 'Cobalt', 'LuckyMouse', and 'OceanLotus'; in particular, $3,845$ 'B-HackOrg' and $1,118$ 'I-HackOrg' entities appear. Out of these $3,845$ 'B-HackOrg' tokens, 477 are unique. According to the context, 140 unique 'HackOrg' single tokens of the validation set have 283 different uses (labels) in the training set. This indicates that the annotation tags of the words are modified according to the context. Moreover, there are also 4308 tokens referring to malware names ('B-Tool' and 'I-Tool') like PlugX or NetTraveler among other categories.

B. *Pre-processing*

Every model is accompanied by its own tokenizer and its own vocabulary, which is essential so that every token can be mapped with a unique code. In this work, the limit of the length of sequences is set to 120 sub-words. The largest lengths of sequences of tokens in our data were in the interval 99 to 125 depending on the different tokenizers, while sequence length up to 512 is supported by BERT, while it is unlimited for XLNet. For example, for BERT base the maximum length of sentences was 115 tokens, the mean length was 46.68 tokens with a standard deviation 27.65. For XLNet the values were 49.39 and 28.72 respectively. Longer sequences were truncated and shorter sequences were padded (post-tokens) to reach the fixed defined size.

The tokenizers and the models are usually offered in cased and uncased variations. Despite the fact that the uncased variants of the models are widely considered to perform better, our strategy was oriented towards focusing on case- sensitive versions which was deemed more suitable for cybersecurity-related Named Entity Recognition. For instance, capitalization can be indicative of phrases referring to a ransomware name and is usually present into the involved words, especially those coming from logs. Hence, it was decided these domain-specific to be taken into consideration. Tokenizers also split complex words into pieces so as to be identified by the vocabulary. To handle this issue, corresponding labels had to be multiplied accordingly during this splitting process.

Finally, Neural Network inputs do not constitute text but numerical values. The tokenizers have core features that convert input tokens to ids (indices numbers), encoding representations according to their vocabulary. IBO/BIO labels are also modified to integer numbers. At this stage special tokens are added. Attention masks are also created as an additional input array to the input ids and labels. They follow their corresponding variables at the training/validation set separation. They are float numbers as signals that inform the model if the respective token is an actual one (1.0) or a padding product to ignore (0.0). Then data loaders need to be set. During training, data were shuffled by a random sampler, while during validation, data were loaded sequentially using a sequential sampler.

C. *Fine-tuning*

The fine-tuning process of BERT, XLNet, and similar transformer-based models includes the insertion of a single extra output layer on top depending on the particular task. For the named entity recognition, a linear classification layer is added. In order to tackle the requirements of the task at hand, the most efficient approach is the token-level classification. The weights of the last hidden state of the networks, derived from the pre-training pipeline, are passed as inputs to the token-level classifier. "Bert For Token Classification" and "XLNet For Token Classification" are two examples of appropriate modifications of the models, in order to make configurations that suit our purpose.

On the other hand, modeling for Sequence Classification classifies entire sentences, in which the desired entities coexist with the adjacent words and it would not be indicative. Thus, sentences containing a desired entity had to be annotated by hand with a '1', otherwise with a '0'. Sequence Classification when applied for NER (e.g., BERT For Sequence Classification) recognises whole sentences instead of words (tokens and entities). The pooler performs specific functions to reduce the dimensionality of the network and the dropout ignores units (neurons) during the training stage to prevent overfitting. A classifier is a linear upper layer.

IV. Experimental Setup

In this work, the BERT base, XLNet base, RoBERTa base and ELECTRA base discriminators, optimised with appropriate hyper-parameters, were selected to be configured and evaluated for NER on CTI. These methods were implemented using PyTorch and evaluationexperiments were set up as follows.

A. *Dataset: Fixed & Custom splits*

As discussed, the DNRTI dataset is released with pre-fixed sets for training and validation, as well as a holdout test set. The initial train/validation split is roughly 89%-11%, while the train/test ratio is also roughly 89%-11%. Table I provides information on the number of sentences and tokens in the pre-fixed sets; the listed percentage reflects the train/test ratio.

Our analysis indicated that 46 complex entities (i.e., entities comprising two or more tokens) that exist in the training set, are repeated with the same annotation in 43 out of the 664 sentences of the test set. In order to avoid over- estimation at the evaluation stage and misleadingly high accuracy, these 43 sentences had to be moved to the training set, resulting in a zero-shot conversion of the test set. As a result, both the

[4] It was observed that the released version of the data set contained some bad lines and typos; to address this,any missing values were removed and the typos were corrected, while some defective entity names were also replaced with the correct ones.

