Cold-start cybersecurity ontology population using information extraction with LSTM

Houssem Gasmi, Jannik Laval, Abdelaziz Bouras

Cold-start cybersecurity ontology population using information extraction with LSTM

Houssem Gasmi
DISP Laboratory
Université Lumière Lyon 2,
Lyon, France
email:houssem.gasmi@univ-lyon2.fr

Jannik Laval
DISP Laboratory
Université Lumière Lyon 2,
Lyon, France
email: jannik.laval@univ-lyon2.fr

Abdelaziz Bouras
Computer Science Department, College of Engineering, Qatar University
Doha, Qatar
email:abdelaziz.bouras@qu.edu.qa

Abstract— **In this paper, we discuss how Long Short Time Memory (LSTM) neural networks can be applied to cyber security knowledge base population. Assuming we have an empty ontology that models the field of vulnerabilities description management using ontology concepts such as classes and properties, we want to populate it from online unstructured textual resources. More precisely, the task involves predicting instances of the classes in the ontology and the semantic relationship between them from a text describing a vulnerability in a software. As opposed to the statistical inference approach, we adopt a neural networks approach to predict the structure of the text. Given an input as a sequence of words, the model predicts the most likely classification of the words and extracts the relationship between the words that are relevant to the domain. The proposed system is decomposed into named entry recognition, relation extraction, ontology population. In this paper, we show how these tasks fit together and how they are implemented as unified framework. Keywords—cybersecurity, named entity recognition, relation extraction, ontology learning.**

I. INTRODUCTION

Knowledge bases are a viable alternative to the natural language as a medium of communication and are becoming an essential source for data analysts [1]. In the cybersecurity domain, some knowledge bases have been established for different purposes. For instance, several antivirus vendors have established massive virus signature bases. In the vulnerability management domain, knowledge bases such as Common Vulnerability Enumeration (CVE) and National Vulnerability Database (NVD). In these knowledge bases, information about vulnerabilities are collected from various sources such as blogs and security bulletins. Vulnerabilities are typically given a unique identifier, threat type, and a threat level. Despite the success of these knowledge bases, they are still semi-structured because the main description of the vulnerability is still in free text. Hence, there is a need to rely on natural language techniques to transform the remaining unstructured textual information into a fully structured knowledge base. This will be helpful for intrusion detection and situational awareness [2].

The manual population of knowledge bases however, is labor intensive and time consuming which make them impractical to keep up-to-date with latest vulnerabilities and prevent the detection of zero day exploits. Automated population recently attracted growing attention. Taking ontologies as a popular example of knowledge bases, the task of establishing such ontologies consists of three main tasks. The definition of the ontology to model the domain of interest which includes mainly the definition of classes and the relations between them, in addition to the rules that would enable the inference of more information beyond the explicit defined knowledge. The second task is the initial population of the ontology with information. This is typically called cold-starting the ontology. It includes information extraction from a source such as NVD using NLP techniques and populating the relevant parts of the ontology. The last task is the automated update of the ontology as daily information is generated.

Compared to traditional statistical-based extraction methods which achieves good results, previous works showed that off-the-shelf NLP tools are not able to extract security-related entities and their corresponding relations with a satisfactory accuracy [3][4]. However, statistical-based method rely on feature engineering which has some disadvantages. Firstly, it needs a person who has a deep knowledge in the domain and a lengthy trial and error process to shape the features of the text. Secondly, feature engineering relies on dictionaries or lookups to identify entities in the domain [5]. These lookups take time and manual effort to build and it is hard to keep them up to date in highly evolving domains such as cybersecurity.

More recently, deep neural networks have been considered as a potential alternative to the traditional statistical methods as they address many of their shortcomings [6]. With neural networks, features can automatically be learned, which considerably decreases the effort needed by human experts in several domains. Moreover, the results achieved in various domains have demonstrated that the features learned by neural networks are better in terms of accuracy than the human-engineered features. RNNs have been studied and proved that they can process input with variable lengths as they have a long-time memory. This property resulted in notable successes with several NLP tasks like speech recognition and machine translation [7]. LSTM further improved the performance of RNNs and allowed the learning between arbitrary long-distance dependencies [8]. With properly annotated large corpus, deep neural networks can provide a viable alternative to the traditional methods, which are labor-intensive and time consuming. Ontology population can be looked at as a function where the input is a set of documents containing unstructured text, the function is a neural network model that processes the input and generates the most probable representation that fits in the ontology model.

