Vol.2021-DPS-186 No.47
Vol.2021-CSEC-92 No.47
2021/3/16

Cyber Threat Intelligence の構造化による分析支援手法の提案

藤井翔太[†1,2]　川口信隆[†1]　重本倫宏[†1]　山内利宏[†2]

概要：サイバーセキュリティの脅威は年々増加するとともに，巧妙化が進んでいる．このような状況下において，CTI（Cyber Threat Intelligence）の収集と最新の脅威情報への追従がより重要となっている．特に STIX（Structured Threat Information eXpression）のような構造化された CTI は，FW・IDS ルールの更新や攻撃傾向の分析など，セキュリティ運用を自動化できるため有用である．一方で，CTI の多くは自然言語の形で記述されており，ドメイン知識も必要であることから，手作業での分析および構造化には多くのコストを要する．

そこで本稿では，CTI を自動的に構造化し，STIX へと変換する手法を提案する．提案手法は，CTI の文脈における固有表現を抽出するとともに，固有表現と IOC 間の関係を抽出することにより，STIX への変換を図る．本稿では，提案手法の設計と実装を述べるとともに，実際の CTI に対し提案手法を適用し，F 値 0.78 の精度で固有表現を抽出可能なことと最大正解率 81.6%で固有表現と IOC 間の関係を抽出可能なことを示す．加えて，提案手法の処理時間が実業務の範囲内であることを示す．

キーワード：CTI，情報抽出，固有表現抽出，関係抽出，STIX

Proposal of Method to Support Analysis by Structuring Cyber Threat Intelligence

SHOTA FUJII[†1,2]　NOBUTAKA KAWAGUCHI[†1]
TOMOHIRO SHIGEMOTO[†1]　TOSHIHIRO YAMAUCHI[†2]

Abstract: Cybersecurity threats have been increasing and are getting more sophisticated year by year. In such circumstances, gathering Cyber Threat Intelligence (CTI) and following up with up-to-date threat information are important. In particular, structured CTI such as STIX (Structured Threat Information eXpression) is useful because it can automate security operations such as updating FW/IDS rules and analyzing attack trends. On the other hand, most CTIs are written in natural language; therefore, it required manual analysis with domain knowledge and it's time-consuming. In this paper, we propose a method for automatically structuring CTI and converting it into STIX format. The proposed method extracts named entities in the context of CTI, and also extracts relations between named entities and IOCs, in order to convert them into STIX. In this paper, we describe the design and implementation of the proposed method. In addition, this paper shows the proposed method can extract named entities with F-measure of 0.78 and extract relations between named entities and IOCs with a maximum accuracy of 81.6%. This paper also shows the processing time of the proposed method is within the range of actual work.

Keywords: CTI, Information Extraction, Named Entity Recognition, Relation Extraction, STIX

1. はじめに

年々サイバー攻撃が増加・高度化しており，サイバー脅威インテリジェンス（CTI: Cyber Threat Intelligence）を収集し，最新の脅威情報に追従することがますます重要となっている．CTI には，新規の脆弱性・マルウェアの情報，攻撃者の手口，およびそれらに対する対策手法等が記載されている．また，攻撃を検知する指標として IOC（Indicator Of Compromise）が含まれることも多い．IOC は，例えば不審サイトの IP アドレスや URL，マルウェアのハッシュ値等から成る．こうした情報をファイアウォールや侵入検知システムの検知ルールに織り込むことにより，攻撃の検知が可能となる．このように，CTI に含まれるマルウェア情報，脆弱性情報，および IOC 等を適切に抽出して活用することにより，検知ルールの構築や攻撃傾向の分析が可能となる．

一方で，CTI は，まずブログ，ニュースサイト，および SNS といった媒体において，非構造データとして配信されることが多く，それらの情報が公開されてから構造化されるまでの間にはタイムラグが存在し，1 ヶ月以上を要する場合もある [1]．このため，最新の脅威情報に追従するには，非構造なデータを分析・活用する必要がある．しかし，例えばセキュリティブログに限定しても月 60,000 件以上 [2] と日々大量の CTI が発行されており，そのすべてを人手で分析することは現実的ではない．また，多くの CTI は自然言語で記述されていることから，単純に機械処理を実施することは困難である．こうした背景から，自然言語で記述された CTI を機械処理可能な形に構造化し，効率的な分析の支援を行うことが重要である．

こうした課題を受けて，辞書やオントロジを作成することによって，非構造データの構造化を試みる研究がある [3-6]．ただし，セキュリティ分野では，新たなマルウェアの出現や脆弱性の発見，コードネームの付与等により，新語が生まれやすいことから，継続的な辞書やオントロジのメンテナンスが容易ではない．また，URL や IP アドレス

†1 株式会社日立製作所
Hitachi Ltd.
†2 岡山大学 大学院自然科学研究科
Graduate School of Natural Science and Technology,
Okayama University

等の IOC は形式が定まっているため，正規表現を用いることで抽出可能ではあるものの，どのようなマルウェアや攻撃者に用いられるものであるかといった文脈情報が欠落してしまうため，それだけでは分析や検知ルールとしての適用可否の判断に活用することが困難である．

