

Label Unification for Cross-Dataset Generalization in Cybersecurity NER

Johan Hausted Schmidt
jhsc@itu.dk

Maciej Jalocho
macja@itu.dk

William Michelseen
wimi@itu.dk

Abstract

The field of cybersecurity NER lacks standardized labels, making it challenging to combine datasets. We investigate label unification across four cybersecurity datasets to increase data resource usability. We perform a coarse-grained label unification and conduct pairwise cross-dataset evaluations using BiLSTM models. Qualitative analysis of predictions reveals errors, limitations, and dataset differences. To address unification limitations, we propose alternative architectures including a multihead model and a graph-based transfer model. Results show that models trained on unified datasets generalize poorly across datasets. The multihead model with weight sharing provides only marginal improvements over unified training, while our graph-based transfer model built on BERT-base-NER shows no significant performance gains compared BERT-base-NER¹.

1 Introduction & Related Work

For cybersecurity NER, previous work has noted the scarcity of high-quality, large datasets in English. Due to the cybersecurity domain-specific entity types and vocabulary, it is difficult to make use of general-purpose NER resources (Srivastava et al., 2023). In recent years a handful more datasets have become available (Wang et al., 2022, 2020; Alam et al., 2022; Deka et al., 2024). As of writing this (2025), the cybersecurity domain has no standard set of NER labels, despite previously being proposed as a way of increasing availability of data resources (Gao et al., 2021a). Many of these newer datasets still use different label sets. This makes it difficult for models to combine and use these datasets effectively (Gao et al., 2021a). It is, for instance, generally not trivial to re-annotate data for NER in cybersecurity (Mouiche and Saad,

2024); no widely applicable, automatic, off-the-shelf annotator exists. Also, cybersecurity often demands familiarity with the domain and manual work for annotation (Evangelatos et al., 2021; Hanks et al., 2022). There are other prevalent problems due to the nature of the cybersecurity domain such as the extensive presence of OOV tokens in cybersecurity reports (Marjan and Amagasa, 2024), which many of the current datasets are based on. Problems such as these are only exacerbated by the domain’s data sparsity (Liu et al., 2022). Various methods have been proposed to remedy the problems caused by, or intimately tied to, the domain’s data sparsity. Some have tried to gather as much information about the tokens as possible, such as semantic, morphological, and contextual features, before any classification to help with novel, unseen tokens (Marjan and Amagasa, 2024). Others have similarly tried to integrate many linguistic features to enhance the representation of tokens, by retrieving the most similar words (Liu et al., 2022). Previous work has also explored merging labels such that some of the available datasets can be combined to train a more generalizable model (Silvestri et al., 2022).

In this work, we investigate the following research question: *How does a unification of NER labels across different datasets impact cross-dataset performance in cybersecurity and how can those limitations be overcome with adjustments to models?*

The main contributions of this investigation will be a systematic analysis of cross-dataset performance for recent cybersecurity datasets, enabled by a unification of labels. This will, to the best of our knowledge, unify a larger set of datasets’ labels than has previously been done in cybersecurity NER. The analysis of the cross-dataset predictions will help us understand the differences between the currently available datasets, and show the viability of such a label unification in practice. Based

¹<https://github.com/macja/CyberNER>

on what has been identified in the analysis, model-based solutions on these datasets will be proposed and evaluated. These contributions can help future research make better use of the available dataset resources in cybersecurity NER.

2 Methodology

This section outlines the datasets used in our study, the preprocessing and label unification steps we applied, and the experimental setup used to evaluate cross-dataset performance.

2.1 Datasets

Our project uses four different NER cyber-security datasets. APTNER - consisting of 260,134 tokens and 21 different entity types. The dataset consists of reports scraped from various network security companies (Wang et al., 2022). CYNER - with 106,991 tokens and 5 different entity types collected from 60 threat intelligence reports (Alam et al., 2022). DRNTI - consisting of 175,220 tokens and 13 entity types. The dataset is comprised of threat intelligence reports from websites of various security companies, government agencies, and GitHub (Wang et al., 2020). ATTACKER - consisting of 78,987 tokens and 18 entity types. The dataset is comprised of a mix of reports, articles, and blogs about previous attacks written by cybersecurity experts (Deka et al., 2024).

