未知語を考慮した固有表現認識による セキュリティインテリジェンスの構造化手法

藤井翔太[†1]　鬼頭哲郎[†1]　重本倫宏[†1]　藤井康広[†1]

概要：年々サイバー攻撃が増加・高度化しており，インテリジェンス収集による最新の脅威情報への追従がますます重要となっている．一方で，その量が膨大かつ非構造であること，セキュリティ分野では新しい語（未知語）が生まれやすいことから，分析コストが大きい．そこで，固有表現認識を用いたインテリジェンスの構造化手法を提案する．本手法は，製品名やマルウェア名等，インテリジェンスとして着目すべき固有表現を抽出・構造化し，分析効率化を狙う．この際，各語の条件付確率を活用し，未知語の見落とし率を抑制する．また，プロトタイプを実装し，固有表現を F 値 0.78 の精度で抽出するとともに処理時間の面でも実用に耐えうることを確認した．

キーワード：インテリジェンス，自然言語処理，固有表現認識，深層学習，SOC/CSIRT

Cybersecurity Intelligence Structuring Method with Named Entity Recognition in Consideration of Unknown Word

SHOTA FUJII[†1]　TETSURO KITO[†1]
TOMOHIRO SHIGEMOTO[†1]　YASUHIRO FUJII[†1]

Abstract: Cybersecurity threats have been increasing and sophisticating year by year. In such circumstance, gathering security intelligence and following up-to-date threat information are important. Meanwhile, its analysis cost is high because intelligence is enormous, most them are unstructured, and unknown words tend to be generated in cybersecurity field. To suppress this cost, we propose cybersecurity intelligence structuring method with named entity recognition. The proposed method extracts and structures cybersecurity-related named entity, e.g., product name or malware name, so that improves intelligence analysis efficiency. Furthermore, the proposed method suppresses misrecognize rate of unknown words by focusing conditional probability of each words. The evaluation shows the f1-measure is 0.78 and the proposed method is practical from the view point of processing time.

Keywords: Intelligence, Natural Language Processing, Named Entity Recognition, Deep Learning, SOC/CSIRT

1. はじめに

年々サイバー攻撃が増加・高度化しており，セキュリティインテリジェンスを収集し，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．National Institute of Standards and Technology（NIST）発行の Computer Security Incident Handling Guide[1]においても，インシデントに備えてインテリジェンスを収集することの有用性が情報源の一覧とともに示されている．ここで，National Vulnerability Database（NVD）[2]や Japan Vulnerability Notes（JVN）[3]等では構造化された脅威情報が配信されており，効率的な分析や脆弱性管理をはじめとした各種自動化に有用である[4]．一方で，最新の脅威情報は，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[5]．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．ただし，その量が膨大かつ非構造であることから，分析コストが大きい．辞書やオントロジを作成することによって，非構造データを構造化する試み[6][7][8]もあるものの，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，新しい語（未知語）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ないことも，本課題を助長している．

これらの課題を緩和するべく，固有表現認識（Named Entity Recognition: NER）を用いたセキュリティインテリジェンスの構造化手法を提案する．本手法は，製品名やマルウェア名のように，インテリジェンスとして着目すべき固有表現を抽出・構造化することにより，分析の効率化を狙う．この際，各語の条件付確率に着目することで，未知語の見逃し率を抑制する．提案手法の貢献は，以下の通りである：

- NER により，セキュリティの脅威情報や脆弱性情報から固有表現を抽出・構造化し，インテリジェンスを自動で

†1 株式会社日立製作所
Hitachi Ltd.

