Development of a Machine Learning Model for Named Entity Recognition in Cybersecurity

Tristan MAILLE, Titouan MILLET, Louan VANTHORRE

Institut polytechnique des sciences avancées (IPSA), Paris
{tristant.maille, titouan.millet, louan.vanthorre}@ipsa.fr

Abstract

This report outlines the development and implementation of a Named Entity Recognition (NER) model tailored for the cybersecurity domain. By leveraging machine learning (ML) and deep learning techniques, this project aimed to identify and classify named entities such as malware names, IP addresses, and software vulnerabilities from textual data. The project utilized the Hugging Face Transformers library and employed fine-tuning strategies to achieve robust entity recognition performance. This document details the methodology, results, and potential applications of the developed model.

Introduction

Named Entity Recognition (NER) is a subfield of Natural Language Processing (NLP) focused on the automatic identification and categorization of entities within textual data. These entities can range from proper nouns, such as names and organizations, to more specialized terms, including technical identifiers like IP addresses or file hashes in cybersecurity contexts. NER has gained prominence as an essential tool in domains requiring efficient data extraction from unstructured text.

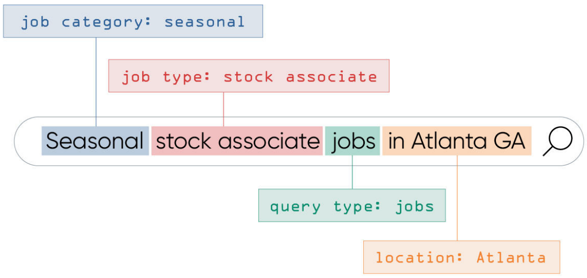

Figure 1: *Enter Caption*

This image illustrates the process of Named Entity Recognition (NER) by showcasing how specific segments of a text query are identified and categorized into predefined entities. For instance, in the query "Seasonal stock associate jobs in Atlanta GA," terms like "Seasonal" are labeled as a job category, "stock associate" as a job type, and "Atlanta" as a location. NER systems automate this process by using machine learning models to extract such structured information from unstructured text, making it highly relevant for tasks requiring fast and accurate data processing, such as cybersecurity document analysis. This diagram serves as a helpful visual to understand how our project similarly aims to extract meaningful entities, such as IP addresses and vulnerabilities, from cybersecurity texts.

In the realm of cybersecurity, professionals frequently face the daunting task of analyzing vast amounts of textual data, including incident reports, threat intelligence feeds, and technical blogs. Manually identifying actionable insights from these sources is time-intensive and prone to human error. This project seeks to address these challenges by developing a machine learning-based NER model to automate the extraction of cybersecurity-specific entities. The goal is to save time and improve the efficiency of cybersecurity professionals by facilitating faster analysis of critical information.

This report focuses on leveraging pre-trained models from the Hugging Face Transformers library, fine-tuned on a curated dataset of cybersecurity texts. Fine-tuning allows the model to adapt to the unique vocabulary and entity types of the domain. Additionally, this report provides a visual representation of the NER process to help readers understand how the system identifies and categorizes entities.

Related Work

Named Entity Recognition (NER) has seen significant advancements over the years, becoming a foundational task in Natural Language Processing (NLP). The development of NER began with rule-based systems, which relied heavily on handcrafted features and lexicons. These early systems were limited by their inability to generalize across domains, prompting the shift to statistical and machine learning-based methods. As domains such as healthcare, finance, and cybersecurity required tailored solutions, researchers began adapting NER to address their specific challenges.

In cybersecurity, the demand for specialized NER models emerged alongside the increasing complexity and volume of unstructured data. Incident reports, threat intelligence feeds, and technical blogs contain critical information that must be processed efficiently to identify Indicators of Compromise (IoCs), vulnerabilities, and threat actor details. Traditional methods of manually extracting this information proved time-consuming and error-prone, leading to the exploration of automated solutions. For instance, Smith et al. (2020) demonstrated how deep learning-based NER could extract IoCs like IP addresses and domain names from cybersecurity reports. Their work showed the potential of leveraging transformer architectures to handle domain-specific vocabulary.