The objective of this paper is twofold. The first is to build an ontology that models vulnerability descriptions and defines rules that demonstrates the capabilities of ontologies in inferring additional knowledge. The second goal is to propose a framework that cold starts the ontology and keep it updated. The first task of the framework is to extract information from unstructured data sources using an LSTM-based neural network model. The model performs Named Entity Recognition (NER) and extracts domain entities from the text using deep learning. To train the model, we tweaked several parameters to see their effect on the accuracy of the model.

This resulted in several potential models that we analyzed. The models are evaluated in terms of training/testing accuracy, precision, recall, and F1 scores. The framework then heuristically infers the relation between entities. Based on the extracted information, the ontology is populated.

The paper is organized as follows: Section II reviews the related work in the field. Section III provides an overview of the proposed framework. The next Section outlines and discusses the results. Finally, Section V concludes the paper.

II. RELATED WORK

A. Cybersecurity ontologies

Efforts to construct ontologies that model the field of cybersecurity focused mainly on modeling the following aspects: attacks, vulnerabilities, threats, policies, and countermeasures. Undercoffer and Joshi [9] proposed ontologies to model cyber-attacks from instruction detection systems and stated the benefits of ontologies using use case scenarios. Undercoffer et al. [10] then analyzed four thousands classes of cyber-attacks and modeled their properties and relationships. Joshi et al [2] extended the ontology proposed by Undercoffer to model the NVD vulnerability descriptions. The resulting ontology consisted of 11 entity types such as product, vulnerability, means.

Do Amaral presented a paradigm to extract knowledge from natural language text presented through an ontology for the information security domain [11]. Wang and Guo [12] built an ontology for security vulnerabilities that defines the main concepts in the vulnerability management domain and demonstrated the capabilities of the ontology to perform vulnerability analysis and assessment through reasoning. Elahi et al. [13] also focused on vulnerability management and aimed at integrating the modeling of vulnerabilities into the security requirements analysis of system development. Bhandari et al. [14] described an ontological approach that models vulnerabilities and attacks to analyze the current state of a network. Lannacone et al. [15] developed a framework that integrates knowledge from a variety of structured and unstructured sources to build an ontology that is derived from several cyber knowledge graphs.

B. Information Extraction

Various methods have been applied to extract cybersecurity entities and their relations in the cybersecurity domain. Joshi et al. [2] developed a framework prototype to spot entities and concepts from heterogeneous data sources. They leveraged the maximum entropy models (MEMs) and trained the CoreNLP off-the-self entity recognition tool on a labeled corpus. With the help of 12 Computer science students who have a good understanding of cybersecurity concepts, the training corpus was painstakingly hand-labeled and contained around 50,000 tokens.

Bridges et al. [16] used the perceptron algorithm, which has been proven to be better than the maximum likelihood estimation techniques [17], and implemented a custom tool for the task, which provided more flexible feature engineering. To automatically build the training corpus, Bridges et al. leveraged the structure of the data in NVD to create a set of heuristics for labelling the text. This resulted in a corpus containing around 750,000 tokens. Compared to Joshi et al., Bridges et al. achieved better accuracy because their training corpus was much larger. However, their corpus is not as varied as the corpus of Joshi et al.

An SVM classifier has been used by Mulwad et al. [5] to separate cybersecurity vulnerability descriptions from non-relevant ones. The classifier uses Wikitology and a computer security taxonomy to identify and classify domain entities. Jones et al. [18] implemented a bootstrapping algorithm that requires little input data consisting of few relation samples and their patterns to extract security entities and the relationship between them from the text.

