Open-CyKG: an open cyber threat intelligence knowledge graph

Sarhan, I.; Spruit, M.R.

Knowledge-Based Systems 233 (2021) 107524

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Open-CyKG: An Open Cyber Threat Intelligence Knowledge Graph

Injy Sarhan [a,b,*], Marco Spruit [b,c,d]

[a] Department of Computer Engineering, Arab Academy for Science, Technology, and Maritime Transport (AAST), Alexandria, Egypt
[b] Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands
[c] Department of Public Health and Primary Care, Leiden University Medical Center(LUMC), Leiden, The Netherlands
[d] Leiden Institute of Advanced Computer Science (LIACS), Leiden University, Leiden, The Netherlands

ARTICLE INFO

Article history:
Received 11 April 2021
Received in revised form 17 September 2021
Accepted 20 September 2021
Available online 21 September 2021

Keywords:
Cyber Threat Intelligence
Knowledge Graph
Named Entity Recognition
Open Information Extraction
Attention network

ABSTRACT

Instant analysis of cybersecurity reports is a fundamental challenge for security experts as an immeasurable amount of cyber information is generated on a daily basis, which necessitates automated information extraction tools to facilitate querying and retrieval of data. Hence, we present Open-CyKG: an Open Cyber Threat Intelligence (CTI) Knowledge Graph (KG) framework that is constructed using an attention-based neural Open Information Extraction (OIE) model to extract valuable cyber threat information from unstructured Advanced Persistent Threat (APT) reports. More specifically, we first identify relevant entities by developing a neural cybersecurity Named Entity Recognizer (NER) that aids in labeling relation triples generated by the OIE model. Afterwards, the extracted structured data is canonicalized to build the KG by employing fusion techniques using word embeddings. As a result, security professionals can execute queries to retrieve valuable information from the Open-CyKG framework. Experimental results demonstrate that our proposed components that build up Open-CyKG outperform state-of-the-art models.[1]

1. Introduction

Cyber threats are developing at a rapid pace, which is driving security analysts to dynamically utilize various Natural Language Processing (NLP) techniques as means to defend, identify, analyze, and possibly mitigate various cybersecurity attacks. These include text memorization [1], information extraction [2,3] and Named Entity Recognition (NER) [4,5]. In order to understand the means and the consequence of different cyber-attacks, security professionals rely on previous reports, such as security bulletins or online reports, to get a better grasp of the threat at hand. Unfortunately, such reports are often stored in an unstructured manner, making efficient information retrieval even more challenging.

Currently, existing information extraction systems lack two essential components. First, a methodology that is capable of extracting valuable information that does not necessitate either a pre-defined set of relations or an existing ontology [6], limiting extraction to a specified set of information, thus increasing the probability of missing out on vital knowledge. Second, a data structure that supports storing extracted data efficiently to allow successful information retrieval and knowledge understanding. The absence of this kind of data structure will prevent security analysts to fully leverage the extracted information.

In this paper we introduce *Open-CyKG*: an open Cyber Threat Intelligence (CTI)[2] Knowledge Graph (KG) constructed from Open Information Extraction (OIE) triples. Open-CyKG is a framework that is capable of efficiently extracting valuable information from unstructured Advanced Persistent Threat (APT) reports and representing the retrieved data in a KG that offers efficient querying and retrieval of threat-related information. Open-CyKG is made up of two main components as shown in Fig. 1. First, an attention-based OIE architecture for extracting domain-independent relational triples from unstructured data. Second, a NER model for automatic labeling of cybersecurity terms. More precisely, we start by extracting structural relation tuples from APT reports using OIE, which are later populated in the KG with the help of the NER task.

Attention mechanisms have had notable success in several deep learning tasks [8–10]. The first building block is an attention-based OIE. We propose a novel attention mechanism that emphasizes the syntactic and semantic features of a given sentence, in a way that words are assigned different weights based on their level of contribution to a sentence. We demonstrate that

[*] Corresponding author at: Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands.
E-mail addresses: i.a.a.sarhan@uu.nl, injy.sarhan@aast.edu (I. Sarhan).

[1] Our implementation of Open-CyKG is publicly available at https://github.com/IS5882/Open-CyKG.

[2] CTI is the outcome of threat information once it has been compiled and analyzed to provide actionable advice regarding previously known or emerging threats that helps with the mitigation process [7].

https://doi.org/10.1016/j.knosys.2021.107524
0950-7051/

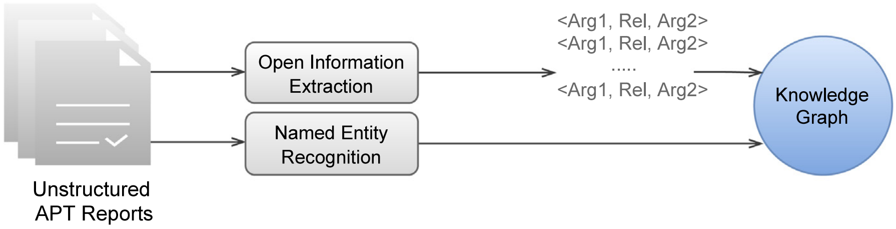


Fig. 1. Main building components of the proposed Open-CyKG framework. A more detailed version is shown in Fig. 2.

an attention-based approach improves the process of identifying semantic relations. The extracted tuples are composed of a predicate and a set of attributes in the form of <*Argument 1, Relation, Argument 2* >.

The second component in Open-CyKG is NER, which acts as both a stand-alone NLP application and a pre-processing phase for several NLP tasks including information extraction, question answering, and KG construction [11,12]. It has been significantly researched in different domains but only a few studies have targeted the cybersecurity domain. We demonstrate the importance of a NER module in KG construction and refinement.

One of the major challenges faced during the building process of a KG is data redundancy and ambiguity. Consequently, to overcome this challenge we employ refinement and canonicalization techniques to fuse information in the KG based on their contextualized word embeddings by using hierarchical agglomerative clustering for entity grouping. Various empirical analyses to validate the components of Open-CyKG were carried out demonstrating that OIE can highly support the development of knowledge bases. We address the above challenges by proposing a novel and open cybersecurity KG model. The contributions presented in this work involving different stages of Open-CyKG are as follows:

- We contribute the first OIE-based KG in the cybersecurity domain that does not limit extractions to a pre-specified set of information. Our model integrates OIE and NER with KG fusion techniques to produce an effective open cyber threat intelligence KG model: Open-CyKG.
- We introduce an attention-based sequence-to-sequence OIE model that outperforms state-of-the-art network architectures, hereby demonstrating its effectiveness in information extraction tasks.
- We develop a cybersecurity NER model to label prominent words in this domain that achieves notable results when compared against several baselines and state-of-the-art models.
- We conduct a refinement and fusion process in Open-CyKG which uses the generated NER labels and contextualized word embeddings to further enhance the quality of the retrieved queries.
- We show that once Open-CyKG is created, information retrieval can be performed efficiently using two sample queries.

The remainder of this paper is structured as follows. Section 2 reviews previous work in OIE, NER, and KG, followed by our proposed framework Open-CyKG in Section 3, while Section 4 presents results and evaluation of Open-CyKG. Finally, Section 5 concludes the paper along with future work discussion.

2. Related work

In this section, we review previous work performed on Open Information Extraction (OIE), Named Entity Recognition (NER), and Knowledge Graphs (KGs) in the literature with underlining previous work of the aforementioned tasks in the field of cybersecurity.

2.1. Open information extraction state-of-the-art

OIE methodologies can be classified into three main categories [13]: machine-learning classifier approaches, hand-crafted rules approaches, and neural network approaches. The first two approaches can be further classified into two categories, either deploying shallow syntactic analyses or dependency parsing techniques. In this section, we focus on neural network approaches and previous work in the cybersecurity domain.

2.1.1. Neural network approaches

Deep neural network approaches have proven their reliability and success on a wide range of NLP tasks and recently made their way towards OIE systems as an alternative to feature-based methods which are considered both time and effort-consuming to correctly capture entities and linguistic features. In 2018, Cui et al. [14] developed a Recurrent Neural Network (RNN) encoder–decoder OIE framework that is constructed from a 3-layer Long Short-Term Memory (LSTM) [15]. By collecting training data from high confidence state-of-the-art OIE systems, a variable-length sequence is inputted to the encoder. Subsequently, the resulting compressed representation vector is used by the decoder to produce the output sequence. Additionally, in the same year, Stanovsky et al. [16] proposed a supervised OIE paradigm that utilizes a Bidirectional LSTM (Bi-LSTM) transducer to train the neural network for tuples extraction, authors also validate that OIE can immensely benefit from an automatic question answering-semantic role labeling extractor. In [17,18] a Bidirectional Gated Recurrent Units (Bi-GRU) OIE model was introduced that leverages contextualized word embeddings.

SpanOIE [19] is the first span OIE model that adapts the same idea of modified span selection that is employed in co-reference resolution, syntactic parsing, and semantic role labeling. The span model's key benefit is visible when applied to token-based sequence labeling models in which span-level syntactic information can be adequately exploited. Authors emphasize that features of span level support better extraction quality. Cabral et al. introduced CrossOIE [20], a multilingual OIE model that deploys convolution neural networks that support extractions in English, Spanish and Portuguese. The cross-language OIE model employs a binary classifier that generates training features from cross-language embeddings, hereby highlighting the importance of developing cross-lingual OIE systems as research is more focused on the English language only.

