Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 449–460

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

Study of Word Embeddings for Enhanced Cyber Security Named Entity Recognition

Smita Srivastava[a,b], Biswajit Paul[b], Deepa Gupta[a]

[a]*Dept of Computer Science and Engineering, Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham, India.*
[b]*Centre for Artificial Intelligence and Robotics, Défense Research and Development Organization, Bengaluru, India.*

Abstract

A vast majority of cyber security information is in the form of unstructured text. A much-needed task is to have a machine-assisted analysis of such information. Named Entity Recognition (NER) provides a vital step towards this conversion. However, cyber security named entities are not restricted to classical entity types like people, location, organisation, miscellaneous etc but comprise a large set of domain-specific entities. Word embedding has emerged as the dominant choice for the initial transfer of semantics to downstream NLP tasks and impacts performance. Though several word embeddings learned using general purpose large corpus like Google News, Wikipedia etc. are available as pre-trained embeddings and have shown good performance on NER tasks; this trend is not consistent when it comes to domain-specific NER. This work explores the relative performances and suitability of prominent word embeddings for cyber security NER task. Embeddings considered include both general-purpose pre-trained word embeddings (non-contextual and contextual) available in the public domain and task-adapted embedding generated by fine-tuning these pre-trained embeddings on a task-specific supervised dataset. The results indicate that when it comes to using pre-trained embeddings for cyber security NER, fastText performs better than GloVe and BERT. However, when embeddings are further fine-tuned for the cyber-NER task, the performance of all the fine-tuned embeddings improved by +2-7%. Further, BERT embedding fine-tuned using position-wise FFN (Feed Forward Network) produced the state-of-the-art 0.974 F1-Score on the cyber security NER dataset.

Keywords: Cyber Security; Named Entity Recognition; Word Embeddings; BiLSTM; CRF; fastText; GloVe; BERT; Fine-Tuning

1. Introduction

Recent years have seen unprecedented growth in cyber-threats like ransomware, sophisticated malware (Advanced Persistent Threat, Remote Access Trojan), DDoS attacks, phishing etc. Interestingly, a massive volume of unstructured cybersecurity-related data is available on multiple platforms: scientific research papers, thesis, security blogs, online forums, service provider bulletins, official news, and social media. These publications give valued cybersecurity-related information about current events, such as software vulnerabilities and insightful analysis covering cyber threats/attacks, mitigations, responses etc. Organisations will be able to make informed decisions and consequently secure their cyber assets and avoid extensive damages, if access to relevant information in a structured, presentable, and inferable form is available. The problem is converting unstructured content into a structured, presentable, and

10.1016/j.procs.2023.01.027

inferable form facilitating complex inference by the security analysts. Named Entity Recognition provides the first step towards this conversion by seeding many downstream NLP tasks like relation extraction, information extraction, knowledge graph construction etc. Various domains, like Biomedical [1, 2], Scientific publications [3], Agriculture domain [4] etc., do use NER for the same and cyber security is no different.

Predominantly, the NER task involves detecting classical entity types like people, location, organisation, miscellaneous [5] etc. and is often trained by initialising with general-purpose word embedding learned from News, Wikipedia, Common crawl etc. Word embedding represents corpus words as points in dense vector space. The basis of the vector space implicitly captures facets of lexical semantics [6] and values as functions of word-word relations present in the corpus, used for learning embedding. NER for the classical entity types usually performs well with various general-purpose pre-trained embeddings. However, for a domain-specific NER task, with the addition of new entity types and domain-specific vocabulary, the performance of general-purpose pre-trained embeddings are not always consistent and superior across domains and tasks [7] and hence requires further investigation. Alternatives to using general-purpose word embeddings include the use of task-adapted, domain-adapted and domain-specific embeddings. Task-adapted embeddings are created by fine-tuning over general-purpose pre-trained embeddings using a relatively small task-specific supervised dataset. In contrast, domain-adapted embeddings are often fine-tuned using a rather large and rich unsupervised domain corpus over the general-purpose pre-trained embeddings. Finally, domain-specific embeddings are created from scratch using a vast unsupervised domain corpus with objectives like masked word prediction, next sentence prediction, span prediction etc. Building domain-specific embeddings from scratch require a large volume of domain corpus [1, 3] and are not readily available for most domains, including cyber security. As compared to domain-specific embedding though domain-adapted embedding requires relatively less domain corpora, it often poses the risk of training instability due to catastrophic forgetting. Hence there is merit in assessing the suitability of general-purpose pre-trained embeddings and task-adapted embeddings for a given domain task before venturing into domain-adaptive or domain-specific embeddings.

The objective of this study is as mentioned below:

- To bring out relative strengths, performances, analysis, and suitability assessment of prominent general-purpose pre-trained and task-adapted fine-tuned word embeddings towards achieving enhanced performance for the cyber-NER task.
- Issues like Out-Of-Vocabulary (OOV) words inherent in a fast-evolving dynamic domain like cyber security, class imbalance and non-availability of large and high-quality annotated Cyber-NER datasets in the public domain are also addressed.

The rest of the paper is organised as follows - Section. 2 provides details of the similar work carried out in various domains, including cyber security, using various models and embeddings. Section 3 gives details of the methodology used and details about experiments. Section 4 provides the results and analysis of various experiments; Section 5 provides the conclusion and Section 6 provides the direction of future work.

