AnnoCTR: A Dataset for Detecting and Linking Entities, Tactics, and Techniques in Cyber Threat Reports

Lukas Lange* **Marc Müller#** **Ghazaleh Haratinezhad Torbati†**
Dragan Milchevski* **Patrick Grau‡** **Subhash Pujari*** **Annemarie Friedrich§**
*Bosch Center for Artificial Intelligence, Renningen, Germany
{lukas.lange, dragan.milchesvki, subhash.pujari}@de.bosch.com
†Max Planck Institute for Informatics, Saarbrücken, Germany
#Hochschule der Medien, Stuttgart, Germany ‡Robert Bosch GmbH, Stuttgart, Germany
§Universität Augsburg, Augsburg, Germany
annemarie.friedrich@informatik.uni-augsburg.de

Abstract

Monitoring the threat landscape to be aware of actual or potential attacks is of utmost importance to cybersecurity professionals. Information about cyber threats is typically distributed using natural language reports. Natural language processing can help with managing this large amount of unstructured information, yet to date, the topic has received little attention. With this paper, we present AnnoCTR, a new CC-BY-SA-licensed dataset of cyber threat reports. The reports have been annotated by a domain expert with named entities, temporal expressions, and cybersecurity-specific concepts including implicitly mentioned techniques and tactics. Entities and concepts are linked to Wikipedia and the MITRE ATT&CK knowledge base, the most widely-used taxonomy for classifying types of attacks. Prior datasets linking to MITRE ATT&CK either provide a single label per document or annotate sentences out-of-context; our dataset annotates entire documents in a much finer-grained way. In an experimental study, we model the annotations of our dataset using state-of-the-art neural models. In our few-shot scenario, we find that for identifying the MITRE ATT&CK concepts that are mentioned explicitly or implicitly in a text, concept descriptions from MITRE ATT&CK are an effective source for training data augmentation.

Keywords: Cybersecurity, Concept Detection, Named Entity Recognition, Entity Linking

1. Introduction

Cyber Threat Intelligence (CTI) necessitates collecting evidence-based knowledge about cyber threats to proactively defend against cyber attacks. Cyber Threat Reports (CTRs), which are usually provided by professional CTI vendors, are unstructured text documents that describe threat-related information such as tactics, techniques, actors, tools, types of systems as well as geographic regions, political entities, or targeted industries. Retrieving and analysing information from CTRs is a tedious and time-consuming yet usually time-critical task (Sarhan and Spruit, 2021; Rahman et al., 2021). Obtaining clean labeled data that ensures replication, validation and extensions of CTI studies constitutes a major technical challenge (Rahman et al., 2021).

Applying information extraction techniques from natural language processing (NLP) to the domain of CTI is promising, yet understudied. MalwareTextDB (Lim et al., 2017; Phandi et al., 2018) focuses on extracting attributes of malware. Other datasets (e.g., Satyapanich et al., 2020; Kim et al., 2020) use custom annotation schemas. In our full-text annotation scenario, we focus on classifying mentions of attack tactics and techniques according to the MITRE ATT&CK taxonomy, a globally accessible database maintained and used by cybersecurity professionals. We choose this taxonomy rather than a custom schema as it fits directly with the daily work of cybersecurity professionals. Prior work using MITRE ATT&CK either only annotated entire documents (Legoy et al., 2020) or individual sentences. Almost all prior datasets are not clearly

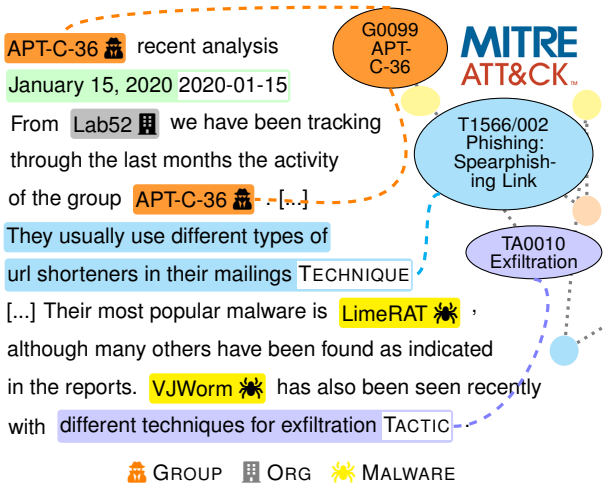

Figure 1: **AnnoCTR** is a CC-BY-SA-licensed dataset of 120 cyber threat reports annotated with MITRE ATT&CK concepts and WikiData entities.

LREC-COLING 2024, pages 1147–1160
20-25 May, 2024.

licensed and hence difficult or even impossible to use without violating copyright. This aspect has particular relevance in application-driven research.

In this paper, we present AnnoCTR[1], a new publicly available dataset consisting of 400 CTRs donated by their copyright holders. We add annotations targeting information extraction and search tasks, including mentions of locations and organizations linked to Wikipedia, and normalized temporal expressions. 120 of the CTRs have been annotated by a domain expert with cybersecurity-specific concepts, explicitly or implicitly mentioned tactics or techniques from MITRE ATT&CK.

We propose a set of NLP tasks based on AnnoCTR. Transformer-based named entity recognition (NER) models for the general-purpose entities achieve macro-average F1 scores of up to 70%. We find entity disambiguation models (Wu et al., 2020; Cao et al., 2021) fine-tuned on our domain to work well for identifying techniques that occur in a document (micro-F1 of around 65%).

Our new dataset will enable researchers in NLP and CTI to research, develop, and apply cutting-edge text understanding, search and analysis technology that will provide a very necessary competitive advantage over threat actors. From an NLP perspective, our contributions are as follows.

- We provide AnnoCTR, a new openly-licensed dataset carefully annotated with named entities (NEs) and cybersecurity concepts, and perform a detailed corpus study (Section 3).
- We propose to model the data from a variety of perspectives using neural sequence tagging, text classification and entity linking models (Section 4), and provide experimental results for state-of-the-art baselines (Section 5).

2. Related Work

In this section, we give a brief overview of NLP work and datasets in the cybersecurity domain.

MITRE ATT&CK[2] is a hierarchical knowledge base (KB) of cyber adversary tactics and techniques compiled based on real-world observations. It is designed to help with managing cyber threat risks. The KB is regularly updated and released under a license permitting research, development, and commercial use. At the time of writing, the Enterprise part of the taxonomy, which we are using in this work, consists of 14 tactics and 193 techniques at the top level and 401 subtechniques. Each technique or tactic comes with a textual description as illustrated in Figure 2, as well as several references to CTRs or technical descriptions. Legoy et al. (2020) crawl the latter to create a dataset (rcATT) annotated with tactics and techniques at document level. The CTRs come from many different sources, hence, licensing is unclear. The Threat Report ATT&CK Mapper (TRAM) is an open-source platform including a web application aiming to advance research into automating the mapping of CTRs to MITRE ATT&CK.[3] The TRAM dataset consists of sentences annotated with multi-label MITRE ATT&CK techniques. Our new dataset and models aim, i.a., to improve the functionality of this open-source endeavor.

