Context-Augmented Key Phrase Extraction from Short Texts for Cyber Threat Intelligence Tasks

Avishek Bose
Department of Computer Science
Kansas State University
Manhattan, Kansas, USA
bosea@ornl.gov

*Huichen Yang
Department of Computer Science
Kansas State University
Manhattan, Kansas, USA
huichen@ksu.edu

*Marissa Shivers
Department of Computer Science
Kansas State University
Manhattan, Kansas, USA
mxs@ksu.edu

Ahat Orazgeldiyev
Department of Industrial and Manufacturing Systems Engineering
Kansas State University
Manhattan, Kansas, USA
ahato@ksu.edu

William H. Hsu
Department of Computer Science
Kansas State University
Manhattan, Kansas, USA
bhsu@ksu.edu

2023 IEEE International Conference on Intelligence and Security Informatics (ISI) | 979-8-3503-3773-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISI58743.2023.10297274

***Abstract*—In this paper, we address contextual limitations of current deep learning-based and heuristic key phrase extraction tools as applied to the domain of cybersecurity. To address these limitations, we develop a hybrid system that augments state-of-the-art (SOTA) transformers for the task of key phrase sequence labeling, using a novel set of part-of-speech (POS) and role-aware tagging rules to generate fine-grained tag sequences from short text corpora. Next, we fine-tune multiple SOTA deep learning (DL) language model (LM) architectures to these transformed sequences. We then evaluate the architectures by measuring the outcomes from respective LMs to select the best-performing underlying transformers for extracting cybersecurity key phrases. This new ensemble achieves very significant predictive gains over SOTA baselines on general cybersecurity corpora, such as F1 scores at least 25% higher than hybrid SOTA transformers fine-tuned using baseline tagging rules on the generic corpus, with a much less significant tradeoff (of less than 5% in F1) on a vulnerability-specific corpus.**

***Index Terms*—Tagging rules, sequence labeling, BERT, BiLSTM, ROUGE score, context, key phrase, cyber threat**

I. Introduction

Extracting cybersecurity-relevant informative tokens from text documents of open-source domains such as CVE and NVD reports and Twitter and then mapping them to respective labels as a sequence labeling task has a particular significance to cyber-security researchers to detect emerging cyber-attacks.Key phrase extraction from such text corpora has been unquestionably prolific but remains highly challenging because open-source short texts have a peculiar lack of structure. Named Entity (NE) Extraction/identification as an information extraction process is used as a pipeline in many other applications [1], [2] of Cyber Threat Intelligence (CTI), mapping a sequence of text tokens to predefined classes. However, existing NE extraction tools can usually only find NEs that are nouns or conjunctions of nouns. Moreover, raw text that has vital information conveyed by corresponding tokens does not qualify as NEs because those tokens may have been tagged as other parts of speech, are wrongly formed, or derived from different languages. Thus, existing NE extraction/identification methods often cannot identify crucial CTI information, triggering the need to develop more robust techniques than current NE extraction processes.

Deep learning neural networks such as sequence-to-sequence models and transformers have predominated among methods for key phrase extraction/identification from text [3] because of their domain-independent adaptability across fields. However, DL methods have trouble extracting rare entities, acronyms, and abbreviations and are limited in learning from text documents relevant to cybersecurity if they were different lengths because information from short texts is condensed, whereas descriptive reports spread across pages. On the other hand, incorporating extraction tagging rules with deep learning (DL) provides the most convenience in a model.

Effective and insightful key phrase extraction from raw texts is not straightforward because contextual information is lacking; key information may remain unobserved by even some robust key phrase identification processes that are either DL-based or heuristic-based. DL-based LM architectures naturally facilitate context-learning of entities by focusing semantic structure of inputting text documents. On the other hand, the semantic role labeling (SRL) process can contextualize text documents based on the central verb. Therefore, a potential solution may be to unify these two context-aware modules to establish a hybrid system that adopts fruitful concepts from both domains. Thus, considering the contextual limitations of inputting text data and the huge effort required in annotating text corpora for sequence labeling tasks, we developed a set of SRL-powered generic extraction tagging rules as a module. We fed text documents into this module to get a fine-grained, tagged sequence from the inputting documents. Next, we adopted a DL-based sequence labeling task of NE extraction [4] along with generalized extraction tagging rules to formulate a hybrid key phrase identification method that supports domain transferability.