2. Related Work

This section presents research on embeddings, deep learning architectures and datasets used for domain-specific NER tasks, including cyber security.

A detailed survey of various word embeddings for domain-specific BioNER tasks is done to compare general-purpose and domain-specific word embeddings from non-contextual and contextual ones using three different pipelines CRF, BiLSTM and BiLSTM-CRF. It is concluded that non-contextual embeddings can be a better choice for general-purpose embedding, whereas contextualised embeddings are a better choice for domain-specific embeddings [7]. It is concluded that “to date, no consensus has been reached as to which kind of word embedding is best for each NLP task, so the selection is based on comparing various experimental setups and each researcher’s own experience” [7]. A detailed comparison of word embeddings with five general-purpose and nine domain-specific embeddings for the BioNER tasks was performed to show that domain-specific embeddings perform better for the health care domain [8]. Modified CRF is used for named entity extraction in text documents [5]. Hence, a detailed

study of these embeddings is required for cyber security, too.

Apart from pre-trained, efforts to create domain-specific embeddings, especially for cyber security, are also attempted. A framework to develop cyber security "Vulnerability" specific word embeddings (SecVulWE) using an open-source dataset was proposed for word similarity and vulnerability discovery tasks which showcased better performance over the general-purpose embeddings for cyber security [9]. But these are not available in the public domain.

A comparison of different architectural models with various word embeddings is carried out for various domains, including cyber security. In the health domain, BiLSTM-CRF is considered as the preferred architecture [8]. For the Cyber-NER task, different deep learning-based approaches are explored and studied with variations on the base model of BiLSTM with CRF [10]. In addition, to get the local context properly, an attention model along with BiLSTM is proposed. Here the use of cyber security, pre-trained word embeddings are used, which provides better semantic context [9] using the benchmarked cyber security dataset [11]. These cyber security pre-trained word embeddings are not available in the public domain for use. The use of regular matching strategies or machine learning techniques using the RDF-CRF model is also proposed for better results [12]. A popular approach to using the CNN layer and the word embedding layer of word2vec to extract character-level features was also explored for better performance [28]. Bidirectional GRU with CNN and CRF model provided good results on the benchmarked cyber security corpus [13]. The same is used for comparison with the proposed models too. This was done as the simple models cannot handle sparse and OOV cyber security type of words, so combining both data and knowledge-based implementation to learn rare entities and vocabulary is also used and showed better results [14]. The same is also compared with the model proposed in this paper. A survey of deep learning architectures was done with a ground truth dataset of 2100 sentences created from various cyber security sources. Detailed experimentation of different pipelines with domain-specific word2vec, domain-independent word2vec and BERT pre-trained embeddings were used along with deep learning pipelines like LSTM+CRF, BiLSTM+CRF etc. to provide a comparative assessment [15, 16]. A Russian Cyber-NER model is also proposed, which has used the pre-trained multilingual model further trained using a huge Russian cyber security corpus and fine-tuned for Cyber-NER using a large domain-specific supervised dataset [29]. This provided a basic framework for creating a domain-specific model which can be further fine-tuned for various downstream tasks.

When it comes to the available datasets for training, the domains like healthcare, transportation systems, cloud data systems etc., have up-to-date publicly available datasets (both unstructured and annotated) [1, 3], but the same is not valid for cybersecurity. A detailed survey on word embedding starting from classical, deep learning based, and the latest domain-specific transformer-based language models have been qualitatively evaluated for their suitability and pros and cons [17]. It is reiterated that these language models, primarily BERT, would provide in-depth learning and better use of unlabelled data for domain-specific tasks. Various aspects of deep learning methodologies, tools and annotated datasets available for NER are studied in detail to provide meaningful insight, along with the tool kits to explore various deep learning models for NER [18].

The primary issue, as explained earlier, for the cyber security NER task is the absence of a high-quality large benchmarked labelled dataset for training models and improvising. An annotated cyber security NER dataset with 13865 sentences and 24 BIO labelled is available in the public domain and is widely used [11]. Even attempts are made to provide cyber security-specific word embeddings to train the models for better performance [19]. A lot of effort has been put into creating a ground truth dataset from open-source unstructured cyber security datasets using various annotation tools like BRAT3 with a set of guidelines [12, 19, 20]. It is concluded that although communities have carried out some work to create domain-specific embeddings and the ground truth dataset for cyber security, these are not available in the public domain for further use by the researchers. Table 1. presents the summary of studies done in Cyber-NER using various architectural models and datasets in the last 2-3 years.