2.1.2. Information extraction in cybersecurity

To the best of our knowledge, no previous work has been performed that specifically explores OIE in the field of cybersecurity. However, several other information extraction techniques targeting the cybersecurity domain were recently proposed. In 2019, Gasmi et al. [4] proposed an LSTM-based Relation Extraction (RE) system for mining predefined cybersecurity entities from text. Contrasting to OIE, in RE task relations must be predetermined prior to extraction. The authors focus mostly on identifying relations that relate to vulnerabilities that link software with vendors or specific files. Another RE framework was introduced by Jones et al. [3] that applies a bootstrapping pattern-based approach for tuple extraction. Their model integrates active learning components that query the user to supply precise input into the system. Similar to Gasmi's RE model, Jones et al. [3] use predefined relations that correspond to attributes of vulnerability and software which are derived from a cybersecurity ontology, both RE models aim for a better understanding of vulnerability-related information.

2.2. Named entity recognition state-of-the-art

NER is a well-researched topic in several different domains, where news and biomedical fields dominated the research in the NER task compared to other less-popular fields like cybersecurity. It is the task of identifying and locating named entities in unstructured text corpora and classify them into a predetermined set of categories like location, person, organization, date, and time expressions. Over the last few years, Neural Network (NN) techniques have taken over the lead in NER systems as they minimize the need for human effort in constructing features and rules necessary for achieving a decent level of accuracy. In this section, we review the innovative NN approaches.

2.2.1. Neural network approaches

Deep learning systems require minimal feature engineering without essentially requiring lexicons or ontologies, thereby making them more domain independent. Amongst the first NN-based NER systems introduced in 2008 is [21], where a single Convolutional Neural Network (CNN) architecture was utilized to create a multi-tasking learning system that predicts named entities, POS tags, and semantically similar words. Additionally, the authors demonstrated that simultaneous task-learning enhances the model's generalization. GRAM-CNN [22] is another NER approach that uses CNN for biomedical entity extraction. The authors of [22] used character embeddings instead of word embeddings to get a more informative representation of the words, where labels are predicted via a Conditional Random Field (CRF) layer.

RNNs paved their way to the NER task by either employing LSTM or GRU networks. CharNER [23] is a character-level tagger that exploits stacked Bi-LSTM for encoding patterns; a decoder is then utilized to transform the generated character-level probability representation to word-level tags. Opposing to the character-level model, in 2016, Yang et al. [24] introduced a multi-tasking, language-independent NER model that concatenates both character-level and word-level features. To encode both of the aforementioned features, a hierarchical GRU is utilized before passing its output to a CRF layer for sequence tagging prediction.

2.2.2. Named entity recognition in cybersecurity

As the demand for automatic text processing and information extraction relatively increased in all domains, NER found its way to the cybersecurity field. Nonetheless, from a technical perspective, NER systems introduced for the cybersecurity domain are highly comparable to the aforementioned systems.

Bridges et al. [5] implemented a maximum entropy model for labeling named entities on three different cybersecurity datasets: Microsoft Security Bulletin, National Vulnerability Database (NVD), and Metasploit Framework database, all of which were made publicly available by the authors. Average perception is trained on a fragment of the datasets, while constantly monitoring successful entity classifications. Moreover, unigram and bigram features were also included.

Kim et al. [25] built a NER system using a deep Bi-LSTM-CRF neural network to automatically extract named entities of cyber threats. The key idea is to incorporate several features in their proposed model, mainly characters based on Bag-Of-Character (BOC) representations. Their predefined named entities consist of a diverse set of cybersecurity terms such as malware, hash, and Common Vulnerabilities and Exposures (CVE). Their model adapted character-level features, and additionally, GloVe [26] was utilized to embed words.

In addition to the cybersecurity RE model proposed by Gasmi et al. [4] – discussed in Section 2.1.2 – they also introduced a NER system that exploits a Bi-LSTM-CRF neural network similar to the model proposed by [25] with the exception of adapting word-level features instead of character-level ones. Alike [5], the NVD dataset was employed for training and testing purposes. Another deep learning approach that combines Bi-GRU with CNN was designed by Simran et al. [27]. The Bi-GRU layer polishes the vectors prior to feeding them to the CNN layer, where features are fine-tuned before passing them to the CRF prediction layer.

2.3. Knowledge graph overview

With the arrival of the automatic information extraction and question answering era, KGs[3] fulfilled the need of effectively mining structured knowledge from exhaustive texts. In this section, we start by briefly presenting the most popular KG applications and other NLP tasks that can benefit from KGs, followed by walking through different methods to build KGs and different canonicalization techniques. Finally, we review previous KG work in the cybersecurity domain.

2.3.1. Knowledge graph applications

The first KG was introduced by Google [28] in 2012, with the main objective of enhancing query results and further enriching the overall search experience of end-users. This was the start that ignited research in KG and the development of other KG-based applications. DBPedia [29] is a well-known multilingual KG project that permits users to retrieve information through semantic queries; data in DBPedia is mainly acquired from Wikipedia infoboxes. Freebase [30] is a collaborative knowledge base where community members compose the data, also described as*"an openly shared database of the world's Knowledge"*. It is worth noting that Freebase powered a part of Google's KG, however, it went offline in 2016 and was succeeded by Wikidata [31].

In addition to the aforementioned applications, KG also aided several NLP tasks, from information extraction [32,33] and question answering [34] to recommendation systems [35].

2.3.2. Knowledge graph construction and canonicalization

There are several manners to construct a KG. It can be curated from existing knowledge bases like YAGO [36] and Wikipedia, where the latter was mainly used in building DBPedia [29], or the KG can be populated and modified by users as in Freebase [30] and Wikidata [31]. A third option is using information extraction techniques to obtain data from unstructured or semi-structured text to create the KG. As stated in [37], whichever of the three methods is utilized to build the KG, it will never be entirely correct or complete. As is the case in DBPedia, although it has almost 4.6 million entities, only half of them include fewer than five relations. Hence, KG canonicalization is required to overcome

[3] Knowledge Graph can be defined as a form of a data structure, which is composed of nodes and edges that are leveraged as a way to manage and illustrate information in such manner that users can efficiently query and obtain data on a specific topic.

this challenge by employing fusion and refinement techniques to improve the overall quality of the KG, which might result in a trade-off between accuracy and coverage of the KG [38].

In [39], an attribute character embedding that is formed on representation learning is created. The model uses the aforementioned embeddings to distinguish similarities between entities in a KG. Additionally, transitivity rules are applied to further enhance the attributes of an entity and assist in the entity linking process. Another way to ensure that similar entities and relations in a KG lie in the same space is by using entity descriptors as deployed by Zhong et al. in [40]. The alignment model produced by the authors of [40] does not require dependencies on specific data sources, therefore it can be integrated into any KG as long as the entities are identified. Entity linking is another way of canonicalization, by mapping entities in the text to existing entities in a KG as in [41,42]. While entity disambiguation is deemed as a sub-task of entity linking, it is the process of linking the identified entity in a KG to a ground truth entity as in [43,44].

2.3.3. Knowledge graphs in cybersecurity

Following the same trend of the two aforementioned NLP tasks, OIE and NER, KG in the cybersecurity domain is one of the most under-researched domains compared to the more popular fields such as news and biomedical domains. Narayanan et al. [45] built a collaborative framework with the help of semantically rich knowledge representations. Their cognitive assistant system designated for early detection of cybersecurity attacks acquires vulnerability-related data from recently published threat intelligence reports from multiple sources such as online blogs and CVE reports, whereas information is later illustrated in a preconstructed KG that is previously loaded with information such as early detected threats, attack patterns, and tools required to carry out an attack.

As an alternative to constructing a KG from data about vulnerabilities, Piplai et al. [12] populate a cybersecurity KG from malware After Action Reports (AAR), as they enclose insightful analyses of cybersecurity incidents, hereby delivering reliable information to security analysts. As AARs provide crucial data about detection and mitigation techniques of attacks, they can also aid in dealing with new unidentified cybersecurity incidents by matching pattern similarities with a predefined incident. Additionally, to ease the extraction phase, a traditional malware entity extractor that is based on Stanford NER [46] was created, that was trained on CVE and security blogs to label each word accordingly.

A further cybersecurity KG that is constantly maintained is SEPSES, introduced by Kiesling et al. [47]. SEPSES encapsulates and relates essential information ranging from vulnerabilities to attack patterns and weaknesses. Data in the KG is populated from several sources and amendments are instantly incorporated in the real-word, for example, CVE data is continuously fed to their model and updated every two hours, which is valuable in capturing alerts caused by intrusion detection systems in parallel with providing updated vulnerability assessments.

Nevertheless, the aforementioned work in this section either depends on structured text to populate the KG or limits extractions to a predefined set of information. For the work of [12], the authors employ RE while authors of [47] completely rely on Apache Jena for the triple formulation of specific relations. This further demonstrates the strength of Open-CyKG, as it employs OIE so it is not restricted to a predetermined set of relations or an ontology to extract information from cybersecurity reports.

3. Open-CyKG framework

Our Open-CyKG framework is presented in Fig. 2. The pipeline is composed of three main modules; a neural OIE system to extract relation triples from unstructured APT reports, a cybersecurity NER model that identifies and classifies each word according to a predefined set of labels, and a KG construction and fusion phase where extracted triples from the OIE phase are illustrated. The KG is constructed such that the extracted entities represent the KG nodes and edges correspond to the extracted relations that couple the entities.