McNeil et al. [19] implemented a semi-supervised learning algorithm. The bootstrapping algorithm learns heuristics to identify cyber entities and recognize additional entities through iterative cycling on a large unannotated corpus. Bridges el al. [20] compared previous MEM models in the field, showing that the training data for these models were unrepresentative of the data in the wild and hence, these models over-fit to the training data. Therefore, they used documents from more diverse security-related resources and crafted three cyber entity extractors based on their set of collected data, which improved the state-of-the-art cyber entity tagging.

The LSTM model has been used since the early days of its introduction for NER and relation extraction. Combining Conditional Random Field (CRF) and LSTM-RNNs has been proposed by Huang et al. [21] by stacking two layers in a single model and showed a performance comparable the state-of-art method used in NER, chunking, and part-of-speech (POS) tagging. In the SemEval-2010 Task 8, several relation neural network-based models have been proposed. Models included CNN-based models, RNN-based models, and embedding-based models [21]. Xu et al. [22] also showed that the performance of LSTM-RNN models is better than the CNN-based models. Li et al. [23] used a basic RNN model to evaluate the different sequence-based and tree-structured LSTM-RNNs models on relation classification. As opposed to the Tai et al. [24] model that do not combine the word sequence information and dependency tree information, Miwa et al. [25] proposed a model that combines both information in one bidirectional model for an end-to-end extraction of relations between entities.

The goal of this paper is to build a framework that would leverage the recent neural network techniques to extract information from cyber text as an alternative to the traditional statistical-based methods and populates an ontology that models the vulnerability descriptions in the cybersecurity domain. The studied LSTM model have been evaluated using general English corpus, but as far as we know, they have not been studied in the specific field of cybersecurity.

III. PROPOSED FRAMEWORK

Figure 1 describes the proposed framework which converts textual descriptions of software vulnerabilities into a more formal representation in the form of an ontology. The framework consists mainly of the following parts: a source of data, LSTM model, information extractor, and the vulnerabilities ontology. This data source is mainly unstructured text with some structured information. The unstructured part of text contains the main text describing a vulnerability in a software. Some databases such as NVD add more structured information to the vulnerability description like an identifier, severity level, etc. The first step performed by the framework is preprocessing the text by extracting it

from the NVD database then format it in the CoNLL2000 format [26] and convert it to word embeddings representation as needed by the LSTM neural network. Word embeddings were introduced by Mikolov et al. [27] as an alternative to the traditional one-hot encoding vectors as a text representation. Compared to the one-hot encoding vectors, word embeddings are vectors that have a small length usually between 200 and 300 which is much shorter than the one hot vectors which are usually as long as the length of the vocabulary of the treated text. The second and most important difference is that word embeddings capture the semantic differences between the words of the vocabulary.

The second step is to train a prediction model that would be able to extract entities from text. To train the model, the NVD processed corpus is divided into 3 corpora, 70% is used to train the model. 10% is used as a holdout cross-validation set and 20% is used to evaluate the model. Once the model is trained, we get a prediction model that is used by a tagger to extract the domain entities from the text. For each paragraph, the LSTM model tags all the words in the text and then keep only the tags of interest in the domain. In the next step, the relation extractor takes the annotated text and assigns a relation between the detected entities from a predefined list of potential relations. Once the entities and relations are extracted, the ontology is populated with the extracted information.