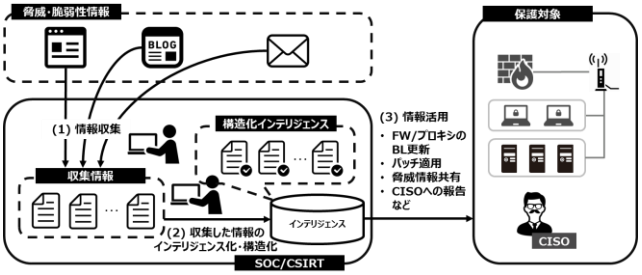


図 1 インテリジェンスに関する SOC/CSIRT 業務の全体像

構築する．この際，NER モデルの単語埋め込み層を教師なしで事前学習することにより，ドメイン固有の教師データを大量に用意することが困難である課題を緩和しつつ，各語の条件付確率を利用することにより，セキュリティ分野で特に頻出な未知語の見逃し率を抑制する．

- 作成したデータセットに対して評価を行い，F 値 0.78 の精度で固有表現を抽出し，セキュリティインテリジェンスを構築・構造化できることと従来手法では見落としていた未知語を抽出できることを示すとともに，処理時間の面でも実用に耐えうることを示した．

以降では，2 章で本研究における背景と課題を述べた後，同課題を解決するための提案手法について 3 章で設計，4 章で実現方式を述べる．5 章では，設計および実現方式に基づいてプロトタイプを実装し，評価を行う．6 章で関連研究について述べ，7 章で本稿のまとめを述べる．

2. 背景と課題

2.1 SOC/CSIRT 業務におけるインテリジェンス収集

1 章で述べたような増加・高度化する攻撃に追従し，適切に対処するために，セキュリティ監視業務を担う Security Operation Center（SOC）や Computer Security Incident Response Team（CSIRT）では，脆弱性情報，攻撃の早期警戒情報，および感染のインジケータ情報（Indicator of Compromise: IOC）等を収集する．SOC/CSIRT では，大きく以下のような手順でインテリジェンスを活用した業務が行われる（図 1）．

(1) 情報収集

サイバーセキュリティに関する情報を外部機関等から収集する．この情報には，例えば，悪性 URL，新規の脆弱性，および流行している攻撃の情報が含まれる．

(2) 収集した情報のインテリジェンス化・構造化

収集した情報を分析し，セキュリティインテリジェンスとしてまとめるとともに，再利用・参照可能な形で保存する．

(3) 情報活用

(2) で収集・構造化したインテリジェンスを各 SOC/CSIRT 業務に活用する．例えば，悪性 URL リストを利用したファイアウォール・プロキシのブラックリスト更新や脆弱性情報を利用したパッチ適用，および攻撃トレンドの共有等が挙げられる．これにより，自組織の資産に関係する脆弱性への対応や流行している脅威への事前対応が可能となる．

従来の SOC/CSIRT では，これら一連の作業は，その多くが人手によるものを占めており，業務効率化のボトルネックとなっていた．特に，(1)(2)で挙げた情報収集とそのインテリジェンス化・構造化は，NVD において 75,000 を超える脆弱性が報告されている上，毎月 60,000 を超えるセキュリティブログ記事が公開されている[9]現状においては，多大なコストとなっている．

2.2 インテリジェンス収集効率化の既存研究とその課題

前節に示したとおり，インテリジェンスの構築は，有用である一方で，そのコストが大きいという課題がある．この課題を緩和すべく，様々な研究がなされている．

例えば，非構造データを構造化することにより，分析を効率化する，あるいは各種自動化に活用するというモチベーションの下，辞書やオントロジを作成する研究がある[6][7][8]．一方で，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，新しい語（未知語）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，機械学習の手法を用いて NER タスクを解くことにより，固有表現を抽出し，インテリジェンスを構造化する研究もある[10][11][12][13][14][15]．固有表現とは，組織名や地名といった固有名詞や日時や金額といった数値表現の総称であり，テキスト中から固有表現を認識・抽出することを NER と呼ぶ．この固有表現として，インテリジェンスからマルウェア名や製品名といった分析に有用な情報を抽出し，構造化を図るのが先述の研究である．一方で，これらの多くは教師あり学習によるものであり，大量のラベル付き（アノテーション済み）コーパスを教師として用いることにより高精度を実現する．しかし，セキュリティ分野をはじめ，ドメイン固有の分野ではアノテーション済みコーパスが存在しておらず，教師データの確保が大きな課題として挙げられている[16]．この場合，自前でアノテーション済みコーパスを用意する必要があるが，このコストが大きいことがインテリジェンス構造化のボトルネックとなっている．具体的には，1 記事へのアノテーションに要する時間は，簡易なニュースでも中央値で約 8 分，場合によっては 1.73 時間を要するうえ，専門的な記事に対してはさらに時間を要するという研究報告[17]もあり，対象の教師データを用意することは容易ではない．