2.2 Data Cleaning & Preprocessing

The datasets we used in this study were originally provided in a variety of formats. To make sure that the different datasets were compatible with our NER models, we converted all the datasets to the standard CoNLL 2003 format using BIOES (Beginning-Inside-Outside) labelling.

Since the datasets came with different labels with varying specificity, we standardize their label sets by carrying out a label unification. We decided to go with the labels presented in the CYNER paper. These are "Organization", "System", "Vulnerability", and "Malware". We chose these coarse-grained labels because they generalize well across domains and fit many of the fine-grained labels found in the other datasets. We also decided to exclude the label "Indicator" from the CYNER paper, as the tokens relevant for the label can be extracted with regular expressions, rather than contextual understanding (Alam et al., 2022).

All labels that could be mapped to one of the four coarse-grained categories were unified accordingly,

for all datasets. Labels that could not be meaningfully mapped were discarded and relabelled as O (non-entity). All labels across the datasets that were successfully relabelled can be found in the appendix, in table 8..

We identified duplicate sentences both within individual datasets (across train, development, and test) and across different datasets. An example can be seen in table 7 in the appendix. So, for each train-test pairing, we remove any overlapping sentences from the training split to avoid data leakage. This applies to all models discussed in the paper.

2.3 Model Architecture

A BiLSTM model setup, using previously explored hyperparameter specifications in cybersecurity NER (Gao et al., 2021b; Ma and Hovy, 2016), was used. See table 9, 10 in appendix. Padding tokens are excluded from loss calculations.

2.4 Cross-Dataset Evaluation Setup

We perform pairwise evaluations by training our model on each training dataset and testing it on all other development datasets to assess cross-dataset generalization.

This approach results in a 4x4 matrix where each entry is the span-F1 score. The off-diagonal is the performance of the models trained on one dataset and evaluated on another, and the diagonal entries are from models trained and evaluated on the same dataset. In addition to span-F1, we also computed recall, precision, and the loose and unlabelled versions of span-F1, which can be found in the appendix in tables 13, 14, 15, and 16. During the analysis, we contrasted the diagonal entry with one set of cross-dataset predictions drawn from a row-adjacent, off-diagonal entry. This was done exactly once for each evaluation dataset. It was also ensured that no off-diagonal model was reused, such that all models were used once for intra-dataset predictions, and once for cross-dataset predictions. We also used Jensen-Shannon divergence (JS-div) to measure similarity between training sets.

3 Results

Table 1 shows the results of our cross-dataset experiment and, perhaps unsurprisingly, no row-adjacent entry outperforms the relevant diagonal entry in span-F1 score, recall, or precision. The precision is also considerably higher than recall for each model. The matrices concerning precision and recall can be found in the appendix.

Train \ Dev	DNRTI	ATTACKER	APTNER	CYNER
DNRTI	0.41	0.16	0.19	0.07
ATTACKER	0.09	0.23	0.01	0.02
APTNER	0.31	0.16	0.41	0.18
CYNER	0.05	0.04	0.06	0.40

Table 1: Cross-dataset evaluation: training datasets are on the rows. Dev datasets are on the columns. Each cell contains span F1 score

APTNER and DNRTI seem to do significantly better when trained on one and evaluated on the other. APTNER and DNRTI likely contain reports scraped from similar websites, as we found by far the highest amount of duplicates between those two datasets, compared to any other pair in our analysis. Also, some of the authors of the two papers are the same, and they use the same annotation tool (Brat), which could lead to similarity in the annotation approach. In addition, we see that when training on CYNER or ATTACKER, the model is performing quite poorly. An explanation for this could be that both datasets contain a smaller amount of training data relative to DNRTI and APTNER.

4 Analysis

The analysis of cross-dataset predictions serves as a proxy for understanding the differences between the datasets involved, and the practicality of the label unification. Therefore, we have focused specifically on how model behaviour and predictions differ, when used on a common evaluation dataset. During the analysis, we treated the diagonal as the intra-dataset baseline. We expected that comparing this baseline to a row-adjacent, off-diagonal entry would facilitate understanding the practicality of the label unification. This is because it would allow greater isolation of errors introduced by the cross-dataset predictions. Initially we wanted to compare the set of predictions on the diagonal to the set which had the lowest span-f1 in that column. However, if this approach had been used, some models would not have had its cross-dataset predictions examined, which would have limited our understanding of the excluded datasets. Therefore, we decided on the approach outlined in section 3.4.