> **T1606: Forge Web Credentials** Adversaries may forge credential materials that can be used to gain access to web applications or Internet services. [...]
>
> **T1606.001 Web Cookies** Adversaries may forge web cookies that can be used to gain access to web applications or Internet services. [...]
>
> **T1606.002 SAML Tokens** An adversary may forge SAML tokens with any permissions claims and lifetimes if they possess a valid SAML token-signing certificate. [...]

Figure 2: MITRE ATT&CK (sub)techniques.

Table 1 gives an overview of manually labeled **NLP datasets in the cybersecurity domain**. MalwareTextDB (Lim et al., 2017; Phandi et al., 2018) contains reports on hacker groups annotated with the 444 attributes of MAEC[4] (Malware Attribute Enumeration and Characterization). The reports are taken from APTnotes[5] which are publicly available but have unclear licensing. Hanks et al. (2022) crawl CTI blog posts from the web and conduct a small annotation study for cybersecurity-specific NEs and linking them to Wikipedia. CASIE (Satyapanich et al., 2020) and CySecED (Man Duc Trong et al., 2020) are annotated with cybersecurity event types and semantic arguments. Kim et al. (2020) annotate CTI-Reports with 20 NE types. Bayer et al. (2022) create a dataset of 3000 tweets annotated for whether they mention a cyber attack.

NER in the cybersecurity domain has been modeled using maximum entropy models with n-gram features (Bridges et al., 2013), and using tf.idf-based features and word2vec (Mikolov et al., 2013) to train a Linear SVM (Cortes and Vapnik, 1995) along with a manually designed confidence propagation procedure (Legoy et al., 2020). In the neural age, recurrent neural networks and convolutional neural networks with learned bag-of-characters embeddings and a CRF layer have been used for the task (Gasmi et al., 2019; Kim et al., 2020; Simran et al., 2020). Sarhan and

[1] https://github.com/boschresearch/anno-ctr-lrec-coling-2024

[2] https://attack.mitre.org, https://github.com/mitre/cti

[3] https://github.com/center-for-threat-informed-defense/tram

[4] https://maecproject.github.io/

[5] https://github.com/aptnotes

Dataset Name & Reference	#Docs.	#Sents.	#Annots.	Labels	License
MalwareTextDB-v1 (Lim et al., 2017)	39	2080	7102	MAEC NE mentions	no license
MalwareTextDB-v2 (Phandi et al., 2018)	85	12,918	8054	MAEC NE mentions	no license
CTI-Reports (Kim et al., 2020)	160	13,750	15,720	custom NE mentions	no license
rcATT (Legoy et al., 2020)	1490	185,000	1490	MITRE ATT&CK at doc. level	no license
CyEnts (Hanks et al., 2022)	380	1339	781	custom NE mentions	no license
CySecED (Man Duc Trong et al., 2020)	292	~7300	8014 events	30 event types (w/o arguments)	not available
CASIE (Satyapanich et al., 2020)	1000	~17,000	8470 events	5 event types + arguments	no license
TRAM	–	2298	3772	MITRE ATT&CK at sent. level	Apache 2.0
AnnoCTR (ours)	120	12,179	13,244	NE, MITRE ATT&CK, TIMEX	CC-BY-SA 4.0

Table 1: Overview of manually labeled **cybersecurity NLP datasets**.

Spruit (2021) use XLM-R (Conneau et al., 2020). Further related work aims at constructing or enhancing cybersecurity KBs from text (Sarhan and Spruit, 2021; Sanagavarapu et al., 2021).

3. AnnoCTR Dataset

In this section, we describe our AnnoCTR dataset.

3.1. Source of Texts and Preprocessing

AnnoCTR consists of 400 CTRs annotated with general-world entities. Out of these, 120 reports are also annotated with cybersecurity categories (see Table 2). All CTRs have been obtained from the blogs of commercial CTI vendors,[6] who have agreed to their re-publication under CC-BY-SA 4.0. The blog entries were published between March 2013 and February 2022. Annotation is performed using the web-based annotation system INCEpTION (Klie et al., 2018).

CTI Vendors. The reports have been kindly donated by Intel471[7], Lab52[8] (the threat intelligence division of S2 Grupo[9]), Proofpoint[10], QuoIntelligence[11], and ZScaler[12].

Preprocessing. First, we retrieve the texts.[13] Because of the numerous URLs, code and image references occurring within the texts, we convert the articles into a format similar to Markdown using BeautifulSoupand Markdownify.[14,15] Off-the-shelf sentence segmenters do not perform well on texts that contain many links or code snippets, hence, we use a custom regular expression tokenizer for sentence segmentation and correct sentence boundaries manually.

[6] Intel471, Lab52 (the threat intelligence division of S2 Grupo), Proofpoint, QuoIntelligence, and ZScaler.
[7] https://intel471.com/blog
[8] https://lab52.io/blog/
[9] https://s2grupo.es/
[10] https://www.proofpoint.com/us/blog
[11] https://quointelligence.eu/blog
[12] https://www.zscaler.com/blogs/security-research
[13] https://github.com/psf/requests
[14] https://www.crummy.com/software/BeautifulSoup/
[15] github.com/matthewwithanm/python-markdownify

3.2. Annotation Scheme

We annotate the reports in our dataset with the following **General Named Entity** (**GNE**) types.

ORG: Organisations including companies.
LOC: Locations, e.g., *California*, *China*.
SECTOR: Industry sectors, e.g., *finance*, *defense*.
TIMEX: Time expressions for dates normalized following TimeML (Saurı et al., 2006), e.g., *July this year* → `2022-07`.
CodeSnippet: Code snippets and command line interface commands.

We annotate mentions of **cybersecurity-specific NEs** (**CyNE**) as follows. The term *software* refers to custom or commercial code, operating system utilities, open-source software, or other tools.

GROUP: Mentions of Advanced Persistant Threats, i.e., hacker groups, e.g., *Fancy Bear*, *Leviathan*, or *APT 40*.
MALWARE: Software that has been written specifically for malicious purposes, e.g., *Terdot*.
TOOL: Software not written for a malicious purpose but used with a malicious intent in a given context, e.g., "a malicious *Microsoft Excel* document builder."
CONCEPT (CON): More general concepts relevant to the cybersecurity that can be linked to Wikipedia (e.g., *malware*, *threat actors*), and non-malicious software that is not used maliciously in a context, e.g., *Kaseya VSA*.
TACTIC: We annotate mentions of tactics as defined by MITRE ATT&CK they capture the adversary's tactical goal (e.g., obtaining credential access), their reason for performing an action.
TECHNIQUE: We mark spans that refer to techniques. MITRE ATT&CK defines them as follows: Techniques represent *how* an adversary achieves a tactical goal by performing an action.