これらの課題を緩和するべく，固有表現抽出と関係抽出を用いた CTI の構造化手法を提案する．提案手法は，マルウェア名や脆弱性名のように，サイバーセキュリティの文脈で着目すべき固有表現を抽出することにより，分析の効率化を狙うとともに，固有表現間の関係を抽出することにより，文脈情報を維持した形で構造化を図る．この際，汎用フォーマットである STIX（Structured Threat Information eXpression [7]）の形式で構造化することにより，活用性の向上を図る．

本稿の貢献は以下の通りである：

- CTI から固有表現や固有表現間の関係性を抽出することにより，自動で STIX 2.1 の形式で構造化を行う．この際，CTI の単位でキーフレーズを抽出し，文末に列挙された IOC と紐づけることにより，従前の関係抽出では想定されていないような，近傍に関係を有する語が存在しないものの関係抽出を実現する
- 作成したデータセットを用いて評価を実施し，最大 F 値 0.78 の精度で固有表現を抽出するとともに，IOC と関係を有する固有表現を最大約 81.6%の正解率で抽出可能なことを示した．また，処理時間の測定を行い，提案手法が想定しているユースケースにおいて問題なく利用可能であることを実証した．

2. CTI の構造化に係る背景

2.1 情報抽出

情報抽出は，非構造化文書から構造化データを抽出するタスクである．様々な技術から構成されるタスクであり，文章から固有表現を抽出する固有表現抽出や固有表現間の関係性を抽出する関係抽出等から成る．

また，近年の情報抽出は，Word2Vec に代表されるような言語モデルによって単語や文章を分散表現と呼ばれる数値表現に変換し，各種タスクのインプットとすることで高い精度が達成されている．特に近年では，BERT（Bidirectional Encoder Representations from Transformers [8]）およびそれらをベースとした応用言語モデルが様々なタスクにおいて高い性能を達成している．

前述の BERT は，双方向の transformer から成るモデルであり，大規模なコーパスを用いて言語モデルを獲得する事前学習と各タスクに合わせてラベル付きデータを用いてファインチューニングする 2 つのステップで構成される．事前学習のステップでは，ランダムにマスクした単語列を予測するというタスク（MLM: Masked Language Model）と 2 つの文章をインプットし，それらが連続した文章か否かを予測するタスク（NSP: Next Sentence Prediction）を解くことにより，言語モデルを獲得する．その後，ファインチューニングのステップにおいて，各タスクのレイヤを追加し，ラベル付きデータで学習を行う．BERT は，共通の事前学習モデルを基に各タスクにファインチューニングすることにより，多数のタスクで高精度を達成している．また，今日では様々な事前学習モデルが公開・共有されており，それらを利用することで手元での事前学習なしに高精度なモデルを構築することが可能となっている．

2.2 サイバーセキュリティにおける構造化フォーマット

サイバーセキュリティ分野において，セキュリティ情報の機械可読化や共通フォーマットでの情報交換を目的に，様々な構造化フォーマットが策定されている．具体的には，IOC に特化した OpenIOC [9] やそれよりも広範な情報を対象とした STIX [7] や MISP [10] 等がある．

3. 提案手法

3.1 提案手法の目的と要件

前述の通り，共通フォーマットに合わせて構造化された CTI は様々な利点がある一方で，構築コストが大きく，非構造の CTI を全て人手で構造化するのは現実的ではない．そこで，本稿における提案手法は，CTI を自動的に共通フォーマットで構造化し，効率的な分析の支援を行うことを目的とする．本目的を達成するための要件を以下に示す．

（要件 1）CTI の自動的な共通フォーマットでの構造化

自然言語で記述された CTI を自動で構造化し，分析を支援する．この際，可視化やセキュリティアプライアンスとの連携等，整備されている機能を活用することができるため，2.2 節に示したような共通フォーマットで構造化することが望ましい．

（要件 2）遠距離にある関係性の抽出

一般ドメインの自然言語では，関係性を有する固有表現は同一あるいは近傍の文章に共起している場合が多く，既存の自然言語処理における関係抽出手法も同一文あるいは近傍数文を関係性の探索範囲としていることが多い [11]．一方で，CTI においては，IOC がこの前提から外れる場合がある．頻繁にみられるのは，特定のマルウェアや攻撃キャンペーンについて本文中で解説を行い，同マルウェアや攻撃キャンペーンに関連する IOC を文末にリストとして列挙するものである．例えば，CTI の本文中で特定のマルウェアについて詳述し，文末に当該マルウェアの IOC を列挙しているものである．この場合，文末の IOC には，関連する固有表現としてマルウェア名等が関連付けられるべきであるが，近傍に当該の語が存在しないため，既存の関係抽出手法では本関係性を抽出することは難しい．このような遠距離にある関係性を抽出する必要がある．