Based on the analysis we identify a set of error trends we believe to be related to the label unification. Divergence measures between the datasets' domains will also be discussed relative to performance to provide additional insights.

4.1 ATTACKER to CYNER compared to CYNER to CYNER

We noticed a tendency for the model trained on ATTACKER to predict tokens that consist of individual symbols as being part of an entity. An example would be parentheses used in a wider context '(' and ')'. This behaviour is unusual relative to the other models. We believe this to be a symptom of the annotation approach for ATTACKER to favor longer spans, where most of the other datasets limit themselves primarily to noun phrases. This is an example of an error we believe to be caused by annotation discrepancies, which is one of the broader trends we identified during this analysis.

As a more general trend, we found tokens that were primarily mapped to system by the ATTACKER model, and primarily to organization by the CYNER model. One such example is the token 'Google'. This is not necessarily a disagreement on the definition of 'Google' relative to the NER labels, since both labels may be reasonable given a certain context, but it may be an indication that the datasets inhabit different sub-domains, as similar tokens are being applied in different contexts.

4.2 CYNER to ATTACKER compared to ATTACKER to ATTACKER

For certain labels in ATTACKER, the label unification mapped specific labels from the original dataset into broader categories defined in CYNER. For example, the label "threat actor" was mapped to the broader category "organisation." This mapping caused issues, likely because the CYNER model had not previously encountered these more specific labels, leading to numerous false negatives. Overall, the CYNER-trained model exhibited notably low recall (0.02 - can be seen in table 16), probably because the original "ATTACKER" labels are more niche. This issue affected the model's performance generally. We consider these errors to be potential examples of a broader trend of label definition discrepancies, where the datasets do not agree on what a label encompasses post unification.

An alternative explanation is that CYNER is one of the smaller datasets, so perhaps it struggles due to what may be a limited vocabulary. It is possible that some of the relevant tokens may have been annotated in a way that agrees with ATTACKER post unification, had they showed up in the CYNER training set more frequently.

4.3 DNRTI to APTNER compared to APTNER to APTNER

The word “sample” tends to be labelled as O or sometimes “FILE” (which was later mapped to O during label unification) in APTNER, while in DNRTI, it originally came from the label “Sam-File,” which got mapped to Malware during merging. So when a model is trained on DNRTI, it learns to associate “sample” with Malware, and ends up overpredicting that label on APTNER data. This appears in other inflected forms of the word. This is an example of what we considered to be definition discrepancies for tokens, which was another trend identified during the analysis.

In one case, both models misclassify the token “Lojax”. It’s labelled as Malware in APTNER but as System in DNRTI. Since neither model gets it right, the error may not just be about annotation mismatches, but instead about context or ambiguity in the input itself.

4.4 APTNER to DNRTI compared to DNRTI to DNRTI

Some of the apparent errors of the APTNER-trained model seem to come from mismatches in how entities are defined rather than from the model misidentifying entities. For instance, the token “backdoor” is labelled as Malware by the APTNER-trained model, but is labelled as System in DNRTI.

We see a similar issue with the token “Dridex,” which is considered Malware in APTNER but labelled as System in DNRTI. The model’s output reflects the definition it was trained on, even if it doesn’t match DNRTI’s label definition.

Label mismatches also appear with broader category terms like “Linux” or “Windows.” The model predicts these as System, which fits with how operating systems are labelled in APTNER (and CYNER). But in DNRTI, only Tool is mapped to System, so those tokens are labelled as O.

4.5 Trends and Frequencies

To summarize, we identified the following error-related trends: Definition discrepancies for tokens, definition discrepancies for labels, and annotation discrepancies (specifically in terms of span lengths). We acknowledge that the first two trends may be difficult to distinguish for any given example. Another trend observed during the manual analysis was that, due to dropping many labels, the models seemed to be more biased towards predicting

O. We found that reference models, using the original label sets, had a 5-15% lower ratio of FNs (on relevant tokens) compared to models using unified labels (see Appendix B for a more thorough description).