For TACTIC and TECHNIQUE mentions, we indicate whether the concept is *explicitly* or *implicitly* mentioned. An explicit mention of a TECHNIQUE means that the descriptive name of the technique is mentioned more or less literally or with a synonym, e.g., as in *send phishing e-mails with malicious attachments* → T0865 (Spearphishing Attachment).

	docs.	sent.	sent/doc	tok./sent
Intel471	30	1907	63.6±55.3	22.3
Lab52	23	1665	72.4±55.3	14.1
ProofPoint	28	2305	82.3±43.4	21.1
QuoIntelligence	12	1541	128.4±60.2	22.4
ZScaler	27	4761	176.3±107.0	21.6
total	120	12,179	101.5±79.1	22.0

Table 2: **Corpus statistics** for AnnoCTR: sentence and token counts (for cyber-security-specific part).

Implicit mentions require more inference on the reader's part, as in *Emotet bots reach out to their controllers and received commands to download and execute Trickbot on victim machines.* → T1105 (Ingress Tool Transfer). Explicit mentions are usually short phrases, while implicit phrases may be any part of the text up to a sentence (as also illustrated by Figure 1).

Entity Disambiguation. ORG, LOC and CON mentions are linked to Wikipedia pages[16], GROUP, TACTIC, and TECHNIQUE to MITRE ATT&CK.

3.3. Corpus Statistics

Table 2 shows that the CTRs from the various vendors differ in their average number of sentences, but that sentence lengths are roughly comparable. Table 3 lists the number of NE mention annotations. Most of them have been annotated with valid links to Wikipedia or MITRE ATT&CK (exact counts are given in Table 4). The distribution of techniques and tactics both have a long tail (details in Appendix B). In total, the 120 documents annotated with both layers contain 13,244 annotations, the full dataset contains 20,855 annotated mentions. Hence, as can be seen in Table 1, AnnoCTR is at least comparable in size to the frequently used MalwareTextDB v2 (Phandi et al., 2018).

3.4. Annotation Process and Agreement

Annotation of the general layer (TIMEX, ORG, LOC, SECTOR, and CodeSnippet) was performed by a team of two annotators with an engineering background who participated in an extensive training phase. The cybersecurity-specific annotations (all others) were created by a graduate student of media informatics who had previously interned at a cybersecurity group and who hence possesses special domain knowledge. The annotator was involved in the design of the annotation scheme. For the **agreement analysis**, we select 9 documents with a total of 416 sentences: one by QuoIntelligence and two of each other vendor.

[16] hhttps://en.wikipedia.org//

NE type	Intel471	Lab52	Proofpoint	QuoIntell.	ZScaler	total
Cyber-security specific + general layer (120 docs.)						
# docs.	*30*	*23*	*28*	*12*	*27*	*120*
TIMEX	294	155	358	164	155	1126
CodeSnippet	18	10	32	0	255	315
LOC	220	503	172	174	86	1155
ORG	306	377	508	124	263	1578
SECTOR	132	89	137	131	63	552
CON	236	66	261	185	820	1568
TOOL	24	25	34	61	76	220
MALWARE	345	100	474	93	657	1669
GROUP	93	102	230	222	66	713
TECHN. (expl.)	260	176	229	121	312	1098
TECHN. (impl.)	301	229	321	198	1243	2292
TACTIC (expl.)	60	55	219	91	271	696
TACTIC (impl.)	65	22	19	14	142	262
General layer (280 additional docs.)						
# docs.	*4*	*5*	*73*	*2*	*196*	*280*
TIMEX	61	62	848	88	731	1790
CodeSnippet	32	5	141	0	1324	1502
LOC	52	196	732	4	271	1255
ORG	41	134	1281	27	1075	2558
SECTOR	13	36	230	8	219	506

Table 3: **Named entity** annotations in AnnoCTR.

NE type	Intel471	Lab52	Proofpoint	QuoIntell.	ZScaler	total
Cyber-security specific + general layer (120 docs.)						
LOC	220	508	172	174	86	1160
ORG	200	363	486	124	263	1436
CON	220	66	261	178	814	1539
TOOL	24	25	34	61	75	219
MALWARE	345	100	474	93	657	1669
GROUP	93	103	230	224	66	716
TECHNIQUE	560	404	550	319	1542	3375
TACTIC	125	77	238	105	413	958
General layer (280 additional docs.)						
LOC	52	201	732	4	271	1260
ORG	22	133	1261	27	1075	2518

Table 4: **Disambiguated entity mentions** in AnnoCTR with links to MITRE ATT&CK or WikiData.

General annotations. The nine documents of the agreement study were marked by an additional annotator, who had not been involved in the first phase, but had already received extensive training on the almost same annotation task in a different domain. When comparing TIMEX annotations, we find 41 exact matches. Six additional cases are annotator lapses, i.e., trivial mistakes. In one other case, one annotator did not include the *late* into the span of *late October*. There are 18 LOC annotations with an observed agreement of 100% on

spans. For 17 of these, the annotators agree on the Wikipedia link. For ORG, 73 and 81 instances are marked by the two annotators, respectively, amounting to precision and recall scores of 78.1 and 70.4% for exact matches. When applying relaxed matching (containment), the scores go up to 89.0 and 80.2%. Out of 57 exact matches and 8 relaxed matches, we found only 3 Wikipedia links to differ. The two annotators mark only 10 and 20 SECTOR mentions, respectively, with a relaxed agreement of 70.0 and 35.0%.

Cybersecurity-specific annotations. This type of annotation requires specialized cybersecurity knowledge and a high familiarity with the MITRE ATT&CK taxonomy. The entire dataset has been marked by single domain expert annotator. In order to provide a rough estimate for the difficulty of the annotation task, we ask another cybersecurity professional with a degree in media informatics and 5 years of professional experience to label our data. As their time constraints did not allow an extended training phase, the agreement presented here in all likelihood underestimates the degree of agreement that is achievable with more training. Precision and recall for identifying entity mentions amount to F1-scores of 31% for CON, 52% for GROUP, and 67% for MALWARE. The second annotator did not mark any TOOL annotations. When comparing the sets of techniques found in a document, average F1 amounts to 54%. While this study does not constitute a proper agreement study (as we acknowledge would be desirable), it still demonstrates that annotations are systematic. An exception may be CON, where the main annotator considered a different set of concepts as relevant. Yet, this set seems to be consistent as in our experiments, the tag can be learned well (F1 68%).