3.1. Neural OIE model

Our OIE model is schematically presented in Fig. 3. It is an upgrade on our previous OIE work described in [17] and [18]. We tackle OIE as a sequence labeling task using the BIO (Beginning, Inside, and Outside) labeling scheme [48] in such a way that the resulting outcome is a set of overlapping tuples for each sentence.

Due to the fact that RNNs have the capacity of storing information in their hidden units, they are considered a suitable choice for handling sequential data when compared to feed-forward artificial neural networks like CNNs that also struggle to capture long-distance dependencies between words. Nevertheless, RNNs are harder to train with longer sequences owing to vanishing and exploding gradient descent complications, which leads the performance to noticeably degrade. As RNNs are trained by back-propagation through time, the further we back-propagate through several time steps, the smaller the gradient gets up until it vanishes or explodes. As a solution, extended versions of RNNs were introduced – LSTMs and GRUs – to mitigate vanishing gradient issues by deploying appropriate gates to permit the gradient to flow effectively while maintaining long-term dependencies [15,49]. GRUs are regarded as a less complex variation of LSTMs, both are built on the gating concept, where LSTM's architecture is compiled of three gates; input, output and forget while GRUs couple the input and forget gates into a single update gate. Our choice of deploying GRUs instead of LSTMs is also motivated by the fact that GRUs utilize fewer training parameters, resulting in quicker execution and training times in contrast to LSTMs.

In addition to embedding the word and its corresponding POS, and passing them as an input to our neural network, the feature vector is further enriched by passing the predicate of the phrase, as predicates are regarded as the building block of any sentence. The input feature vector $(F.V)$ is represented in Eq. (1).

$$F.V = (Emb(w) \oplus Emb(POS(w)) \oplus Emb(w_{Pred}))|w \in S \qquad (1)$$

Where $\oplus$ represents the concatenation of the 3 inputs: word denoted as w, its corresponding POS obtained using the NLTK toolkit [50] $POS(w)$ and the predicate of the sentence w_{Pred}, where each word belongs to a sentence S. All the prior mentioned inputs are embedded Emb using contextualized word embeddings as discussed in Section 4.2.1.

The embedded feature vector is then passed to two Bi-GRU layers. As shown in Fig. 3, Bi-GRU is composed of two GRUs operating in a reverse direction, the significance in utilizing a Bi-GRU rather than a single direction GRU is that information is captured in both directions, forward and backward during each time-step to efficiently perform sequence labeling.

The outputted tensor from the Bi-GRU layer is then passed through an attention layer [51] that is based on an additive attention module. Most of the proposed OIE neural models discussed in Section 2.1.1 are formulated in a way such that all words have the same level of importance, however, it is important to highlight that not all words in a sentence have an equivalent level of contribution in the OIE task. To address this issue, we employ the attention mechanism in our network to learn the varying significance of words in each phrase and aggregate the output to the following double-layered Time Distributed Dense (TDD) layer that employs a consistent dense layer to the passed tensor at each time step. As a final point, the tensor produced by the TDD layers is fed into a SoftMax layer, where the output is an individual probability distribution covering all possible tags.

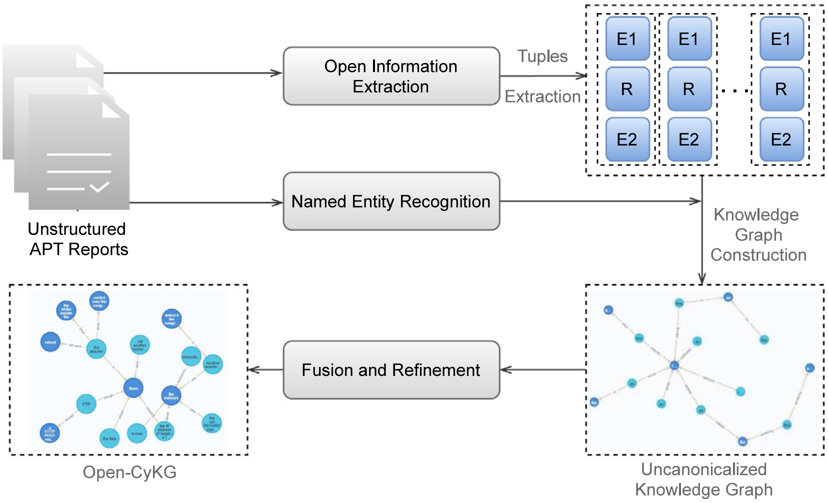


Fig. 2. The Open-CyKG pipeline is composed of three primary building blocks: OIE, NER modules, as well as fusion and KG refinement techniques that result in a canonicalized open KG.

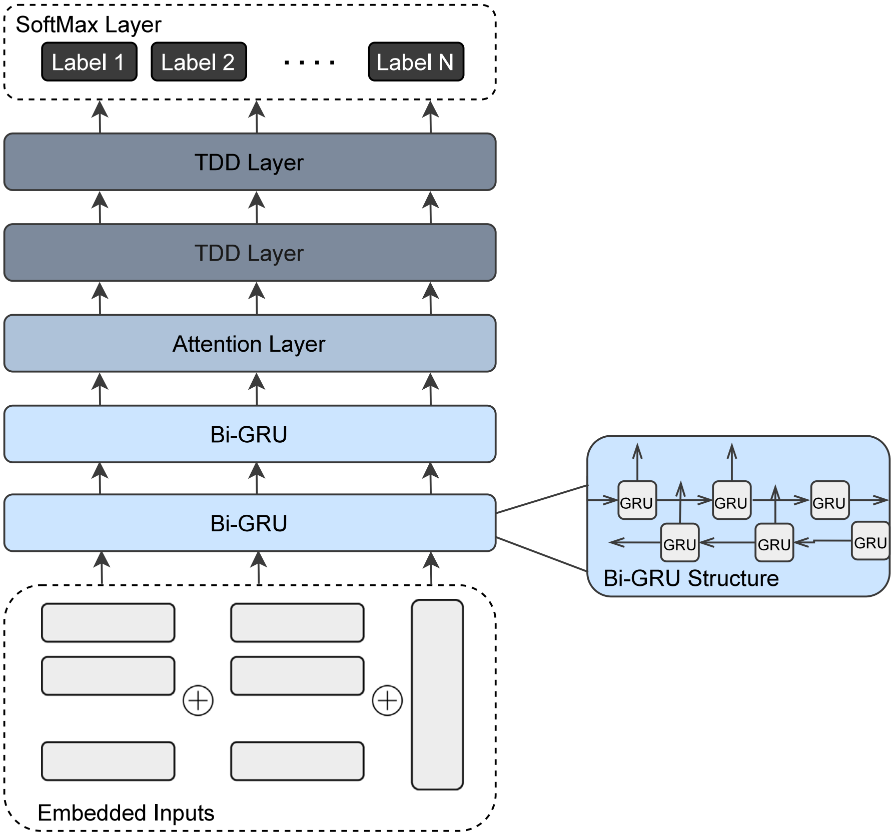


Fig. 3. Our OIE model takes the concatenation of all inputs and passes the input to the two Bidirectional Gated Recurrent Units (Bi-GRU) layers, followed by an attention layer, two Time Distributed Dense (TDD) layers, and finally, a SoftMax layer for prediction.

3.2. Cybersecurity-NER

The task of classifying cybersecurity entities in our dataset resembles the efforts of prior research discussed in Section 2.2.1, however, with a differently designed neural network as illustrated in Fig. 4. In a similar manner to our OIE approach, we formulate the NER task as a sequence labeling problem with BIO taggers, as it is considered the most suitable tagging module for NER specifically in neural models employing CRF [52], where each word in the dataset is labeled according to a set of predefined entities based on its position.

Initially, words are translated into their respective embeddings and are progressed directly to the following layer. Since long dependencies modeling in NER is essential we also opted to deploy an RNN. To capture both, backward and forward information, our proposed cybersecurity-NER deploys a Bi-GRU layer which outputs a tensor that is later passed to a TDD layer. Finally, a CRF prediction layer labels each word in our dataset by generating likelihood distributions over every available tag.

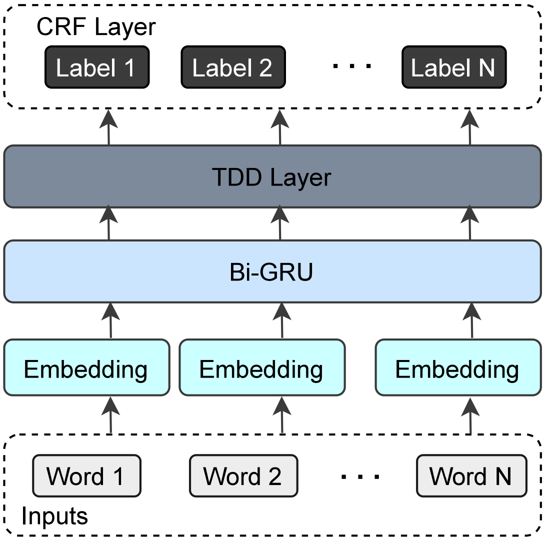


Fig. 4. Our implemented cybersecurity NER neural model architecture that is composed of four main layers; Embedding, Bidirectional Gated Recurrent Units (Bi-GRU), Time Distributed Dense (TDD), and Conditional Random Field (CRF) for label prediction.