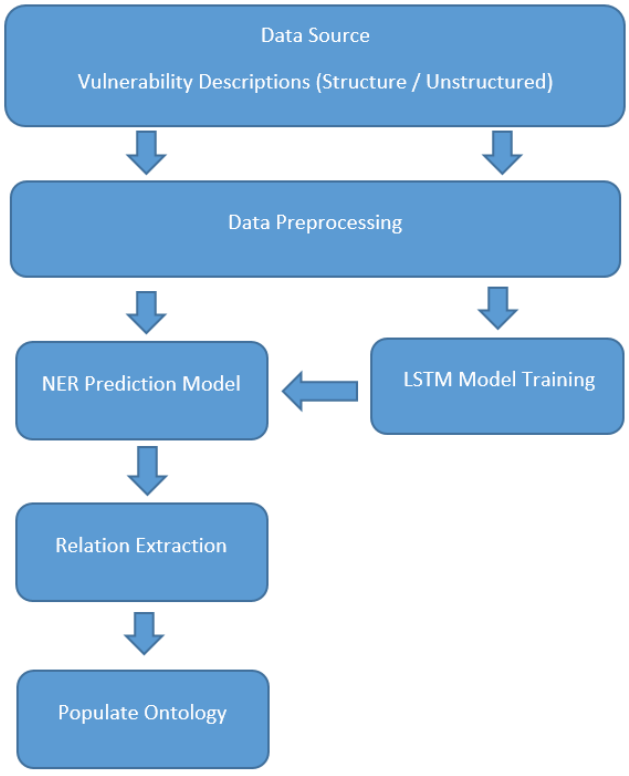

Fig 1. Framework Architecture

A. *Ontology*

An ontology is an explicit specification of a conceptualization. A conceptualization is an abstract and simplified view of the reality portion that interests us: objects, concepts and other units that exist in some area of interest and the relationships among them. (Gruber, 1995).

Figure 2 shows the vulnerabilities ontology which consists of the classes representing the domain and their relations; it is described using the OWL semantic web language to represent the knowledge describing the entities of the domain and the relationship between them. The purpose of each of these classes is explained in the following section.

- **Vendor.** A vendor of a product.
- **Product.** A software developed by a vendor which can be an application or an operating system.
- **Version.** The version(s) of the product affected by the vulnerability.
- **File.** A file that is part of a product and is related to the described vulnerability.
- **Function.** A programmatic function that is part of a product and is related to the described vulnerability.
- **Vulnerability.** Each vulnerability in the ontology has a unique identifier that corresponds to the CVE ID from NVD.
- **Vulnerability Level.** A level of 'Low', 'Medium', and 'High' on the level of criticality in terms of vulnerabilities in a product.

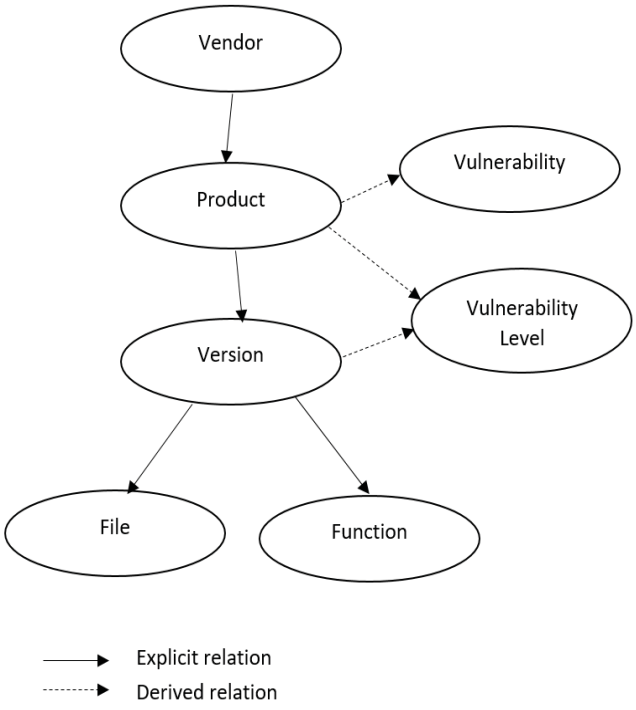

Fig 2. Vulnerability Management Ontology

As shown in Figure 2, two types of relations exist in the ontology. The first type is the explicit relations we define in the ontology and are extracted from the text. The second type is implicit relations that are derived from existing information in the ontology using inference rules. The ontology defines these rules using the SWRL language to infer more knowledge using an inference engine. Examples of these rules are:

```
Vendor(?v) ^ is_vendor_of(?v, ?p) ^ has_version (?p,
?version) ^ is_vulnerable_in_file(?version, ?f) ^
Vulnerability(?vulnerability)
-> has_vulnerability(?p, ?vulnerability)
```

This rule infers based on the existing fact that if a product from a vendor has a vulnerability in a file this implies that the product is vulnerable and the same applies for vulnerabilities in functions.

```
Version(?v), (is_vulnerable_in_function min 5
                Function)(?v)
-> has_vulnerability_level(?v, High)
```

This rule classifies a version of a product as having a high vulnerability level if it has a minimum of five vulnerable functions.