TABLE I
NUMBER OF SENTENCES AND TOKENS IN THE TRAINING, VALIDATION , AND TEST SETS OF THE THREE CONSIDERED DATASET SPLITS

	Fixed split (89%-11%)		**Custom split I** **(89.7%-10.3%)**		**Custom split II** **(82%-18%)**	
	# sent.	**#tok.**	**# sent.**	**#tok.**	**# sent.**	**#tok.**
Training	5261	140345	5304	141777	4876	129660
Validation	662	17600	662	17600	662	17600
Test	664	17715	621	16283	1039	28400

TABLE II
PARAMETERS OF THE FINE-TUNING FOR BERT BASE, XLNET, ROBERTA AND ELECTRA.

Number of Epochs	4
Lower Case	False
Learning Rate	1e-4
Batch Size	16
Sentence Length	120
Epsilon	1e-12
Max Gradient Norm	1.0

training and test sets were slightly revised in comparison to the original data set resulting in the *Custom split I* listed in Table I; in this case, the train/test ratio is 89.7%-10.3%.

Moreover, after concatenating the fixed training and test sets, alternate train/test splits were also tested, because they were considered to be more stable, accurate, and consistent. In particular, after an 80%-20% split was performed, we moved the sentences of the test set that contained repeated entities to the training set (similarly to above), resulting in the *Custom split II* (Table I) where the train/test ratio is 82%-18%. In this case, a 5-fold cross validation was applied at the evaluation.

Last, it was observed that some types of entities were similar to each other and therefore could be merged without loss of specificity. The 'Tool' category that contains mostly malware names entities (e.g., NetTraveler, Triton) was merged with the 'SamFile' category that contained malware files entities (e.g., avemaria, balkandoor). The 'Idus' and 'Org' categories were also merged, as they contain similar entities related to organisations like government, universities, or corporations in various sectors. This merging was performed on the fixed split and resulted in 11 (compared to 13) entity categories.

B. Annotation schemes

Every token of a sentence is associated with a label in a particular format in order to be recognised by the transformer-based models. Apart from the IOB/BIO format (discussed above), we also employed the BILOU format. BILOU is similar to IOB/BIO, but every individual unit-length relevant token is notated as *U-label* (Unique, e.g. 'U_HackOrg'), while the last I-label of a chunk is tagged as *L-label* (Last, e.g. 'L_HackOrg'). We converted our data set to the specific format and also conducted experiments with the increased number of labels. For example, now the 4963 'Hacking Organisations' tokens were allocated as 1033 'B-HackOrg', 115 'B-HackOrg', 1003 'L-HackOrg', and 2812 'U-HackOrg'.

C. Hyper-parameters

Deep learning models comprise many hyper-parameters that need to be tuned in the validation set so as to achieve the optimal outcome. Table II summarizes the hyper-parameters used during testing. The AdamW optimiser was selected and weight decay was chosen as a regularization technique to penalize weight matrices of the nodes; regularization adds a term to the loss function that penalizes overfitting. A batch size of either 16 or 32 is recommended in the BERT paper [10], where also a number of epochs between 2 and 4 during the fine-tuning stage is considered sufficient since additional epochs add nothing beyond a slight gain.

D. Evaluation

The main evaluation metrics taken into account to compare models and measure their performance were Precision, Recall, and F1 score of predicted tokens and entities. These metrics are reported for each of the different 'B' and 'I' classes (BIO classification), as well as their average values (i.e., unweighted mean). The 'O' class is of no importance to the task, since it designates simple entities, thus it was not considered in the results. Accuracy is not that considerable because the majority 'O' class is included and most models predict it quite well.

Except for the token level, the recognition at entity level was examined too. For example, "The United States of America" is correctly identified if every token it is composed of is predicted correctly. A rational and fair policy is to allow only the perfect exact matches of a full entity to be taken into consideration. Two entities with two or more words, where B-label or I-label tags have been assigned to each of them, are considered equal only when all their internal equivalent tags match one another exactly (thus deemed as an exact match) [23]. This policy was adopted and applied at the evaluation stage between the predicted entities and their true validation mapping labels.

V. RESULTS AND DISCUSSION

This section presents the experimental results derived through a series of steps where the employed models were fine-tuned, trained, and evaluated in the downstream NER task.