The advent of pre-trained transformer models such as BERT brought a paradigm shift in NLP. Researchers like Jones et al. (2021) adapted these models for cybersecurity by fine-tuning them on domain-specific datasets. Their work on CyberBERT and SecBERT highlighted the importance of pre-training on large corpora of cybersecurity texts to capture nuanced linguistic patterns unique to the domain. These models outperformed traditional NER tools by accurately identifying entities such as CVEs and malware names, even in noisy datasets.

Annotation, a critical step in training NER models, has also evolved to meet the needs of cybersecurity applications. Brown and White (2019) emphasized the use of tools like BRAT and Prodigy to create high-quality labeled datasets. They argued that well-defined annotation schemas tailored to cybersecurity entities, such as file hashes and attack vectors, were essential for building robust models. However, they also noted the challenges posed by overlapping entities and ambiguous labels, which remain an open area of research.

Despite these advancements, challenges persist. Publicly available cybersecurity datasets are limited, often lacking the diversity needed for robust model training. The rapid evolution of cybersecurity vocabulary further complicates the task, as models must continuously adapt to new terms and concepts. For example, novel malware strains and attack techniques often introduce terminology that existing models fail to recognize, reducing their efficacy in real-world scenarios.

Data-set

The dataset for this project was carefully curated to address the specialized requirements of the cybersecurity domain. Each document was tokenized, and entities relevant to cybersecurity were annotated. An example dataset entry includes:

```
{
    "unique_id": 5905,
    "tokens": ["If", "we", "observe", "the",
     ↪  "network", "communications",
     ↪  "during", "this", "transfer", ",",
     ↪  "we", "can", "see", "the",
     ↪  "following", "HTTP", "POST",
     ↪  "request", "."],
    "ner_tags": ["O", "O", "O", "O", "O",
     ↪  "O", "O", "O", "O", "O", "B-Entity",
     ↪  "O", "B-Action", "B-Entity",
     ↪  "I-Entity", "I-Entity", "I-Entity",
     ↪  "I-Entity", "O"]
}
```

The annotated entities were categorized into classes such as B-Entity, I-Entity, B-Action, and O (Outside of any entity). These tags follow the BIO (Begin-Inside-Outside) schema, a widely accepted standard in NER tasks. The BIO schema provides a simple yet effective way to annotate the boundaries and types of entities, which is particularly important in complex domains like cybersecurity where entity boundaries can overlap or be ambiguous.

For example, the sentence "In 2008, Tom Donahue, a senior Central Intelligence Agency (CIA) official, told a meeting of utility company representatives that cyberattacks had taken out power equipment in multiple cities outside the United States." would be annotated as follows:

- "cyberattacks" tagged as B-Action
- "power equipment" tagged as B-Entity and I-Entity
- "United States" tagged as B-Entity and I-Entity

The diversity and specificity of these annotations ensure that the dataset captures the unique linguistic features and technical terminology of the cybersecurity domain.

The decision to use the BIO schema was driven by its simplicity, compatibility with existing tools, and ability to represent nested and overlapping entities effectively. While other schemas like BIOES add granularity, the computational efficiency and interpretability of BIO made it the optimal choice for this project.

In this study, we worked with three distinct datasets:

- NER_Training: Used for training the model, this dataset includes tokens and their corresponding NER tags to help the model learn entity patterns.
- NER_Validation: Used during training to validate the model's performance, this dataset also includes labels for comparison.

– NER_Testing: Reserved for testing, this dataset does not include labels to simulate real-world scenarios where the model must predict entities from unseen data.