B. *Information Extraction*

The main responsibility of the proposed framework is extracting information from the textual descriptions of vulnerabilities. Information extraction includes the extraction of domain entities or Named Entity Recognition (NER) and relations extraction. NER is performed using LSTM neural networks which are a type of RNNs that have the ability to detect and learn patterns in a sequence of input data. Sequences of data can be stock market time series, natural language text or voice, genomes, etc. RNNs combines the current input (e.g., current word in the text) with the knowledge learned from the previous input (e.g., previous words in the text). While RNNs perform well with short sequence, they suffer from an issue called vanishing or exploding grading issue when the processed sequence becomes too long. When the input becomes long, RNNs become difficult to train especially when the number of model parameters becomes large. LSTMs solved this issue and can process an arbitrary long sequence of input.

We applied the LSTM architecture to the domain of cybersecurity NER (Figure 3). This architecture combines LSTM, word2vec models, and CRFs [28]. The main characteristic of this method is that it is domain and entity type agnostic and can be applied to any domain. All it needs as input is an annotated corpus in the same format as the CoNLL-2000 dataset. Unlike domains such as the biomedical domain, annotated corpora in the field of cybersecurity are not widely available. The corpora used to train the model were generated as part of the work of Bridges et al [1].

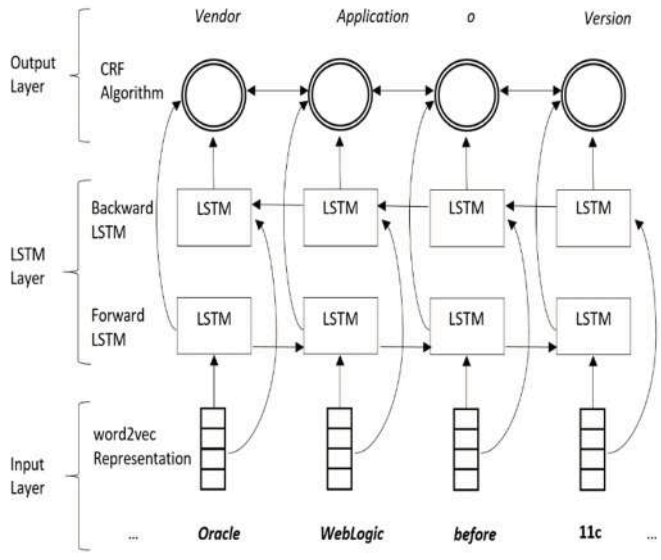

Fig 3. LSTM Architecture for NER

To train the model that would yield the most accurate results, we need to find the best parameters combination for the model. The model provides a number of parameters that can be tuned. We analyzed the set of model parameters and selected the set of parameters that would be relevant to our context. We then verified our assumptions with experiments. The parameter set we chose to tune when training the model are as follows:

- Lowercase words. A flag specifying whether words of the training corpus are converted to lowercase or not.
- Replace digits with 0. A flag that converts all the digits in the corpus to 0s.
- LSTM hidden layer size. The number of the stacked LSTM cells in the hidden layer.
- Use a bidirectional LSTM for words. A flag specifying whether to use a bidirectional LSTM.
- Use CRF: the model can apply the CRF algorithm before predicting the tag of a word. This can be disabled.
- Dropout on the input. Dropout is a regularization method where input and recurrent connections to LSTM units are probabilistically excluded from activation and weight updates while training a network. This has the effect of reducing overfitting and improving model performance.
- Learning method (SGD, Adadelta, Adam..). The optimization algorithm (or optimizer) is the approach used for training a machine learning model to minimize its error rate.

C. *Evaluation Criteria*

The evaluation metrics used for the LSTM model evaluation are the precision, recall, and F1 score. Precision and recall are the two main performance indicators and F1 is called the harmonic mean of precision and recall and further assists in the evaluation of the model. The evaluation metrics are calculated as follows:

$$P = \frac{TP}{TP+FP}, R = \frac{TP}{TP+FN}, F1 = \frac{2 \times P \times R}{P+R}.$$

The metrics are defined in terms of true positives, false positives, and false negatives that are defined as follows:

- True positives (TP): the outcome where the model correctly predicts the positive class
- False positives (FP): the outcome where the model incorrectly predicts the positive class
- False negatives (FN): the outcome where the model incorrectly predicts the negative class.