4. Task Definitions and Modeling

In this section, we define several NLP tasks based on AnnoCTR and describe neural models that we propose as strong baselines for solving them.

4.1. Named Entity Recognition

The first processing step consists of detecting entity mentions and tagging them with the NE types listed in Section 3.2. We represent label sequences using the BIO scheme. First, the model processes the input sequence with a pre-trained transformer model, applies a linear classification layer to the transformer output to compute the logits for each potential NE label, and predicts the label corresponding to the maximum logit for each first wordpiece token of each "real" token.

We compare the following pre-trained transformer models: BERT (Devlin et al., 2019), which has been trained on Google Books (Zhu et al., 2015) and the English Wikipedia, SciBERT (Beltagy et al., 2019), a BERT-style model trained on scientific text, and CodeBERT (Feng et al., 2020), which has been trained on programming code. We also use RoBERTA (Liu et al., 2019), which is trained on Google Books, Wikipedia, and news articles, and which uses an optimized training procedure compared to BERT.

As a recent baseline for NER in the cybersecurity domain, Gasmi et al. (2019) and Kim et al. (2020) use a BiLSTM-CRF (Lample et al., 2016) with GloVe (Pennington et al., 2014) embeddings, essentially following the architecture of Huang et al. (2015). We compare to a reimplementation of the model by Kim et al. (2020) in our experiments.

4.2. Temporal Tagging

We compare two methods for temporal expression extraction and normalization. First, we use the rule-based system Heideltime (Strötgen and Gertz, 2010) limited to DATE rules. We experiment with adapting HeidelTime to the cybersecurity domain by automatically selecting a subset of rules. We sequentially remove randomly selected rules and keep only rules that influence the performance positively. We start this procedure with 5 different random seeds and keep the best-performing subset of rules based on the dev set score.

Second, we use a multilingual neural temporal tagging model (Lange et al., 2022). This model first detects TIMEXes using a neural sequence tagger trained on gold standard corpora of news and Wikipedia, and then, for temporal expression normalization, uses a masked language modeling approach to fill slots in XML-like templates:

`The attack happened <TIMEX type="DATE" value="`**`YEAR`**`-`**`MONTH`**`-`**`DAY`**`">yesterday</TIMEX>.`

`YEAR`, **`MONTH`**, and **`DAY`** are masks, whose values are predicted by the language model. The normalization model has been trained on a multilingual weakly-supervised dataset created using HeidelTime. In our experiments, we substitute the sequence tagger for TIMEX detection with our cybersecurity-specific NER model (Section 4.1).

4.3. Entity Disambiguation

NE mentions in AnnoCTR are, depending on their type, linked to Wikipedia or MITRE ATT&CK. We restrict the search space for the linking models depending on the extracted NE type. LOC/ORG/CON are linked against Wikipedia. MALWARE and TOOL are linked to the `/software/` subtree of MITRE ATT&CK. Accordingly, we map GROUPS to `/groups/`, TACTIC to `/tactics/` and TECHNIQUES to `/techniques/` only. In this section,

we describe several models that we use to identify the KB entry disambiguating the mention.

BLINK (Wu et al., 2020) uses a bi-encoder model to compute embeddings for all entries of the KB and for the input sentence with the mention to be disambiguated. A set of candidates is selected based on the similarity of the target sentence encoding to the KB encodings. In a second step, a cross-encoder computes a final ranking score for the concatenation of the text strings representing the entity in the KB and the input sentence. In our experiments, we found that in our setting, the cross-encoder does not provide any benefit and hence only use the bi-encoder part of the model. BLINK has been trained on 9M examples of document-mention-entity triples from Wikipedia.

GENRE (Cao et al., 2021) is a constrained language model based on BART (Lewis et al., 2020). Given an input sentence with an entity to be linked (marked using special tags), the model is asked to generate the most likely KB entry. The search space is constrained by the entries of the target KB, e.g., all titles of Wikipedia pages for the general-domain linking, or the titles or MITRE ATT&CK entries for mentions of cybersecurity entities. GENRE has first been trained on the same Wikipedia data as BLINK and then further fine-tuned on AIDA-CoNLL (Hoffart et al., 2011).

BLINK works out of the box for linking to MITRE ATT&CK as the pre-trained bi-encoder can be applied to any KG. We use the ATT&CK descriptions (as in Figure 2) to generate entity representations. GENRE can also be applied in a zero-shot setting on MITRE ATT&CK as the set of KG entities that constrains decoding is part of the model configuration, which are the ATT&CK titles in our setting. In order to compare general-purpose entity disambiguation models and cybersecurity-specific models, we fine-tune the models on AnnoCTR. In cases where BLINK or GENRE predict a subtechnique, we change the prediction to the parent technique.

4.4. Sentence-based Tactic and Technique Classification

From a practical point, it matters to correctly detect the set of tactics and techniques that occur in a document. In particular, implicitly mentioned techniques are often marked as long phrases and hence NER models are not well-suited. We here address the problem using few-shot text classification methods. In order to comprehensively detect and link explicit and implicit cyber attack tactics and techniques to MITRE ATT&CK, we first detect sentences mentioning a TACTIC or TECHNIQUE, and then classify these sentences into the set of concepts as defined by ATT&CK.

Detection. First, we train two four-way sentence classifiers using RoBERTA for detecting TACTICs and TECHNIQUEs, respectively. The classifiers predict whether an input sentence contains an explicit, and implicit, both an explicit and an implicit, or no mention of TECHNIQUE/TACTIC. This setup performed better or on par with a binary classifier. The second step takes as an input sentences that are predicted to contain TECHNIQUE/TACTIC mentions and the linking model links the sentence to a node in ATT&CK.

Classification/Disambiguation. For classifying a sentence into a set of tactics or techniques, we compare the following models.

GENRE: We use the entity linking model described in Section 4.3, but do not mark entity mentions in the input sentence.

TMM: The Transformer-based Multi Task Model (Pujari et al., 2021) encodes the input sentence using SciBERT, and feeds the CLS embedding into a set of binary classification heads which each predict whether a particular technique occurs in the sentence or not.

TRAM: For comparison, we run the model provided by the TRAM project on our data. The model consists of an ensemble of a logistic regression and a Naive Bayes model using n-gram features and has been trained on the TRAM dataset for technique detection and linking.

5. Experiments

This section describes our experimental results on our AnnoCTR corpus for the NLP tasks and models introduced in the previous sections.