これらインテリジェンス構造化に関する既存研究の課題をまとめると，以下の 2 つとなる．

(課題1) インテリジェンス構築におけるコストが大きい

辞書やオントロジに関しては，初期作成コストに加え，継続的なメンテナンスコストが，機械学習に関しても

教師データの用意コストが大きい．

(課題2) 未知語を見落としてしまう可能性がある

辞書やオントロジに関しては，事前に定義していない未知語は認識することが出来ない．機械学習に関しては，教師データにない単語であっても，文脈情報等から未知語を認識することも可能である．しかし，未知語の認識精度は，教師データに出現する固有表現の認識精度に比べ大きく劣ることが示されている[18]．本課題は，未知語が頻出するセキュリティ分野では，その影響が顕著であると言える．

3. 設計

3.1 提案手法の要件

2.2 節で述べた各課題を解決するための要件として，以下の 2 つが挙げられる．

(要件1) NER の実施とその際のコスト抑制

インテリジェンスの構築・構造化を実現するために，既存研究においてその有用性が示されている NER を利用する．ただし，辞書・オントロジによるアプローチでは，それぞれの初期作成コストとメンテナンスコストが，教師ありな機械学習のアプローチでは，教師データの用意コストが大きいという課題がある（課題 1）．インテリジェンス分析コストの低減を狙う本研究において，その手段にあたる NER でもコストを抑制することが要件として挙げられる．

(要件2) 未知語の見落とし率抑制

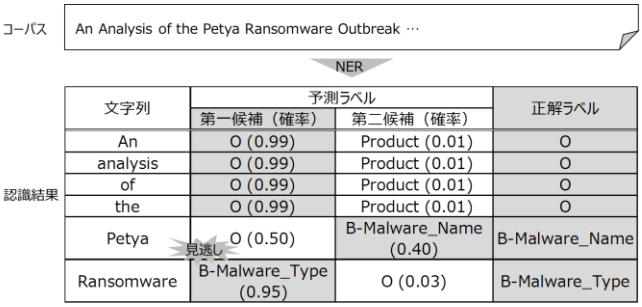
辞書・オントロジによるアプローチ，機械学習によるアプローチのいずれにおいても未知語の認識精度が比較的低いという課題がある（課題 2）．この課題は，特にセキュリティ分野での悪影響が大きく，未知語の見落とし率を抑制することが要件として挙げられる．

3.2 設計方針

（要件 1）を満足するにあたり，まず，先述の通り，NER を用いてインテリジェンス構築・構造化を実現し，SOC/CSIRT 業務におけるインテリジェンス構築を支援する．また，辞書ベースのアプローチでは，（要件 2）でも挙げている未知語への対応が難しいため，機械学習によるアプローチを採る．この際，機械学習によるアプローチにおける教師データの用意コストが大きいという課題は，NER モデルにおける単語埋め込み層を教師なしで事前学習することにより緩和する．単語埋め込み層では，単語を実数ベクトル（分散表現）として獲得する．分散表現により，近い意味の単語を近いベクトルで表現できる．また，分散表現をモデルの入力として用いることにより，自然言語処理タスクの精度が向上することが報告されている[19]．分散表現は，word2vec として知られる skip-gram や CBOW をはじめとして，教師なし学習で獲得することができ，これにより，全ての教師データにアノテーションが必要な教師ありモデルよりもアノテーションコストを抑制できる．具体的に利用するモデルについては，4 章で詳述する．また，インテリジェンスを構築・構造化するにあたり，NER でどういった固有表現を抽出すべきかを導出する必要がある．これについては 3.3 節で後述する．