IV. Experimental Results and Discussion

We evaluated the performance of the framework that is based on the LSTM architecture. For the NER task, the evaluation was performed on the full set of 40 entity tags in the corpus as well as for a subset of the eight common entities that appear frequently in the cybersecurity vulnerability descriptions. The entities considered are vendor, application, version, file, operating system (os), hardware, edition, and Vulnerability ID.

The initial model parameters are as follows:

TABLE 1. INITIAL MODEL PARAMETERS

Parameter	Value
CRF	True
Lowercase words	False
Replace digits with 0	False
LSTM hidden layer size	100
Use a bidirectional LSTM for words	True
Dropout	0.5
Learning method	SGD

The NER results for Models 1 are shown in Table 2. The table shows that the F1 scores of 'file' and 'Vulnerability ID' are relatively high while 'edition' achieved the lowest F1 score. The probable explanation for this behavior is that 'Vulnerability ID's have a specific format such as CVE-2019-12623 which make it easy for the LSTM model to learn whereas the 'edition' tag is usually a general English word in the text such as: 'second edition'. Table 3 shows the average performance of the model across all the tags.

TABLE 2. NER RESULTS FOR MODEL 1 FOR SPECIFIC TAGS

Entity type	Precision	Recall	F1
Vendor	0.95	0.90	0.92
Application	0.86	0.88	0.87
Version	0.99	0.99	0.99
Edition	0.91	0.34	0.45
OS	0.95	0.93	0.94
File	1.00	0.99	0.99
Vulnerability ID	1.00	1.00	1.00

TABLE 3. NER RESULTS FOR MODEL1 FOR ALL TAGS

Entity type	Precision	Recall	F1-score
All tags (average)	0.84	0.77	0.79

The following table shows the average performance of the model variations across all the entity types. In each model, a parameter has been changed.

TABLE 4. MODEL VARIATIONS RESULTS

Parameters	Value	Precision	Recall	F1-score
Default values	-	0.84	0.77	0.79
Dropout	0	0.87	0.82	0.83
Hidden layer size	25	0.81	0.79	0.79
Hidden layer size	50	0.83	0.78	0.78
Leaning method	adam	0.87	0.86	0.84
Leaning method	adadelta	0.81	0.76	0.77
Replace digits with 0	True	0.83	0.83	0.83
CRF	False	0.53	0.53	0.53
Bidirectional LSTM	False	0.53	0.54	0.54
Lowercase words	True	0.82	0.79	0.79

The two parameters that had the biggest impact on the performance of the model are CRF and word bidirectional parameters. When we disabled these parameters, the F1 score dropped considerably. The results show also that 'Adam' optimization algorithm achieved the best performance among the other algorithms. As expected, removing the dropout improved the result but increases overfitting to the training data. Varying the remaining parameters did not have a big impact on the performance of the model.

V. CONCLUSION AND FUTURE WORK

In this paper, we built an ontology population framework for the cybersecurity vulnerability management domain. The framework is mainly based on LSTM neural network to train an entity extraction model to extract entities of the domain. The model needs further validation with more varied sources as it could suffer from overfitting. The ontology needs further improvement with more classes, relations, and inference rules to make it more practical. This gives cybersecurity professionals the necessary tools that grant them rapid access to the information needed for a better understanding of the threats and decision-making

In future, our work will focus on enriching the ontology and improving the framework by diversifying the sources of information.

VI. ACKNOWLEDGEMENTS

This publication was made possible by NPRP grant # NPRP 11S-1227-170135 from the Qatar National Research Fund (a member of Qatar Foundation). This work was also supported by DISP Laboratory, Université Lumière Lyon 2. The statements made herein are solely the responsibility of the authors.