5.1. Settings

For our experiments, we split the AnnoCTR corpus into three parts (train/dev/test) with 60/15/25% of the documents, respectively. We perform a temporal split ensuring that documents from each vendor end up in each split. The temporal split simulates a real-world scenario as older documents are used for training and more recent documents are processed during inference. The dev set is used for model picking. Results are reported on the test set. For the NER experiments, we train 5 models with different random seeds and report the average scores and standard deviation of all runs.

Hyperparameters. We did not perform a hyperparameter search, as we used the suggested default values, i.e., the Huggingface training script for NER,[17] and the model-specific scripts for BLINK and GENRE from the respective repositories. For GENRE, we set the number of beams and output

[17] https://github.com/huggingface/.../run_ner.py

Model	GNE	GNE^{+X}	CyNE
BiLSTM-CRF (Kim et al.)	$62.2_{\pm 0.6}$	$65.2_{\pm 1.1}$	$43.7_{\pm 1.6}$
BERT	$77.2_{\pm 1.7}$	$80.4_{\pm 1.3}$	$51.0_{\pm 2.3}$
SciBERT	$76.3_{\pm 1.8}$	$80.4_{\pm 1.4}$	$53.0_{\pm 2.3}$
CodeBERT	$75.7_{\pm 2.0}$	$80.3_{\pm 1.9}$	$53.7_{\pm 3.5}$
RoBERTa	$\mathbf{80.2}_{\pm 1.6}$	$\mathbf{82.3}_{\pm 1.2}$	$\mathbf{57.8}_{\pm 1.9}$

Table 5: **NER results**. Metric: average Micro F1 of 5 runs for general-world named entities (GNE) and *explicit* cybersecurity entities (CyNE) trained on 120 documents. GNE^{+X}: training includes data from 280 additional documents.

sequences to 10 during decoding to match the number of candidates of BLINK.

5.2. NER Results

Table 5 contains the results for our NER experiments. In this experiment, we only use explicit mentions of TECHNIQUE and TACTIC, as we found in preliminary experiments that the NER models do not work well for implicit mentions. The BiLSTM-CRF baseline (Kim et al., 2020) performs between 12 and 15 F1 points worse than the transformers. Differences between the transformer models are smaller. The RoBERTa models perform best for general and cybersecurity entities with 2-3 points improvements compared to the other BERT models. A possible explanation is that the pretraining data of RoBERTa is closer to AnnoCTR. For example, in contrast to the pretraining data of the other models, it includes news articles, which are comparable in structure to CTRs. Moreover, training all models on the extended corpus with general entities ($^{+X}$) consistently improves the performance by 2 points.

5.3. Temporal Tagging Results

The results for temporal tagging are given in Table 6. We use the TempEval3 evaluation script (UzZaman et al., 2013) and report relaxed/strict F1 and value F1 for the extraction and normalization of temporal expressions, respectively.

Both out-of-the-box models, HeidelTime and the neural model of Lange et al. (2022), have rather low scores for the extraction step. Thus, we experimented with domain adaptation methods and find that both can be greatly improved. First, the HeidelTime model benefits from the reduced set of rules for the cybersecurity domain (+12.0/10.5/9.2 pp.), which results in a decrease of false positive matches. These are often caused by differences in the annotation schemes, e.g., imprecise expressions like *now* or *soon* are not annotated in AnnoCTR. Second, the neural model can be improved by substituting the extraction component

	Extraction		*Norm.*
Model	Strict	Relaxed	Value
HeidelTime	57.5	69.3	69.3
+ optimized rules	69.5	79.8	78.5
NER+MLM (Lange et al.)	68.7	83.2	81.6
+ cysec. NER model	$84.3_{\pm 1.7}$	$93.4_{\pm 0.8}$	$89.2_{\pm 0.8}$
+ cysec. NER model^{+X}	$\mathbf{87.2}_{\pm 0.7}$	$\mathbf{94.2}_{\pm 0.4}$	$\mathbf{92.2}_{\pm 0.3}$

Table 6: **Temporal tagging** results. Metrics: F1 sores for the extraction and normalization.

	ORG		LOC		CON	
Model	Acc.	R@10	Acc.	R@10	Acc.	R@10
BLINK	50.4	**66.4**	85.7	97.8	64.5	**87.3**
GENRE	**57.2**	60.0	**93.7**	**97.9**	**64.7**	77.9

Table 7: **Entity disambiguation** results on gold-standard general entities for linking against **Wikipedia**. Metrics: accuracy (R@1) / recall@10.

with our domain-specific cybersecurity NER model (+18.5/11.0 pp.). As a result, the normalization relying on the extracted expressions performs better as well (+10.6 pp.).

5.4. Entity and Concept Disambiguation

We report the results for entity disambiguation when linking against Wikipedia in Table 7. While precision is higher for the top-1 predictions of GENRE, it suffers from lower recall compared to BLINK for ORG and CON entities. This means that likely, GENRE prunes away good candidates too early. The scores for ORG are lower compared to the other two types, indicating that selecting the correct entity is harder for ORGs, often due to the presence of several likely candidates.

Results for linking against MITRE ATT&CK are shown in in Table 8. In our zero-shot experiments, we use the standard BLINK and GENRE models trained on Wikipedia (+ AIDA-CoNLL) and evaluate them for linking to MITRE ATT&CK. We also experiment with fine-tuning the models on our corpus and we retrain a GENRE model from an initial BART checkpoint without any previous entity disambiguation training. We find that the zero-shot models perform already reasonably well on GROUP, MALWARE, and TOOL. Fine-tuning on AnnoCTR for these entity types mostly reduces the performance, which may be caused by catastrophic forgetting of the original training or overfitting.

By contrast, fine-tuning on AnnoCTR is essential for correctly linking tactics and techniques. In particular, the retrained GENRE model outperforms all other models for techniques by a large margin. This task is quite different from standard entity disambiguation, as the textual surface form often notably differs from the KB concept's title in con-

Model	GROUP		MALWARE		TOOL		TACTIC		TECHNIQUE	
	Acc.	R@10	Acc.	R@10	Acc.	R@10	Acc.	R@10	Acc.	R@10
BLINK (zero-shot)	82.6	85.5	91.0	96.7	**100.0**	**100.0**	57.7	82.0	11.6	29.2
+ finetuned	82.2	87.8	89.1	96.9	98.1	**100.0**	**85.1**	95.7	45.1	68.6
GENRE (zero-shot)	**86.4**	**89.2**	**95.0**	**97.1**	88.5	88.5	66.7	75.7	14.6	19.8
+ finetuned	11.5	28.8	64.8	64.8	9.9	27.3	64.6	80.6	39.7	65.0
retrained	49.8	89.2	64.4	97.0	46.2	**100.0**	83.5	**96.5**	**66.9**	**87.8**

Table 8: **Entity disambiguation** results on gold-standard cybersecurity entities for linking against **MITRE ATT&CK**. Metrics: accuracy (R@1) / recall@10. Explicit and implicit TACTICs and TECHNIQUEs.

trast to when linking against Wikipedia or to linking GROUP/MALWARE/TOOL. The experiments in Table 8 use gold standard entity spans and types.