Table 1. Summary of Literature Survey for Cyber-NER

Authors	Architectural Model	Word Embeddings	Dataset	Performance (F1-Score)
Ma et al. (2021)	X-BiLSTM-CRF, Dynamic RNN-CRF, StanfordNER	One hot encoded	Self-collated dataset from cyber security domain (English)	0.8938
Gao et al. (2021)	BiLSTM-DomainDictionary-Attention-CRF	Pre-trained embeddings by Roy et al. (2017)	Open-source cyber security dataset (Bridges et al. (2013) (English)	0.8836
Zhou et al. (2021)	BERT-BiLSTM-CRF	Word2vec, BERT, BERTwwm	Bridges et al. (2013) (English)	0.9687
Yi et al. (2020)	RDF-CRF (Regular expression and known entity dictionary), LSTM-CRF, FT-CNN-BiLSTM-CRF	Integer embeddings	Self-collated dataset from cyber security domain (English)	0.8191
Tikhmirov et al. (2020)	Multilingual BERT model	BERT	Sec.col (SecurityLab.ru) (Russian)	0.6882
Dasgupta et al. (2020)	LSTM+CRF BiLSTM+CRF CNN+LSTM	Domain-independent and domain-specific Word2Vec and pre-trained BERT	Self-collated dataset from cyber security domain (English)	0.7700 - 0.8860
Simran et al. (2019)	BiGRU +CNN+CRF	(Keras Embedding(128dim))	Bridges et al. (2013) (English)	0.9340
Gasmi et al. (2018)	LSTM+CRF	word2vec	Bridges et al. (2013) (English)	0.8337

3. Proposed Methodology

This section covers a brief on a dataset acquired, word embeddings considered, deep learning architectures experimented with, and evaluation measures employed. Fig. 1. shows a high-level diagram of the proposed methodology.

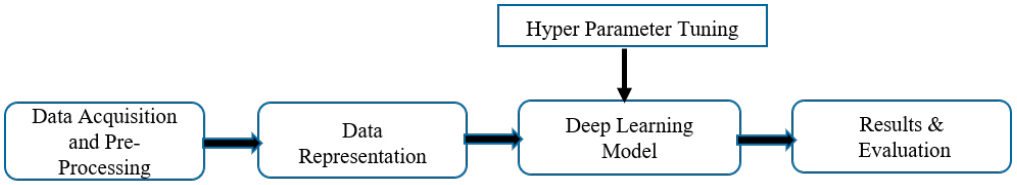

Fig. 1. Proposed Methodology for Experimentations

3.1 Data Acquisition and Pre-Processing

As the study pertains to Cyber-NER, we have used the publicly available Cybersecurity NER corpus. There are only two cyber-NER annotated datasets publicly available in English, namely the Cybersecurity NER corpus 2019 Saganowski et al. [21, 27] and the auto-labelled corpus provided by Bridges et al. [11, 28]. The auto-labelled-corpus dataset is selected as it has a more significant number of examples and NER tags. It contains data from Microsoft Security Bulletin, Metasploit and NVD (National Vulnerability Database) corpus and 13865 sentences, 24 BIO labelled named entities with 15 unique entities as listed in Fig. 2 (a). Out of these sentences, 10500 is used for training and the remaining 3365 for testing. Due to the auto-labelling strategy, we observed that the dataset is not free from

NER tag assignment errors. The dataset is in JSON format and is pre-processed to provide a CoNLL2000 format for use in the downstream task of cyber security NER.

As shown in Fig. 2 (a), the dataset is severely imbalanced as B-language has 7 entities, compared to 77114. Hence, for this study, some low-represented entities were merged with other named entities to create a dataset with dense tagged representation. The entities like B-method were merged with B-function, and B-edition and I-edition were merged with B-version and I-version, respectively. This reduced the number of BIO labels to 14, as shown in Fig. 2. (b). After merging, the dataset contains 14 BIO labelled and 9 unique named entities, as shown in Fig. 2(b). For the experimentation, both datasets were used.

```
[ ]  data['labels'].value_counts()

O                          563157
B-relevant_term             77114
I-relevant_term             50051
B-version                   29880
I-version                   25321
B-application               20525
I-application               13191
B-vendor                    11433
B-update                     4260
B-os                         4244
B-file                       3222
B-cve id                     3158
I-os                         2778
B-function                   1468
B-parameter                   657
B-edition                     595
B-hardware                    587
I-hardware                    586
I-update                      204
B-method                      175
B-programming language        168
I-vendor                       55
I-edition                      29
B-language                      7
Name: labels, dtype: int64
```

```
[ ]  data['labels'].value_counts()

O                  563157
B-relevant_term     77114
I-relevant_term     50051
B-version           30475
I-version           25350
B-application       20525
I-application       13395
B-vendor            12661
B-os                 4419
B-update             4260
B-file               3222
B-cve id             3158
I-os                 2778
B-function           2300
Name: labels, dtype: int64
```

Fig. 2. (a) Distribution of Cyber NER tags in the dataset, (b) Distribution of Cyber NER tags in reduced label dataset after merging

3.2 Word Embeddings

A variety of word embeddings were proposed in the literature, mainly differing in the purpose and ways embeddings are learned. Given embeddings' involvement in the initial transfer of semantics to downstream tasks, impacting overall task performance, empirical evaluation is necessary.