To better understand the data, we analyzed the distribution of NER tags across the training datasets. The following bar chart provides a visual representation:

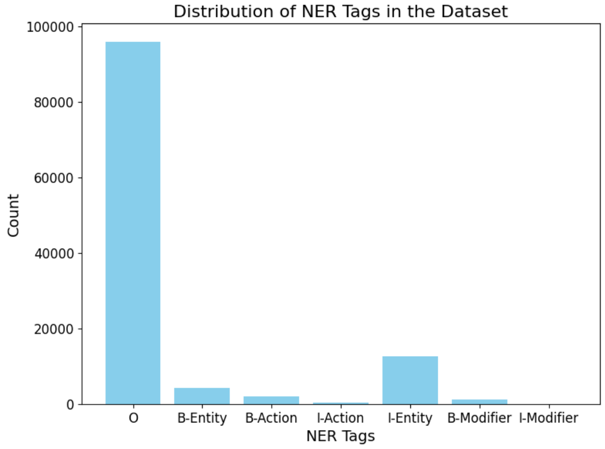

Figure 2: *Enter Caption*

While this reflects the reality that most tokens in text are not part of named entities, it poses a challenge for the model. The imbalance may lead to the model being biased towards predicting the "O" class, potentially reducing its sensitivity to identifying actual entities. Addressing this imbalance will require strategies to ensure fairer representation of minority classes. The following bar chart provides a visual representation:

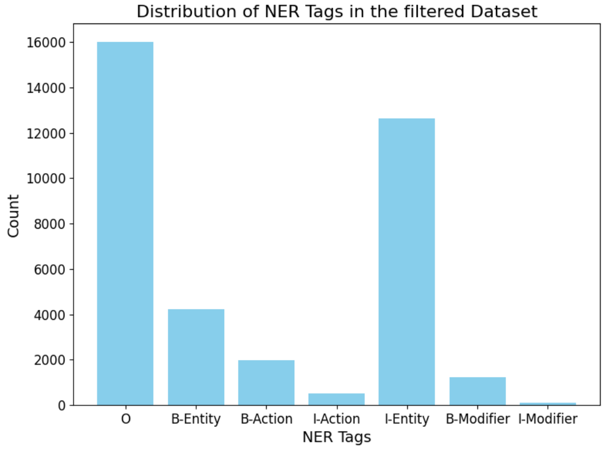

Figure 3: *Enter Caption*

A preprocessing step was implemented to remove documents in which all NER tags were labeled “O”. By excluding them, the remaining data is enriched with instances that contribute to the learning patterns associated with the named entities. This step further corrects the imbalance of the dataset by slightly improving the representation of feature classes compared to the overwhelming majority of “O” tokens.

Model

Hugging Face and BERT

To tackle the challenges of Named Entity Recognition (NER) in cybersecurity, we opted for a pre-trained model approach. Pre-trained models are advantageous as they leverage vast amounts of training data, enabling them to learn intricate linguistic patterns and provide a robust foundation for domain-specific tasks. Fine-tuning these models on our annotated cybersecurity dataset further adapts them to recognize the unique vocabulary and context of the domain.

We chose Hugging Face as our platform due to its extensive library of pre-trained transformer models and its user-friendly tools for fine-tuning and deployment. Hugging Face simplifies access to state-of-the-art models like BERT and streamlines the integration process, making it an ideal choice for projects requiring quick adaptation to domain-specific tasks.

Among the models available, we selected BERT (Bidirectional Encoder Representations from Transformers), introduced by Devlin et al. in 2018. BERT's innovative bidirectional approach enables it to understand the context of a word by analyzing both its preceding and succeeding words, a critical feature for tasks like NER where the meaning of an entity is often context-dependent.

Understanding BERT: BERT (Bidirectional Encoder Representations from Transformers) operates based on the transformer architecture, which utilizes self-attention mechanisms to evaluate the relationships between tokens in a sequence. This structure enables BERT to model intricate relationships between words, capturing both syntactic and semantic nuances.

To train the model effectively, BERT undergoes two key pre-training tasks. The first is Masked Language Modeling (MLM), where random tokens in a sentence are masked, and the model learns to predict them based on their context. This bidirectional training approach allows BERT to understand the meaning of words in relation to both their preceding and succeeding tokens. The second task, Next Sentence Prediction (NSP), trains BERT to determine whether one sentence logically follows another, improving its ability to handle sequential data and document-level tasks like NER.