4.6 Language Metrics

Datasets	DNRTI	ATTACKER	APTNER	CYNER
DNRTI	0	0.23	0.04	0.05
ATTACKER	0.22	0	0.24	0.19
APTNER	0.04	0.24	0	0.07
CYNER	0.05	0.19	0.07	0

Table 2: JS-div between training datasets for distributions over span length of entities

Information-theoretic measures like JS-div estimate differences between probability distributions (Kashyap et al., 2020). In NLP, these are often based on relative frequencies (Lu et al., 2021). Our span-length divergence estimates (Table 2) shows ATTACKER stands out, aligning with observations in the analysis.

Divergence in word and POS tag distributions correlated negatively with model performance in table 1 (Pearson: -0.74 and -0.71, see Table 6). For many of the errors highlighted in the analysis, distributional shifts in words or POS tags may be simpler explanations.

4.7 Unified-datasets model

We hypothesise that if we train the model on all four unified datasets, that the scores on the individual evaluation datasets should be higher than a model trained only on the individual datasets. Table 3 shows the difference in span-F1 performance between predicting a development dataset with a model trained on all four datasets compared to a model only trained on a single dataset.

val/train	Combined	Original
Combined	0.38	-
DNRTI	0.49	0.41
ATTACKER	0.33	0.23
Aptner	0.30	0.41
Cyner	0.37	0.39

Table 3: Span-F1 on development datasets given a model trained on all datasets, compared to one

Opposite to the expectation, the performance on CYNER and APTNER is hindered by the combined dataset. It suggests that the noise introduced by the unification is so large that it dominates the benefits of a larger dataset.

5 Addressing Unification Limitations

5.1 Multi-head Model

Dataset	Shared: LSTM	Shared: EMB	Shared: Both	Reference
DNRTI	0.43	0.52	0.52	0.45
ATTACKER	0.01	0.19	0.21	0.04
APTNER	0.36	0.38	0.37	0.35
CYNER	0.34	0.41	0.39	0.40

Table 4: Span-F1 results for different variants of the multi-head model using a given dataset as evaluation, along with the reference-model performance for that dataset

Some of the identified error trends from the analysis likely explain the disappointing results in table 3. Here, we present the performance of a model that uses all available datasets without applying a label unification to the label sets. It is a multi-headed model where each head predicts labels from a specific dataset’s label set. While the weights associated with the output heads are individualised, those associated with the LSTM cell, and embedding matrix, may be shared. If the tasks demanded by the datasets, share enough characteristics to where building similar context representations in the LSTM layer, or numerical representations of words, would be beneficial, then such a model architecture could improve performance.

Three variations of the model were trained and evaluated on each dataset: one sharing both LSTM and embedding weights, one sharing only the embedding weights, and one sharing only the LSTM weights. A reference model was also trained for each dataset, which do not share any weights, but do use the same label sets as the multi-head model, for a baseline point of comparison.

Looking at table 4, the best performing multi-head variant, in terms of span-F1, performed significantly better than the reference model, for two of the datasets (DNRTI & ATTACKER). In general, the greatest improvements in span-F1 came from the variants that shared only the embedding weights, or both sets of weights. Lastly, for most of the datasets, the difference between the best multi-head variant and the best unification-model counterpart in table 3 is not significant. We would argue that similar performance to the models that use unified labels is generally positive, because one can keep the specificity of the original label set with comparable performance.

5.2 LST-NER

The last model we evaluated was taken from the paper "Cross-domain Named Entity Recognition

val\train	BERT-base-NER	LST-NER (ours)
DNRTI	0.70	0.71
ATTACKER	0.41	0.40
Aptner	0.64	0.65
Cyner	0.79	0.78

Table 5: comparing Span-F1 between our graph model and BERT-base-NER

via Graph Matching" (Zheng et al., 2024). Unlike our label unification process, this architecture takes a different approach by preserving the original label sets and leveraging structural knowledge from a rich-resource domain.