3.3. Knowledge graph construction and canonicalization

To produce Open-CyKG, relation triples extracted from the OIE stage are processed and outlined in the KG as defined in Eq. (2):

$$KG = \{(nh, e, nt) | nh, nt \in E, e \in R\} \tag{2}$$

Where the set of the extracted OIE triples (nh, e, nt) are composed of a node head nh and a node tail nt in which both belong to the entities E, both nodes are linked together using an edge e that represents the relation R that lies between the two entities. Additionally, named entity tags are allocated to each node as a property. An uncanonicalized sample of the generated KG using Neo4J [53] is illustrated in Fig. 5.

Several sources of information are used when constructing a KG, which will possibly prompt duplication. As a result, it is essential to apply refinement and fusion techniques to address this matter. The leading step is triple refinement: this two step-process involves removing redundant and vague information, and entity blending where identical entities are merged together after identifying and removing non-essential words to preserve only informative entities. The filtration task also involves eliminating uninformative triples generated from the OIE phase, in which all words forming the three extracted components are not assigned any named entity labels from the cybersecurity NER phase.

Another common setback in the construction process that is not captured in the previous step is entity disambiguation, which can be perceived in two contradictory ways, the first is ensuring that an entity represents the same semantic concept to all its connected nodes, while the second is unifying and merging entities that represent identical concepts. Entity disambiguation in KG is considered a research problem on its own that is out of the scope of this paper. Nevertheless, we briefly attend to this issue by performing entity fusion using contextualized word embeddings to capture the semantics of entities. In our work, we experiment with several word embeddings discussed in detail in Section 4.2.1.

The potential of using word embeddings in Open-KG canonicalization to address ambiguity in the generated knowledge graph has been demonstrated by the works of [38,54]. By first averaging the generated word embeddings of all subjects in an entity, we cluster entities by carrying out Hierarchical Agglomerative Clustering (HAC) by employing cosine similarity as a distance metric. Our choice behind HAC is motivated by the fact that it does not require predefining the number of clusters in advance. Additionally, it supports complete linkage clustering, similar to the concept of farthest neighbor clustering, where initially each average embedding is a cluster on its own, and in each step the two clusters having the smallest maximum pairwise distance are merged. The complete linkage clustering is fitting in KG canonicalization as demonstrated in [38] as small-sized clusters are expected as opposed to single and average linkage clustering. Fig. 6 illustrates the canonicalized version of Fig. 5, where nodes are merged after the clustering process.

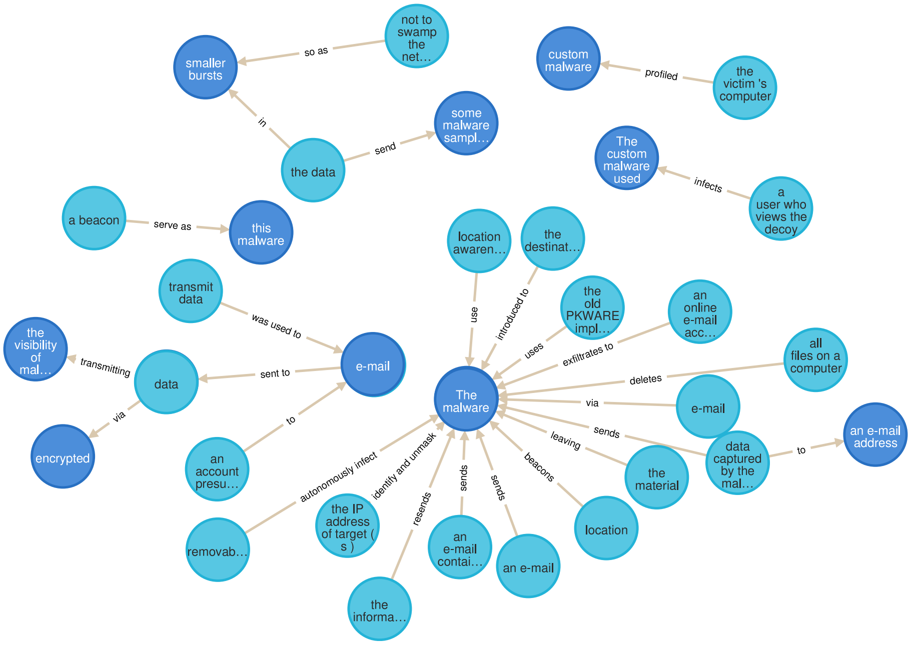


Fig. 5. An uncanonicalized sample of the created KG using Neo4J.

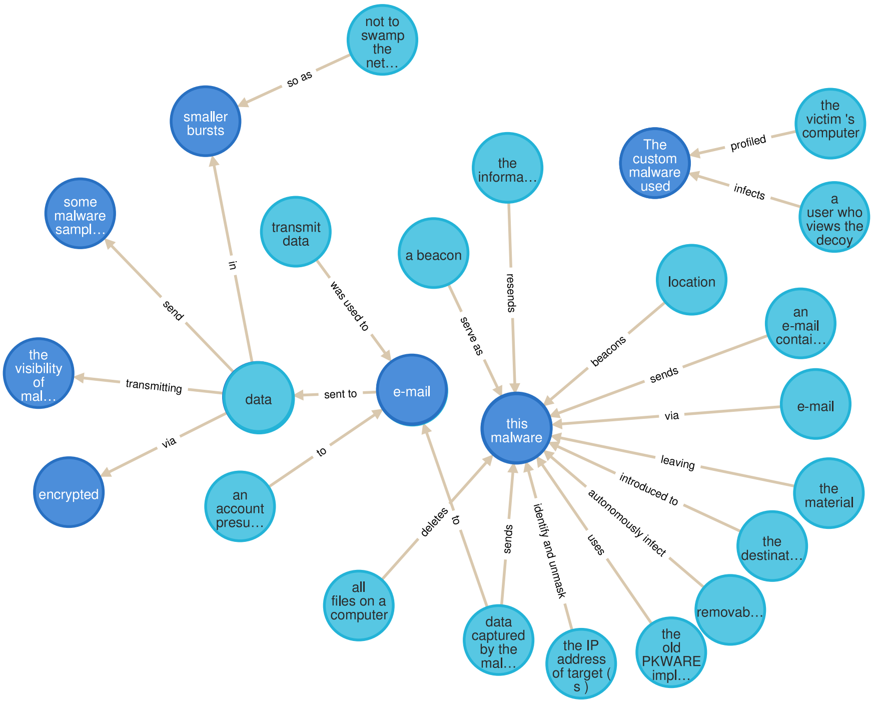


Fig. 6. Canonicalized version of Fig. 5 using Neo4J.

The final phase is determining a representative for each cluster. In line with the work of [38] we calculate the mean of all the generated elements' embeddings weighted by the number of occurrences of each element in the input. The entity with the minimum distance to the weighted cluster mean is selected as a representative.

To further clarify the importance of canonicalization in addressing ambiguity and redundancy, consider the following two triple extractions: <*Barack Obama, born in, Hawaii*> and <*Obama, served as, 44th U.S President*>. In an uncanonicalized version of the KG, the two extractions would be included separately without any connecting edges, as *Barack Obama* and *Obama* are perceived as two distinct entities. This may lead to a remarkable impact when querying data from the KG as it will not return all information linked with Barack Obama. Such KGs will also suffer from redundant facts, which is undesired. Canonicalizing KGs using HAC clustering as described above guarantees relation transitivity, that both entities —*Barack Obama* and *Obama*— are fused to represent a single entity. Several other canonicalization approaches and entity linking techniques are proposed in [55,56].

4. Results and evaluation

In this section, we report the utility of Open-CyKG. KG curation is the task of assessing the value of the constructed KG, this process is commonly fulfilled by experts in the field. However, nowadays, it is deemed a tedious task to be done by humans, especially with densely populated KGs, or even more complicated ones incorporating several domains [57]. Nonetheless, KG validation is still an open challenge, hence we address this matter by evaluating each component in our model separately to reflect the quality of Open-CyKG. We also present a set of auxiliary experiments to further analyze our proposed OIE and NER models. All our experiments were implemented using the Keras framework [58] with the TensorFlow backend [59]. We start by describing the dataset used to build the KG in detail along with its inherent constraints.

4.1. MalwareDB dataset

As the world is digitally growing, devices are more prone to malware attacks which might lead to misfortunate events ranging from unauthorized access to personal data to device damage. MalwareDB [60] is an annotated dataset based around Malware Attribute Enumeration and Characterization (MAEC) vocabulary that primarily outlines malware characteristics gathered from 39 APT reports. Fig. 7 shows a sample from the aforementioned dataset with the extracted triples from the OIE task.

OIE is the primary building block in Open-CyKG to construct the KG, so our main objective is to effectively identify relation triples necessary for successful querying. Hence, training data is crucial in our work. Unfortunately, one of the ongoing challenges is the lack of BIO-labeled data, specifically in understudied domains such as cybersecurity. Although the 39 APT reports that constitute the MalwareDB dataset originally contain 6,819 sentences, we were only able to classify 1,910 sentences as informative sentences, which challengingly formed our training, test, and validation sets. Uninformative sentences can be defined as:

- Sentences that are composed of 'O': Outside labels only.
- Phrases without any relationship labels.
- Sentences that contain only a single entity.

