Extracting threat intelligence relations using distant supervision and neural networks

Yali Luo, Shengqin Ao, Ning Luo, Changxin Su, Peian Yang, Zhengwei Jiang

Chapter 10

EXTRACTING THREAT INTELLIGENCE RELATIONS USING DISTANT SUPERVISION AND NEURAL NETWORKS

Yali Luo, Shengqin Ao, Ning Luo, Changxin Su, Peian Yang and Zhengwei Jiang

Abstract Threat intelligence is vital to implementing cyber security. The automated extraction of relations from open-source threat intelligence can greatly reduce the workload of security analysts. However, implementing this feature is hindered by the shortage of labeled training datasets, low accuracy and recall rates of automated models, and limited types of relations that can be extracted.

This chapter presents a novel relation extraction framework that employs distant supervision for data annotation and a neural network model for relation extraction. The framework is evaluated by comparing it with several state-of-the-art neural network models. The experimental results demonstrate that it effectively alleviates the data annotation challenges and outperforms the state-of-the-art neural network models.

Keywords: Threat intelligence, relation extraction, machine learning

1. Introduction

Threat intelligence is evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice, about an existing or emerging menace or hazard to assets that can be used to inform decisions about the response to the menace or hazard [15]. Security analysts have become proficient at extracting indicators of compromise (IOCs). Indicators of compromise such as URLs, IP addresses, email addresses, domain names and hashes are easy to extract, but they are easily modified by attackers to bypass security measures. In any case,

indicators of compromise on their own cannot be expected to provide adequate cyber security.

Security analysts and policymakers need high-level threat intelligence to make critical decisions. High-level threat intelligence, such as tactics, techniques and procedures (TTPs), is extracted from a variety of sources and expressed in a form that enables further analysis and decision making. The sources are typically referred to as open-source intelligence (OSINT), which includes unstructured information collected from public resources such as research papers, newspapers, magazines, social networking sites, wikis, blogs, etc. [23].

The higher the level in the threat intelligence pyramid [21], the greater the difficulty of extracting information. Moreover, open-source intelligence resources are massive and complex, and require considerable manual analysis to obtain high-level threat intelligence. Some researchers have attempted to automate this process. However, as described below, these methods tend to focus on identifying cyber threat entities and ignore the relations between the entities. Additionally, threat intelligence relation extraction approaches often rely on rules and features developed by experts, making it difficult to deal with new entities and relations.

The proposed framework for threat intelligence relation extraction leverages distant supervision and neural networks. Distant supervision is a popular method for collecting and generating training datasets in the natural language processing domain [18]. The proposed framework uses distant supervision to generate a large amount of annotation data needed for machine learning relatively quickly. With adequate training data, a neural network can be created to effectively extract relations from unstructured, text-based, open-source intelligence resources.

The proposed framework is the first to produce a dataset for threat intelligence relation extraction. The framework is evaluated by comparing it against several state-of-the-art neural network models. The experimental results demonstrate that it effectively alleviates the data annotation challenges and outperforms the state-of-the-art neural network models.

2. Related Work

This section describes related work in the areas of threat intelligence datasets and threat intelligence information extraction.

2.1 Threat Intelligence Datasets

The demand for actionable threat intelligence, including datasets for extracting threat intelligence, is increasing. Mulwad et al. [20] designed

a framework for extracting vulnerabilities and attack information from web text and translating them to a machine-understandable format, The datasets were drawn from 107 vulnerability description documents, but they are not open source.

McNeil et al. [16] developed a novel entity extraction and guidance algorithm that extracts valuable network security concepts. They acquired information from an online open-source website containing ten documents with seven entity types, and manually annotated the information to produce their dataset.

Jones et al. [7] specified a bootstrapping algorithm that extracts security entities and their relationships from textual information. They created a dataset using 62 document corpora from various cyber security websites, but the dataset is not publicly available.

Joshi et al. [9] extracted network-security-related link data from text documents and constructed experimental datasets via professional annotation. Their training dataset comprises 3,800 entities and 38,000 instances and their testing dataset contains 1,200 entities and 9,000 instances. However, the datasets are not publicly available.

Lal [10] has researched the extraction of security entities and concepts from unstructured text. More than 100 open-source reports were processed using screening and factual sampling methods to produce a dataset containing 60 common vulnerabilities and exposures (CVEs), 12 Microsoft announcements and 12 Adobe announcements. However, this dataset is not available to the public.

The literature survey reveals that threat intelligence datasets are rare and very few of them are publicly available. Therefore, this research has sought to develop an automated annotation method based on distant supervision that would enable security analysts to label open-source intelligence data quickly and efficiently as a precursor to creating threat intelligence datasets.

2.2 Threat Intelligence Information Extraction

Threat intelligence information extraction is a hot research topic. Liao et al. [12] have devised an automated technique that extracts indicators of compromise from security blogs and generates a machine-readable version for discovering inherent relationships in threat intelligence.