（要件 2）の未知語の見落とし率抑制は，NER において，未知語が見落とされる際の条件付確率に着目し，未知語の可能性がある語を拾得する手法によって満足する．本手法は，未知語をはじめとして，本来であれば何らかの固有表現としてラベル付けされるべきであったにも関わらず，NER では見落とされたものを拾得するものである．見落とされた固有表現を抽出するために，各語の条件付確率を利用する．NER による固有表現の見落とし例を図 2 に示す．NER では，各語がどういった確率でどのラベルの固有表現かを算出する．この際，通常であれば最も確率の高い候補が同語のラベルとして採用される．つまり，見逃された語には，第一候補が「どの固有表現でもない」というラベルが付与されている．このとき，真に固有表現でないものは，「どの固有表現でもない」というラベルの条件付確率が比較的高く，固有表現たりうる語は，同ラベルの条件付確率が比較的低くなる．また，第二候補として真のラベルが挙がることが多い．これは，多くの場合コーパスは固有表現以外の語の割合が多く，過去のデータからその語の情報が直接得られない未知語や出現頻度の少ない語に対しては，固有表現で無いというラベルを付与することで正解率が高くなりやすいこと，および語の情報を除くとその文脈的な特徴から，正解のラベルが上位候補として現れやすいことに起因する．これらの現象を利用し，第一候補が「固有表現でない（O）」というラベルかつ第二候補の条件付確率が閾値よりも高いものを検出し，第二候補を固有表現の可能性があるとして，拾得する．

本手法の具体的な挙動を図 2 で示した従来の NER であれば見逃していた「Petya」を用いて説明する．図中では，第一候補が「O（固有表現ではない）」，第二候補が「Malware_Name（正解ラベル）」と続いている．ここで，第一候補が「固有表現ではない」こと，および他の真に固有表現ではない語の第二候補に比べ，その条件付確率が比較的高いことから，「Petya」は，固有表現の候補として拾得される．本手法により，固有表現の見落とし率を抑制する．



文字列	予測ラベル		正解ラベル
	第一候補（確率）	第二候補（確率）	
An	O (0.99)	Product (0.01)	O
analysis	O (0.99)	Product (0.01)	O
of	O (0.99)	Product (0.01)	O
the	O (0.99)	Product (0.01)	O
Petya	O (0.50)	B-Malware_Name (0.40)	B-Malware_Name
Ransomware	B-Malware_Type (0.95)	O (0.03)	B-Malware_Type

図 2 NER による未知語の見逃し例

表 1 抽出する固有表現の一覧

固有表現	意味	正規表現抽出
Alert_ID	アラートの識別子	○
Attack_Method	攻撃手法	×
Attacker	攻撃者・攻撃グループ名	×
Campaign_Name	攻撃のキャンペーン名	×
CVE_ID	CVE 番号	○
File_Hash	ハッシュ値	○
File_Name	ファイル名	○
Ip_v4/Ip_v6	ipv4/ipv6 アドレス	○
Mail_Address	メールアドレス	○
Malware_Name	マルウェア名	×
Malware_Type	マルウェアの種類	×
Mitigation_Word	Mitigation のための語	×
MS_ID	MS 更新プログラムの ID	○
Organization_Name	組織の名称	×
Port	ポート番号	○
Product	製品，システム	×
Reference	URL，FQDN	○
Region	国・地域名	×
Segment	業界区分	×
Site_Type	悪性 Web サイトの性質	×
Time	時間・時期	○
Version	バージョン名	○
Vulnerability	脆弱性	×