5.5. Text-Classification-based Tactic and Technique Disambiguation Results

We evaluate the identification of tactics and techniques mentioned in a document in an end-to-end setting, as proposed in Section 4.4. As a first step, we train a RoBERTA-based text classification model to identify sentences containing a TECHNIQUE or TACTIC mention. The model achieves F1 scores of 77.0% and 59.4% for the two labels, respectively.

The sentences that are predicted to mention a tactic or technique are then classified with regard to which MITRE ATT&CK concepts they mention. We retrain several GENRE models, as this model has shown the most promising results for these classes in our previous experiments. Moreover, we augment the training data with an additional 7972 sentences taken from the technique/tactic descriptions as found in ATT&CK, labeled with the corresponding tactic/technique. For techniques, we also add the TRAM dataset to the training. As a baseline for comparison, we train the GENRE model with a negative class for sentences that do not contain a TACTIC or TECHNIQUE. This model receives all sentences as input. In order to estimate the impact of the first sentence classification step, we compare to using the sentences containing a TECHNIQUE or TACTIC annotation in the gold standard.

Table 9 shows the results in terms of the average F1 score of detecting the set of ATT&CK techniques/tactics mentioned in a document. We find that our domain-specialized models perform a lot better than the zero-shot model and the baselines. In particular, the models leveraging the extra training data from TRAM or the MITRE ATT&CK descriptions work best. For techniques, we find that the approach utilizing our text classification model for the sentence filtering performs the best. In contrast, this method performs worse than the linking model using a negative class for tactics. As

Model	**Techn.**	**Tactic**
TRAM baseline	24.8	-
TMM (Pujari et al., 2021)	35.3	36.1
Sentence-level linking models with negative class		
GENRE (AnnoCTR)	42.5	45.8
GENRE (AnnoCTR +KG desc.)	45.2	47.7
GENRE (AnnoCTR +TRAM)	47.1	-
GENRE (AnnoCTR +TRAM+KG desc.)	43.7	-
GENRE (TRAM)	25.3	-
Sentence-level linking models + text classification		
GENRE (zero-shot)	23.4	21.9
GENRE (AnnoCTR)	52.9	**39.2**
GENRE (AnnoCTR +KG desc.)	**56.6**	36.7
GENRE (AnnoCTR +TRAM)	56.0	-
GENRE (AnnoCTR +TRAM + KG desc.)	56.5	-
GENRE (TRAM)	25.9	-
Sentence-level linking models + gold techn. detection		
GENRE (zero-shot)	24.8	51.7
GENRE (AnnoCTR)	58.9	**84.2**
GENRE (AnnoCTR +KG desc.)	**65.7**	83.3
GENRE (AnnoCTR +TRAM)	65.4	-
GENRE (AnnoCTR +TRAM+KG desc.)	63.7	-
GENRE (TRAM)	27.7	-

Table 9: **Document-level technique and tactic detection** results: F1 scores micro-averaged over documents.

indicated by the high scores when using oracle sentence selection (lower part of the table), this is caused by the worse performance of the text classification model for TACTIC, which has less than a third training examples compared to TECHNIQUE. We assume that a better text classification model for TACTIC entities will also improve the performance of our GENRE models as suggested by the gold-standard sentence selection results.

6. Conclusion and Outlook

In this resource paper, we have described a new large-scale dataset in the cybersecurity domain annotated with general-world NEs and cybersecurity concepts including tactics and techniques. We have proposed several NLP tasks and provided an

extensive set of experimental results using neural transformer models for NER and linking entities and concepts to Wikipedia and MITRE ATT&CK, demonstrating that the corpus is consistently annotated. Our work lays the foundation for developing cybersecurity-specific NLP models using freely available and permissively licensed data.

Ethical Considerations

The annotators participating in our project were aware of the goal of the annotations and gave their explicit consent to the publication of their annotations. The annotators were paid considerably above the respective country's minimum wages.

While our work attempts to help fighting cybercrime, it is like most NLP work not exempt from the risk of dual use.

Limitations

Our experiments are focused on the AnnoCTR dataset that we describe in this paper. We could not perform larger-scale multi-task or transfer learning with other datasets due to licensing issues as mentioned in Section 2. The exception for which we could try transfer learning was TRAM. While the experiments in Section 5.5 resemble a real-world evaluation, the linking models in Section 5.4 take the gold-standard entities as inputs, which assumes a perfect extraction model. The training of any of our neural models requires a considerable number of computational resources (up to 12 GPU hours), which might not be available for every person/organization.

As knowledge bases are typically continually updated, the recognition and linking models have to deal with unseen classes during real-world inference setups. While the corpus itself cannot cover these new concepts without constant updates, our models are adaptable to such changes. For the entity recognition task, our annotated data guides the model to identify contexts in which an entity usually appears, and hence a full enumeration of possible values is not even the case at present. For entity linking, we require a snapshot of the database, from which we extract a list of all concepts, techniques, tactics, etc. with corresponding textual descriptions. These descriptions are used in the models as targets for decoding, i.e., the model has to output valid entries from the given snapshot. We can exchange or update the knowledge base, such that the model can also output new entities.

7. Bibliographical References

Markus Bayer, Tobias Frey, and Christian Reuter. 2022. Multi-level fine-tuning, data augmentation, and few-shot learning for specialized cyber threat intelligence. *CoRR*, abs/2207.11076.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Robert A. Bridges, Corinne L. Jones, Michael D. Iannacone, and John R. Goodall. 2013. Automatic labeling for entity extraction in cyber security. *CoRR*, abs/1308.4941.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Mach. Learn.*, 20(3):273–297.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou. 2020. CodeBERT: A pre-trained model for programming and natural languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1536–1547, Online. Association for Computational Linguistics.

Houssem Gasmi, Jannik Laval, and Abdelaziz Bouras. 2019. Information extraction of cybersecurity concepts: An lstm approach. *Applied Sciences*, 9(19).

Casey Hanks, Michael Maiden, Priyanka Ranade, Tim Finin, Anupam Joshi, et al. 2022. Recognizing and extracting cybersecurity entities from text. In *Workshop on Machine Learning for Cybersecurity, International Conference on Machine Learning*.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Gyeongmin Kim, Chanhee Lee, Jaechoon Jo, and Heuiseok Lim. 2020. Automatic extraction of named entities of cyber threats using a deep Bi-LSTM-CRF network. *International Journal of Machine Learning and Cybernetics*, 11(10):2341–2355. Funding Information: Funding was provide by Korea Creative Content Agency (Grant No. R2017030045). Publisher Copyright: © 2020, Springer-Verlag GmbH Germany, part of Springer Nature.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Lukas Lange, Jannik Strötgen, Heike Adel, and Dietrich Klakow. 2022. Multilingual normalization of temporal expressions with masked language models. *CoRR*, abs/2205.10399.