Lee et al. [11] have focused on discovering valuable security information and identifying emerging security event topics. Their system leverages modified topic graphs and topic discovery algorithms to discover information from open-source threat intelligence resources.

Mittal et al. [19] have attempted to obtain timely network security threats and vulnerabilities in an automated manner. Their system discovers, extracts and analyzes threat intelligence from Twitter feeds. A database in the WWW Resource Description Framework (RDF) format maintains the collected intelligence, and inference rules specified in the Semantic Web Rule Language (SWRL) process the data to produce network security threat and vulnerability information.

Tao et al. [24] have focused on the timely sharing of threat intelligence and responding to threat alerts. They applied an immune factor network algorithm based on a classification model to actively access and extract useful information from a large quantity of raw security information.

Gascon et al. [3] have attempted to discover potential relationships between pieces of different threat intelligence. They proposed a similarity algorithm based on attribute graphs to perform similarity correlations of text at different levels of granularity. However, their results were not very promising.

Traditional research on threat intelligence information extraction has focused on indicators of compromise. New research should focus on developing robust techniques for extracting high-level threat intelligence, such as tactics, techniques and procedures, from multiple sources and express it in a form that enables further analysis and decision making. Relation extraction leveraging deep learning theory from the natural language processing domain can significantly advance this line of research.

3. Proposed Framework

This section describes the threat intelligence relation extraction framework developed in this research.

3.1 Overview

The proposed framework is designed to extract relations efficiently from unstructured open-source intelligence. It incorporates a distant supervision module that annotates unstructured data efficiently and a neural network module that extracts relations from unstructured threat intelligence. Figure 1 provides an overview of the framework workflow.

3.2 Problem Specification

Relation extraction is the elicitation of semantic relationships from unstructured text. Figure 2 shows an example of relation extraction.

The objective of relation extraction is to create a function $F : (c, R) \mapsto (e_1, e_2, r)$. The input data c is an unlabeled sentence to be processed

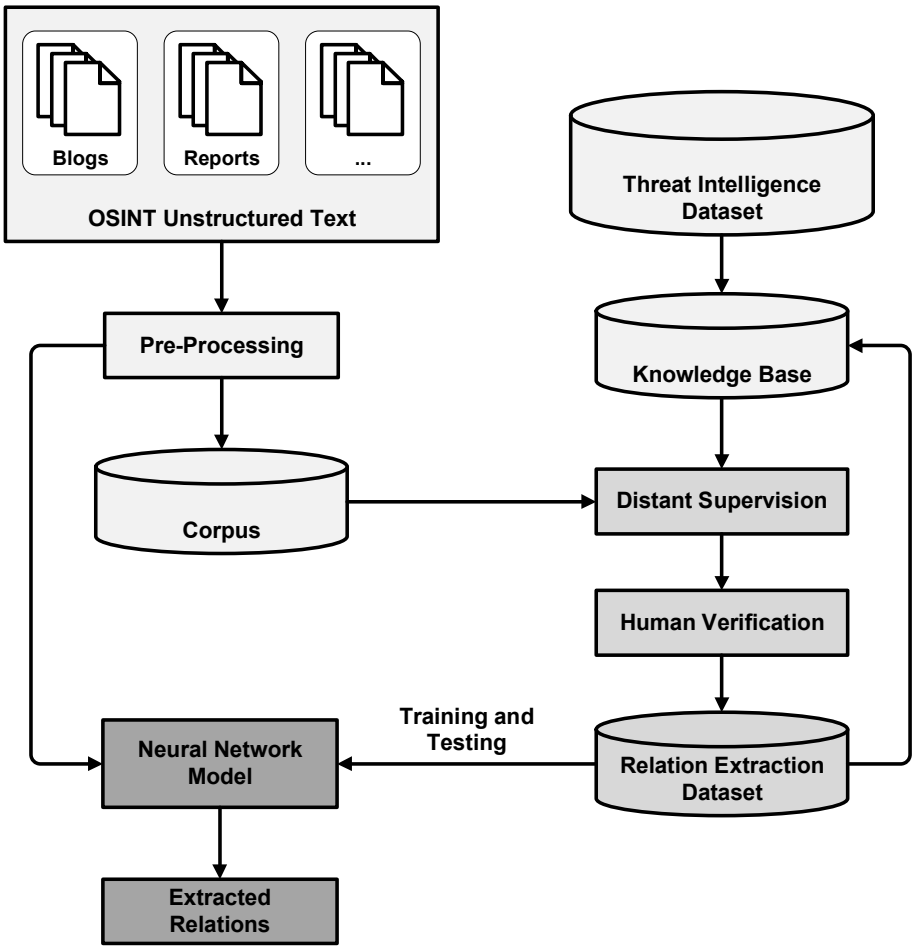

Figure 1. Framework workflow.

and $R = \{r_1, r_2, ..., r_n\}$ is a set of relations contained in sentences. The output triple, which corresponds to the prediction of c given R, comprises a head entity e_1, tail entity e_2 and relation $r \in R$ between the two entities.