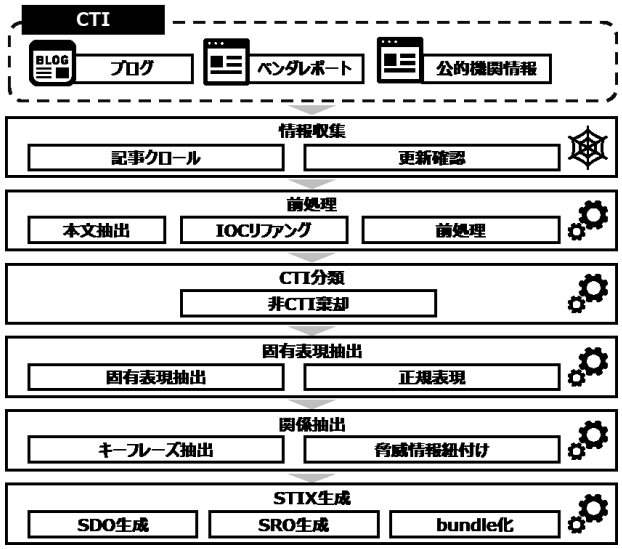

図 1 提案手法の全体像

3.2 基本方針と全体像

まず，（要件 1）を満たすために，固有表現抽出や関係抽出といった情報抽出技術を活用して CTI の構造化を図る．この際, 共通フォーマットとして 2021 年 2 月現在における最新版である STIX 2.1 を利用する．STIX 2.1 はサイバーセキュリティの文脈におけるドメイン語のオブジェクト（SDO: STIX Domain Object）と同オブジェクト間の関係（SRO: STIX Relationship Object）から成る．そこで，自然言語で記述された CTI から前者を固有表現抽出，後者を関係抽出で抽出し，STIX へと変換する．

また，（要件 2）を満たすために，文末に列挙されている IOC は，当該 CTI を代表する語と関係があるという仮定を置き，キーフレーズ抽出を活用することにより，IOC に係る遠距離の関係性を抽出する．具体的には，CTI からキーフレーズを抽出し，固有表現抽出で抽出した固有表現と一致するものが当該 IOC を表す語と判断して関係性を結ぶ．これにより，IOC を表す語が近傍に存在しない場合でも，関係性を抽出することが期待される．

上記を踏まえた提案手法の全体像を**図 1** に示す．まず，CTI を発信しているサイトから記事を収集し，後段の処理のための前処理を行う．その後，記事の分類によって CTI のみを抽出し，そうでないものを棄却する．その後，固有表現抽出や関係抽出によって, STIX として記載すべき情報を抽出する．最後に，抽出した情報を STIX フォーマットとして整形し，出力する．

以降の節では各ステップにおける処理の詳細を述べる．

3.3 情報収集

まず，CTI を発信しているサイトに対してクロールを行い，既存の記事を全て収集する．その後，当該サイトの記事更新有無を定期的に確認し，更新された新しい記事のみを収集する事により，記事の重複や当該サイトへの可不可を抑制する. なお, RSS を提供しているサイトであれば RSS を利用して更新記事を取得し，そうでないサイトに関しては，トップページを解釈し，新規記事の有無を判定と取得を行う．

3.4 前処理

後段の自然言語処理に先駆け，収集した記事に対して前処理を行う．Web ページをクローリングした際，CTI を含む本文の他に，広告やナビゲーション等の不要な情報も含まれていることが多い．そこで，前処理フェーズにおいては，まず本文抽出を行い，不要な情報を除去する．本機構は，BeautifulSoup [12] によって html タグを解析し，本文部分を抽出することにより実装した．

次に，IOC のリファングを行う．CTI においては，URL や IP アドレスに関する IOC を記載する際，誤クリックによる意図しない悪性サイトへのアクセスを抑制するために，デファングされていることが多い（例：example.com→example[.]com）．ただし，このままの形では，固有表現として抽出するべきである URL や IP アドレスが正規表現で抽出できない．そこで，デファング状態の IOC をデファング前の状態に戻す（リファングする）ことにより，本来のフォーマットに戻し，正規表現で抽出できるようにする．事前に”[.]”を”.”に置換する等のリファングルールを定義し，ルールベースでの検索と置換により，リファング機構を実現した．

ここまでの前処理を行った上で，収集した情報を後段の自然言語処理を実施するために適した形にするための処理を実施する．まず，言語モデルで処理できるように，抽出した本文を文章ごとに分割する．また，不要な情報による固有表現抽出への悪影響を抑制するため，ストップワードの除去を行う．これらの実装には NLTK [13] を用いた．

3.5 CTI 分類

CTI を提供するブログや公式ページの中には，製品やセミナの紹介記事を含むものもある．それらは CTI ではないため，CTI か否かを判定する 2 値分類器を構築することにより, 棄却する. 今回は, huggingface [14] の学習済み BERT をファインチューニングすることにより，入力された文章を基に，CTI か否かを判定する 2 値分類器を構築した．

3.6 固有表現抽出

ここまでの処理で自然言語処理に適した形となったコーパスに対し，固有表現抽出を実施する．固有表現の抽出に際して，抽出する項目を拡張固有表現として定義した（**表 1**）．前述の通り，提案手法は STIX 2.1 の形式に合わせて構造化を実施する．そこで，STIX におけるオブジェクト（SDO）と対応する形で固有表現を定義した．

まず，正規表現で抽出可能な IP アドレスや URL，CVE 番号や各種ハッシュ値等については, 正規表現で抽出する．それ以外の固有表現に，固有表現抽出モデルによって抽出する．また，固有表現抽出モデルについては，今回は，CTI 分類と同様に huggingface の事前学習済み BERT を固有表現抽出用にファインチューニングすることで実装した．