3.3 抽出する固有表現

既存研究，STIX[20]，および実務者へのヒアリングを基に，インテリジェンスを構築すべき情報は何かという観点から， 表 1 に示す固有表現を導出した．ここで，表 1 で挙げた固有表現には，CVE 番号やハッシュ値のように，フォーマットが定まっており，正規表現で抽出できるものとそうでないものに大別できる．提案手法では，正規表現で抽出できるものは正規表現で抽出し，そうでないものを NER で抽出する．

3.4 提案手法の全体像

これまで述べてきた設計方針に基づいた提案手法の全体像を図 3 に示す．

(1) 情報収集

外部機関が公開している脅威情報や脆弱性情報を収集する．

(2) 前処理

収集した情報に対し，ノイズの除去や NER で利用できる形への変換といった前処理を実施する．

(3) 固有表現認識

収集した情報から 3.3 節で定義した固有表現を抽出する．この際，前述したとおり，正規表現で抽出できるものは正規表現で,そうでないものは NER で抽出する．

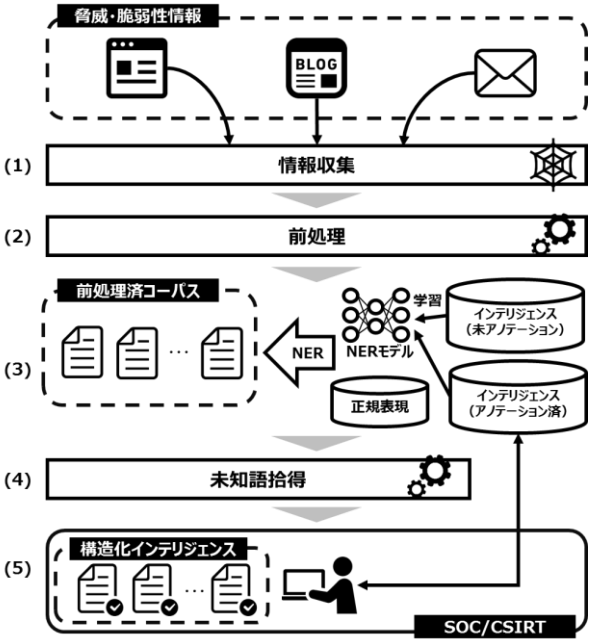


図 3 提案手法の全体像

また，NER モデルを構築するための教師データのうち，アノテートされた教師データは，構造化された（アノテートされた）インテリジェンスと同等である．そこで，人手でアノテートした教師データに加え，提案手法によって構造化されたインテリジェンスも同様に教師データとして利用する．

(4) 未知語抽得

(3)において，固有表現として認識されなかったものから，未知語の可能性があるものを抽出する．

(5) 分析者への提示

ここまでの処理フローで構築されたインテリジェンスを SOC/CSIRT の分析者に提示し，分析に活用する．

4 章において，上記の各ステップにおけるコンポーネントの実現手法を述べる．

4. 実現方式

4.1 情報収集

提案手法で扱う情報としては，US-CERT[21]・ICS-CERT[22]をはじめとする公的組織や各種セキュリティブログ[23]によって Web サイトで公開されているものを主に想定している．ここで，日々の SOC/CSIRT 業務における情報収集では，前回収集時から更新された新規の情報のみを取得する．今回は，RSS で配信されている各 Web ページの更新情報を利用して新規の情報を取得した．

4.2 前処理

提案手法で扱う Web ページでの公開情報には，本文以外のヘッダ・フッダ等に加え，本文の中にも html タグ等のノイズが含まれる．また，ストップワード等，後段の NER においてノイズとなる情報が含まれている．そこで，それらのノイズを除去するために，下記の前処理を実施する．

(1) 本文抽出

Web ページの内容をテキスト化し，ヘッダとフッダを取り除くことにより，インテリジェンスとなり得る本文部分を抽出するとともに，NER でノイズとなる html タグ等を除去する．

(2) 文/単語分割

今回の手法は，文ごとの系列ラベリングにより固有表現を抽出する．また，単語単位でラベル付けを行う．そこで，(1)で抽出した本文を文ごとに分割した上で，それぞれ単語ごとに分割する．