Our contributions are as follows:

1) We developed a set of generalized extraction tagging rules for key phrase extraction;
2) We fed the tagged data set to some prominent DL-based transformers and statistical language model architectures to determine the suitable learning architectures for specific use cases;
3) We validated the applicability of extraction tagging rules by applying the ROUGE metric calculation between a sample tagged data set and its corresponding annotated

Author with * mark contributed equally

data set;

4) We designed, implemented, and experimented with an end-to-end framework that combined the developed extraction tagging module and learning framework module which achieved better generalization performance;
5) We validated our hypothesis by demonstrating context information from external sources (e,g. SRL tagging) enhances transformers' performance through our experiments.

II. Related Work

This section gives information on earlier approaches to key phrase extraction and NE extraction from text documents specifically in the cybersecurity domain and that apply heuristic, statistical, ML/DL-based techniques.

Statistical Learning Approaches as a basic technique of keyphrase extraction, statistical Learning approaches compute the probabilities of a sequence of tokens in a text that enables labeling the tokens to their suitable particular classes. Examples of such approaches are Support Vector Machine (SVM) frameworks [5] or Conditional Random Field (CRF) frameworks [6] for extracting entity and context information about security vulnerabilities and attacks. On the other hand, the scarcity of labeled open-source text data sets introduced rule-based unsupervised approaches to extract NEs [7] that require further domain-specific knowledge. In addition, ML architectures [8] such as vanilla BERT for NE extraction are limited because they also depend upon external knowledge bases. More recently, NLP and cybersecurity researchers have used DL-based approaches applied to various DL architectures for key phrase extraction because DL is feature-independent. BiLSTM-CRF settings [9] and its descendant approaches such as [10] for NE extraction have become state-of-the-art (SOTA) techniques because they effectively apply word context as a primary step in token sequence tagging. The implications associated with extracting cybersecurity key phrases [11] can be well addressed using a combination of DL methods [12] that leverage the connection between the character vector model and the word vector model with feature templates. The attention mechanism [13] and its combination with feature templates [14] facilitate extracting cyber security-relevant rare tokens from a text corpus. A joint module of a DL model and domain dictionary for generalized applicability and correctness for security entity extraction has been proposed [15], and in another study [2], a framework combined Stanford NER and Regular Expressions to detect cybersecurity NEs. Other research [16] uses vector space representations. DL architectures such as LSTM and BERT, have been frequently used in cybersecurity key phrase extraction tasks [17] that collect features locally and globally from the corpus.

This earlier research has shown incremental improvement in cybersecurity key phrases and entity extraction, but they did not provide more effective results through combining both context augmentation and DL techniques for further improvement. We have results of different, open source, key phrase extraction tools that are very popular (see Table I).

III. Cybersecurity Keyphrase Identification Framework

Despite having many common ideas and processes, the difference between NE extraction and Key Phrase extraction lies in their purposes within a particular domain. Unlike NE extraction, which focuses only on extracting and identifying noun phrases, the key phrase extraction process tries to extract all key information that makes it appropriate for downstream CTI tasks. Nearly all generic NE extraction methods fail to detect text tokens mentioning emerging cyber threats and foreign-language software entities because it is nearly impossible to assign a tag to each valid NE. On the other hand, key phrase extraction is important in the CTI domain because it accumulates subtle information from both structured long text and unstructured short text in which non-noun tokens also contain crucial information. Moreover, the purpose of CTI information extraction, even characterizing entity types, has significant value in CTI, although obtaining this vital information is a priority. Considering the requirement of the current needs in CTI, we designed a generalized set of extraction tagging rules to obtain key phrases by keeping their previously assigned types. Figure 1 presents our implemented framework for key phrase identification, which consisted of six steps.

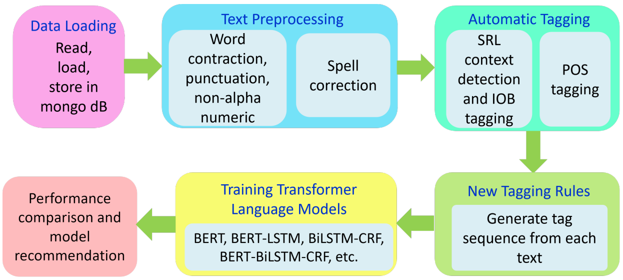
Data Loading
Read, load, store in mongo dB
Text Preprocessing
Word contraction, punctuation, non-alpha numeric
Spell correction
Automatic Tagging
SRL context detection and IOB tagging
POS tagging
New Tagging Rules
Generate tag sequence from each text
Training Transformer Language Models
BERT, BERT-LSTM, BiLSTM-CRF, BERT-BiLSTM-CRF, etc.
Performance comparison and model recommendation

Fig. 1. Workflow diagram of new context-augmented key phrase extractor consists of six major steps.