表 1 提案手法における固有表現

STIX 項目	抽出項目	説明	例	抽出方法
Attack Pattern	name	攻撃パターン名	Spear Phishing	固有表現抽出
Campaign	name	キャンペーン名	Operation Aurora	固有表現抽出
Grouping	name	脅威アクタ名	APT10	固有表現抽出
Identity	name	名前	Hitachi, Ltd.	固有表現抽出
Indicator	pattern	IOC	URL，ハッシュ値，ファイル名等	正規表現
Malware	labels	マルウェアの種類	ransomware	固有表現抽出
	name	マルウェア名	WannaCry	固有表現抽出
Tool	name	ツール名	metasploit	固有表現抽出
Vulnerability	name	脆弱性名	CVE-2014-0160 HeartBleed	正規表現 固有表現抽出

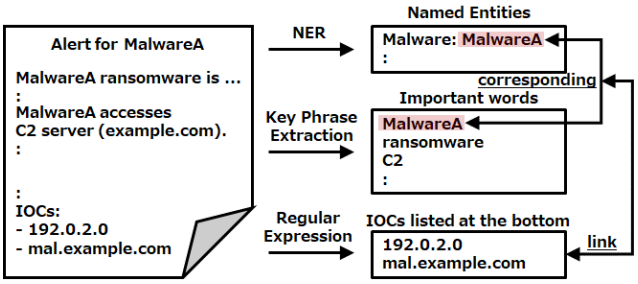

図 2 文末の IOC に係る関係抽出手法

表 2 関係抽出のルール

主体	客体	関係性
indicator - ハッシュ値 - ファイル名	attack_pattern campaign_name malware_name threat_actor_name	indicates
malware_name	indicator - URL - IP アドレス	communicates-with

3.7 関係抽出

前段で抽出した固有表現間の関係性を抽出することで，文脈情報を確保する．ここで，3.2 節で述べた方針に基づいて，遠距離に存在する IOC と固有表現の関係性の抽出を図る（図 **2**）．具体的な処理の流れは以下の通りである：

(1) 固有表現抽出の際に正規表現で抜き出した IOC のうち，文末に単独で存在するものを関係性抽出候補として抽出する．

(2) キーフレーズ抽出により，CTI を代表するキーフレーズ上位 10 件を抽出する．

(3) 抽出済みの固有表現のうち，IOC と関係性を有しうる性質の固有表現を性質毎にキーフレーズ群と比較し，最も上位のキーフレーズと一致したものを事前に定義した関係性を有するとして IOC と紐づける．なお，IOC と関係性を有しうる固有表現の性質は，attack_pattern，campaign_name，malware_name，threat_actor_name である．即ち，一つの IOC につき，最大 4 つの固有表現との関係性が付与される．

上記の処理により，既存手法では抽出困難な IOC に係る遠距離の関係性を抽出する．なお，IOC と固有表現の関係性の定義は，**表 2** に示すとおりである．本定義についても，拡張固有表現と同様に，STIX 2.1 の SRO に合わせる形で定義した．

また，キーフレーズ抽出の手法としては，幾つかの手法を実装・比較した中から，今回のテストデータに対して最も精度の高かった MultipartieRank [15] を利用した．

3.8 STIX 生成

ここまでの処理で抽出した固有表現と固有表現間の関係を用いて STIX を生成する．具体的は，まず固有表現抽出で抜き出した固有表現を対応する SDO に変換する．次に，関係抽出で抜き出した関係があれば，STIX オブジェクト間の関係を定義する SRO に変換する．最後に，STIX のオブジェクト群をグループとしてひとまとめにする bundle を用いて，CTI 毎に一つの STIX オブジェクトとして生成する．本機構の実装には，cti-python-stix2 [16] を用いた．

以上の機能群により，CTI の構造化を実現し，それを用いることによる SOC/CSIRT 業務における CTI 分析の自動化・効率化を目指す．

Vol.2021-DPS-186 No.47
Vol.2021-CSEC-92 No.47
2021/3/16

4. 評価

4.1 評価項目

先述の設計に沿って提案手法のプロトタイプを実装し，以下に示す 4 つの評価を実施した：

（評価 1）CTI 分類精度

提案手法は，STIX 生成に先駆けて，収集した情報のうち CTI でないものを棄却する．本評価ではこの分類の正解率を評価する．

（評価 2）固有表現抽出精度

提案手法は，SDO 抽出用にファインチューニングした BERT モデルを用いて，表 1 に示す固有表現の抽出を試みる．本評価では，固有表現抽出の精度を適合率，再現率，および F 値の軸から評価する．

（評価 3）関係抽出精度

提案手法は，固有表現抽出の結果とキーフレーズ抽出の結果を突き合わせることにより，文末の IOC に関する関係を抽出する．本評価では，同抽出の正解率を評価する．

（評価 4）処理時間

提案手法は日々の分析業務で活用することを想定している．そこで，各処理の処理時間を測定し，日々の運用と照らし合わせて実用の範囲内に収まっているかを評価する．本測定は，GUI マルチユーザモード下において Python の time モジュールを用いて行った．

4.2 データセット

前節で述べた各評価を実施するにあたり，既存研究や実務者へのヒアリングを基に CTI を発信している 34 のサイトを選定した．また，各サイト用にクローラを実装し，2001 年 6 月～2020 年 12 月の間に公開された 75,652 件の CTI 候補を収集した後，以下に示す 3 つのデータセットを評価用に構築した．

・分類用データセット

CTI 分類の精度を評価するため，収集した CTI をランダムにピックアップし，CTI/非 CTI それぞれを 200 ずつ合計 400 件のラベル付きデータを用意した．