The key insight in the paper is that a pre-trained NER model from a general domain can provide structural knowledge about entity relationships even if the labels do not match those of the target domain. The model does this by representing label relationships as graphs and using Gromow-Wasserstein distance for structure matching. In theory, this should allow for the transfer of knowledge from the resourceful dataset to the more sparse dataset without requiring label correspondence. Since this model leverages BERT-base-NER as a backbone model, we thought that a fair comparison would be to compare it to that model without the addition of graphs proposed in the paper. Table 5 demonstrates the results on each evaluation dataset. As the table shows, we found no significant difference when introducing the graph proposed in the paper.

Hyperparameters for the models can be found in table 17 in the appendix.

6 Conclusion

We investigated the effects of label unification across cybersecurity NER datasets, by evaluating cross-dataset performance. While unification enabled combined training datasets, it introduced errors due to annotation differences, label mismatches, and increased bias toward predicting non-entities. Unified models often underperformed compared to single-dataset baselines.

Multi-head models showed slight gains by preserving original label sets, while LST-NER, despite its complexity, offered no clear advantage over BERT-base-NER. Overall, our results suggest that simple label merging is not sufficient for robust generalization, and future work should explore more targeted domain adaptation strategies.

References

- Md Tanvirul Alam, Dipkamal Bhusal, Youngja Park, and Nidhi Rastogi. 2022. Cyner: A python library for cybersecurity named entity recognition. *arXiv preprint arXiv:2204.05754*.
- Pritam Deka, Sampath Rajapaksha, Ruby Rani, Amira Almutairi, and Erisa Karafili. 2024. Attacker: towards enhancing cyber-attack attribution with a named entity recognition dataset. In *International Conference on Web Information Systems Engineering*, pages 255–270. Springer.
- Pavlos Evangelatos, Christos Iliou, Thanassis Mavropoulos, Konstantinos Apostolou, Theodora Tsikrika, Stefanos Vrochidis, and Ioannis Kompatsiaris. 2021. Named entity recognition in cyber threat intelligence using transformer-based models. In *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, pages 348–353. IEEE.
- Chen Gao, Xuan Zhang, Mengting Han, and Hui Liu. 2021a. A review on cyber security named entity recognition. *Frontiers of Information Technology & Electronic Engineering*, 22(9):1153–1168.
- Chen Gao, Xuan Zhang, and Hui Liu. 2021b. Data and knowledge-driven named entity recognition for cyber security. *Cybersecurity*, 4(1):9.
- Casey Hanks, Michael Maiden, Priyanka Ranade, Tim Finin, Anupam Joshi, and 1 others. 2022. Recognizing and extracting cybersecurity entities from text. In *Workshop on Machine Learning for Cybersecurity, International Conference on Machine Learning*.
- Abhinav Ramesh Kashyap, Devamanyu Hazarika, Min-Yen Kan, and Roger Zimmermann. 2020. Domain divergences: A survey and empirical analysis. *arXiv preprint arXiv:2010.12198*.
- Peipei Liu, Hong Li, Zuoguang Wang, Jie Liu, Yimo Ren, and Hongsong Zhu. 2022. Multi-features based semantic augmentation networks for named entity recognition in threat intelligence. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 1557–1563. IEEE.
- Jinghui Lu, Maeve Henchion, and Brian Mac Namee. 2021. Diverging divergences: Examining variants of jensen shannon divergence for corpus comparison tasks.
- Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.
- Md Abu Marjan and Toshiyuki Amagasa. 2024. Domain-adaptive entity recognition: unveiling the potential of cser in cybersecurity and beyond. *International Journal of Machine Learning and Cybernetics*, pages 1–19.
- Inoussa Mouiche and Sherif Saad. 2024. Ti-nermerger: Semi-automated framework for integrating ner datasets in cybersecurity. *International Conference on Security and Cryptography*.
- Stefano Silvestri, Giuseppe Felice Russo, Giuseppe Tricomi, and Mario Ciampi. 2022. A dataset for the fine-tuning of llm for the ner task in the cyber security domain.
- Smita Srivastava, Biswajit Paul, and Deepa Gupta. 2023. Study of word embeddings for enhanced cyber security named entity recognition. *Procedia Computer Science*, 218:449–460.
- Xuren Wang, Songheng He, Zihan Xiong, Xinxin Wei, Zhengwei Jiang, Sihan Chen, and Jun Jiang. 2022. Aptner: A specific dataset for ner missions in cyber threat intelligence field. In *2022 IEEE 25th international conference on computer supported cooperative work in design (CSCWD)*, pages 1233–1238. IEEE.
- Xuren Wang, Xinpei Liu, Shengqin Ao, Ning Li, Zhengwei Jiang, Zongyi Xu, Zihan Xiong, Mengbo Xiong, and Xiaoqing Zhang. 2020. Dnrti: A large-scale dataset for named entity recognition in threat intelligence. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1842–1848. IEEE.
- Junhao Zheng, Haibin Chen, and Qianli Ma. 2024. Cross-domain named entity recognition via graph matching. *arXiv preprint arXiv:2408.00981*.