A. Experiments on the Fixed DNRTI Split

Table III shows the results of the analysis for the various models on the fixed test set, compared to the models applied in the DNRTI paper; these results were acquired directly from the paper [7]. It is apparent that modern models considerably outperform previous approaches. Their attention-based architecture permits them to better exploit the available information found in the context of the whole sentence, learning simultaneously from both directions. Moreover, concerning the best model for single token level (XLNet), Table V provides the Precision, Recall and F1-score for every class except for the 'O' class and their macro average. With regards to entity level results, Table VI presents the exact and partial match metrics for the complex entities (groups of tokens) of all used models.

Table IV shows the results for the two best performing models using the BILOU annotation format, instead of the

TABLE III
EVALUATION RESULTS (AVERAGE: 'MACRO') FOR DIFFERENT SOTA MODELS USING DNRTI CORPUS AND IOB SCHEME COMPARED TO THE PROPOSED MODELS FROM DNRTI [7] AS A BASELINE.

Models	F1- Score	Precision	Recall
BERT base	0.875	0.852	0.902
ELECTRA base	0.831	0.809	0.858
RoBERTa base	0.842	0.806	0.886
XLNet base	**0.883**	**0.863**	**0.906**
LSTM [7]	0.671	-	-
BiLSTM [7]	0.713	-	-

TABLE IV
EVALUATION RESULTS FOR DIFFERENT SOTA MODELS USING DNRTI CORPUS AND BILOU SCHEME.

Models	F1- Score	Precision	Recall
BERT base	0.838	0.818	**0.871**
XLNet base	**0.846**	**0.833**	0.870

TABLE V
DETAILED RESULTS FOR THE MOST EFFECTIVE MODEL USING DNRTI CORPUS AND IOB SCHEME.

XLNet			
Models	**F1- Score**	**Precision**	**Recall**
B-HackOrg	0.88	0.85	0.90
I-HackOrg	0.81	0.78	0.85
B-OffAct	0.86	0.90	0.83
I-OffAct	0.88	0.90	0.86
B-Features	0.95	0.91	0.99
I-Features	0.94	0.89	1.00
B-Purp	0.88	0.84	0.92
I-Purp	0.86	0.78	0.96
B-Way	0.97	0.95	0.99
I-Way	0.96	0.93	0.99
B-SecTeam	0.96	0.97	0.94
I-SecTeam	0.81	0.72	0.93
B-SamFile	0.91	0.92	0.90
I-SamFile	0.90	0.91	0.89
B-Idus	0.91	0.91	0.91
I-Idus	0.84	0.83	0.85
B-Tool	0.92	0.91	0.94
I-Tool	0.83	0.81	0.84
B-Area	0.90	0.88	0.92
I-Area	0.81	0.78	0.83
B-Time	0.831	0.809	0.858
I-Time	0.89	0.84	0.94
B-Org	0.75	0.75	0.76
I-Org	0.78	0.72	0.85
B-Exp	0.99	0.99	1.00
I-Exp	0.98	0.97	1.00

TABLE VI
ENTITY- BASED EVALUATION RESULTS WHERE ONLY THE 948 TEST SET'S COMPLEX ENTITIES WERE EXAMINED AS GROUPS OF TOKENS FOR DIFFERENT SOTA MODELS USING DNRTI CORPUS AND IOB SCHEME.

Models	Exact Match	Partial Match
BERT base	0.93	0.07
ELECTRA base	0.91	0.09
RoBERTa base	0.90	0.10
XLNet base	0.92	0.08

original IOB/BIO one. Contrary to findings reported in similar works [24], this annotation schema failed to ameliorate results in the current task, actually performing worse than the IOB.

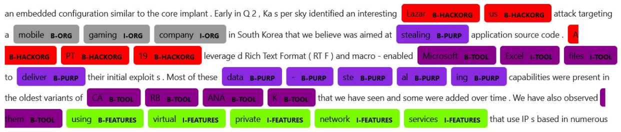
Overall, the BERT base and XLNet base models had comparable performance in all metrics and achieved the best F1, Precision, and Recall scores at token and entity level on the DNRTI corpus. The results are balanced, comparable and very close and outscored the best baseline model (BiLSTM) F1 score by 17%. Moreover, XLNet outperformed, even if it was just marginally higher, every other model at almost any level. It has achieved F1 just over 88% for B and I classes' average, although it seemed to be slightly lower than the BERT base for the exact entity match predictions of complex entities on the DNRTI test set. XLNet has also achieved great individual B and I tags' F1 scores, as well the best mean Precision score and Recall score. As an indicative example, the best baseline model achieves B-HackOrg and B-Tool F1 score 74% and 60%, respectively, while the XLNet 88% and 92%. BERT base is consistently ranked second, scoring equivalent, but slightly inferior results. The RoBERTa base follows above ELECTRA base that also performed well. All the models broadly predict the 'B' class of most categories better, even to a small extent, than the 'I' equivalent class.