Valentine Legoy, Marco Caselli, Christin Seifert, and Andreas Peter. 2020. Automated retrieval of att&ck tactics and techniques for cyber threat reports. *CoRR*, abs/2004.14322.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Swee Kiat Lim, Aldrian Obaja Muis, Wei Lu, and Chen Hui Ong. 2017. MalwareTextDB: A database for annotated malware articles. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1557–1567, Vancouver, Canada. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Hieu Man Duc Trong, Duc Trong Le, Amir Pouran Ben Veyseh, Thuat Nguyen, and Thien Huu Nguyen. 2020. Introducing a new dataset for event detection in cybersecurity texts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5381–5390, Online. Association for Computational Linguistics.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

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)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Peter Phandi, Amila Silva, and Wei Lu. 2018. SemEval-2018 task 8: Semantic extraction from CybersecUrity REports using natural language processing (SecureNLP). In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 697–706, New Orleans, Louisiana. Association for Computational Linguistics.

Subhash Chandra Pujari, Annemarie Friedrich, and Jannik Strötgen. 2021. A multi-task approach to neural multi-label hierarchical patent classification using transformers. In *European Conference on Information Retrieval*, pages 513–528. Springer.

Md. Rayhanur Rahman, Rezvan Mahdavi-Hezaveh, and Laurie A. Williams. 2021. What are the attackers doing now? automating cyber threat intelligence extraction from text on pace with the changing threat landscape: A survey. *CoRR*, abs/2109.06808.

Lalit Mohan Sanagavarapu, Vivek Iyer, and Y. Raghu Reddy. 2021. Ontoenricher: A deep learning approach for ontology enrichment from unstructured text. *CoRR*, abs/2102.04081.

Injy Sarhan and Marco Spruit. 2021. Open-CyKG: An Open Cyber Threat Intelligence Knowledge Graph. *Knowledge-Based Systems*, 233:107524.

Taneeya Satyapanich, Francis Ferraro, and Tim Finin. 2020. Casie: Extracting cybersecurity event information from text. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8749–8757.

Roser Saurı, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. Timeml annotation guidelines version 1.2. 1.

K. Simran, S. Sriram, R. Vinayakumar, and K. P. Soman. 2020. Deep learning approach for intelligent named entity recognition of cyber security. In *Advances in Signal Processing and Intelligent Recognition Systems*, pages 163–172, Singapore. Springer Singapore.

Jannik Strötgen and Michael Gertz. 2010. HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324, Uppsala, Sweden. Association for Computational Linguistics.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA. Association for Computational Linguistics.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society.

Appendix

A. Preprocessing

The texts were retrieved using Python Requests[18]. Due to the URLs, code and image references within the texts, we convert the articles to a format similar to Markdown using BeautifulSoup[19] and Markdownify[20]. Because off-the-shelf sentence segmenters do not perform well on texts that contain many links or code snippets, we use a custom regular expression tokenizer for sentence segmentation and correct sentence boundaries manually before uploading the texts to the web-based annotation system INCEpTION (Klie et al., 2018).

B. Corpus Statistics

Figure 3 and Figure 4 show the distributions over technique and tactic annotations in AnnoCTR, respectively.

Table 10 shows details on the training/dev/test splits of our corpus grouped by CTI vendor.

C. NER results per class

The detailed NER results per class are shown in Table 11.

D. Computational experiments

In the following, we report further information on our computational experiments.

Computing infrastructure. We use V100 gpus for all experiments with neural models.

Number of parameters. The base-sized transformer models (BERT, SciBERT, CodeBERT, RoBERTa) have 110M parameters and take 20 minutes to train for NER. The two BERT-large model used in BLINK the BLINK encoder have 340M parameters each and take 10 hours to train. The GENRE model has 406M parameters and train for 12 hours. The BiLSTM-CRF baseline for NER has 41M parameters and takes 2 hours to train. The TMM model has 145M parameters and takes 1 hour to train.

[18] https://github.com/psf/requests
[19] https://www.crummy.com/software/BeautifulSoup/
[20] github.com/matthewwithanm/python-markdownify

		Intel471	Lab52	ProofPoint	QuoIntelligence	ZScalar	total
Train (GNE+CyNE)	#docs.	18	13	16	7	16	70
	Start date	2016-06-17	2019-04-02	2021-04-15	2018-11-29	2020-06-26	
	End date	2021-04-07	2002-04-14	2021-08-31	2021-01-16	2021-05-11	
Train add.^{X} (GNE)	#docs.	4	5	73	2	196	280
	Start date	2020-04-01	2019-04-02	2017-11-29	2020-01-27	2013-03-08	
	End date	2021-10-20	2021-12-14	2022-02-15	2020-07-20	2021-07-20	
Dev (GNE+CyNE)	#docs.	4	3	4	1	4	16
	Start date	2021-04-19	2020-05-14	2021-09-29	2021-01-27	2021-05-21	
	End date	2021-05-15	2020-06-09	2021-10-27	2021-01-27	2021-07-14	
Test (GNE+CyNE)	#docs.	8	7	8	4	7	34
	Start date	2021-07-14	2020-08-26	2021-10-28	2021-02-16	2021-09-09	
	End date	2021-12-09	2022-01-24	2022-02-03	2021-06-30	2021-11-16	

Table 10: Detailed information on datasplits.