When it comes to the exploration of word embeddings, often the first choice goes to general-purpose embeddings because of the availability of ready-to-use, rich pre-trained embeddings covering a large general-purpose vocabulary set. The embeddings considered here are no different and include non-contextual GloVe [22], fastText [23] and contextual BERT [24] embeddings.

a) ***Word2Vec embedding:*** Word2Vec proposed by [6] learns embedding from the task of either predicting words given their context or predicting the context given the word using FFN. In the embedded space, the cosine similarity of words provides a degree of semantic similarity between the words, which holds even for word vector arithmetic. The Word2vec (both domain-specific and domain-independent) has been studied extensively for cyber security domains [13, 15, 26]; hence, it is not included in our study.
b) ***GloVe embedding:*** As most of the word embeddings can capture the semantic and the syntactic regularities using word vectors, GloVe goes a step ahead and also finds out the reason for such regularities using global matrix factorisation and local context window. The training uses weighted least squares of the global word-word co-occurrence counts, thereby providing efficient substructures and embeddings. We experimented with GloVe embeddings (*50 dim and 300 dim)*. Only 16.8% of words from the cyber security NER data were present in the pre-trained Glove vocabulary list.
c) ***fastText embeddings:*** It is a word embedding suitable for addressing the problem of generating embeddings for OOV words by representing the word as a combination of character n-grams and then dynamically generating embedding as a function of constituent n-gram embeddings. The training happens on words and word n-grams like Word2Vec. This is very apt for the cyber security domain, where we have a fast evolved

dynamic vocabulary set with a large number of rare words. For experimentation, 300 dim embeddings were used. 37.06% of the words from the cyber security NER data were present in the pre-trained fastText vocabulary list. However, fastText could generate embeddings covering 100% of words by using embeddings of the constituent n-grams in the case of OOV words.

d) ***BERT embeddings:*** BERT provides contextualised embeddings by training on both the left and the right context for Masked Language Model (MLM) and also on Next Sentence Prediction (NSP) objectives. For our experimentation, we have extracted BERT embeddings for the cyber security sentences, which were further averaged for the unique words of the vocabulary. All three pre-trained models, bert-base-cased, bert-large-cased and bert-large-cased-wwm (whole word masking), were used to provide embeddings of 768, 1024 and 1024 dimensions, respectively. Here 100% of words from the cyber security NER data were present in BERT vocabulary using a sub-words vocabulary list.

3.3 Deep Learning Training Pipelines

Experiments were carried out with two pipelines to evaluate word embeddings on the Cyber NER task. The de-facto standard BiLSTM+CRF pipeline is used for pre-trained and task-adapted fine-tuned embedding evaluations. Additionally, a position-wise linear feed-forward neural network is used to evaluate fine-tuned contextual BERT embedding.

3.3.1 BiLSTM+CRF pipeline for pre-trained and task-adapted fine-tuned embedding evaluation:

Figure 3. shows the first deep learning architecture, where a word-embedding layer provides input to the BiLSTM layer (Dense (256)) along with a wrapper of the Time Distributed Layer (Dense (50)). The Time Distributed Layer applies the same Dense layer, meaning the same weights to the LSTM outputs one-time step at a time. In this way, the output layer needs one connection to each LSTM unit, as shown in Fig. 3. The CRF layer is added to generate the tag sequence optimally, considering the correlations amongst the tags.

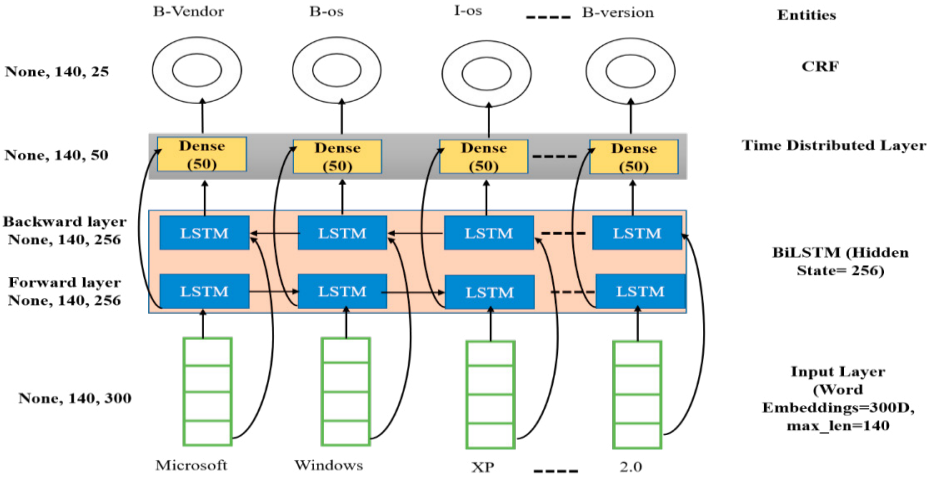

Fig. 3. Deep learning architecture (BiLSTM+CRF) used for pre-trained and task-adapted fine-tuned embeddings evaluation

For the evaluation of pre-trained embeddings, embedding layer weights were fixed, and learning is allowed only in the BiLSTM+CRF layers, whereas for task-adapted fine-tuned, embedding weights are updated along with learning parameters of BiLSTM+CRF layers. The auto-labelled supervised task-specific dataset is used for the learning.

3.3.2 Position-wise linear FFN pipeline for task-adapted fine-tuned BERT embedding evaluation:

The second deep learning pipeline consists of a position-wise collection of FFN and softmax for multiclass token classification, as shown in Fig. 4. This pipeline is used for fine-tuning BERT using the same auto-labelled task-specific supervised dataset, where the values of BERT embeddings are updated along with learning weights of FFN. Three variants of general-purpose pre-trained BERT models considered for fine-tuning are bert-base-cased, bert-large-cased and bert-large-cased-whole-word-masking. BertForTokenClassification method from HuggingFace Transformer package is used.