・固有表現抽出用データセット

収集した CTI をランダムにピックアップし，固有表現に関してアノテーションした CTI を 100 件分用意した．なお，本データセットは，13,479 文・193,027 単語から成り，のべ 4,562 の固有表現を含む．

・IOC 関係性抽出用データセット

収集した CTI の中から IOC を一つ以上含むものをランダムにピックアップし，IOC についてその性質をラベル付けした CTI を 100 件分用意した．なお，本データセットには，2,371 件の IOC が含まれる．

以降の評価は上記のデータセットを用いて実施したものである．なお，評価はいずれも**表 3** に示す環境で行った．

表 3 評価環境

CPU	Intel Xeon E5-2698 v4 (2.2 GHz, 20 cores)
メモリ	256 GB 2133 MHz DDR4 LRDIMM
GPU	Tesla V100 (VRAM 32 GB) ×4 (VRAM 128 GB)
OS	Ubuntu 18.04

表 4 固有表現抽出のモデルごとの精度

モデル	適合率	再現率	F 値
bert_base_uncased	0.84	0.71	0.77
bert_base_cased	0.84	0.73	0.78

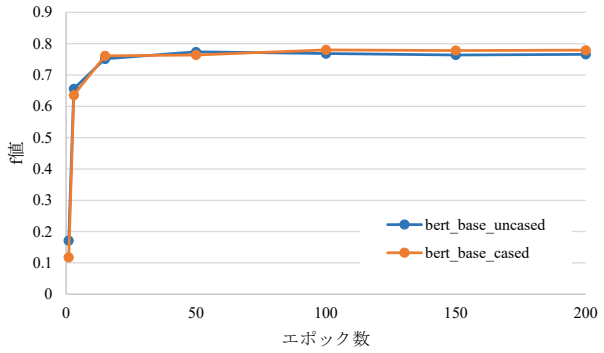

図 3 エポックごとの F 値の推移

4.3 評価結果

4.3.1 評価 1：CTI 分類精度

データセットのうち，50%にあたる 100 件を学習に，残り 100 件を評価用に用いた．この際，学習用データを更に 70%の訓練用セットと 30%検証用セットに分割し，cased な BERT をファインチューニングする形で学習を実施した．評価の結果，評価用データに対して 97.0%の正解率で分類することができた．以降の評価は，本分類器を用いて CTI と判断したものを用いて実施している．

4.3.2 評価 2：固有表現抽出精度

本評価では，70%にあたる 70 記事分を学習に，30 記事分を検証用に用いた．この際，学習用データを更に 70%の訓練用セットと 30%検証用セットに分割して学習を実施した．また，モデルには huggingface で利用可能な事前学習モデルのうち，大文字小文字を区別する bert_base_cased と全て小文字として扱う bert_base_uncased を利用し，200 エポック学習を実施した．両モデルの最終的な精度を**表 4** に，エポックごとの F 値の推移を**図 3** に示す．

両モデル共に 100 エポック付近で飽和するまではエポックごとに精度が向上しており，最終的にはわずかではあるが cased な BERT の方がいずれの指標においても同等，あるいは高い精度を示している．これは，マルウェア名や脆弱性名等，CTI の文脈での固有表現が大文字で表現されることがあることに起因すると推察される．

Vol.2021-DPS-186 No.47
Vol.2021-CSEC-92 No.47
2021/3/16

表 5 固有表現抽出の項目ごとの精度

項目	適合率	再現率	F 値	固有表現数
attack_pattern_name	0.92	0.92	0.92	63
campaign_name	1.00	1.00	1.00	1
grouping_name	0.92	0.67	0.78	278
identity_name	0.86	0.82	0.84	327
malware_label	0.86	0.96	0.91	373
malware_name	0.76	0.52	0.62	174
tool_name	0.54	0.71	0.61	31
vulnerability_name	0.85	0.82	0.84	102
平均/合計	0.84	0.73	0.78	1,349

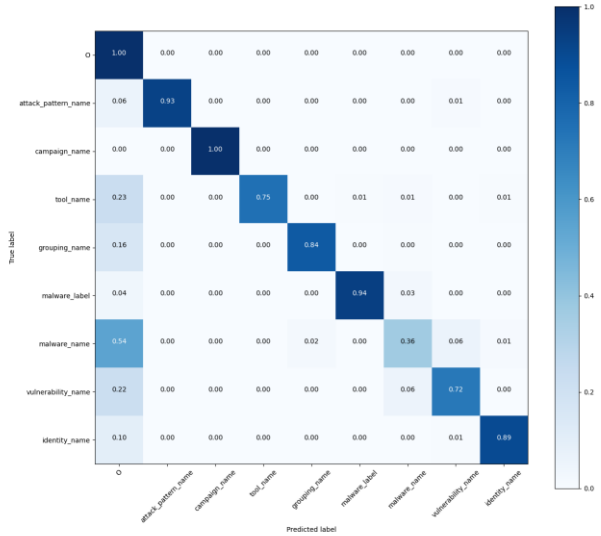

図 4 固有表現抽出における混同行列

以降の評価における固有表現抽出に関しては，精度の高かった bert_base_cased の BERT を用いている．

次に，項目ごとの精度を**表 5** に，混同行列を**図 4** に示す．表 6 から，malware_name と tool_name の抽出精度がそれ以外の項目と比較してやや低い傾向であると言える．これは，マルウェア名やツール名は命名規則が自由なためにバリエーションが多いことに加え，学習用データに存在しない新語も比較的多く含まれることが原因であると推察される．**図 4** を見てもそれらの項目の多くが O（固有表現ではない項目）に誤って分類されていることが見受けられる．