3.3 Dataset

Dataset construction involves three steps: (i) knowledge base and corpus creation, (ii) distant supervision and (iii) human verification:

- **Knowledge Base and Corpus Creation:** Structured threat intelligence data conforming to the STIX II [8] specification was

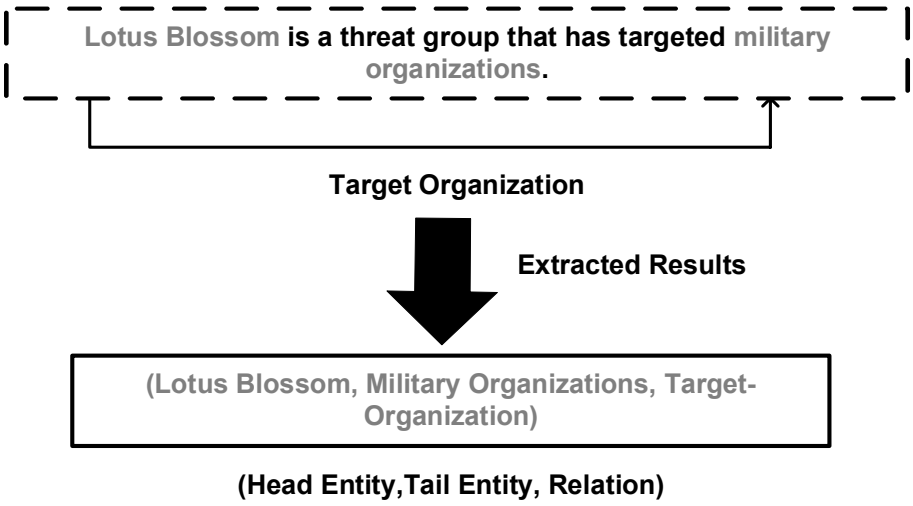

Figure 2. Relation extraction.

converted to triples of the form (e_1, e_2, r). A triple represents a relational fact. For example, (APT28, Russia, Attribution) means that the APT28 hacker group is from Russia. The knowledge base, which contains a total of 60 relations, supports the distant supervision step.

After creating the knowledge base, 2,153 threat intelligence documents from open-source intelligence resources such as cyber security blogs and APT group reports were used to create a raw corpus. The collected HTML and PDF data was processed to obtain clean text. Named entity recognition was employed to find potential entities and co-reference resolution was used to reduce noise in the text by applying natural language processing tools such as NLTK [2], Stanford CoreNLP [14] and spaCy [22]. If a sentence contained more than one entity, then a potential relation was deemed to exist between the entities and the sentence was stored in the corpus. The final corpus contained 41,835 valid sentences.

- **Distant Supervision:** Most deep learning models require a labeled training dataset. Traditionally, a labeled training dataset is created by manually annotating training data, but this is a very time-consuming task. Another approach is to generate training data using distant supervision [18]. Distant supervision assumes

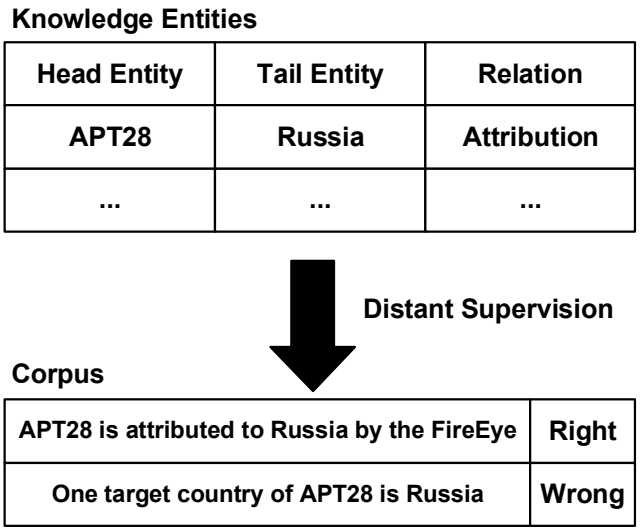

Figure 3. Distant supervision and noisy labeling.

that, if two entities participate in a relation, then any sentence that contains the two entities might express the relation.

The candidate set was created by considering each sentence c in the corpus. If c contained a head entity e_1 and tail entity e_2, and a triple (e_1, e_2, r) existed in the knowledge base, then it was assumed that the sentence c mentioned relation r and the tuple (e_1, e_2, r, c) was added as an instance in the candidate set. The final candidate set contained 11,906 instances.

Although distant supervision is effective at labeling data automatically, it suffers from the noisy labeling problem (shown in Figure 3). Unlike natural language processing, the cyber security domain requires strict data labeling, so all the instances in the candidate set had to be manually verified. Fortunately, the manual verification workload was reduced considerably because the candidate set was generated via distant supervision.

- **Human Verification:** Human annotators with expertise in computer science were recruited to eliminate incorrectly-labeled instances. A crowdsourcing verification platform similar to Amazon's Mechanical Turk [1] was employed.