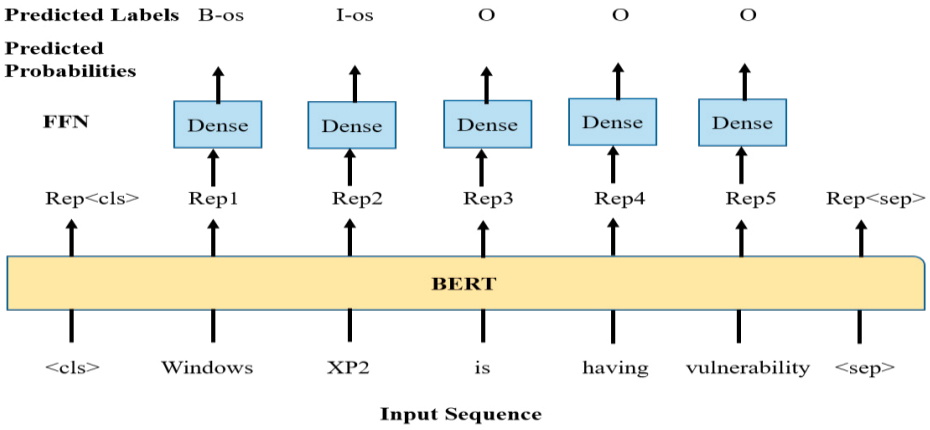

Fig. 4. Deep learning architecture (BERT+FFN) used for task-adapted fine-tuned BERT embedding evaluation

3.4 Selection of Hyper Parameters and optimisation

The model parameters need to be tuned to provide the best-performing models and performance. The following parameters were tuned using a series of exhaustive experimentations to arrive at the parameter configuration of the best-performing model.

- **Batch size**

Batch size is varied from 16 to 128. Here the maximum sentence length is kept as 140, with the learning rate at 0.0005. Batch size 32 and 64 provided the best results. The smaller batch size of 32 is selected as fastText, and BERT both provided the best results for the same. To prevent overfitting, a recurrent dropout of 0.1 is used with a learning rate of 0.0005 and Adam optimiser. These final figures were arrived at after exhaustive experimentations.

- **Max Sequence Length**

The dataset used for the experimentation has a skewed sentence length distribution with high variance, minimum of 14 words and a maximum of 8090 words. Statistical evaluations were done to arrive at the best-performing maximum sequence length. The experiments spanned sentence length percentile values between 75-90 and corresponding sentence length between 50 to 250. The 90% percentile value, having a sentence length of 140, was found to give the best result and hence chosen as the max-sequence length. Sentences less than 140 words were padded with zero, and sentences with a length greater than 140 were segmented recursively and added to the training set. Here the number of epochs is fixed at 32, with a learning rate of 0.0005. Fig. 5 shows the sentence length distribution after splitting bigger sentences into 140-word lengths.

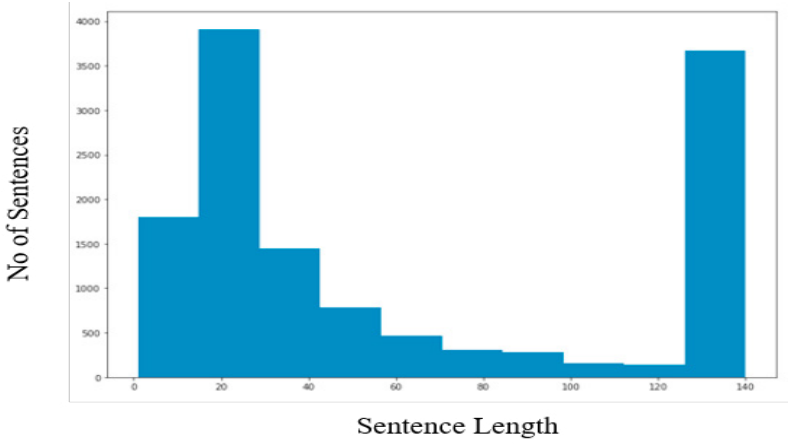

Fig. 5. Density plot of Sentences after splitting

- **Units in BiLSTM hidden layers**

The value of Units in BiLSTM hidden layers is a function of the complexity of the underlying output label generation process. Often assignments are based on the size of the output tag set. As a hyperparameter, we experimented with Unit values between 32 and 1024, and embedding size varied between 50 and 1024. It is found that the best performing model works out to be having 256 units in the hidden layers. The higher values add more to the complexity of the model, and the results were not significantly improved. So, a single BiLSTM with 256 units and a TDL with 50 units provided the best combination for the dataset.

3.5 Evaluation Metric

The standard metrics of F1-Score, Precision and Recall are used for performance. Further, to capture the effect of class imbalance, Micro, Macro and Weighted Average of F1-Score are also reported. For better error analysis entity wise break-up of results is also presented.

4. Experimental Results and Analysis

This section covers details of the experiments carried out and analysis of the experimental outcomes. The hardware includes one NVIDIA Quadro RTX 6000 with 24 GB of GPU memory and 256 GB of RAM. All experiments were done using the optimised parameters with batch size 32, learning rate of 0.0005 and maximum sentence length of 140 words.