A. *Incorporating Heuristic Rules*

1) Preparing Initial Data: To create primary test beds for applying extractive tagging rules we developed, we first apply part of speech (POS) tagging and semantic role labeling (SRL) to every text document from each of the benchmark corpora [18], [19]. POS tagging defines micro-level feature information for each token present in a text. On the other hand, semantic role labeling (SRL) defines verbs as predicates and associates other tokens with semantic roles represented as class-typed arguments. In other words, SRL maps the verb or predicate arguments as functions and other tokens to specific relations that correspond to a specific function in a given sentence. NLTK's POS tagger tags each token in an English text as the following CC: coordinating conjunction, CD: cardinal digit, DT: determiner, FW: foreign word, IN: preposition/subordinating conjunction, JJ+: adjective, MD: modal, NN+: nouns, RB+: adverbs, VB+: verbs. The symbol + showed all types of similar POS tags are included. On the other hand, SRL-produced tagging includes the following three terms: (i) numbered arguments such as ARG0, ARG1, etc. to plot a middle course among possible different theoretical analyses and ensure consistent annotation, (ii) IOB tagging to locate their position according to the verb, (iii) mnemonic ArgM modifier tags are LOC: location, CAU: cause, TMP: time, PNC: purpose, NEG: negation marker, DIR: direction, etc. SRL usually generates two to four role arguments, but as many as six can appear based on the verbs and other tokens remain adjuncts. A reference example can be shown as the following *i appreciate the love for hacking guys but let*

TABLE I

COMPARISON OF OUR DEVELOPED HYBRID SYSTEM AGAINST THE CURRENT TOOLS FOR IDENTIFYING KEY PHRASES FROM A SHORT TEXT

Clean Structured Text	A cyber attack in CSEDU has been committed by North Korean hackers	
Existing Tools	**Different test case of the given text**	**Remarks**
Stanford NER	A cyber attack in csedu has been committed by north korea hackers	Stanford NER failed to detect any key phrase from the given lower case letter transformed text
	A cyber attack in Csedu has been committed by north korean hackers	Stanford NER detected only part of key phrase after making the token upper case *csedu* ⟶ Csedu
	A cyber attack in Csedu has been committed by North Korea Hackers	Stanford NER detected two key phrases only after nounifying and making the tokens upper case *csedu* ⟶ Csedu and *north korean* ⟶ North Korea
Spacy	A cyber attack in CSEDU has been committed by north korean hackers	Spacy detected only one after making the token upper case csedu ⟶ CSEDU
	A cyber attack in csedu has been committed by North Korean Hackers	Spacy failed to detect any key phrase from the given lower case letter transformed text
	A cyber attack in CSEDU has been committed by North Korea Hackers	Spacy failed to detect any part of a certain key phrase *North Korea Hackers* even though it was in upper case and nounified
	A cyber attack in CSEDU has been committed by North Korean Hackers	Spacy detected two key phrases, one after making the first NE token upper case; the second was only partly detected
Our developed tagging rules	A cyber attack in csedu has been committed by north korean hackers	All the potential key phrases are tagged after modifications
Along with BERT-BiLSTM-CRF Model	A cyber attack in CSEDU has been committed by North Korean hackers	Indentification of potential key phrases is compatible to the tagging rules result even after modifications

[ARG0: us] [ARGM-NEG: not] [V: dos] [ARG1: our own fun stream rip techno therapy] [ARG2: an above beyond].

However, more than one verb in a sentence can cause ambiguity because of multiple SRL assignments. To solve this problem, we exclude all light verbs from being predicates and prioritized the verb written at the end of a sentence as a predicate. At the end of this step, we have POS tags and role labels for each token present in a given text.