Yet, currently there is no alternative dataset available in the cybersecurity domain with BIO labeling.

4.2. Experimental results and analysis: OIE

In this section we assess the outcome of our proposed attention-based OIE model, we follow the framework configuration and dataset as discussed in Sections 3.1 and 4.1 respectively.

4.2.1. Word embeddings

In recent years various types of embeddings have been proposed, varying from character and word embeddings to sentence and document embeddings. Nonetheless, they all provide the

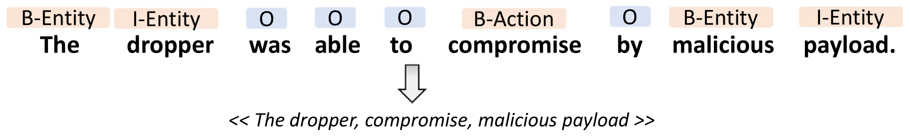


Fig. 7. An example of OIE performed on a sentence from APT reports, where the Action tag represents the relation that links the two entities together.

Table 1
Four embedding techniques and their respective dimensionality employed in our OIE model.

Embedding technique	Vector dimensionality
GloVe [26]	100
BERT [62]	3072
XLNet [63]	2048
XLM-RoBERTa [64]	1024

same function of mapping textual input to semantically meaningful vector representations. The innovative contextualized word embeddings are capable of capturing dense semantic and syntactic features of a word, by incorporating context into the generated embeddings. In our OIE task, three inputs are embedded and fed to the network, making the choice of embedding fundamental, hereby we opted to experiment using different word embedding techniques, by selecting one conventional non-contextualized embedding – GloVe [26] – and three contextualized embeddings with varying dimensionality and trained on a diverse set of domains. In order to carry out our experiments, we utilized Flair [61], a powerful open-source framework developed by Zalando Research that provides a unified interface for a wide range of state-of-the-art word, document, and sentence embeddings. The employed embeddings along with their dimensionality are shown in Table 1.

4.2.2. Experiment and evaluation on MalwareDB

To assess the competence of our model we have analyzed it rigorously with different experimental setups and word embeddings. As observed in Table 2, Bi-GRU achieves an overall higher score than Bi-LSTM neural network models. More precisely, our Bi-GRU + Attention model scores an F-measure of 59.4% which is 2.2% higher than when using a Bi-LSTM + Attention network, both achieved the highest score with XLM-RoBERTa embeddings.

We performed an ablation study to measure the effectiveness of the attention mechanism by testing on a Bi-GRU network which resembles the model introduced by Sarhan et al. [17] and [18]. By removing the attention component, the Bi-GRU model achieved an F-measure score of 56.8%, verifying the impact of deploying the attention mechanism as it contributed to a 2.6% increase in F-measure. Despite the fact that GRUs and LSTMs capture long-range dependencies better than traditional RNNs, they do not have the ability to direct the focus to some of the input words to point out the words that are important to our task, which further demonstrates the importance of deploying attention mechanisms in information extraction tasks.

To further evaluate the potential of our attention-based OIE model, we compare our model against yet another prior state-of-the-art neural OIE network that is composed of Bi-LSTM as proposed by Stanovsky et al. [16]. Alike the comparison to the previous state-of-the-art, our model was able to achieve a higher F-measure by 4.2%. BERT embeddings attained the highest results in both networks, Bi-LSTM and Bi-GRU.

The rationale behind Bi-GRUs performing better than Bi-LSTMs is due to GRUs' ability to expose the complete memory, unlike LSTMs. Additionally, LSTMs have more gates than GRU, which causes the gradients to flow through which leads to steady progress being more complex to maintain after many epochs [65]. Another interesting conclusion that can be drawn from our ablation study is the adoption of an attention mechanism to fully leverage the bidirectional context information as it is also elaborated by the authors of [66] and [8].

It should be emphasized that despite the fact that achieving a decent recall and precision would be the optimal situation, precision is more crucial in our work as it reflects the certainty of the extracted information. In KG false positives are expensive to maintain, in a setting of a high-scoring recall and a lower precision, the KG would be populated with uninformative or false information which will result in a less-efficient querying experience. Nevertheless, it is important to note that the highest precision achieved by our model was 62.9% using BERT, trading-off to a lower recall of 54.7%, which resulted in a decrease of 0.9% in F-measure when compared to our highest achieving score of 59.4% reported in Table 2.

Our model's results are sensitive to hyperparameter alternations, thus a grid search was performed to single out the optimal number of training epochs and batch size. The hyperparameter configurations that realized the best results are reported in Table 3. The hidden size of all Bi-GRU layers is set to 128, which is also the same number of units used in the two TDD layers. For regularization reasons to prevent over-fitting, the dropout rate is adjusted to 0.1. Moreover, early stopping is employed to terminate training based on the model's performance on the development set. Furthermore, a linear activation function, Rectified Linear Unit (ReLU) [67] was applied in the two TDD layers, while the Adam optimizer [68] was utilized to train our model.

Additionally, as the limited size of the training set influences the neural network's performance, we were able to further evaluate our proposed attention-based OIE model by experimenting on a larger annotated news dataset [69], which is composed of 2906 training sentences. An increase of 3.2% in F-measure was achieved, emphasizing that the limited size of training data indeed plays an important role.

4.3. Experimental results and analysis: NER

As the MalwareDB dataset has no named entities annotation we could not train or evaluate our model based on MalwareDB extractions. In this section, we will discuss the datasets used to train and validate our NER neural network described in Section 3.2.

4.3.1. Dataset

To perform NER, necessary for refinement and labeling of KG nodes, we used two different training sets, each containing a diverse set of labels. The output from the two-pass process is then merged to assign labels from the two datasets to named entities. The first dataset Microsoft Security Bulletins, utilized in [5], is also annotated and made widely accessible by the authors. It is composed of 5072 sentences discussing vulnerabilities and security flaws in Microsoft's software products along with various patch and mitigation information. A random sample drawn from the Microsoft Security Bulletin dataset is shown in Fig. 8. The second dataset employed in [25] is a malware-specific dataset collected from various CTI reports resulting in a total of 3450 sentences with predefined named entities that relate to malware,

Table 2
Results of our Open-CyKG: OIE model (Open-CyKG$_{\text{Bi-GRU+Att}}$). Recall, precision, and F-measure are used as evaluation metrics. Along with the deployed word embedding that resulted in the highest scores.

Model	Network architecture	Word embedding	Results		
			Recall	Precision	F-Measure
Sarhan et al. [17], [18]	Bi-GRU	BERT	54.9%	58.9%	56.8%
Stanovsky et al. [16]	Bi-LSTM	BERT	53.0%	57.5%	55.2%
Open-CyKG$_{\text{Bi-LSTM+Att}}$	Bi-LSTM + Attention	XLM-RoBERTa	55.7%	58.7%	57.2%
Open-CyKG$_{\text{Bi−GRU+Att}}$	**Bi-GRU + Attention**	**XLM-RoBERTa**	**57.2%**	**61.8%**	**59.4%**

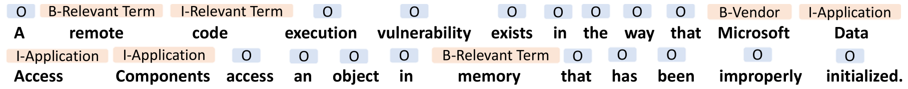


Fig. 8. Sample from Microsoft Security Bulletin dataset with the named entity tags following the BIO tagging schema.

Table 3
Hyperparameter settings used in our OIE model.

Hyperparameter	Value
Epochs	100
Batches	50
Bi-GRU	128 units
TDD Activation Function	ReLU
TDD units	128 units
Dropout Rate	0.1
Optimizer	Adam

Table 4
Results of the Open-CyKG: NER model (Open-CyKG$_{\text{Bi-GRU}}$). Both training and testing are done on the Microsoft Security Bulletin dataset. Along with the original dataset baseline results as reported in [5] and Bi-LSTM + CRF network. Recall, precision, and F-measure are used as evaluation metrics.

Model	Method	Results		
		Recall	Precision	F-Measure
Bridges et al. [5]	Hand-crafted Heuristic	75.3%	**99.4%**	77.8%
Open-CyKG$_{\text{Bi-LSTM}}$	Bi-LSTM + CRF	96.6%	97.4%	97.0%
Open-CyKG$_{\text{Bi−GRU}}$	**Bi-GRU + CRF**	**98.7%**	99.2%	**98.9%**

Table 5
Results of our Open-CyKG:NER model (Open-CyKG$_{\text{Bi-GRU}}$). Both training and testing are done on the CTI dataset, the original baseline results in [25] are reported along with the CNN-based network. Recall, precision, and F-measure are used as evaluation metrics.

Model	Network Architecture	Results		
		Recall	Precision	F-Measure
Simran et al. $_{\text{BOC}}$ [25]	BOC: Bi-LSTM + CRF	70.5%	**80.3%**	75.1%
Simran et al. $_{\text{CNN}}$[25]	CNN + Bi-LSTM + CRF	71.0%	78.9%	75.0%
Open-CyKG$_{\text{Bi-LSTM}}$	Bi-LSTM + CRF	70.4%	71.9%	71.1%
Open-CyKG$_{\text{Bi−GRU}}$	**Bi-GRU + CRF**	**80.8%**	78.9%	**79.8%**

which are considered key elements in our malware-based APT reports. The predefined tags in both datasets with their respective ratio in the training set are illustrated in Fig. 9. It is worth noting that when two labels are assigned to a single word, we select the malware-specific CTI report label since it is more closely related to our APT reports dataset.