VII. REFERENCES

[1] J. Piskorski and R. Yangarber, "Information extraction: Past, present and future," in *Multi-source, multilingual information extraction and summarization*, Springer, 2013, pp. 23–49.

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

[3] X. Liao, K. Yuan, X. Wang, Z. Li, L. Xing, and R. Beyah, "Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 755–766.

[4] S. More, M. Matthews, A. Joshi, and T. Finin, "A knowledge-based approach to intrusion detection modeling," in *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*, 2012, pp. 75–81.

[5] V. Mulwad, W. Li, A. Joshi, T. Finin, and K. Viswanathan, "Extracting information about security vulnerabilities from web text," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, 2011, vol. 3, pp. 257–260.

[6] Y. Goldberg, "A primer on neural network models for natural language processing," *Journal of Artificial Intelligence Research*, vol. 57, pp. 345–420, 2016.

[7] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, 2013, pp. 6645–6649.

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

[9] J. Undercoffer, A. Joshi, and J. Pinkston, "Modeling computer attacks: An ontology for intrusion detection," in *International Workshop on Recent Advances in Intrusion Detection*, 2003, pp. 113–135.

[10] J. Pinkston, J. Undercoffer, A. Joshi, and T. Finin, "A target-centric ontology for intrusion detection," in *In proceeding of the IJCAI-03 Workshop on Ontologies and Distributed Systems. Acapulco, August 9 th*, 2004.

[11] F. N. Do Amaral, C. Bazilio, G. M. H. Da Silva, A. Rademaker, and E. H. Haeusler, "An ontology-based approach to the formalization of information security policies," in *2006 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW'06)*, 2006, pp. 1–1.

[12] J. A. Wang and M. Guo, "OVM: an ontology for vulnerability management," in *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*, 2009, p. 34.

[13] G. Elahi, E. Yu, and N. Zannone, "A modeling ontology for integrating vulnerabilities into security requirements conceptual foundations," in *International Conference on Conceptual Modeling*, 2009, pp. 99–114.
[14] P. Bhandari and M. S. Gujral, "Ontology based approach for perception of network security state," in *2014 Recent Advances in Engineering and Computational Sciences (RAECS)*, 2014, pp. 1–6.
[15] M. Iannacone *et al.*, "Developing an ontology for cyber security knowledge graphs," in *Proceedings of the 10th Annual Cyber and Information Security Research Conference*, 2015, p. 12.
[16] 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.
[17] M. Collins, "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, 2002, pp. 1–8.
[18] C. L. Jones, R. A. Bridges, K. M. Huffer, and J. R. Goodall, "Towards a relation extraction framework for cyber-security concepts," in *Proceedings of the 10th Annual Cyber and Information Security Research Conference*, 2015, p. 11.
[19] N. McNeil, R. A. Bridges, M. D. Iannacone, B. Czejdo, N. Perez, and J. R. Goodall, "Pace: Pattern accurate computationally efficient bootstrapping for timely discovery of cyber-security concepts," in *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, 2013, vol. 2, pp. 60–65.
[20] R. A. Bridges, K. M. Huffer, C. L. Jones, M. D. Iannacone, and J. R. Goodall, "Cybersecurity Automated Information Extraction Techniques: Drawbacks of Current Methods, and Enhanced Extractors," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, pp. 437–442.
[21] I. Hendrickx *et al.*, "Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals," in *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, 2009, pp. 94–99.
[22] K. Xu, Y. Feng, S. Huang, and D. Zhao, "Semantic relation classification via convolutional neural networks with simple negative sampling," *arXiv preprint arXiv:1506.07650*, 2015.
[23] Q. Li and H. Ji, "Incremental joint extraction of entity mentions and relations," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, vol. 1, pp. 402–412.
[24] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *arXiv preprint arXiv:1503.00075*, 2015.
[25] M. Miwa and M. Bansal, "End-to-end relation extraction using lstms on sequences and tree structures," *arXiv preprint arXiv:1601.00770*, 2016.
[26] E. F. Tjong Kim Sang and S. Buchholz, "Introduction to the CoNLL-2000 shared task: Chunking," in *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7*, 2000, pp. 127–132.
[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.
[28] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 2003, pp. 188–191.