2) Applying Extractive Tagging Rules: A crucial component of our new keyphrase extraction system is a set of tagging rules we have developed that maps POS and SRL-tagged sequences (the output of the steps discussed in the above subsection) to the input of our transformer-based language model. The set of tagging rules is designed according to the generic syntactic and semantic information of sentences which does not include any cybersecurity-specific task or its associated information but works better for both structured and unstructured texts. These rules are formulated empirically to synthesize the intrinsic structure of sentences obtained from POS and SRL tags. These also avoid using overconstrained, domain-specific pattern-matching techniques such as gazetteers that are too brittle to adapt to new domains such as different cybersecurity-related domains and thus omit key role-specific information in context. For example, for the text document *A cyber attack in CSEDU has been committed by North Korean hackers* SRL tagged representation is A cyber attack in CSEDU[B-ARG1, I-ARG1, I-ARG1, I-ARG1, I-ARG1] has[O] been[O] committed[B-V] by North Korean hackers[B-ARG0, I-ARG1, I-ARG1, I-ARG1]. Here, the role verb *committed* created the relation context that rule-based tagger could use with a POS tag to extract the key phrases by tagging respective tokens.

SRL synthesizes a text document into multiple contexts (examples given in Section III.A.1 and Section III.A.2) according to the roles based on the central verb of the text document. Our rules mentioned in Section III.A.2 consider the position of different arguments in the contexts and extract noun phrases from them. If informative tokens in the context map to other POS except for nouns, the rules transform them into Noun phrases. The resulting tagging sequence is mapped to the respective token where a tag determines a token's presence in the set of key phrases. Unlike existing tagging rules (too brittle for generic Benchmark 2 [19] data set), that attempt to only tag specific CTI text tokens, our new rules consider the POS tag sequence of tokens inside the context (generated by SRL for tagging text tokens of a text document) for key phrase extraction tasks. Our new set of rules does not constrain itself to a particular domain data set(Benchmark 1 [18]), it supports cross-domain transfer. In the following itemized points, we describe the set of rules that we design for tagging key phrases.

For any set of tokens T and SRL argument tags A, the set of all functions $f_{SRL}(T)$:T→A is denoted as A = $f_{SRL}(T)$ = *SRL(T)*

For any set of tokens T and POS tag P, the set of all functions $f_{POS}(T)$:T→P is denoted as P = $f_{POS}(T)$ = *POS(T)*

For any set of tokens T and Nounified token N, the set of all functions $f_{NOUN}(T)$:T→N is denoted as P = $f_{NOUN}(T)$ = *NOUN(T)*

1) Up to two consecutive noun phrases (NN+) where the first one is an initial argument (B-ARG) and the second one is an intermediate argument (I-ARG) preceded by a determiner (DT/IN) is jointly considered as a key phrase.

$$\begin{aligned} key_j = \{T_iT_{i+1} : i \in \{1, ..., |S_k|\}, \\ f_{SRL}(T_i) \in \{NN+\}, \\ f_{SRL}(T_{i+1}) \in \{NN+\}, \\ f_{POS}(T_i) \in \{B-ARG\}, \\ f_{POS}(T_{i+1}) \in \{I-ARG\}, \\ f_{POS}(T_{i-1}) \in \{DT/IN\}\}. \end{aligned} \tag{1}$$

2) A noun phrase (NN+) having tagged by an initial argument (I-ARG) preceded by a determiner (DT/IN) is considered a key phrase.

$$\begin{aligned} key_j = \{T_i : i \in \{1, ..., |S_k|\}, \\ f_{SRL}(T_i) \in \{NN+\}, \\ f_{POS}(T_i) \in \{I - ARG\}, \\ f_{POS}(T_{i-1}) \in \{DT/IN\}\}. \end{aligned} \tag{2}$$

3) If any determiner (DT/IN) is present just before an adjective (JJ), the adjective is an intermediate argument (I-ARG), and a noun (NN+) is located just after the adjective (JJ+), the adjective and noun jointly considered as a key phrase.

$$\begin{aligned} key_j = \{T_i T_{i+1} : i \in \{1, ..., |S_k|\}, \\ f_{SRL}(T_i) \in \{JJ+\}, \\ f_{SRL}(T_{i+1}) \in \{NN+\}, \\ f_{POS}(T_i) \in \{I - ARG\}, \\ f_{POS}(T_{i+1}) \in \{I - ARG/B - ARG\}, \\ f_{POS}(T_{i-1}) \in \{DT/IN\}\}. \end{aligned} \tag{3}$$

4) If up to two adjectives (JJ) followed by a noun phrase (NN) altogether is considered as a key phrase.