4.3.2. Experiments and evaluation

For both datasets, we split the corpus into two partitions, 80% for training and 20% for testing, with setting apart a fraction of 0.1 of the training set for validation purposes to assess the loss at the end of each epoch. To certify the quality of our model, we performed five-fold cross-validation. We opted for stratified K-fold as it takes the cross-validation process one step further by preserving the distribution of the class in both training and test splits, to avoid unbalanced labels' distribution. To validate the efficiency of our NER network architecture, we compared our model's performance against the results of different baselines and state-of-the-art models.

Table 4 shows the complete results of our NER model that is employed in Open-CyKG on the Microsoft Security Bulletins dataset. We can clearly find that the proposed model outperforms both the baselines and state-of-the-art model by scoring a 98.5% F-measure. More precisely, when comparing our Bi-GRU + CRF results to the baseline reported by Bridges et al. [5] that employed a traditional approach using hand-crafted rules, we can see that our model outperforms by more than 20% of the F-measure score. In addition, there is an increase of 1.9% in the F-measure when we compare our proposed Bi-GRU model to the Bi-LSTM network architecture with 50 batches and 30 epochs.

It is observed that all models obtain decent precision, however the overall performance of NN models significantly outperforms hand-crafted rules. The low recall achieved by Bridges et al. [5] model contributed to this decrease, the rationale behind this is due to the large variation of expressions in natural language specifically reflected in tasks such as NER and RE as it is also shown in the work of [70].

The performance of our NER model on the second dataset – CTI reports – is presented in Table 5. Three comparisons are carried out in this experiment; the first is the baseline model that originally annotated and constructed the CTI reports dataset which employed a BOC-based Bi-LSTM + CRF network architecture proposed by Simran et al. [25]. The second is a character-based CNN that also uses a Bi-LSTM architecture. A pure Bi-LSTM + CRF network is our third comparison that is trained with the same hyperparameters of our model. As it is observed when comparing our model with the baseline, our model scored an F-measure of 79.8% which is 4.7% higher than that of Simran et al. [25], the increase was mainly reflected by an increase in the recall of our model by 10.3%. In addition to reporting the findings of their model on the CTI reports dataset, the authors of [25] reported the score of using a CNN-based Bi-LSTM + CRF network – the second comparison – which resulted in almost the same score as the BOC system and 4.8% decrease in F-measure when compared against the score of our model. Furthermore, in line with the evaluation we carried out on the Microsoft Security Bulletin dataset, we implemented a state-of-the-art Bi-LSTM+CRF network which achieved the lowest score among all other models reported in Table 5 by scoring an F-measure of 71.1% on 10 training epochs with a batch size of 50.

In addition to the reasons mentioned in Section 4.2.2 on why Bi-GRUs outperform Bi-LSTMs, in this experiment, we attribute the increase to the nature of the CTI dataset used in this NER task, which is a small dataset with long sentences. This phenomenon is also observed in the work of [71].

Table 6 states our hyperparameter configurations that realized the reported scores in Tables 4 and 5 for both datasets. Bi-GRU

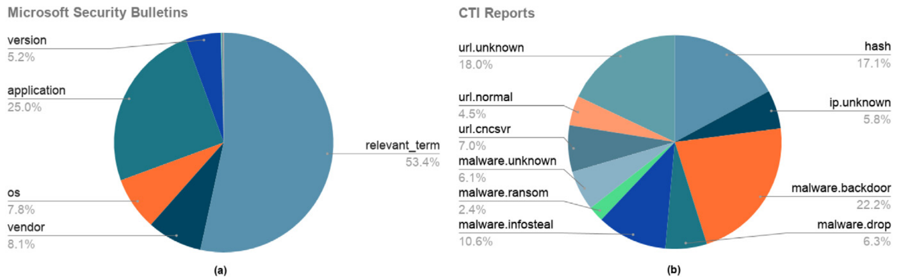


Fig. 9. (a) Labels distribution in Microsoft Security Bulletin training dataset [5]. (b) Labels distribution in CTI training dataset [25]. Note that label 'O': Outside, takes up the majority of the words' labels but it is removed from the chart for better illustration of the distribution of informative tags.

Table 6
Hyperparameter settings used in our NER model on both datasets.

Hyperparameter	Microsoft security bulletins	CTI reports
Epochs	30	10
Batches	50	30
Bi-GRU	50 units	50 units
TDD Activation Function	ReLU	ReLU
TDD units	50 units	50 units
Dropout Rate	0.1	0.1
Embedding	Keras	Keras
Optimizer	Adam	Adam

Table 7
Canonicalization results.

Metric	Results		
	Recall	Precision	F-Measure
Macro	86.5%	78.9%	82.6%
Micro	90.5%	74.4%	81.7%
Pairwise	79.6%	54.7%	64.8%

layers and TDD layers have a mutual number of units – 50 – with ReLU selected as an activation function in the TDD layer. Similar to our OIE network, the drop-out rate is set to 0.1 to prevent overfitting and early stopping is utilized. All models are trained with the Adam optimization algorithm.

Although the MalwareDB dataset has no annotated named entity tags, the results indicate that our model can effectively label cybersecurity-related terms. To further validate the performance of our NER model, a random sample of 10% was selected from the MalwareDB test set and manually annotated to compare against the model's predicted labels. When training on Microsoft Security Bulletins dataset our model achieved a recall, precision, and F-measure of 85.5%, 87.7%, and 86.6% respectively. While training on CTI reports resulted in a recall, precision, and F-measure of 83.6%, 82.3%, and 82.9% respectively.

4.4. Canonicalization evaluation

To evaluate canonicalization using contextualized word embeddings carried out in Open-CyKG, we manually construct a gold standard of clusters that represents the ground truth clusters of all extracted entities. We follow the work of [38,54,72] by using *macro*, *micro* and *pairwise* metrics to evaluate canonicalization. We concisely explain these metrics below. Let C be the clusters produced by Open-CyKG canonicalization, and G denotes the gold standard clusters.

Macro: Macro precision (P_{macro}) can be defined as a fraction of pure clusters in C formed by our approach that are linked to the same gold standard G. While Macro recall (R_{macro}) is the inverse of (P_{macro}), by interchanging the roles of C and G as seen in Eqs. (3) and (4).

$$P_{macro}(C, G) = \frac{|\{c \in C : \exists g \in G : g \supseteq c\}|}{|C|} \tag{3}$$

$$R_{macro}(C, G) = P_{macro}(G, C) \tag{4}$$

Micro: Micro precision (P_{micro}) measures the purity of the clusters C under the assumption that the most frequent gold entity of the mentions in a cluster is the correct entity [73], as depicted in Eq. (5), where N denotes the number of mentions in the input. In a similar manner, micro recall (R_{micro}) is the inverse of (P_{micro}) as shown in Eq. (6).

$$P_{micro}(C, G) = \frac{1}{N} \sum_{c \in C} max_{g \in G} |c \cap g| \tag{5}$$

$$R_{micro}(C, G) = P_{micro}(G, C) \tag{6}$$

Pairwise: A *hit* in cluster C indicates that two mentions refer to the same gold entity. Pairwise precision ($P_{pairwise}$) measures the ratio of the number of hits ($\#hits_c$) in C to total possible pairs ($\#pairs_c$) in C [38], where $\#pairs_c = |c| * (|c| - 1)/2$. Eqs. (7) and (8) define pairwise precision ($P_{pairwise}$) and pairwise recall ($R_{pairwise}$) respectively.

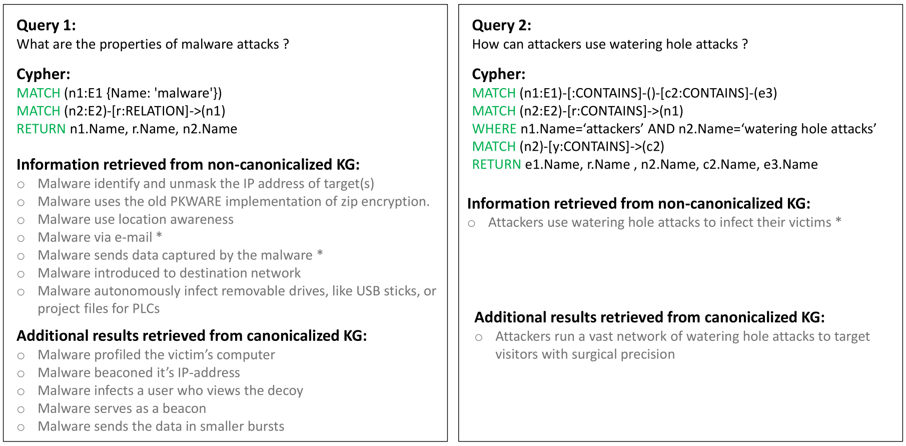
$$P_{pairwise}(C, G) = \frac{\sum_{c \in C} \#hits_c}{\sum_{c \in C} \#pairs_c} \tag{7}$$

$$R_{pairwise}(C, G) = \frac{\sum_{c \in C} \#hits_c}{\sum_{g \in G} \#pairs_g} \tag{8}$$

In all cases, F-measure is defined as the harmonic mean of the model's precision and recall. The optimal threshold value chosen for HAC clustering was decided upon using a grid search on the validation set. Due to the fact that XLM-RoBERTa was the highest-scoring language model in our OIE phase, we used the generated embeddings to compute the distance metric. It should be emphasized that the word embeddings used in the clustering phase are generated based on the whole input sentence to fully leverage the concept of contextual embeddings. Results are shown in Table 7. We observe that the recall achieved in all metrics is moderate to good, in line with canonicalization results reported in related works [38,54,72]. Nonetheless, If we look at pairwise precision, we can notice it is relatively low, which indicates that not all pairs of entities in C refer to the same gold entity.