  Figure 4 shows the human verification system interface. It implements checks on the sentences and their relation triples that were labeled by distant supervision. Each instance was verified

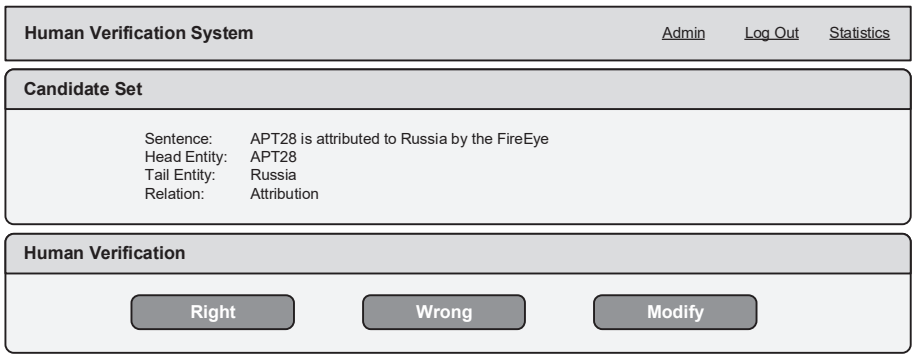

Figure 4. Human verification system interface.

by at least two human annotators. If the judgments of the two annotators were inconsistent, then the instance was passed to a third individual for proofreading and the final label determined according to the majority principle.

After the human verification, relations with less than 200 instances were eliminated to create a clean relation extraction dataset. The final dataset contained 9,277 instances, 7,035 unique entities, 18 unique relations, 8,027 entity pairs and 8,150 relational facts.

3.4 Neural Network Model

A neural network model was employed for relation extraction. The backbone of the model is a bidirectional long short-term memory (Bi-LSTM) network [4] with selective attention [13] to learn the representations of text-expressing relations. Figure 5 shows the neural network model for relation extraction. The neural network model has four layers: (i) embedding layer, (ii) encoding layer, (iii) selection layer and (iv) classification layer:

- **Embedding Layer:** For each sentence c, the Word2vec technique [17] was used to train word embeddings to project each word token onto d_w-dimensional space. The words that appeared more than 10 times in the corpus were retained as vocabulary. Position embedding [25] was also employed for all the words in each sentence to create d_p-dimensional vectors with entity position information.

- **Encoding Layer:** A Bi-LSTM neural network model was employed for each sentence encoding. Hochreiter and Schmidhu-

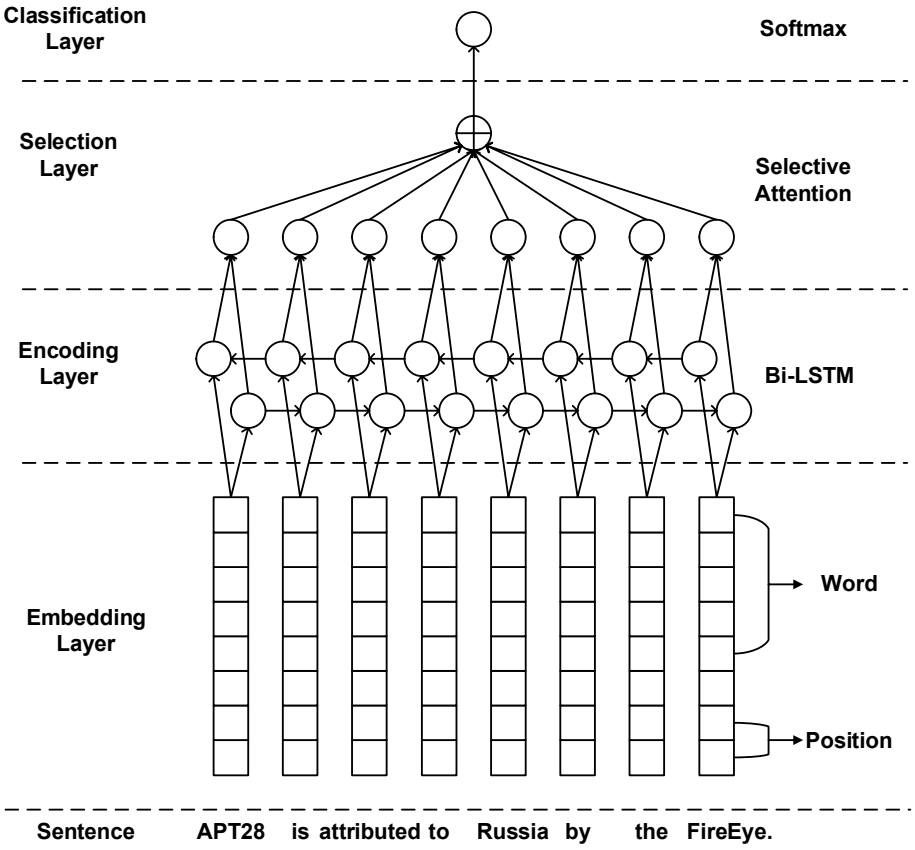

Figure 5. Neural network model for relation extraction.

ber [5] proposed the LSTM (long short-term memory) neural network model to address the gradient vanishing problem. However, a standard LSTM model only process sequences in temporal order. The Bi-LSTM neural network model [4] improves on the standard LSTM model by incorporating a second layer to obtain information from the past and future. The model is well-suited to sequence-oriented tasks such as name entity recognition and relation extraction.