$$\begin{aligned} key_j = \{T_{i-1} T_i T_{i+1} : i \in \{1, ..., |S_k|\}, \\ f_{SRL}(T_{i-1}) \in \{JJ+\}, \\ f_{SRL}(T_i) \in \{JJ+\}, \\ f_{SRL}(T_{i+1}) \in \{NN+\}\}. \end{aligned} \tag{4}$$

5) We nounify adjective (JJ+), adverb (RB), and verb (VB) and check to see if any of the noun forms are related to cybersecurity or not.

$$\begin{aligned} key_j = \{T_i : i \in \{1, ..., |S_k|\}, \\ f_{NOUN}(T_i) \in \{NN+\}\}. \end{aligned} \tag{5}$$

6) Any foreign language token or token length of more than 3 written with alphanumeric letters are considered key phrases.

$$\begin{aligned} key_j = \{T_i : i \in \{1, ..., |S_k|\}, \\ f_{POS}(T_i) \in \{FW\} \cup |T_i| \geq 3\}. \end{aligned} \tag{6}$$

Here, *i* represents the token number of a token part of a key phrase that can be any number from 1 to $|S_k|$ where $|S_k|$ is the length of a key phrase.

DL transformer language models do not learn well if they receive highly constrained target labels against training data. Following this concept, too many type-specific CTI tags as labels would reduce the learning of LMs to predict key phrases from a text document. However, our contextualized set of rules produces generic tagging of tokens that does not limit the performance of LMs and results in a higher overall accuracy boost. So, new tagging rules are effective in tandem with transformers, generating better results, which means hybridizing two modules is transformer-friendly.

B. Adopted statistical and neural network models

1) BERT pre-trained language model: Pre-trained language models of Bidirectional Encoder Representations from Transformers (aka BERT) [**?**] show better performance than Word2Vec to generate contextualized embeddings for words present in a sentence by introducing two new ideas (i) masked language model (MLM) and next sentence prediction (NSP). These two ideas leverage BERT as an embedding to include attention-focused, multi-layer, bidirectional, and nonlinear correlation constraint layers for sequence labeling tasks. BERT has used contextual information learning and transferability, so it can be certainly used as an embedding layer for key phrase extraction as a sequence labeling task.

2) BiLSTM layer: The applicability of Bidirectional LSTM or BiLSTM in sequence processing tasks leads to further improvement that simultaneously analyzes both forward (future) and backward (past) contextual information for a given text.

3) BERT-BiLSTM-CRF model: This model [4] comes with the added advantage of BERT, which is used as a contextualized embedding layer. The BiLSTM model gets its embedding input from the BERT embedding layer, where BiLSTM's hidden states concatenate outputs from forward and backward LSTM networks together to generate a feature vector matrix for the CRF layer.

IV. Experiment and Evaluation

Block five and block six in Figure 1 present model training and experiment steps.

A. Data sets

Open-source cybersecurity data sets are significant to supporting new CTI applications. Here, we work with two benchmark data sets (i) benchmark 1 [18] and (ii) benchmark 2 [19] to train and evaluate different machine learning language models. The benchmark 1 [18] data set comprises cybersecurity information from three different sources: NVD, Metasploit, and Microsoft Security Bulletins. The data set includes the following 15 entity types, among them, ”software vendor”, ”software product”, ”software version”, ”software language”, ”vulnerability name”, “software symbol”, ”OS”, and ”hardware”, where the tagging rules are strictly constrained to tag entities from the data set. However, important cybersecurity information beyond these 15 entity types, like rare entities, non-noun entities, and misspelled entities, cannot be detected using these tagging rules. This tagging rule thus works for a structured data set such as the benchmark 1 data set but is severely limited for an unstructured data set such as the benchmark 2 data set that was generated by a crawl from Twitter using security-related keywords. The data set was manually annotated based on the relevancy of each tweet to cybersecurity. The data set initially had 21368 clean tweet texts but, of those tweets, 11111 are related to cybersecurity. We used only cybersecurity-related tweets. We used the full text of a tweet if the tweet was not quoted or retweeted and the original tweet if the tweet was retweeted or quoted. We applied our extraction tagging rule to both data sets ([18] and [19]), tagging key phrases by KeyB, KeyI, KeyO, and KeyNone. To evaluate the performance of the developed tagging rules, we annotated 400 randomly sampled tweets (discussed in subsection IV-C) by extracting all possible combinations of key phrases from the tweet texts. Then we compared tagged tweet texts from the

400-tweet sample data set against the annotated extracted key phrases from the same data set by calculating rouge scores.