4.5. Demonstrating information retrieval using open-CyKG

In this section, we present two sample queries, a general one targeting malware, while the other is a more specific query that focuses on watering hole attacks. To further illustrate the effect of the applied fusion technique, we perform an ablation analysis by solely applying the first phase of our refinement process as explained in Section 3.3. The ablation analysis is supported by



Query 1:
What are the properties of malware attacks ?

Cypher:

```
MATCH (n1:E1 {Name: 'malware'})
MATCH (n2:E2)-[r:RELATION]->(n1)
RETURN n1.Name, r.Name, n2.Name
```

Information retrieved from non-canonicalized KG:

- Malware identify and unmask the IP address of target(s)
- Malware uses the old PKWARE implementation of zip encryption.
- Malware use location awareness
- Malware via e-mail *
- Malware sends data captured by the malware *
- Malware introduced to destination network
- Malware autonomously infect removable drives, like USB sticks, or project files for PLCs

Additional results retrieved from canonicalized KG:

- Malware profiled the victim's computer
- Malware beaconed it's IP-address
- Malware infects a user who views the decoy
- Malware serves as a beacon
- Malware sends the data in smaller bursts

Query 2:
How can attackers use watering hole attacks ?

Cypher:

```
MATCH (n1:E1)-[:CONTAINS]-()-[c2:CONTAINS]-(e3)
MATCH (n2:E2)-[r:CONTAINS]->(n1)
WHERE n1.Name='attackers' AND n2.Name='watering hole attacks'
MATCH (n2)-[y:CONTAINS]->(c2)
RETURN e1.Name, r.Name , n2.Name, c2.Name, e3.Name
```

Information retrieved from non-canonicalized KG:

- Attackers use watering hole attacks to infect their victims *

Additional results retrieved from canonicalized KG:

- Attackers run a vast network of watering hole attacks to target visitors with surgical precision

Fig. 10. Two sample query results as retrieved from Open-CyKG..

analyzing the results of the queries to inspect the outcome of word embeddings in Open-CyKG canonicalization.

Cypher [74] is the official supported query language in Neo4j. It is an SQL-inspired declarative querying language that permits users to retrieve data from a graph database. The queries performed along with their Cypher translations are shown in Fig. 10.

In the first query, Open-CyKG was able to retrieve diverse information as long as it is directly connected to a 'malware' node. The majority of the retrieved data captures valuable insights on malware threats, whereas in some cases the extraction can be considered less informative such as*'Malware via e-mail'* , or uninformative such as 'Malware sends data captured by the malware'. In the second query, the retrieved data had to relate to both 'attackers' and 'watering hole attacks', since this query has a higher level of specificity it only results in two extractions, although *'Attackers use watering hole attacks to infect their victims.'* might be interpreted as an uninformative extraction. Nevertheless, when the queries are performed on the canonicalized version of Open-CyKG the KG was able to deliver more insights on the requested data. Additionally, as the generated named entity tags discussed in Section 4.3.1 are assigned as properties to nodes and edges, they can be leveraged to eliminate uninformative or ambiguous extractions while querying.

As observed, canonicalization using contextualized word embeddings aids in capturing more information, As a result, security analysts can query Open-CyKG to retrieve data on a specific cyber entity albeit being expressed differently among various APT reports.

5. Conclusion and future work

We introduced Open-CyKG: a novel framework that combines features from several components along with fusion techniques using contextualized embeddings to generate a knowledge graph from advanced persistent threat reports. Our proposed framework is developed from two core components, an attention-based OIE and a cybersecurity NER system.

We validated the quality of the generated KG by evaluating each component separately. We evaluated our cybersecurity NER model against several baselines and state-of-the-art models on two different datasets. Our model was able to deliver the best performance. The attention mechanism is a revolutionary theory that transformed the way researchers design neural networks. Not only does it have an essential role in various neural network-based NLP tasks to enhance performance, but it also offers important insights on how the models are operating. This has motivated the development of our attention-based OIE framework, which we validated by performing an ablation study and compared against state-of-the-art OIE models. In both cases, our attention-based model achieved the best results. Another interesting option would be a transformer-based model. A transformer-based model relies on self-attention, at each time step there is direct access to all other steps, which practically means that there is no room for information loss. However, most transformer-based models have a quadratic complexity, which limits the token length as a trade-off between performance and memory usage, resulting in truncating training sentences [75]. The authors of [76] introduced a transformer-based OIE model, however, its performance was not evaluated against any state-of-the-art neural network model. Thus as a future direction, we intend to further evaluate different neural OIE research trends including transformer-based models on benchmark datasets.

Despite their value and practicality, KGs usually suffer from incompleteness, redundancy, and ambiguity that might translate to uninformative query results. First and foremost, we are in the process of acquiring more cybersecurity data from AAR reports to carry out a large-scale experiment. In future work we will shift our attention to KG completion and link prediction to further enhance the strength of the generated KG. Additionally, we would like to explore the possibility of extending the KG model to include a dynamic reasoning component instead of completely relying on static information to build the KG. A final interesting addition would be to construct a multi-lingual or cross-lingual KG to support machine translation-based applications. All in all, we foresee that in the near future KGs will become sufficiently mature to provide added value to daily practices in cybersecurity and beyond.

CRediT authorship contribution statement

Injy Sarhan: Conceptualization, Methodology, Software, Data curation, Writing – original draft. **Marco Spruit:** Conceptualization, Writing - review & editing, Validation, Funding acquisition, Investigation, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was made possible with funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 883588 (GEIGER). The opinions expressed and arguments employed herein do not necessarily reflect the official views of the funding body.

References

[1] E.R. Russo, A. Di Sorbo, C.A. Visaggio, G. Canfora, Summarizing vulnerabilities' descriptions to support experts during vulnerability assessment activities, J. Syst. Softw. 156 (2019) 84–99.

[2] H. Gasmi, J. Laval, A. Bouras, Information extraction of cybersecurity concepts: An LSTM approach, Appl. Sci. 9 (19) (2019) 3945.

[3] C.L. Jones, R.A. Bridges, K.M. Huffer, 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, pp. 1–4.

[4] T.-M. Georgescu, B. Iancu, A. Zamfiroiu, M. Doinea, C.E. Boja, C. Cartas, A survey on named entity recognition solutions applied for cybersecurity-related text processing, in: Proceedings of Fifth International Congress on Information and Communication Technology, Springer, 2021, pp. 316–325.

[5] R.A. Bridges, C.L. Jones, M.D. Iannacone, K.M. Testa, J.R. Goodall, Automatic labeling for entity extraction in cyber security, URL https://www.osti.gov/biblio/1143555.

[6] I. Muhammad, A. Kearney, C. Gamble, F. Coenen, P. Williamson, Open information extraction for knowledge graph construction, in: International Conference on Database and Expert Systems Applications, Springer, 2020, pp. 103–113.

[7] R. McMillan, Open threat intelligence, 2013, Online; https://www.gartner.com/doc/2487216/definition-threat-intelligence/. (Accessed 19-February-2021).

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L.u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, 30, Curran Associates, Inc., 2017, URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[9] S. Gao, M.T. Young, J.X. Qiu, H.-J. Yoon, J.B. Christian, P.A. Fearn, G.D. Tourassi, A. Ramanthan, Hierarchical attention networks for information extraction from cancer pathology reports, J. Am. Med. Inform. Assoc 25 (3) (2018) 321–330.

[10] G.-H. Liu, J.-Y. Yang, Z. Li, Content-based image retrieval using computational visual attention model, Pattern Recognit. 48 (8) (2015) 2554–2566.

[11] V. Yadav, S. Bethard, A survey on recent advances in named entity recognition from deep learning models, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 2145–2158.

[12] A. Piplai, S. Mittal, A. Joshi, T. Finin, J. Holt, R. Zak, Creating cybersecurity knowledge graphs from malware after action reports, IEEE Access 8 (2020) 211691–211703.

[13] I. Sarhan, M. Spruit, Uncovering algorithmic approaches in open information extraction: A literature review, in: 30th Benelux Conference on Artificial Intelligence, Springer CSAI/JADS, 2018, pp. 223–234.

[14] L. Cui, F. Wei, M. Zhou, Neural open information extraction, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2018, pp. 407–413.

[15] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[16] G. Stanovsky, J. Michael, L. Zettlemoyer, I. Dagan, Supervised open information extraction, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 885–895.

[17] I. Sarhan, M.R. Spruit, Contextualized word embeddings in a neural open information extraction model, in: Natural Language Processing and Information Systems, Springer International Publishing, Cham, 2019, pp. 359–367.

[18] I. Sarhan, M. Spruit, Can we survive without labelled data in NLP? Transfer learning for open information extraction, Appl. Sci. 10 (17) (2020) 5758.

[19] J. Zhan, H. Zhao, Span model for open information extraction on accurate corpus, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34 (05), 2020, pp. 9523–9530.