- **Selection Layer:** The selection layer employs the selective attention mechanism proposed by Lin et al. [13]. It uses sentence-level attention to select sentences that express the associated relation and de-emphasize noisy sentences. The representation of a sen-

tence x_i is obtained by concatenating the word and position embeddings [25].

Suppose a set S contains n sentences for an entity pair (e_1, e_2). Then, the set vector s is computed as the weighted sum of the sentence vectors x_i:

$$s = \sum_i \alpha_i x_i$$

$$\alpha_i = \frac{exp(x_i A r)}{\sum_k exp(x_k A r)}$$

where α_i is the weight of the sentence vector x_i, A is a weighted diagonal matrix and r is the query vector associated with the relation.

- **Classification Layer:** The final classification layer employs the softmax loss function defined by Lin et al. [13]. The conditional probability $p(r|S, \theta)$ (θ denotes the model parameters) is given by:

$$p(r|S, \theta) = \frac{exp(o_r)}{\sum_{k=1}^{n_r} exp(o_k)}$$

where n_r is the total number of relations and o is the final output of the neural network model, which is given by:

$$o = Ms + d$$

where $d \in R^{n_r}$ is a bias vector and M is the representation matrix of relations.

4. Experiments and Results

This section describes the experiments conducted to demonstrate that the proposed neural network model can effectively extract relations in unstructured threat intelligence. The held-out evaluation was employed and the aggregate precision/recall graphs are provided. The results reveal that the proposed neural network model has the best performance.

4.1 Experiment Details

This section provides details about the experiments, including the data sources, experimental dataset and parameter settings:

- **Data Sources:** A web crawler based on the Scrapy framework was developed to harvest unstructured threat intelligence information

Table 1. Unstructured threat intelligence sources.

Source	Content	Format
FireEye	Security blog, security report	HTML, PDF
Symantec	Security blog, security report	HTML, PDF
Kaspersky	Security blog, security report	HTML, PDF
Cisco	Security blog	HTML
McAfee	Security blog	HTML
UNIT 42	Security blog	HTML
Twitter	Security tweet	HTML

from public security blogs and security reports released by prominent network security companies. In addition, the Twitter API was employed to harvest unstructured threat intelligence information from security practitioner tweets. Table 1 lists the threat intelligence sources. A total of 2,153 threat intelligence documents were collected.

Table 2. Top five relations in the dataset.

Relation	Instances
`/hackgroup/tool/use_tool`	1,160
`/hackgroup/organization/target_org`	1,140
`/hackgroup/location/target_loc`	1,094
`/hackgroup/method/texhnique/attack_method`	715
`/organization/hackgroup/investigate_hackgroup`	687
Total	4,796

- **Experimental Dataset:** Table 2 lists the top five relations extracted from the dataset. For each relation, 500 instances were randomly generated to produce the experimental dataset that contained 2,500 instances.

  Data associated with each relation was randomly partitioned into training (80%) and testing (20%) datasets. The training dataset contained 2,000 instances whereas the testing dataset contained 500 instances. The entire dataset is available at `github.com/luoluoluoyl/relation_extract_dataset.git`.

- **Parameter Settings:** Cross-validation of the training dataset was used to tune the neural network models. Table 3 shows all the

Table 3. Parameter settings.

Parameter	Setting
Word embedding dimensions (d_w)	50
Position embedding dimensions (d_p)	5
Window size (l)	3
Batch size (b)	100
Maximum training iterations (max_{TI})	60
Learning rate (λ)	0.1
Dropout (p)	0.5

parameter settings. In particular, the number of word embedding dimensions d_w was set to 50, position embedding dimensions d_p to 5 and window size l to 3. The batch size B was set to 100. The maximum number of iterations for training max_{TI} was set to 60. The learning rate λ was set to 0.1 and dropout rate p to 0.5.

4.2 Comparison with Baseline Models

The neural network model developed in this research was compared with several state-of-the-art neural network models:

- **Bi-LSTM+ATT+NOPOS:** Zhou et al. [26] proposed the ATT-Bi-LSTM neural network model. Experimental results obtained for the SemEval-2010 relation classification task demonstrated the effectiveness of the ATT-Bi-LSTM model. Since the baseline ATT-Bi-LSTM model uses word embedding but not position embedding (NOPOS), it is named Bi-LSTM+ATT+NOPOS for comparison purposes.

- **Bi-LSTM+ONE:** Zeng et al. [25] proposed a neural network model based on the at-least-one assumption (ONE). The model incorporates a piecewise convolutional neural network with multi-instance learning for distant supervised relation extraction. In the experiments, the ONE assumption was applied in a Bi-LSTM model to create the Bi-LSTM+ONE baseline neural network model.