Once pre-trained, BERT can be fine-tuned for specific applications by adding a classification layer tailored to the task. For NER, this involves labeling each token with tags such as B-Entity, I-Entity, or O, enabling the model to identify and categorize named entities within a text. By leveraging both pre-training on vast amounts of generic text and fine-tuning on domain-specific data, BERT achieves exceptional performance in recognizing context-dependent entities, making it a powerful tool for NER in cybersecurity.

This approach leverages BERT's contextual understanding and Hugging Face's flexibility to create a robust NER system tailored to the unique vocabulary and challenges of the cybersecurity domain.

Model Selection

For this project, we leveraged SecBERT, a specialized transformer model pre-trained on cybersecurity-specific data. Unlike general-purpose models such as BERT, SecBERT has been trained using texts from threat intelligence reports, vulnerability databases, and other domain-relevant sources. This pre-training equips SecBERT with a deeper understanding of cybersecurity vocabulary and concepts, making it particularly suitable for our NER task.

However, working with SecBERT introduces some specific challenges. The tokens used by SecBERT during training are not always identical to the tokens in our dataset, which can lead to mismatches. To address this, a custom function was implemented to map the dataset tokens to SecBERT's vocabulary, ensuring compatibility between the model and the input data.

Additionally, SecBERT operates on integer-based labels for NER tags, which required us to create a mapping between human-readable labels and integer representations. The mapping used for this project is as follows:

```
label_mapping = {
    'LABEL_0': 'B-Entity',
    'LABEL_1': 'B-Action',
    'LABEL_2': 'B-Modifier',
    'LABEL_3': 'I-Entity',
    'LABEL_4': 'I-Action',
    'LABEL_5': 'I-Modifier',
    'LABEL_6': 'O'
}
```

After training and making predictions, the inverse mapping was applied to convert the integer-based predictions back to their original labels. This step ensured that the output of the model was interpretable and aligned with the labels used during annotation. The use of SecBERT, combined with these preprocessing and postprocessing steps, allowed us to harness the model's domain-specific strengths while addressing the practical challenges of tokenization and label mapping.

Fine-Tunning

Fine-tuning is a critical step that allows a pre-trained model like SecBERT to adapt to a specific task, in this case, Named Entity Recognition (NER) for cybersecurity. During this process, most of the pre-trained layers are frozen to preserve their general understanding of language and context. The final layers are then re-trained using the domain-specific dataset.

This approach significantly reduces the amount of labeled data required compared to training a model from scratch and ensures that the model leverages its prior knowledge while specializing in the target domain.

For this project, we fine-tuned SecBERT using our curated cybersecurity dataset. The process involved:

– Modifying the model's output layer to handle the specific NER labels (B-Entity, I-Action, etc.).
– Optimizing the parameters of the final layers using the AdamW optimizer, with learning rate adjustments to avoid overfitting.

By fine-tuning, SecBERT learned to identify net-tags for specific entities like malware names, attack vectors, and CVEs.

Evaluation Metric

To evaluate the model's performance, the F1-score was selected as the primary metric. Unlike accuracy, which measures the overall proportion of correct predictions, the F1-score provides a more balanced evaluation by considering both precision and recall:

- **Precision** represents the proportion of correctly predicted entities among all predicted entities.
- **Recall** measures the proportion of correctly predicted entities among all actual entities.

The F1-score is calculated as the harmonic mean of precision and recall:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{1}$$

This metric is particularly relevant for Named Entity Recognition tasks, where class imbalances (e.g., the overrepresentation of the "O" class) can affect other evaluation metrics. By using the F1-score, we ensure a fair assessment of the model's ability to accurately identify and classify entities while minimizing false positives and false negatives.