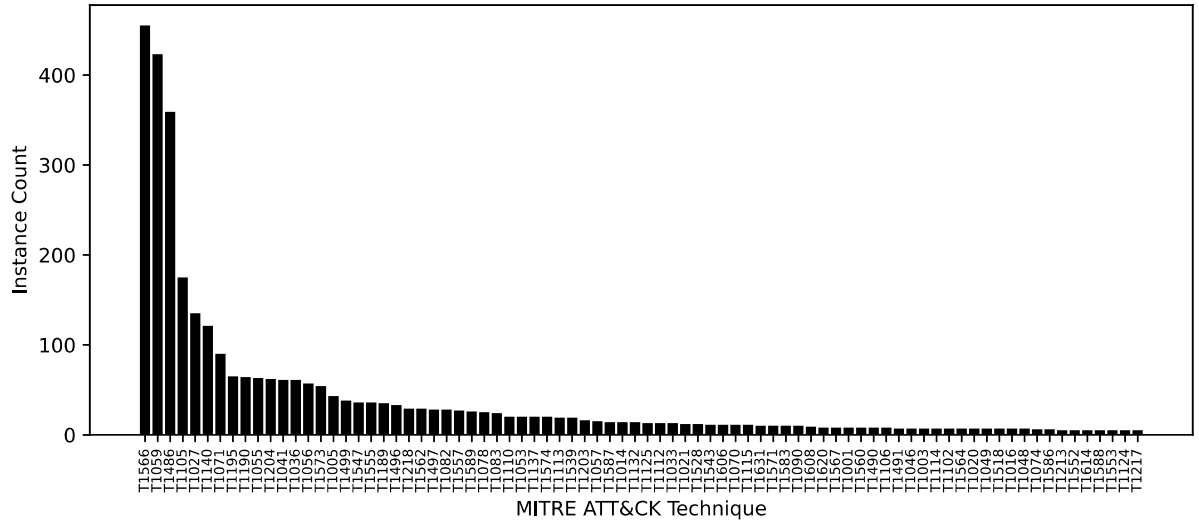

Figure 3: Distribution of technique links annotated in AnnoCTR, showing techniques that occur at least 5 times. In addition, there is a long tail of 136 instances annotated with techniques occurring less frequently.

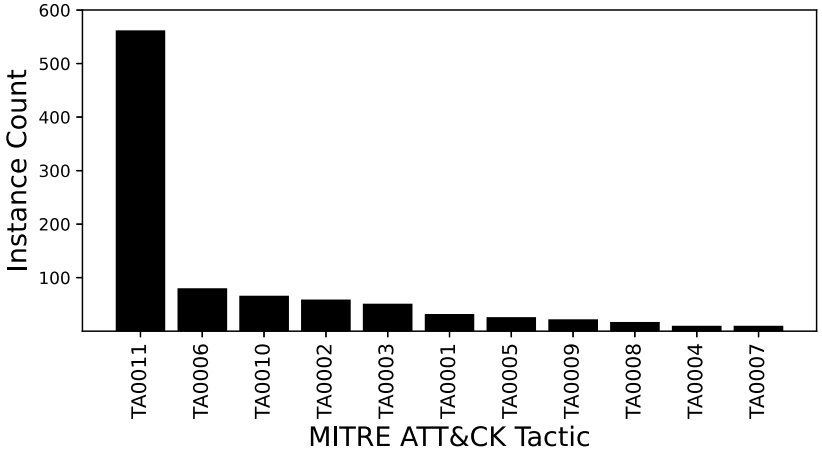

Figure 4: Distribution of tactic links annotated in AnnoCTR.

	BERT	$+X$	SciBERT	$+X$	CodeBERT	$+X$	RoBERTa			$+X$
	F1	F1	F1	F1	F1	F1	Pre.	Rec.	F1	F1
CodeSnippet	$37.6_{\pm 3.9}$	$41.7_{\pm 5.0}$	$34.5_{\pm 5.6}$	*42.0*$_{\pm 6.3}$	$28.9_{\pm 6.7}$	$37.1_{\pm 5.3}$	$30.7_{\pm 4.2}$	$53.3_{\pm 5.6}$	**38.9**$_{\pm 4.7}$	$37.0_{\pm 3.6}$
DATE	**86.8**$_{\pm 1.1}$	$86.6_{\pm 1.0}$	$85.9_{\pm 1.1}$	$86.2_{\pm 0.9}$	$85.3_{\pm 0.8}$	*87.2*$_{\pm 1.2}$	$81.8_{\pm 1.7}$	$87.2_{\pm 1.2}$	$84.4_{\pm 1.2}$	$86.8_{\pm 0.7}$
LOC	$82.5_{\pm 1.0}$	*86.6*$_{\pm 1.4}$	$80.4_{\pm 0.5}$	$84.5_{\pm 0.6}$	$81.2_{\pm 1.9}$	$84.2_{\pm 2.5}$	$83.2_{\pm 2.4}$	$84.7_{\pm 0.5}$	**83.9**$_{\pm 1.1}$	$86.2_{\pm 0.9}$
ORG	$80.9_{\pm 2.0}$	$86.4_{\pm 0.8}$	$79.1_{\pm 2.1}$	$85.6_{\pm 0.8}$	$80.4_{\pm 1.9}$	$86.4_{\pm 0.8}$	$81.2_{\pm 1.5}$	$91.0_{\pm 1.1}$	**85.8**$_{\pm 1.2}$	*88.5*$_{\pm 1.1}$
SECTOR	$52.8_{\pm 2.3}$	$55.1_{\pm 2.1}$	$56.3_{\pm 3.2}$	*60.4*$_{\pm 3.5}$	$50.7_{\pm 3.2}$	$55.2_{\pm 4.5}$	$56.4_{\pm 3.5}$	$56.1_{\pm 3.8}$	***60.4***$_{\pm 3.4}$	$63.3_{\pm 2.1}$
CON	$61.8_{\pm 1.0}$	-	$63.4_{\pm 2.6}$	-	$64.6_{\pm 1.3}$	-	$69.5_{\pm 2.8}$	$66.8_{\pm 1.4}$	**68.1**$_{\pm 1.2}$	-
GROUP	$43.6_{\pm 3.6}$	-	$45.6_{\pm 2.6}$	-	$40.5_{\pm 3.5}$	-	$70.6_{\pm 5.7}$	$37.3_{\pm 1.0}$	**48.7**$_{\pm 1.6}$	-
MALWARE	$58.9_{\pm 3.6}$	-	$62.9_{\pm 2.9}$	-	$65.5_{\pm 5.0}$	-	$69.1_{\pm 2.7}$	$69.6_{\pm 5.1}$	**69.3**$_{\pm 3.2}$	-
TOOL	$8.1_{\pm 2.3}$	-	$16.6_{\pm 3.3}$	-	**18.0**$_{\pm 15.2}$	-	$8.5_{\pm 4.5}$	$8.1_{\pm 2.8}$	$8.2_{\pm 3.5}$	-
TACTIC	$54.1_{\pm 2.9}$	-	$57.0_{\pm 2.0}$	-	$54.7_{\pm 3.4}$	-	$60.0_{\pm 4.3}$	$57.5_{\pm 3.8}$	**58.6**$_{\pm 3.4}$	-
TECHNIQUE	$37.1_{\pm 1.3}$	-	$36.0_{\pm 0.8}$	-	$38.5_{\pm 2.8}$	-	$39.8_{\pm 5.0}$	$42.9_{\pm 1.6}$	**41.2**$_{\pm 3.3}$	-

Table 11: Sequence Labeling results (average Micro F1 and the standard deviation of 5 runs) We report recall and precision for the best model (RoBERTa). $^{+X}$ marks models trained with extra data.