- **Bi-LSTM+CROSS MAX:** Jiang et al. [6] proposed a multi-instance multi-label neural network model for distant supervised relation extraction. It relaxes the at-least-once assumption (ONE) and uses cross-sentence max-pooling (CROSS MAX) to enable information sharing across different sentences. Overlapping relations are handled using multi-label learning with a neural network clas-

Table 4. Baseline neural network model results.

Neural Network Model	F1-Score	AUC	Accuracy
Word Embedding			
LSTM+ATT+NOPOS	0.5719	0.5991	0.7491
Bi-LSTM+ATT+NOPOS	0.6406	0.6695	0.8083
Word + Position Embedding			
LSTM+CROSS MAX	0.6641	0.7253	0.8750
LSTM+ONE	0.7047	0.7362	0.8674
LSTM+ATT	0.7312	0.7995	0.9253
Bi-LSTM+CROSS MAX	0.7069	0.7643	0.8977
Bi-LSTM+ONE	0.7730	0.8253	0.9300
Bi-LSTM+ATT	0.8207	0.9004	0.9784

sifier. In the experiments, CROSS MAX was combined with a Bi-LSTM model to create the Bi-LSTM+CROSS MAX baseline neural network model.

Cross-validation was performed on the three baseline neural network models. In order to compare the LSTM and Bi-LSTM models, four additional neural network models were specified as baselines: LSTM+ATT+NOPOS, LSTM+ATT, LSTM+CROSS MAX and LSTM+ONE.

Table 4 shows the F1-score, AUC (area under curve) and accuracy values for the baseline neural network models that only use word embedding and the baseline neural network models that use word and position embeddings. Figure 6 shows the precision/recall graphs obtained for all the baseline neural network models.

The experimental results motivate the following observations:

- The Bi-LSTM+ATT neural network model developed in this research significantly outperforms all the baseline neural network models with regard to relation extraction.
- As shown in Figures 6(a) and 6(b), the LSTM and Bi-LSTM neural network models that incorporate the ATT method have better performance than the LSTM and Bi-LSTM neural network models that incorporate the ONE and CROSS MAX methods.
- As shown in Figures 6(c), 6(d) and 6(e), a Bi-LSTM neural network model outperforms the baseline LSTM neural network models for all three methods (ATT, ONE and CROSS MAX).
- Table 4 shows that the LSTM+ATT+NOPOS and Bi-LSTM+ATT +NOPOS baseline neural network models that only use word em-

Table 5. Analysis of the extraction results of four hacker groups.

Group	Tool Used	Target Org.	Target Location	Attack Method	Investig. Org.
Lazarus	KillDisk, PapaAlfa, Rising Sun, AIX	Sony Pictures Entertainment, South Korean Aerospace Companies	Africa, Europe, South Korea	Watering Hole, Phishing Email, Spear Phishing	McAfee. Kaspersky, Symantec, Novetta
Lucky Mouse	hTan, HyberPro	Data Center, Financial Services Company	Central Asia, Southeast Asia, America	Watering Hole, Phishing Email	Kaspersky, Palo Alto
APT10	Poison Ivy, PlugX, Quasar	Laoying Baichen Instruments Equipment Company, MSPs	Southeast Asia, United States, France, Germany	DLL Injection, Previous Credentials	FireEye, PwC UK, Recorded Future, BAE Systems
Nitro	Poison Ivy, Legitimate Compromised Websites	Chemical Company, Defense Company	South Korea	Spear Phishing, Remote Access	Symantec, Cyber Squared

bedding yield the worst results. This indicates that the position embedding proposed in this research is necessary and beneficial. This is also confirmed by the graphs in Figure 6(f).

4.3 Extraction Results

Five relations associated with hacker groups were extracted: Tool Used, Target Organization, Target Location, Attack Method and Investigating Organization. The extracted relations can be used to conduct a behavioral analysis of hacker groups. In addition to discovering the behavioral characteristics of hacker groups, it is possible to obtain correlations between the hacker groups.

Table 5 shows an analysis of four hacker groups. Using the extracted (head entity, tail entity, relation) tuples, it was possible to identify the tools used by the hacker groups, their targets and target locations, their attack methods and the organizations that investigated their attacks. For example, the Lazarus hacker group used a variety of attack methods (watering holes, phishing email and spear phishing) and, in addition to

targets in South Korea, attacked targets in several countries in Africa and Europe.

The extracted information also enables commonalities between hacker groups to be discerned. For example, the Lazarus and Lucky Mouse hacker groups used watering holes and phishing email in their attacks, and the APT10 and Nitro hacker groups used the Poison Ivy tool to launch their attacks.

Such extraction results could be useful in an incident investigation. After an attack by an unknown hacker group is discovered, the tools and methods used in the attack would be determined during the investigation. Information about the tools and methods could then be compared with the extracted data. Hacking groups that match the tools and methods could be identified as suspects. In short, automatically extracting relationships from open-source intelligence helps build a knowledge base, which reduces the manual workload in investigations.