Appendix

A Limitations

It should be noted that extensive tuning of learning-algorithm-hyperparameters (specifying this to contrast with below) was not attempted for any of these models just discussed, so it is possible that different configurations might make this model solution more, or less, effective. Also, we acknowledge that our unification of labels is likely suboptimal. We are not cybersecurity experts and some of the label-unification-related issues may be lessened if cybersecurity experts conducted the unification. Lastly, if the motivations for the model match a given problem, we expect one can see greater improvements in performance by moving to a different, more complex architecture since the internal parts of the model will have access to more data during training.

B O-class Imbalance Experiment

To gauge the effect of dropping many labels, we compared the false negative rate for reference models, that train and evaluate using the original set

of labels, with the models trained on the unified labels. For the reference models, the rate was only computed for tokens whose label are apart of the actual unification. We found that the reference models had a 5-15% lower ratio. While not enough to say anything conclusively, we caution that this is a problem.

C Data Cleaning

This appendix summarises the deterministic cleaning rules applied to the original datasets before label-unification.

C.1 APTner (CoNLL)

- Sentences are forced to break after the literal line . 0, reproducing the sentence count reported in the original paper. Original file breaks are not considered.
- A line that contains only 0 is discarded.
- Any other single-token line is retained with its label changed to 0.
- A line with ≥ 3 whitespace-separated fields is treated as corrupt; the first token is preserved and labelled 0.
- A two-token line whose second field is *not* in the official labels set is kept, but its label is replaced by 0.

C.2 DNRTI (CoNLL)

- A line that contains only 0 is discarded.
- Any other single-token line is retained with its label changed to 0.
- For A line with ≥ 3 whitespace-separated fields the first token is preserved and labelled 0.

C.3 Attacker (JSON)

- Tokens that are an explicit space character (i.e. the string " ") are skipped. *Note:* this check occurs inside the label-unification function, so no separate *cleaned* intermediate file is produced.

All these operations are reproducible using the cleaning functions for DNRTI & Aptner given their raw files. The cleaned files are produced.

D Contributions

Order meaningless:

- Maciej Pawel jalocha - cross-dataset models, combined-dataset model, data processing & cleaning, analysis
- William Michelseen - multi-head model, analysis, consolidation of analysis, language metrics, limitations, introduction & related work
- Johan Hausted Schmidt - graph model, analysis, consolidation of analysis, general methodology section, introduction & related work

All authors state that the workload was not significantly uneven.

E Tables

Distributions	Pearson Correlation
Words	-0.74
POS labels	-0.71
Entity span lengths	-0.36
Entity span-based counts	0.00

Table 6: Distributions for which JS-div was calculated and corresponding estimated correlation between cross-dataset span f1 performance and JS-div between datasets

Sentence	APT32 actors continue to deliver the malicious attachments via spear-phishing emails.
labels A	B-Organization O O O O B-Malware I-Malware O B-Vulnerability O O
labels B	B-Organization O O O O O O O O

Table 7: Comparison of token-level annotation of the same sentence in two different datasets.