B. Environment setup

We used Python, PyTorch, and NLTK to implement our source code. We used two embedding layers, word2Vec, and BERT in two different learning models, to compare performance. For the BiLSTM-CRF model, we set the maximum sequence length at 256, batch size at 8, learning rate at 0.00005, and the round of training at 20 epochs. For the BERT-related models (BERT, BERT-CRF, BERT-BiLSTM-CRF), we used 512 for maximum sequence length, 32 for batch size, 0.00005 for learning rate, and 10 epochs for training rounds. The BERT pre-trained language models were tuned as the BERT embedding layer during the training process. All models were trained with a single Nvidia A40 GPU.

C. Compatibility of Tagging Rules

To evaluate the effectiveness of the tagging rules introduced above for enhancing transformer LMs' performance and henceforth for improving CTI key phrase extraction, we fine-tune our transformer models on a training data set **without** the tagging/extraction rules as a baseline, and **with** them as an alternative treatment. Then we validate the resulting predicted sequences. Higher performance metric (e.g. Precision, Recall, F1) scores of a model obtained by fine-tuning with the tagged data set, validate the importance of contextualized tagging of texts for transformers. Conversely, lower scores obtained from the model after fine-tuning with a sample data set represent a lack of expressiveness or generalization in the model. In other words, contextualized tagging can improve transformer model learning by guiding the transformer models to learn SRL information of texts. For this analysis, we use a pre-trained BERT-base-uncased model with two different model setups (I) fine-tuning the model with the rule-tagged data set from Twitter cybersecurity corpus, and (II) fine-tuning it without tagged data set. We annotated a sample dataset of 400 tweets where we used the sample of the first 200 tweets for fine-tuning, and the remaining 200 tweets for testing. For the first model setup, we fine-tuned the whole tweet corpus and predicted the sequences for the 200 test tweet samples. For the second model setup, we use the first 200 labeled tweets for fine-tuning and the 200 test tweet samples for prediction. Table II shows the BERT-base-uncased model result for two experimental setups (batch size 4 and batch size 2) run on 15 epochs. The model performs worse when it is fine-tuned on 200 annotated tweets to predict sequence labels for 200 tweets. On the other hand, the model predicts a relatively correct sequence label for the 200 test tweet sample if this is fine-tuned on the tagged corpus data set generated by the newly introduced tagging rules.

D. Our developed tagging rule validation

The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score [20] provides a set of performance evaluation metrics for various text sequence matching tasks such as summary generation, entity matching, and machine translation. We adopted the ROUGE score to evaluate how effective the developed extraction tagging rules were in tagging key phrases compared to the baseline tagging rules on the annotated sample of benchmark 2 [19] data set. Table III provides the results of two ROUGE metrics ROUGE-N (where N=1) and Rouge-L with their respective performance indicators such as precision, recall, and F1 score for both developed and the baseline sets of rules. ROUGE-1 computed 1-gram matching performance and ROUGE-L computed the Longest Common Sub-sequence performance for the generated tagged data set and annotated data set. We can observe that our developed set of rules outperformed the baseline tagging rules by a significant margin in tagging key phrases in a text. The tagged data set was then fed as input to the machine-learning models for sequence labeling.

TABLE II
RESULTS FOR BERT-BASE-UNCASED MODEL ON TWITTER GENERAL CYBERSECURITY CORPUS

Experiment Setup	Model Setup	Prec	Rec	F1
Batch size 4	Without tagging Rules	48.52	55.75	51.88
	With Tagging Rules	**65.45**	**63.78**	**64.60**
Batch size 2	Without tagging Rules	48.45	49.83	49.13
	With Tagging Rules	**62.40**	**66.90**	**64.57**

TABLE III
ROUGE-1 AND ROUGE-L SCORE COMPARISON OF GENERATED KEY PHRASES AGAINST ANNOTATED DATA SET FOR EACH OF THE TWO SET OF RULES(OUR AND BASELINE)

ROUGE Metric	sample set from Benchmark 2 data set	Prec	Rec	F1
ROUGE-1	Tagged with developed rule	**52.19**	**35.05**	**40.09**
	Tagged with baseline rule	19.40	28.52	21.49
ROUGE-L	Tagged with developed rule	**31.96**	**27.61**	**31.96**
	Tagged with baseline rule	20.45	26.93	20.45