B. Experiments on Custom Splits

Next, the models were tested on the zero-shot unseen sentences of the Custom splits. The sentences were fed to the fine-tuned models and the CTI-related candidate entities were identified. The sentences were prepared and encoded (tokenized and special tokens appended to them) to be fed as new inputs. Examples are given in Figures 1 and 2. he results presented in Table VII, indeed show a slight over-estimation of the result values in comparison to the initial fixed one.

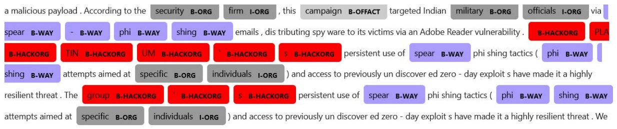
C. Experiments on Reduced Number of Labels

Finally, in Table VIII, the experiments conducted with reduced classes of entities, as an abstraction in order to decrease the complexity caused by overlapped tags, seem to fulfill their purpose and achieve F1 scores just under 90%. The information presented in this table concerns the two best performing models; further testing was not expanded to all considered models due to time constraints.



an embedded configuration similar to the core implant . Early in Q 2 , Ka s per sky identified an interesting Lazar B-HACKORG us B-HACKORG attack targeting
a mobile B-ORG gaming I-ORG company I-ORG in South Korea that we believe was aimed at stealing B-PURP application source code . A
B-HACKORG PT B-HACKORG 19 B-HACKORG leverage d Rich Text Format (RT F) and macro - enabled Microsoft B-TOOL Excel I-TOOL files I-TOOL
to deliver B-PURP their initial exploit s . Most of these data B-PURP - B-PURP ste B-PURP al B-PURP ing B-PURP capabilities were present in
the oldest variants of CA B-TOOL RB B-TOOL ANA B-TOOL K B-TOOL that we have seen and some were added over time . We have also observed
them B-TOOL using B-FEATURES virtual I-FEATURES private I-FEATURES network I-FEATURES services I-FEATURES that use IP s based in numerous

Fig. 1. Entities identification example by XLNet tested on zero-shot sentences.



a malicious payload . According to the security B-ORG firm I-ORG , this campaign B-OFFACT targeted Indian military B-ORG officials I-ORG via
spear B-WAY - B-WAY phi B-WAY shing B-WAY emails , dis tributing spy ware to its victims via an Adobe Reader vulnerability . B-HACKORG PLA
B-HACKORG TIN B-HACKORG UM B-HACKORG ' B-HACKORG s B-HACKORG persistent use of spear B-WAY phi shing tactics (phi B-WAY
shing B-WAY attempts aimed at specific B-ORG individuals I-ORG) and access to previously un discover ed zero - day exploit s have made it a highly
resilient threat . The group B-HACKORG ' B-HACKORG s B-HACKORG persistent use of spear B-WAY phi shing tactics (phi B-WAY shing B-WAY
attempts aimed at specific B-ORG individuals I-ORG) and access to previously un discover ed zero - day exploit s have made it a highly resilient threat . We

Fig. 2. Entities identification example by XLNet tested on zero-shot sentences.

TABLE VII
EVALUATION RESULTS FOR THE BEST PERFORMING MODELS USING THE CUSTOM SPLITS, WITHOUT THE OVER-ESTIMATION FACTOR OF REPEATED COMPLEX ENTITIES ON BOTH TRAIN AND TEST SETS, ON DNRTI CORPUS WITH IOB SCHEME.

Models	F1- Score	Precision	Recall
	Custom split I (89.7%- 10.3%)		
BERT base	0.868	0.842	0.899
XLNet base	**0.872**	**0.842**	**0.907**
	Custom split II (82%- 18%)		
BERT base	0.868	0.845	0.897
XLNet base	**0.872**	**0.851**	**0.898**

TABLE VIII
EVALUATION RESULTS FOR BEST PERFORMING MODELS USING THE FIXED SPLIT BUT WITH 11 LABELS (RATHER THAN 13) PLUS THE 'O' ON DNRTI CORPUS WITH IOB SCHEME.