4.1 Performance Comparison of General-purpose Pre-trained Embeddings

Table 2. shows the performance comparison of general-purpose pre-trained embeddings, GloVe, fastText, and BERT. The model training was done without fine-tuning the embeddings (trainable parameters were false for the embedding layer). Learning happened in the BiLSTM, CRF layers with the supervised NER Cyber dataset Bridges et al. [11]. Among general-purpose pre-trained embeddings, fastText embedding produced the best results.

Table 2. Comparison of the General-purpose Pre-trained (15 named entities)

SL No.	Deep Learning Pipeline	Embeddings Dimension	Performance		
			Precision	Recall	FI-Score
1	GloVe + BiLSTM + CRF	300	0.883	0.884	0.883
2	FastText + BiLSTM + CRF	300	0.910	0.916	0.914
3	bert-base-cased + BiLSTM + CRF	768	0.908	0.899	0.904
4	bert-large-cased + BiLSTM + CRF	1024	0.928	0.890	0.908
5	bert-large-cased-wwm + BiLSTM + CRF	1024	0.905	0.888	0.897

4.2 Performance Comparison of Task-adapted Fine-Tuned Embeddings

Table 3. shows the comparison of task-adapted fine-tuned embeddings. As the fine-tuning results showed better performance, we also experimented with the reduced labelled set, as shown in Table 4.

Table 3. Comparison of the General-purpose Fine-tuned embeddings (15 named entities)

SL No.	Deep Learning Pipeline	Embeddings Dimension	Performance		
			Precision	Recall	FI-Score
1	GloVe + BiLSTM + CRF	300	0.945	0.942	0.943
2	FastText + BiLSTM + CRF	300	0.930	0.946	0.938
3	bert-base-cased + BiLSTM + CRF	768	0.933	0.930	0.931
4	bert-large-cased + BiLSTM + CRF	1024	0.9294	0.930	0.929
5	bert-large-cased-wwm + BiLSTM + CRF	1024	0.932	0.936	0.934
7	bert-base-cased + FFN	768	0.969	0.977	0.973
8	bert-large-cased + FFN	1024	0.967	0.978	0.972
9	bert-large-cased-wwm + FFN	1024	0.967	0.973	0.970

Table 4. Comparison of General-Purpose Fine-Tuned embeddings (with 9 named entities)

SL No.	Deep Learning Pipeline	Embeddings Dimension	Performance		
			Precision	Recall	FI-Score
1	GloVe + BiLSTM + CRF	300	0.933	0.933	0.933
2	FastText + BiLSTM + CRF	300	0.919	0.942	0.931
3	bert-base-cased + BiLSTM + CRF	768	0.920	0.938	0.929
4	bert-large-cased + BiLSTM + CRF	1024	0.924	0.939	0.931
5	bert-large-cased-wwm + BiLSTM + CRF	1024	0.920	0.938	0.929
6	bert-base-cased + FFN	768	0.965	0.976	0.970
7	bert-large-cased + FFN	1024	0.973	0.975	0.974
8	bert-large-cased-wwm + FFN	1024	0.967	0.974	0.970

Here the general purpose pre-trained embeddings were further fine-tuned on the task-specific Cyber NER dataset Bridges et al. [11]. Learning happened in the embedding layer along with the BiLSTM+CRF and position-wise FFN layer. Table 3. shows that performance of all the task-adapted fine-tuned embeddings improved by +2-7% over the general-purpose pre-trained versions. This is because, during task adaptation, parameters of the pre-trained embeddings are updated, capturing and transferring better domain semantics. BERT fine-tuned with an added trainable FFN layer produced the best and state-of-the-art F1-Score of 0.974 on the same auto-labelled cyber security NER dataset. BERT with BiLSTM+CRF pipeline failed to produce competitive results as BERT embedding for this pipeline is generated by averaging the vectors of tokens across context.

Table 5. (a) shows the details of the tag-wise evaluation metric for the best model and BERT+FFN models. It is evident that tags having a low amount of labelled data set in the corpus have resulted in lower performance. E.g., the vendor has a low score compared to relevant-term as both contain 55 and 77114 entries in the dataset. The detailed classification report in Table 5. (b) shows that the macro average has improved by reducing the tag set. It shows that the merging of low count tags has resulted in a better macro average, indicating that the class imbalance problem is addressed to some extent (an increase of almost +3% in F1-Score, precision and recall).