5. Conclusions

The automated extraction of relations from open-source threat intelligence can reduce the workload involved in security analyses and incident investigations. The framework described in this chapter employs distant supervision for data annotation and a Bi-LSTM neural network model to automatically extract threat intelligence relations. It effectively and efficiently alleviates the challenges involved in manual data annotation to create a high-quality labeled dataset for training neural network models. The Bi-LSTM neural network model used by the framework provides significant improvements in relation extraction performance over state-of-the-art neural network models.

Future research will create an open-source threat intelligence relation extraction benchmark dataset. It will also define more sophisticated relations and expand the annotation dataset. Efforts will also be made to further alleviate the incorrect data labeling problem and reduce the manual label verification workload.

Acknowledgement

This research was supported by the National Key Research and Development Program of China under Grant nos. 2016YFB0801004 and 2018YFC0824801.

References

[1] Amazon, Amazon Mechanical Turk, Seattle, Washington (www.mturk.com), 2021.

[2] S. Bird, NLTK: The Natural Language Toolkit, *Proceedings of the Twenty-First International Conference on Computational Linguistics and Forty-Fourth Annual Meeting of the Association for Computational Linguistics: Interactive Presentation Sessions*, pp. 69–72, 2006.

[3] H. Gascon, B. Grobauer, T. Schreck, L. Rist, D. Arp and K. Rieck, Mining attributed graphs for threat intelligence, *Proceedings of the Seventh ACM Conference on Data and Application Security and Privacy*, pp. 15–22, 2017.

[4] A. Graves and J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, *Neural Networks*, vol. 18(5-6), pp. 602–610, 2005.

[5] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Computation*, vol. 9(8), pp. 1735–1780, 1997.

[6] X. Jiang, Q. Wang, P. Li and B. Wang, Relation extraction with multi-instance multi-label convolutional neural networks, *Proceedings of the Twenty-Sixth International Conference on Computational Linguistics: Technical Papers*, pp. 1471–1480, 2016.

[7] C. Jones, R. Bridges, K. Huffer and J. Goodall, Towards a relation extraction framework for cyber security concepts, *Proceedings of the Tenth Annual Cyber and Information Security Research Conference*, article no. 11, 2015.

[8] B. Jordan and J. Wunder (Eds.), STIX 2.0 Specification, Core Concepts, Version 2.0 Draft 1, OASIS Cyber Threat Intelligence Technical Committee (www.oasis-open.org/committees/download.php/58538/STIX2.0-Draft1-Core.pdf), 2017.

[9] A. Joshi, R. Lal, T. Finin and A. Joshi, Extracting cybersecurity-related linked data from text, *Proceedings of the Seventh IEEE International Conference on Semantic Computing*, pp. 252–259, 2013.

[10] R. Lai, Information Extraction of Security-Related Terms and Concepts from Unstructured Text, M.S. Thesis, Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore, Maryland, 2013.

[11] K. Lee, C. Hsieh, L. Wei, C. Mao, J. Dai and Y. Kuang, Sec-Buzzer: Cyber security emerging topic mining with open threat intelligence retrieval and timeline event annotation, *Soft Computing*, vol. 21(11), pp. 2883–2896, 2017.

[12] 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, *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 755–766, 2016.

[13] Y. Lin, S. Shen, Z. Liu, H. Luan and M. Sun, Neural relation extraction with selective attention over instances, *Proceedings of the Fifty-Fourth Annual Meeting of the Association for Computational Linguistics*, pp. 2124–2133, 2016.

[14] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard and D. McClosky, The Stanford CoreNLP Natural Language Processing Toolkit, *Proceedings of the Fifty-Second Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60, 2014.

[15] R. McMillan, Definition: Threat Intelligence, Gartner, Stamford, Connecticut, 2013.

[16] N. McNeil, R. Bridges, M. Iannacone, B. Czejdo, N. Perez and J. Goodall, PACE: Pattern accurate computationally efficient bootstrapping for timely discovery of cyber security concepts, *Proceedings of the Twelfth International Conference on Machine Learning and Applications*, pp. 60–65, 2013.

[17] T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient estimation of word representations in vector space, presented at the *First International Conference on Learning Representations*, 2013.

[18] M. Mintz, S. Bills, R. Snow and D. Jurafsky, Distant supervision for relation extraction without labeled data, *Proceedings of the Joint Conference of the Forty-Seventh Annual Meeting of the Association for Computational Linguistics and Fourth International Joint Conference on Natural Language Processing of the Asian Federation of National Language Processing*, pp. 1003–1011, 2009.

[19] S. Mittal, P. Das, V. Mulwad, A. Joshi and T. Finin, CyberTwitter: Using Twitter to generate alerts for cyber security threats and vulnerabilities, *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 860–867, 2016.

[20] V. Mulwad, W. Li, A. Joshi, T. Finin and K. Viswanathan, Extracting information about security vulnerabilities from web text, *Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, pp. 257–260, 2011.

[21] J. Smith, Cyber threat intelligence sharing – Ascending the pyramid of pain, *APNIC Blog*, June 23, 2016.