Dataset	Original Labels	Unified Label	Notes
APTNER	APT, SECTEAM	Organization	
	OS	System	
	VULNAME	Vulnerability	
	MAL	Malware	
DNRTI	HackOrg, SecTeam, org	Organization	
	Tool	System	
	Way, exp	Vulnerability	
	SamFile	Malware	
ATTACKER	THREAT_ACTOR, GENERAL_IDENTITY	Organization	
	INFRASTRUCTURE, GENERAL_TOOL, ATTACK_TOOL	System	
	VULNERABILITY	Vulnerability	
	MALWARE	Malware	
CyNER	Indicator	O (removed)	Not used in learning phase

Table 8: Mapping of original entity labels to unified schema used across all four datasets.

Component	Specification
Embedding Layer	Embedding dimension = 100
LSTM Layer	Bidirectional LSTM with hidden size = 100 per direction (200 total)
Dropout Layers	Dropout $p = 0.5$ applied before and after LSTM
Output Layer	Linear layer mapping to label predictions

Table 9: Model architecture for the BiLSTM baseline

Hyperparameter	Value
Loss Function	Cross-entropy loss
Optimizer	Adam
Learning Rate	0.001
Batch Size	32
Epochs	15
Gradient Clipping	Max Norm = 5

Table 10: Training hyperparameters for the BiLSTM baseline

Metric	combined	aptner
l_precision	0.34	0.50
l_recall	0.46	0.53
precision	0.26	0.40
recall	0.35	0.42
ul_precision	0.38	0.48
ul_recall	0.50	0.51

Table 11: Precision and recall metrics comparison

Metric	combined	cyner
l_precision	0.55	0.49
l_recall	0.35	0.42
precision	0.48	0.43
recall	0.31	0.37
ul_precision	0.60	0.51
ul_recall	0.38	0.43

Table 12: Precision and recall metrics comparison

Train \ Dev	DNRTI	ATTACKER	APTNER	CYNER
DNRTI	0.68	0.24	0.19	0.16
ATTACKER	0.49	0.58	0.02	0.08
APTNER	0.50	0.24	0.40	0.33
CYNER	0.38	0.31	0.25	0.43

Table 13: Cross-dataset evaluation: training datasets are on the rows, while development (evaluation) datasets are on the columns. Each entry has the recorded PRECISION

Train \ Dev	DNRTI	ATTACKER	APTNER	CYNER
DNRTI	0.30	0.12	0.19	0.04
ATTACKER	0.05	0.15	0.01	0.01
APTNER	0.23	0.12	0.42	0.12
CYNER	0.03	0.02	0.04	0.37

Table 14: Cross-dataset evaluation: training datasets are on the rows, while development (evaluation) datasets are on the columns. Each entry has the recorded RECALL

Train \ Dev	DNRTI	ATTACKER	APTNER	CYNER
DNRTI	0.46	0.21	0.26	0.07
ATTACKER	0.09	0.24	0.06	0.02
APTNER	0.40	0.27	0.50	0.21
CYNER	0.06	0.05	0.07	0.47

Table 15: Cross-dataset evaluation unlabelled span-F1: training datasets on rows, development datasets on columns.

Train \ Dev	DNRTI	ATTACKER	APTNER	CYNER
DNRTI	0.48	0.24	0.29	0.09
ATTACKER	0.13	0.32	0.05	0.09
APTNER	0.39	0.22	0.52	0.21
CYNER	0.06	0.05	0.09	0.45

Table 16: Cross-dataset evaluation loose span-F1: training datasets on rows, development datasets on columns.

Hyperparameter	Value
Learning rate	5e-5
Batch Size	16
Epochs	3
Max Length	128
Temp	4
Edge Threshold	1.5
gwd_lambda	0.01

Table 17: Training hyperparameters for BERT-base-NER and LST-NER (only the first four applies to BERT-base-NER, whereas all hyperparameters apply LST-NER)

Disclosure of Chatbot Use (Required by ACL 2023)

In accordance with the ACL 2023 policy on the use of generative AI tools, we disclose the following usage of a chatbot (OpenAI’s ChatGPT) in the development of this project:

Implementation Support

ChatGPT was used to assist in the implementation of an LST-NER model. The original research paper (to the best of our knowledge) did not provide a public code repository. The chatbot was used to generate initial code snippets and to help debug issues encountered during development.

All code was critically reviewed, tested, and adapted by the authors to ensure alignment with the methodology described in the paper.

The final results and interpretations presented in this work are entirely the responsibility of the authors.