Table 5. (a) Classification Report of bert-base-cased +FFN model (15 Named Entities)

	Precision	Recall	F1-Score
application	0.89	0.93	0.91
cve id	1.00	1.00	1.00
edition	0.79	0.84	0.81
file	0.97	0.98	0.98
function	0.97	0.97	0.97
hardware	0.59	0.65	0.62
language	0.00	0.00	0.00
method	0.91	0.94	0.92
os	0.95	0.96	0.95
parameter	0.97	0.97	0.97
programming language	1.00	1.00	1.00
relevant_term	1.00	1.00	1.00
update	0.92	0.93	0.93
vendor	0.96	0.97	0.96
version	0.98	0.99	0.98
micro avg	0.97	0.98	0.97
macro avg	0.92	0.94	0.93
weighted avg	0.97	0.98	

Table 5. (b) Classification Report of bert-large-cased + FFN model (9 Named Entities)

	Precision	Recall	F1-Score
application	0.90	0.92	0.91
cve id	1.00	1.00	1.00
file	0.97	0.99	0.98
function	0.96	0.99	0.98
os	0.94	0.96	0.95
relevant_term	1.00	1.00	1.00
update	0.95	0.94	0.94
vendor	0.95	0.93	0.94
version	0.98	0.98	0.98
micro avg	0.97	0.97	0.97
macro avg	0.96	0.97	0.96
weighted avg	0.97	0.97	0.97

4.3 Performance Comparison with Published Results

Table 6. compares published results with the proposed best result on the benchmark cyber-NER dataset by Bridges et al. [11] for the last 4 years covering from 2018-2021. Overall performance of task-adapted fine-tuned BERT embedding with position-wise FFN has produced the state-of-art F1-Score of 0.974. The previous best result of 0.968 F1-Score, published in [25], used fine-tuned BERT on cyber security corpus and BiLSTM+CRF pipeline. The dataset details and strategy used for fine-tuning are not available in the public domain. We used the Bridges et al. [11] dataset and position-wise FNN for fine-tuning. Bridges et al. [11] dataset has 15 named entities (24 BIO labelled entities). However, the tag sets reported in some of these literature are of varied length, the maximum being 40, which is again not known and is not available in the public domain.

Table 6. Performance comparison with published results

SL No.	Model	Methodology	No of the Named Entities	Performance		
				Precision	Recall	FI-Score
1	Gao et al. (2021)	BiLSTM + Dictionary + Attention +CRF (Cyber Security +Dictionary Embeddings)	7	0.9019	0.8660	0.8836
2	Zhou et al. (2021)	BERTwwm (1024dim) + BiLSTM + CRF	40	0.9703	0.9671	0.9687
3	Simran et al. (2019)	BiGRU +CNN+CRF (Keras Embedding(128dim))	40	0.9080	0.9620	0.9340
4	Gasmi et al. (2018)	word2vec(300dim) +LSTM+CRF	7	0.8516	0.8070	0.8337
5	Proposed Model	bert-base-cased (768dim) +FFN (fine-tuned)	15	0.9690	0.9770	0.9730
6	Proposed Model	bert-large-cased(1024dim) +FFN (fine-tuned)	9	0.9730	0.9750	0.9740

5. Conclusion

This study created a base for further development of robust domain-specific cyber security NER techniques, especially under low resource settings. The study established that task-adapted fine-tuning of embeddings consistently produced better results over the general-purpose pre-trained embeddings for the Cyber NER task. Empirical results show that the performance of all the fine-tuned embeddings improved by +2-7% over the pre-trained versions. The BERT embedding fine-tuned using position-wise FFN architecture produced the state-of-the-art 0.974 F1-Score on the auto-labelled cyber security NER dataset. Further, it is concluded that position-wise FFN architecture produced an improvement of 4.2% over BiLSTM+CRF pipeline under embedding fine-tuning setting. The OOV vocabulary issue is better handled using fastText and BERT word embeddings, as GloVe had only 16.8%-word coverage as compared to 100% coverage by fastText and BERT. It is observed that the class imbalance problem is addressed to a reasonable extent after merging of low count tags, as this has resulted in an +3% increase in F1-Score, precision and recall for the macro average values.

.

6. Future work

The next step would be to explore other strategies and architectures for task-adapted fine-tuning. However, significant community attention is sought to create publicly available, relatively larger and high-quality supervised training and evaluation datasets for advancing research in the field of cyber-NER. Standardisation of NER tag sets, probably complying with Structured Threat Information Expression (STIX™) or other standards, is the need of the hour. Additionally, a large unsupervised cyber domain corpus required for domain-adapted and domain-specific embedding creation is likely to enhance the performance of cyber-NER tasks further.

References

[1]. Lee Jinhyuk, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So and Jaewoo Kang. (2020) "BioBERT: a pre-trained biomedical language representation model for biomedical text mining." *Bioinformatics* 36: 1234 - 1240.

[2]. Gopalakrishnan, Athira, KP Soman and B. Premjith. (2019) "A Deep Learning-Based Named Entity Recognition in Biomedical Domain." Emerging Research in Electronics, Computer Science and Technology, vol. 545, pp. 517-526, 10.1007/978-981-13-5802-9_47.

[3]. Beltagy Iz, Kyle Lo and Arman Cohan. (2020) "SciBERT: A Pretrained Language Model for Scientific Text*." Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP) and the 9th International Joint Conference on Natural Language Processing*, pages 3615–3620, Hong Kong, China.