E. Analysis of Results

Table IV shows the performance evaluation of all models and the two different data sets ([18] and [19]). Clearly, any learning model using a generalized word2Vec embedding performs the worst for both data sets. The BERT-BiLSTM-CRF model is the best-performing model overall, so we recommend this LM be used with our generic contextualized tagging rules for sequence labeling prediction tasks. All learning models BERT, BERT-CRF, BiLSTM-CRF, and BERT-BiLSTM-CRF trained on the benchmark 2 data set tagged by our new tagging rules outperform the same models trained on the same data set tagged by the baseline tagging rules. The BERT F1 scores are 60% higher, BERT-CRF results are 41.37% higher, BiLSTM-CRF 54.48% higher, and BERT-BiLSTM-CRF 25.35% higher. On the other hand, the models trained on the benchmark 1 data set tagged by the baseline rules outperform the same models trained on the benchmark 1 data set tagged by our newly developed set of rules. The F1 scores are 6.41%, 6.05%, 11.07%, and 5.7% higher (see Table IV). Baseline tagging rules slightly outperform our newly developed rules on the benchmark 1 data set because this data set is domain-dependent and has some repetitive and similar pattern security keywords (CVE-NVD-specific cyber threat), easily identified by the highly-specific baseline rules. However, our generic set of rules worked much better for generic CTI data sets such as the benchmark 2 data set. The trade-off in using our new tagging rules includes slight performance loss in results on highly domain-dependent data sets. Attempting to improve the resulting performance on

TABLE IV
Results for baseline tagging rules and our rules tandem with different statistical and transformer-based models [4] applied to CVE & NVD-specific and general cybersecurity corpora

Corpora	Model	Prec	Rec	F1
Merged CVE & NVD-specific corpus (Benchmark 1) tagged using baseline rules	BERT	96.40	97.13	96.77
	CRF	84.00	77.00	79.00
	BERT-CRF	97.36	97.29	97.33
	BiLSTM-CRF (Word2Vec)	94.78	89.70	92.17
	BERT-BiLSTM-CRF	**97.82**	**97.37**	**97.59**
Merged CVE & NVD-specific corpus (Benchmark 1) tagged using our new rules	BERT	90.34	91.54	90.94
	CRF	88.00	86.00	87.00
	BERT-CRF	91.39	92.16	91.77
	BiLSTM-CRF (Word2Vec)	82.30	83.67	82.98
	BERT-BiLSTM-CRF	92.36	92.21	92.28
Twitter general cybersecurity corpus ((Benchmark 2) tagged using baseline rules	BERT	51.42	49.18	50.28
	CRF	70.00	54.00	60.00
	BERT-CRF	66.97	52.14	58.63
	BiLSTM-CRF (Word2Vec)	44.58	45.83	45.19
	BERT-BiLSTM-CRF	63.63	70.68	66.97
Twitter general cybersecurity corpus (Benchmark 2) tagged using our new rules	BERT	80.44	80.55	80.49
	CRF	82.00	81.00	82.00
	BERT-CRF	82.38	83.41	82.89
	BiLSTM-CRF (Word2Vec)	70.02	69.60	69.81
	BERT-BiLSTM-CRF	**84.01**	**83.89**	**83.95**

a domain-dependent data set would require the rules to be over-constrained, which in turn will reduce performance on more generic data. Because emerging cyber threat incidents are ceaseless and have drawn the most scrutiny, limiting rules to a particular domain(s) would only make room for the potential loss of security information.

V. Conclusion and Future Work

In this research, we presented a two-stage hybrid system combining a newly developed extraction tagging rule module with a recommended DL-based sequence label prediction module (BERT-BiLSTM-CRF) for key phrase extraction in the cyber-threat intelligence domain. We have also demonstrated in detailed experiments with different machine learning and statistical learning models that our claim is supported by the evaluation of their performances on two benchmark data sets tagged by both the newly developed and baseline tagging rules. The ROUGE score validated the applicability of the newly developed set of tagging rules for key phrase extraction against a small sample annotated data set. Although our system works better for generic CTI (benchmark 2) data sets, this hybrid system can be further improved for particular data sets containing entities of specific patterns. We intend to apply respective knowledge bases in developing the tagging rules. Efficiency could also be increased by adding more layers to the current DL LMs to extract more discriminative features. We will continue developing our set of rules to obtain better performance indicator scores.