[20] B.S. Cabral, R. Glauber, M. Souza, D.B. Claro, CrossOIE: Cross-lingual classifier for open information extraction, in: International Conference on Computational Processing of the Portuguese Language, Springer, 2020, pp. 368–378.

[21] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th International Conference on Machine Learning, 2008, pp. 160–167.

[22] Q. Zhu, X. Li, A. Conesa, C. Pereira, GRAM-CNN: a deep learning approach with local context for named entity recognition in biomedical text, Bioinformatics 34 (9) (2018) 1547–1554.

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

[24] Z. Yang, R. Salakhutdinov, W.W. Cohen, Multi-task cross-lingual sequence tagging from scratch, 2016, CoRR abs/1603.06270, 2016. arXiv:1603.06270, URL http://arxiv.org/abs/1603.06270.

[25] G. Kim, C. Lee, J. Jo, H. Lim, Automatic extraction of named entities of cyber threats using a deep Bi-LSTM-CRF network, Int. J. Mach. Learn. Cybern. 11 (10) (2020) 2341–2355.

[26] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[27] K. Simran, S. Sriram, R. Vinayakumar, K. Soman, Deep learning approach for intelligent named entity recognition of cyber security, in: International Symposium on Signal Processing and Intelligent Recognition Systems, Springer, 2019, pp. 163–172.

[28] S. Amit, Introducing the knowledge graph: Things, not strings, Official Blog (of Google), 2012 (2012).

[29] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, et al., Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia, Semantic Web 6 (2) (2015) 167–195.

[30] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: A collaboratively created graph database for structuring human knowledge, in: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, 2008, pp. 1247–1250.

[31] T. Pellissier Tanon, D. Vrandečić, S. Schaffert, T. Steiner, L. Pintscher, From freebase to wikidata: The great migration, in: Proceedings of the 25th International Conference on World Wide Web, 2016, pp. 1419–1428.

[32] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, D.S. Weld, Knowledge-based weak supervision for information extraction of overlapping relations, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 541–550.

[33] H. Fei, Y. Ren, Y. Zhang, D. Ji, X. Liang, Enriching contextualized language model from knowledge graph for biomedical information extraction, Briefings Bioinform (2020).

[34] A. Bordes, S. Chopra, J. Weston, Question answering with subgraph embeddings, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 615–620.

[35] H. Wang, M. Zhao, X. Xie, W. Li, M. Guo, Knowledge graph convolutional networks for recommender systems, in: The World Wide Web Conference, 2019, pp. 3307–3313.

[36] M. Fabian, K. Gjergji, W. Gerhard, et al., Yago: A core of semantic knowledge unifying wordnet and wikipedia, in: 16th International World Wide Web Conference, WWW, 2007, pp. 697–706.

[37] A. Bordes, E. Gabrilovich, Constructing and mining web-scale knowledge graphs: KDD 2014 tutorial, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 1967–1967.

[38] S. Vashishth, P. Jain, P. Talukdar, Cesi: Canonicalizing open knowledge bases using embeddings and side information, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1317–1327.

[39] B.D. Trisedya, J. Qi, R. Zhang, Entity alignment between knowledge graphs using attribute embeddings, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33 (01), pp. 297–304.

[40] H. Zhong, J. Zhang, Z. Wang, H. Wan, Z. Chen, Aligning knowledge and text embeddings by entity descriptions, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 267–272.

[41] P. Radhakrishnan, P. Talukdar, V. Varma, Elden: Improved entity linking using densified knowledge graphs, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 1844–1853.

[42] A. Thawani, M. Hu, E. Hu, H. Zafar, N.T. Divvala, A. Singh, E. Qasemi, P.A. Szekely, J. Pujara, Entity linking to knowledge graphs to infer column types and properties, SemTab@ ISWC 2019 (2019) 25–32.

[43] H. Huang, L.P. Heck, H. Ji, Leveraging deep neural networks and knowledge graphs for entity disambiguation, CoRR abs/1504.07678, 2015 (2015) arXiv:1504.07678, URL http://arxiv.org/abs/1504.07678.

[44] I.O. Mulang', K. Singh, C. Prabhu, A. Nadgeri, J. Hoffart, J. Lehmann, Evaluating the impact of knowledge graph context on entity disambiguation models, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 2157–2160.

[45] S.N. Narayanan, A. Ganesan, K. Joshi, T. Oates, A. Joshi, T. Finin, Early detection of cybersecurity threats using collaborative cognition, in: 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), IEEE, 2018, pp. 354–363.

[46] J.R. Finkel, T. Grenager, C.D. Manning, Incorporating non-local information into information extraction systems by gibbs sampling, in: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), 2005, pp. 363–370.

[47] E. Kiesling, A. Ekelhart, K. Kurniawan, F. Ekaputra, The SEPSES knowledge graph: an integrated resource for cybersecurity, in: International Semantic Web Conference, Springer, 2019, pp. 198–214.

[48] L.A. Ramshaw, M.P. Marcus, Text chunking using transformation-based learning, in: Natural Language Processing using Very Large Corpora, Springer, 1999, pp. 157–176.

[49] R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, in: International Conference on Machine Learning, PMLR, 2013, pp. 1310–1318.

[50] S. Bird, NLTK: the natural language toolkit, in: Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, 2006, pp. 69–72.

[51] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, ICLR (2015).

[52] N. Reimers, I. Gurevych, Reporting score distributions makes a difference: Performance Study of LSTM-networks for sequence tagging, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 338–348.

[53] J.J. Miller, Graph database applications and concepts with Neo4j, in: Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA, Vol. 2324 (36), 2013.

[54] T.-H. Wu, B. Kao, Z. Wu, X. Feng, Q. Song, C. Chen, Mulce: Multi-level canonicalization with embeddings of open knowledge bases, in: Z. Huang, W. Beek, H. Wang, R. Zhou, Y. Zhang (Eds.), Web Information Systems Engineering – WISE 2020, Springer International Publishing, Cham, 2020, pp. 315–327.

[55] B. Hachey, W. Radford, J. Nothman, M. Honnibal, J.R. Curran, Evaluating entity linking with wikipedia, Artificial Intelligence 194 (2013) 130–150.

[56] D. Ceccarelli, C. Lucchese, S. Orlando, R. Perego, S. Trani, Dexter: an open source framework for entity linking, in: Proceedings of the Sixth International Workshop on Exploiting Semantic Annotations in Information Retrieval, 2013, pp. 17–20.

[57] D. Fensel, U. Şimşek, K. Angele, E. Huaman, E. Kärle, O. Panasiuk, I. Toma, J. Umbrich, A. Wahler, Knowledge Graphs, Springer, 2020.

[58] F. Chollet, Keras, 2015, Online; https://github.com/fchollet/keras/. (Accessed 02-February-2021).

[59] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, TensorFlow: A system for large-scale machine learning, in: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, in: OSDI'16, USENIX Association, USA, 2016, pp. 265–283.

[60] S.K. Lim, A.O. Muis, W. Lu, C.H. Ong, Malwaretextdb: A database for annotated malware articles, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 1557–1567.

[61] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, R. Vollgraf, FLAIR: An easy-to-use framework for state-of-the-art NLP, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), 2019, pp. 54–59.

[62] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186, http://dx.doi.org/10.18653/v1/N19-1423, URL https://www.aclweb.org/anthology/N19-1423.

[63] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q.V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, 2019, pp. 5753–5763,

[64] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8440–8451.

[65] R. Dey, F.M. Salem, Gate-variants of gated recurrent unit (GRU) neural networks, in: 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), IEEE, 2017, pp. 1597–1600.

[66] S. Bao, H. He, F. Wang, H. Wu, H. Wang, PLATO: Pre-trained dialogue generation model with discrete latent variable, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 85–96.

[67] V. Nair, G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, in: ICML'10, Omni Press, Madison, WI, USA, 2010, pp. 807–814.

[68] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, ICLR, San Diego, CA, USA, Conference Track Proceedings, 2015.

[69] G. Stanovsky, I. Dagan, Creating a large benchmark for open information extraction, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 2300–2305.

[70] D. Nadeau, S. Sekine, A survey of named entity recognition and classification, Lingvist. Investig. 30 (1) (2007) 3–26.

[71] S. Yang, X. Yu, Y. Zhou, Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example, in: 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI), 2020, pp. 98–101, http://dx.doi.org/10.1109/IWECAI50956.2020.00027.

[72] L. Galárraga, G. Heitz, K. Murphy, F.M. Suchanek, Canonicalizing open knowledge bases, in: Proceedings of the 23rd Acm International Conference on Conference on Information and Knowledge Management, 2014, pp. 1679–1688.

[73] H. Schütze, C.D. Manning, P. Raghavan, Introduction to information retrieval, vol. 39, Cambridge University Press Cambridge, 2008.

[74] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, P. Selmer, A. Taylor, Cypher: An evolving query language for property graphs, in: Proceedings of the 2018 International Conference on Management of Data, 2018, pp. 1433–1445.

[75] A. Fan, T. Lavril, E. Grave, A. Joulin, S. Sukhbaatar, Addressing some limitations of transformers with feedback memory, 2020, ArXiv Preprint ArXiv:2002.09402, 2020.

[76] J. Han, H. Wang, Transformer based network for open information extraction, Eng. Appl. Artif. Intell. 102 (2021) 104262.