[4]. Veena Gangadharan, Deepa Gupta. (2020) "Recognizing Named Entities in Agriculture Documents using LDA based Topic Modelling Techniques". *Procedia Computer Science*. 171. 1337-1345. 10.1016/j.procs.2020.04.143.
[5]. Veena G, Deepa Gupta, Lakshmi S and Jacob J.T. (2018) "Named Entity Recognition in Text Documents Using a Modified Conditional Random Field", *Advances in Intelligent Systems and Computing*, vol 709. Springer, Singapore.
[6]. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. (2013) "Distributed representations of words and phrases and their compositionality." *in Advances in neural information processing systems*, pp. 3111–3119.
[7]. Ramos-Vargas RE, Román-Godínez I, Torres-Ramos S. (2021) "Comparing general and specialized word embeddings for biomedical named entity recognition." *PeerJ Comput. Sci*. 7:e384.
[8]. Unanue IJ, Borzeshi EZ, Piccardi M. (2017) "Recurrent neural networks with specialised word embeddings for health-domain named-entity recognition." *Journal of Biomedical Informatics* 76(5):102–109.
[9]. Mumtaz Sara, Rodriguez Carlos, Benatallah Boualem, Al-Banna Mortada, Zamanirad Shayan. (2020) "Learning Word Representation for the Cyber Security Vulnerability Domain." *International Joint Conference on Neural Networks (IJCNN)* 10.1109/IJCNN48605.2020.9207140.
[10]. Gasmi H, Bouras A, Laval J. (2018) "Lstm recurrent neural networks for cyber security named entity recognition." *Proceedings of the Thirteenth International Conference on Software Engineering Advances, Nice*.
[11]. Bridges R, Jones C, MD. Iannacone KT, Goodall J. (2013). "Automatic labelling for entity extraction in cyber security." arXiv preprint arXiv:1308.4941
[12]. F. Yi, B. Jiang, L. Wang and J. Wu. (2020) "Cybersecurity Named Entity Recognition Using Multi-Modal Ensemble Learning." in *IEEE Access*, vol. 8, pp. 63214-63224.
[13]. Ketha Simran, Srinivasan Sriram, Ravi Vinayakumar and KP Soman. (2019) "Deep Learning Approach for Intelligent Named Entity Recognition of Cyber Security." *Advances in Signal Processing and Intelligent Recognition Systems. SIRS 2019.* Communications in Computer and Information Science, vol 1209. Springer 10.13140/RG.2.2.23104.28169.
[14]. Gao, C., Zhang, X. Liu, H. (2021) "Data and knowledge-driven named entity recognition for cyber security." *Cybersecurity* 4, 9.
[15]. Dasgupta, Soham, Aritran Piplai, Anantaa Kotal and Anupam Joshi. (2020) "A Comparative Study of Deep Learning based Named Entity Recognition Algorithms for Cybersecurity." *2020 IEEE International Conference on Big Data (Big Data):* 2596-2604.
[16]. Vinayakumar R, Soman K. P, Poornachandran P, and Akarsh S. (2019) "Application of Deep Learning Architectures for Cyber Security. *In Cybersecurity and Secure Information Systems."* Springer, Cham, (pp. 125-160).
[17]. Usman Naseem, Imran Razzak, Shah Khalid Khan, Mukesh Prasad. (2020) "A Comprehensive Survey on Word Representation Models: From Classical to State-Of-The-Art Word Representation Language Models." *Transaction on Asian and low-resource Language Information Processing*, abs/2010.15036.
[18]. li Jing, Sun Aixin, Han Ray and Li Chenliang. (2020) "A Survey on Deep Learning for Named Entity Recognition." *IEEE Transactions on Knowledge and Data Engineering*, PP. 1-1. 10.1109/TKDE.2020.2981314.
[19]. Roy Arpita, Youngja Park and Shimei Pan. (2017) "Learning Domain-Specific Word Embeddings from Sparse Cybersecurity Texts." ArXiv abs/1709.07470
[20]. P. Ma, B. Jiang, Z. Lu, N. Li and Z. Jiang. (2021) "Cybersecurity named entity recognition using bidirectional long short-term memory with conditional random fields." in Tsinghua Science and Technology, vol. 26, no. 3, pp. 259-265.
[21]. Saganowski, Stanisław. (2020) "Cybersecurity NER corpus 2019 Harvard Dataverse." V1
[22]. Jeffrey Pennington, Richard Socher, and Christopher Manning. (2014) "GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)." *Doha, Qatar. Association for Computational Linguistics* pages 1532–1543,
[23]. Piotr Bojanowski, Edouard Grave, Armand Joulin and Tomas Mikolov. (2017) "Enriching Word Vectors with Subword Information." *Transactions of the Association for Computational Linguistics* 2017; 5 135–146.
[24]. Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. (2019) "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (202618).
[25]. S. Zhou, J. Liu, X. Zhong and W. Zhao. (2021) "Named Entity Recognition Using BERT with Whole World Masking in Cybersecurity Domain." *2021 IEEE 6th International Conference on Big Data Analytics (ICBDA)*, pp. 316-320.
[26]. T. Li, Y. Guo and A. Ju. (2019) "A Self-Attention-Based Approach for Named Entity Recognition in Cybersecurity." *15th International Conference on Computational Intelligence and Security (CIS)*, pp. 147-150
[27]. Dataverse dataset: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/1TCFII
[28]. Auto-labelled-corpus: https://github.com/stucco/auto-labeled-corpus
[29]. Tikhomirov, M., Loukachevitch, N., Sirotina, A., Dobrov, B. (2020) "Using BERT and Augmentation in Named Entity Recognition for Cybersecurity Domain." Natural Language Processing and Information Systems. NLDB 2020. vol 12089. Springer, Cham.