4.3.3 評価 3：関係抽出精度

本評価では，キーフレーズ抽出手法として PositionRank [17], TopicRank [18], および MultipartieRank [15] を選定し，各手法を用いて 3.7 節で述べた関係抽出を実施した際の正解率を測定した．この際，いずれの手法においてもキーフレーズの候補としては，名詞，固有名詞，および形容詞のみを利用する．その後キーフレーズとして抽出されたもの固有表現一覧を比較し，一致するものを関係語として抽出する．この評価での正解率を**表 6** に示す．

表 6 関係抽出の精度

キーフレーズ抽出手法	正解率（%）
PositionRank [17]	77.22
TopicRank [18]	69.59
MultipartieRank [15]	81.65

表 7 提案手法の 1 記事に対する処理時間（秒）

前処理	0.42
CTI 分類	1.12
固有表現抽出	0.87
関係抽出	1.92
STIX 生成	0.41
合計	4.74

MultipartieRank を用いた場合が，81.65%と最も高い正解率であった．また，PositionRank も 77.22%とほぼ同等の精度となっている．両手法ともに文の先頭に近い語を優遇する手法であり，上記の評価結果は，CTI での IOC に係るキーフレーズは，文章の先頭に近い語との共起性が高いことを示唆するものであると言える．

4.3.4 評価 4：処理時間

CTI の前処理から STIX 生成までに要する時間を**表 7** に示す．表に示した値は，（評価 1）で構築した分類器を利用し，75,652 件のうち CTI であると判断された 52,187 件に対する処理の平均時間である．なお，文章の平均長は，987 行であった．

表 7 に示した通り，各処理の合計で，1 記事につき約 4.74 秒を要する．月 60,000 件 [2] の CTI に対応するには，1 日に約 2,000 件対応する必要があるが，1 記事に約 4.74 秒かかる場合，2,000*4.74=9,480 秒（約 2.63 時間）であることから，対応可能である．

また，上記の処理の前段として，CTI 分類と固有表現抽出に利用する BERT のファインチューニングにも処理時間が発生する．そこで，それぞれについて 200 エポックで学習を実施して処理時間を計測したところ，CTI 分類に約 2,100 秒（1 エポック約 10.5 秒），固有表現抽出に約 3,400 秒（1 エポック約 17.4 秒）の時間を要した．これらのファインチューニングは日時や週次でバッチ処理を行うことが想定されることから，こちらも実業務に耐えうると推察される．

以上のように，日時や週次で行うファインチューニングに合計で約 5,500 秒（約 1.5 時間），STIX 化に 1 記事平均約 4.74 秒，想定される 1 日の記事量 2,000 に対しても約 2.63 時間と，いずれも実用の範囲内であると言える結果となった．

Vol.2021-DPS-186 No.47
Vol.2021-CSEC-92 No.47
2021/3/16

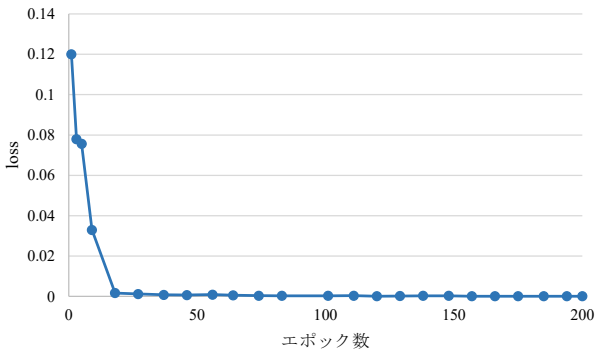

図 5 固有表現抽出におけるエポック毎の loss 値

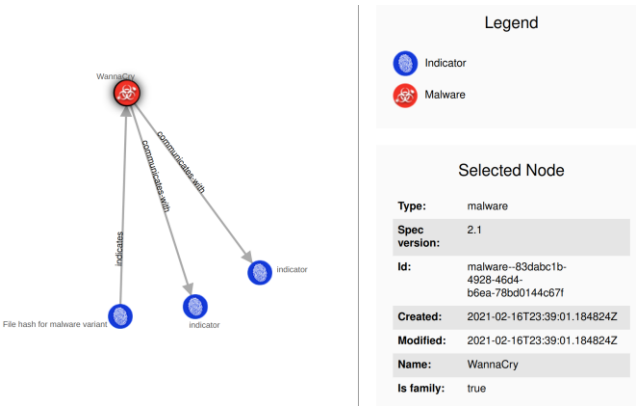

図 6 提案手法で生成した STIX の可視化

5. 議論

5.1 精度

各種タスクの精度については，本結果をもって実業務に充分であるかは検討する必要がある．また，固有表現の見落としが発生した場合，SDO として STIX に含むことができないだけでなく，後段の関係抽出も不可能になることから，影響範囲が大きい．本タスクにおいては，再現率を優先する方向にパラメータを調整する事が望ましいと推察される．