References

[1] Mittal, Sudip, Prajit Kumar Das, Varish Mulwad, Anupam Joshi, and Tim Finin. "Cybertwitter: Using twitter to generate alerts for cybersecurity threats and vulnerabilities." In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 860-867. IEEE, 2016.

[2] Piplai, Aritran, Sudip Mittal, Anupam Joshi, Tim Finin, James Holt, and Richard Zak. "Creating cybersecurity knowledge graphs from malware after action reports." *IEEE Access 8 (2020)*: 211691-211703.

[3] Sirotina, Anastasiia, and Natalia Loukachevitch. "Named entity recognition in information security domain for Russian." In Proceedings of the *International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pp. 1114-1120. 2019.

[4] Yang, Huichen, and William H. Hsu. "Named Entity Recognition from Synthesis Procedural Text in Materials Science Domain with Attention-Based Approach." In *SDU@ AAAI. 2021.*

[5] Mulwad V, Li W, Joshi A, Finin T, Viswanathan K (2011) Mulwad, Varish, Wenjia Li, Anupam Joshi, Tim Finin, and Krishnamurthy Viswanathan. "Extracting information about security vulnerabilities from web text." In *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, vol. 3, pp. 257-260. IEEE, 2011.

[6] Joshi, Arnav, Ravendar Lal, Tim Finin, and Anupam Joshi. "Extracting cybersecurity related linked data from text." In *2013 IEEE Seventh International Conference on Semantic Computing*, pp. 252-259. IEEE, 2013.

[7] Etzioni, Oren, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. "Unsupervised named-entity extraction from the web: An experimental study." *Artificial intelligence 165*, no. 1 (2005): 91-134.

[8] Liang, Chen, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. "Bond: Bert-assisted open-domain named entity recognition with distant supervision." In Proceedings of the *26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1054-1064. 2020.

[9] Huang, Zhiheng, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF models for sequence tagging." arXiv preprint arXiv:1508.01991 (2015).

[10] Gasmi, Houssem, Abdelaziz Bouras, and Jannik Laval. "LSTM recurrent neural networks for cybersecurity named entity recognition." *ICSEA 11 (2018)*: 2018.

[11] Georgescu, Tiberiu-Marian, Bogdan Iancu, and Madalina Zurini. "Named-entity-recognition-based automated system for diagnosing cybersecurity situations in IoT networks." *Sensors 19*, no. 15 (2019): 3380.

[12] Qin, Ya, Guo-wei Shen, Wen-bo Zhao, Yan-ping Chen, Miao Yu, and Xin Jin. "A network security entity recognition method based on feature template and CNN-BiLSTM-CRF." *Frontiers of Information Technology & Electronic Engineering 20*, no. 6 (2019): 872-884.

[13] Li, Tao, Yuanbo Guo, and Ankang Ju. "A self-attention-based approach for named entity recognition in cybersecurity." In *2019 15th International Conference on Computational Intelligence and Security (CIS)*, pp. 147-150. IEEE, 2019.

[14] Zhou, Shengping, Zi Long, Lianzhi Tan, and Hao Guo. "Automatic identification of indicators of compromise using neural-based sequence labelling." arXiv preprint arXiv:1810.10156 (2018).

[15] Wu, Han, Xiaoyong Li, and Yali Gao. "An effective approach of named entity recognition for cyber threat intelligence." In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, vol. 1, pp. 1370-1374. IEEE, 2020.

[16] Mittal, Sudip, Anupam Joshi, and Tim Finin. "Thinking, fast and slow: Combining vector spaces and knowledge graphs." arXiv preprint arXiv:1708.03310 (2017).

[17] Gasmi, Houssem, Jannik Laval, and Abdelaziz Bouras. "Information extraction of cybersecurity concepts: an LSTM approach." *Applied Sciences 9*, no. 19 (2019): 3945.

[18] Bridges, Robert A., Corinne L. Jones, Michael D. Iannacone, Kelly M. Testa, and John R. Goodall. "Automatic labeling for entity extraction in cybersecurity." arXiv preprint arXiv:1308.4941 (2013).

[19] V. Behzadan, C. Aguirre, A. Bose and W. Hsu, "Corpus and Deep Learning Classifier for Collection of Cyber Threat Indicators in Twitter Stream," *2018 IEEE International Conference on Big Data (Big Data), 2018*, pp. 5002-5007, doi: 10.1109/BigData.2018.8622506.

[20] Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." In *Text summarization branches out*, pp. 74-81. 2004.