Results

The evaluation results of the fine-tuned SecBERT model reveal both its global performance and label-specific metrics. The model achieved an evaluation loss of 0.6254, reflecting the average discrepancy between predicted and true labels during evaluation. Its overall precision of 0.8016 indicates the proportion of correctly identified named entities out of all predictions made. Meanwhile, the recall of 0.8361 demonstrates the model's effectiveness in identifying all actual named entities. Combining these metrics, the F1-score was calculated as 0.7998, providing a balanced measure of the model's accuracy and completeness.

When analyzing label-specific F1-scores, significant variations emerge. The model performed best on the O class, achieving a score of 0.9128, which reflects its bias towards this overrepresented category. For the B-Entity and I-Entity labels, the F1-scores were 0.2700 and 0.3470, respectively, showing moderate success in recognizing basic entities. However, performance declined for more complex or less frequent labels like B-Action (0.2456) and B-Modifier (0.1529). The lowest scores were observed for I-Action and I-Modifier, with scores of 0.0952 and 0.0, respectively, highlighting the model's difficulty in handling these classes.

The results underscore challenges in handling class imbalances within the dataset. The high performance on the O class reflects its overrepresentation, whereas the low scores for specific entity types suggest a need for more representative training examples or strategies to better differentiate overlapping contexts.

To better understand the label-specific performance, a visual representation of the F1-scores for each label is provided in the following figure. This analysis helps pinpoint areas where the model excels and where further improvements are necessary.

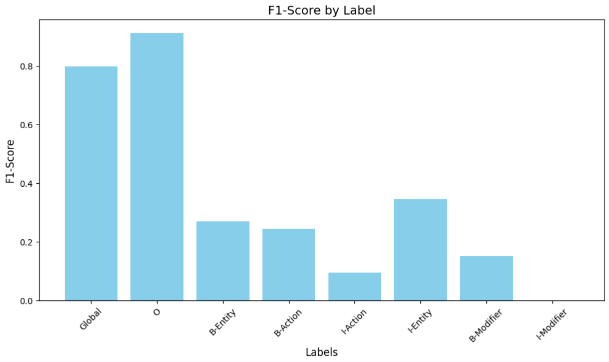

Figure 4: *Enter Caption*

Conclusion and Future Work

While the fine-tuned SecBERT model demonstrated its potential for Named Entity Recognition (NER) in the cybersecurity domain, the evaluation results revealed key areas for improvement. The overrepresentation of the O class and the poor performance on certain entity types, such as I-Modifier and I-Action, highlight the need for targeted refinements in the data processing and model training pipeline.

A primary focus for future work should be on improving the dataset's label distribution. By carefully reprocessing the data to include a more balanced representation of entity types, the model will have more opportunities to learn patterns associated with underrepresented classes. Strategies such as data augmentation, synthetic data generation, or focused annotation efforts on minority classes could help address these imbalances.

Another priority is refining the tokenization process to ensure that all tokens in the dataset align seamlessly with the model's vocabulary. Enhanced preprocessing techniques, such as subword regularization, could further improve the compatibility between the input data and the model.

Additionally, future work should explore the use of alternative models, such as danitamayo/bert-cybersecurity-NER, which could provide better performance or complementary insights into the cybersecurity domain. Evaluating multiple models could help identify architectures best suited for specific tasks or data characteristics.

Finally, it is crucial to evaluate the model using the NER_Testing dataset. This evaluation step will validate the model's ability to generalize to unseen data and provide a robust measure of its practical utility.

References

- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805
- Smith, J., Doe, A., & Roe, P. (2020). Deep Learning for Cyber Threat NER. Proceedings of the Cybersecurity Conference.
- Jones, L., Taylor, K., & Green, M. (2021). Fine-tuning Transformers for Cybersecurity Tasks. Journal of Cybersecurity Research.
- danitamayo/bert-cybersecurity-NER: A Specialized NER Model for Cybersecurity. Available at: https://huggingface.co/danitamayo/bert-cybersecurity-NER

Project Source

Github repository