5.2 処理時間

今回，処理時間の評価は，記事毎に直列に処理を行って計測したが，実際は記事毎の並列化や処理毎のパイプライン化が可能なため，さらに高速化可能である．また，ファインチューニングにおいても，固定で 200 エポック分学習を実施したが，**図 5** に示す通り 90 エポック付近で収束しているため，Early-stopping [20] 等を用いて学習を打ち切ることにより，より短い時間で処理を完了することができる．

5.3 ユースケース

前述の通り，提案手法を用いて CTI を構造化することにより，検知ルールの構築や攻撃傾向の分析の支援に資する効果が期待できる．また，共通フォーマットである STIX として構造化しているため，**図 6** に示すような可視化（STIX Visualizer [21] による可視化）やセキュリティアプライアンスとの自動連係等，整備されている機能の活用も期待できる．

5.4 制限事項

URL や IP アドレスが未知の手法でデファングされていた場合，リファングすることができず，正規表現で抽出出来ない可能性がある．ただし，多くの場合 CTI 提供サイトの単位ではデファングの方式は統一されているため，サイトごとに一度リファングルールを整備すれば対応可能であると推察される．

また，提案手法における関係抽出では，一つの CTI に含まれる IOC 群はすべてフラットに取り扱い，全ての IOC が CTI を代表する語へと紐づけられる．即ち，週次レポートのような一つの CTI に複数のトピックが含まれるものには適合しない可能性が高い．

5.5 研究倫理

本稿における評価用の CTI を収集する際，同一のサイトから情報を取得する場合は，アクセス毎に一定の間隔を置いている．加えて，設計の章で述べた通り，記事の更新有無を確認し，更新がない場合はそれ以上のアクセスを試みないようにしている．これらの施策により，CTI 配布サイトに対する必要以上の負荷を低減できる．

6. 関連研究

辞書やオントロジを作成することによって，非構造データの構造化を試みる研究がある [3-6]．ただし，セキュリティ分野では，新たなマルウェアの出現や脆弱性の発見，コードネームの付与等により，新語が生まれやすいことから，継続的な辞書やオントロジのメンテナンスが容易ではない．こうした課題を緩和するべく，提案手法と同様に機械学習ベースの自然言語処理によって非構造データの構造化を試みる研究もある [22-25]．特に，iACE [25] は，固有表現の抽出だけでなく．グラフマイニングを用いて IOC に係る文脈情報の抽出を試みるものである．ただし，本稿で言及したような遠距離にある IOC との関係性抽出は行っていない．

また，構造化以外にも CTI の活用に焦点を当てた研究が多数実施されている．FeatureSmith [26] は，CTI をテキストマイニングすることにより特徴量を生成し，Android マルウェアを検出するモデルを自動構築する．文献 [27] も CTI から検出ルールを自動構築する研究である．TTPDrill [28] は，CTI をテキストマイニングし，記載内容を攻撃手口（TTPs）や Cyber Kill Chain に割り当てるものであり，ChainSmith [29] は，CTI から抽出した IOC の役割を推定するものである．また，POIROT [30] は，audit ログと CTI をそれぞれグラフ化して比較することにより，Threat Hunting を行うものである．これらは，目的は違うものの，提案手法と同様に自然言語処理によって CTI を分析し，活用するものである．

7. おわりに

本稿では，自然言語で記述された CTI の分析を効率化することを目的とし，自動で STIX へと変換する手法を提案した．提案手法は，CTI から固有表現や固有表現間の関係性を抽出することにより，自動で STIX 2.1 の形式で構造化を行う．この際，CTI の単位でキーフレーズを抽出し，文末に列挙された IOC と紐づけることにより，従前の関係抽出では想定されていないような，近傍に関係を有する語が存在しないものの関係抽出を実現する．

評価では，提案手法のプロトタイプを実装し，最大 F 値 0.78 の精度で固有表現を抽出するとともに，IOC と関係を有する固有表現を最大約 81.6%の正解率で抽出可能なことを示した．また，処理時間の測定を行い，提案手法が想定しているユースケースにおいて問題なく利用可能であることを実証した．

今後の課題としては，各タスクの精度向上やより大規模なデータセットを用いた評価が挙げられる．また，STIX 化した CTI の活用方法についても検討を進める予定である．