[22] spaCy, spaCy – Industrial-Strength Natural Language Processing in Python (`spacy.io`), 2021.

[23] R. Steele, Open-source intelligence, in *Handbook of Intelligence Studies*, L. Johnson (Ed.), Routledge, Abingdon, United Kingdom, pp. 129–147, 2007.

[24] Y. Tao, Y. Zhang, S. Ma, K. Fan, M. Li, F. Guo and Z. Xu, Combining big data analysis and threat intelligence technologies for the classified protection model, *Cluster Computing*, vol. 20(2), pp. 1035–1046, 2017.

[25] D. Zeng, K. Liu, S. Lai, G. Zhou and J. Zhao, Relation classification via convolutional deep neural networks, *Proceedings of the Twenty-Fifth International Conference on Computational Linguistics: Technical Papers*, pp. 2335–2344, 2014.

[26] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao and B. Xu, Attention-based bidirectional long short-term memory networks for relation classification, *Proceedings of the Fifty-Fourth Annual Meeting of the Association for Computational Linguistics: Volume 2 Short Papers*, pp. 207–212, 2016.

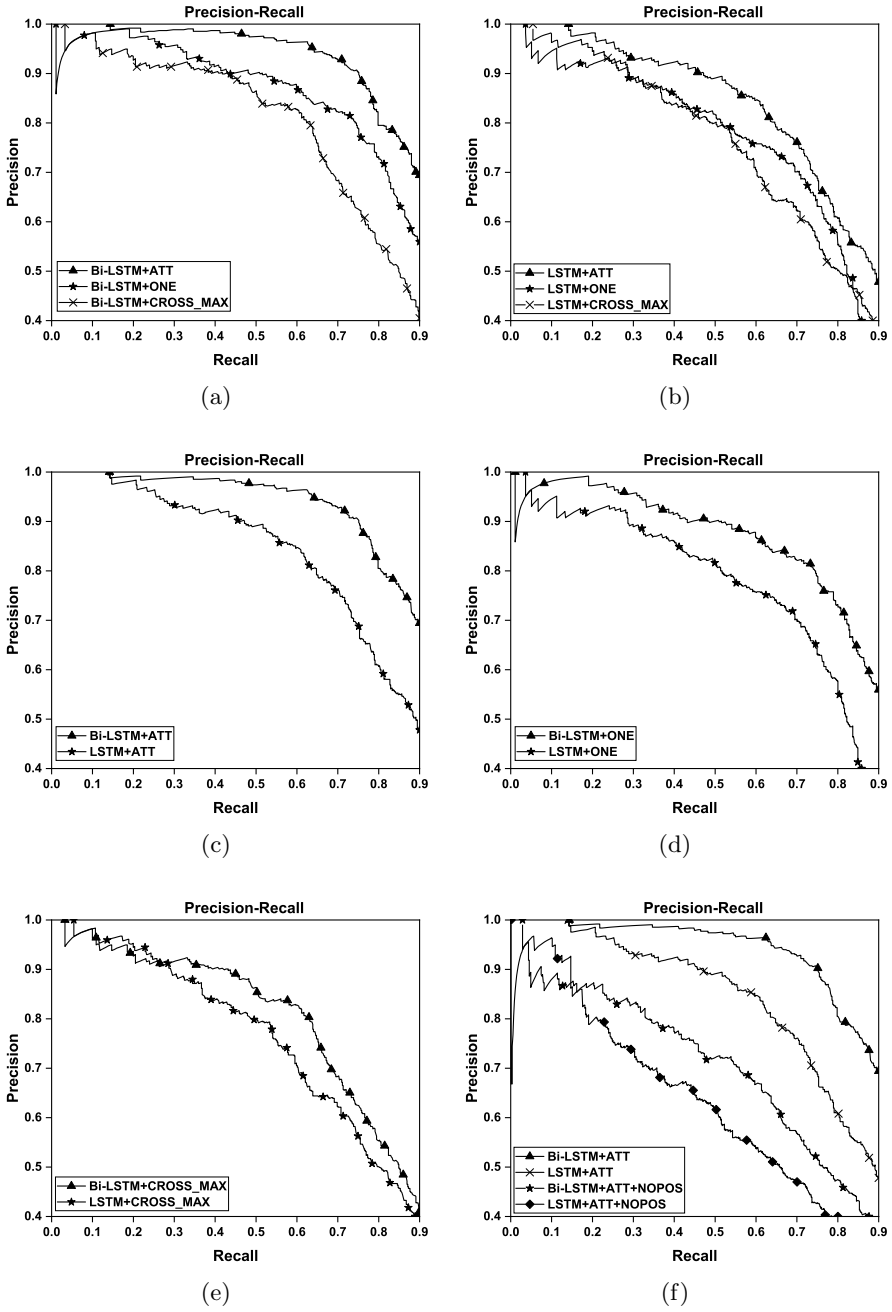

Figure 6. Aggregate precision/recall graphs for the baseline neural network models.