Models	F1-Score	Precision	Recall
BERT base	**0.899**	**0.874**	0.928
XLNet base	0.898	0.870	**0.929**

VI. CONCLUSIONS

This work investigated the effectiveness of transformers-based models for the extraction of Named Entities on CTI context and showed the improvements achieved in performance compared to deep neural network methods previously applied on the DNRTI dataset. Further investigation is required in order to clarify the influence that a more extensive annotated corpus or an enhancement of DNRTI could have and how steeply the evaluation results could be affected and both the training and validation errors/ losses could drop. Moreover, Relation Extraction will be applied as the next step in order to gain inferences about possible relations between the entities.

ACKNOWLEDGMENT

This work was supported by the FORESIGHT (H2020 833673) and ECHO (H2020 830943) projects, funded by the European Commission.

REFERENCES

[1] S. Skylaki, A. Oskooei, O. Bari, N. Herger, and Z. Kriegman, "Named entity recognition in the legal domain using a pointer generator network," *ArXiv*, vol. abs/2012.09936, 2020.

[2] E. Leitner, G. Rehm, and J. Moreno-Schneider, "Fine-grained named entity recognition in legal documents," in *International Conference on Semantic Systems*. Springer, 2019, pp. 272–287.

[3] A. Ritter, S. Clark, O. Etzioni *et al.*, "Named entity recognition in tweets: an experimental study," in *Proc. the 2011 conference on empirical methods in natural language processing*, 2011, pp. 1524–1534.

[4] L. Derczynski, D. Maynard, G. Rizzo, M. Van Erp, G. Gorrell, R. Troncy, J. Petrak, and K. Bontcheva, "Analysis of named entity recognition and linking for tweets," *Information Processing & Management*, vol. 51, no. 2, pp. 32–49, 2015.

[5] T. Rocktäschel, M. Weidlich, and U. Leser, "Chemspot: a hybrid system for chemical named entity recognition," *Bioinformatics*, vol. 28, no. 12, pp. 1633–1640, 2012.

[6] S. Atdağ and V. Labatut, "A comparison of named entity recognition tools applied to biographical texts," in *2nd International conference on systems and computer science*. IEEE, 2013, pp. 228–233.

[7] X. Wang, X. Liu, S. Ao, N. Li, Z. Jiang, Z. Xu, Z. Xiong, M. Xiong, and X. Zhang, "Dnrti: A large-scale dataset for named entity recognition in threat intelligence," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020, pp. 1842–1848.

[8] V. Yadav and S. Bethard, "A survey on recent advances in named entity recognition from deep learning models," in *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 2145–2158. [Online]. Available: https://www.aclweb.org/anthology/C18-1182

[9] J. Giorgi and G. Bader, "Transfer learning for biomedical named entity recognition with neural networks," *Bioinformatics (Oxford, England)*, vol. 34, 06 2018.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[11] S. Wu and M. Dredze, "Beto, bentz, becas: The surprising cross-lingual effectiveness of bert," 04 2019.

[12] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. of NAACL*, 2018.

[13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *ArXiv*, vol. abs/1907.11692, 2019.

[14] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 5753–5763. [Online]. Available: http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding.pdf

[15] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=r1xMH1BtvB

[16] P. Ma, B. Jiang, Z. Lu, N. Li, and Z. Jiang, "Cybersecurity named entity recognition using bidirectional long short-term memory with conditional random fields," *Tsinghua Science and Technology*, vol. 26, no. 3, pp. 259–265, 2020.

[17] A. Joshi, R. Lal, T. Finin, and A. Joshi, "Extracting cybersecurity related linked data from text," in *2013 IEEE Seventh International Conference on Semantic Computing*. IEEE, 2013, pp. 252–259.

[18] H. Gasmi, A. Bouras, and J. Laval, "Lstm recurrent neural networks for cybersecurity named entity recognition," *ICSEA*, vol. 11, p. 2018, 2018.

[19] R. A. Bridges, C. L. Jones, M. D. Iannacone, K. M. Testa, and J. R. Goodall, "Automatic labeling for entity extraction in cyber security," *arXiv preprint arXiv:1308.4941*, 2013.

[20] S. K. Lim, A. O. Muis, W. Lu, and C. H. Ong, "MalwareTextDB: A database for annotated malware articles," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1557–1567. [Online]. Available: https://www.aclweb.org/anthology/P17-1143

[21] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Huggingface's transformers: State-of-the-art natural language processing," *ArXiv*, vol. abs/1910.03771, 2019.

[22] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 260–270. [Online]. Available: https://www.aclweb.org/anthology/N16-1030

[23] A. Symeonidou, "Transfer learning for biomedical named entity recognition with biobert," 06 2019.

[24] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, 2009, pp. 147–155.