本稿中で使われているシステム・製品名は，各社の商標または登録商標です．

参考文献

[1] McNeil, N., Bridges, R.A., Iannacone, M.D. Czejdo, B., Perez, N. and Goodall, J.R.: PACE: Pattern Accurate Computationally Efficient Bootstrapping for Timely Discovery of Cyber-security Concepts, 12th International Conference on Machine Learning and Applications, pp. 60－65 (2013).
[2] IBM: IBM Watson to Tackle Cybercrime, available from <https://www-03.ibm.com/press/us/en/pressrelease/49683.wss> (2021-02-07 accessed).
[3] Obrst, L., Chase, P. and Markeloff, R.: Developing an Ontology of the Cyber Security Domain, CEUR Workshop Proceedings, Vol. 96, pp.49－56 (2012).
[4] Iannacone, M., Bohn, S., Nakamura, G., Gerth, J., Huffer, K., Bridges, R., Ferragut, E. and Goodall, J.: Developing an Ontology for Cyber Security Knowledge Graphs, In Proceedings of the 10th Annual Cyber and Information Security Research Conference (CISR 2015), pp. 1－4 (2015).
[5] Lim, S.K., Muis, A.O., Lu, W. and Ong, C.H.: MalwareTextDB: A Database for Annotated Malware Articles, Proceedings of the 55th Annual Meeting of the Association for Computational, Vol. 1, pp.1557－1567 (2017).
[6] Mavroeidis, V. and Bromander, S.: Cyber Threat Intelligence Model: An Evaluation of Taxonomies, Sharing Standards, and Ontologies within Cyber Threat Intelligence, 2017 European Intelligence and Security Informatics Conference (EISIC 2017), pp. 91－98 (2017).
[7] OASIS: Introduction to STIX, available from <https://oasis-open.github.io/cti-documentation/stix/intro.html> (2021-02-07 accessed).
[8] Devlin, J. and Chang, M.W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019), pp.4171－4186 (2019).
[9] Mandiant: OpenIOC, available from <https://github.com/mandiant/OpenIOC_1.1> (2021-02-07 accessed).
[10] MISP project: MISP - Open Source Threat Intelligence Platform & Open Standards For Threat Information Sharing, available from <https://www.misp-project.org/> (2021-02-07 accessed).
[11] Gupta, P., Rajaram, S., Schütze, H. and Runkler, T.: Neural Relation Extraction Within and across Sentence Boundaries, Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2019), pp. 6513－6520 (2020).
[12] Beautiful Soup Documentation, available from <https://www.crummy.com/software/BeautifulSoup/bs4/doc/> (2021-02-07 accessed).
[13] Natural Language Toolkit, available from <https://www.nltk.org/> (2021-02-07 accessed).
[14] Huggingface: Transformers, available from <https://huggingface.co/transformers/> (2021-02-07 accessed).
[15] Boudin, F.: Unsupervised Keyphrase Extraction with Multipartite Graphs, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2018), pp.667－672 (2018).
[16] OASIS: STIX 2 Python API Documentation, available from <https://stix2.readthedocs.io/en/latest/> (2021-02-07 accessed).
[17] Florescu, C. and Caragea, C.: PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017), pp.1105－1115 (2017).
[18] Bougouin, A., Boudin, F. and Daille, B.: TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction, Proceedings of the Sixth International Joint Conference on Natural Language Processing (IJCNLP2013), pp.543－551 (2013).
[19] Mulwad, V. Li, W. Joshi, A., Finin, T. and Viswanathan, K.: Extracting Information about Security Vulnerabilities from Web Text, In Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 03 (WI-IAT 2011), Vol. 3, pp. 257－260 (2011).
[20] Caruana, R., Lawrence, S. and Giles, L.: Overfitting in neural nets: Backpropagation conjugate gradient, and early stopping, Proceedings of the 13th International Conference on Neural Information Processing Systems (NIPS 2000), pp.381 － 387 (2000).
[21] OASIS: STIX Visualizer, available from <https://oasis-open.github.io/cti-stix-visualization/> (2021-02-07 accessed).
[22] Joshi, A., Lal, R., Finin, T. and Joshi, A.: Extracting Cybersecurity Related Linked Data from Text, 2013 IEEE Seventh International Conference on Semantic Computing, pp. 252－259 (2013).
[23] Jones, C.L., Bridges, R.A., Huffer, K.M.T. and Goodall, J.R.: Towards a Relation Extraction Framework for Cyber-Security Concepts, In Proceedings of the 10th Annual Cyber and Information Security Research Conference, pp.1－4 (2015).
[24] Ramnani, R.R., Shivaram, K., Sengupta, S. and Annervaz. K.M.: Semi-Automated Information Extraction from Unstructured Threat Advisories, In Proceedings of the 10th Innovations in Software Engineering Conference (ISEC 2017), pp.181－187 (2017).
[25] Liao, X., Yuan, K., Wang, X., Li, Z., Xing, L. and Beyah, R.: Acing the IOC Game: Toward Automatic Discovery and Analysis of Open-Source Cyber Threat Intelligence, The ACM Conference on Computer and Communications Security (CCS 2016), pp. 755–766 (2016).
[26] Zhu, Z. and Dumitraş, T.: FeatureSmith: Automatically Engineering Features for Malware Detection by Mining the Security Literature, The ACM Conference on Computer and Communications Security (CCS 2016), pp. 767–778 (2016).
[27] Feng, X, Liao, X., Wang, X., Wang, H., Li, Q., Yang, K., Zhu, H. and Sun, L.: Understanding and securing device vulnerabilities through automated bug report analysis, The 28th USENIX Security Symposium (Security 2019), pp. 887–903 (2019).
[28] Husari, G., Al-Shaer, E., Ahmed, M., Chu, B. and Niu, X.: TTPDrill: Automatic and Accurate Extraction of Threat Actions from Unstructured Text of CTI Sources, Annual Computer Security Applications Conference (ACSAC 2017), pp. 103–115 (2017).
[29] Zhu, Z. and Dumitras, T.: ChainSmith: Automatically Learning the Semantics of Malicious Campaigns by Mining Threat Intelligence Reports, The 3rd IEEE European Symposium on Security and Privacy (EuroS&P 2018), pp. 458–472 (2018).
[30] Milajerdi, S. M., Eshete, B., Gjomemo, R. and Venkatakrishnan, V.N.: POIROT: Aligning Attack Behavior with Kernel Audit Records for Cyber Threat Hunting, The ACM Conference on Computer and Communications Security (CCS 2019), pp. 1795–1812 (2019).