(3) 正規化

表記ゆれや不要な情報による NER への悪影響を抑制するため，正規化を行う．今回は，全ての文字を小文字に統一した上で，語の語幹を取り出すステミングと見出し語へ変換するレンマ化を行った．また，URL は，ある程度自由に設定できるものの，その表記の多様性自体はラベル付与に寄与しない．そこで，全 URL に対し，URL を意味するタグへの置換を行った．

(4) ストップワード除去

ストップワードとは，冠詞や前置詞のように，多くの文に高頻度で出現するために，ラベル付与に寄与しない語のことである．今回は，Natural Language Toolkit[24]で定義されている英語のストップワード 179 個を除去した．

4.3 固有表現認識

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，NER を実施する．正規表現で抽出可能な CVE 番号や各種ハッシュ値等を正規表現で抽出した後，それ以外の固有表現を 3.2 節でも述べたとおり，NER モデルによって抽出する．具体的には，文献[25]で提案されている文字の Convolutional Neural Network（CNN）による表現と単語の埋め込み表現を結合したベクトルを入力とし，Bi-directional Long Short-Term Memory（Bi-LSTM）層，結合層，および Conditional Random Field（CRF）層によって NER を行うモデルを用いる（図 4）．本モデルは，NER タスクにおいて state-of-the-art を達成しており，同等のモデルは多くの研究で用いられている[16][26]ことから，提案手法においても採用した．この際，単語埋め込み表現を教師なしで学習することにより，アノテーションコストを抑制しつつ，固有表現の認識精度向上を図る．

4.4 未知語拾得

未知語を拾得するために，3.2 節で述べたとおり，固有表現認識結果における条件付確率を参照し，第一候補のラベルが O かつ第二候補の条件付確率が閾値以上のものを拾得する．今回の提案手法においては，4.3 節で述べた CRF 層の結果を受け取り，未知語か否かを判定する層を NER モデルの終端に追加する形で実現した（図 4）．

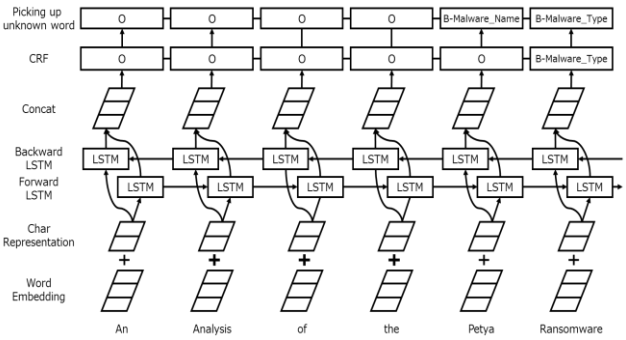
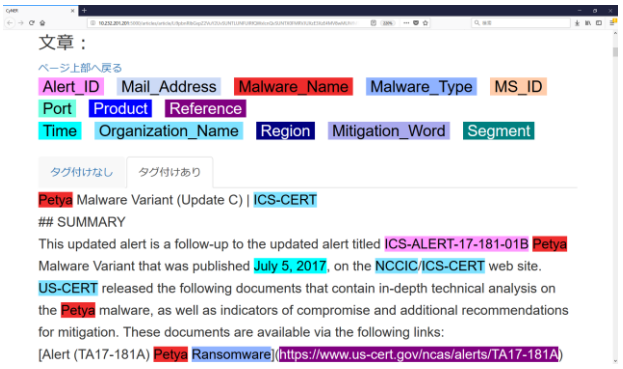


図 4 NER モデルの全体像



(A) タグ付けされた文章

(B) 構造化されたインテリジェンス

図 5 提案手法の出力

4.5 分析者への提示

分析者は最終的にタグ付けされた文章や構造化された情報を確認する．本手法によって，ICS-CERT による Petya の Alert[27]に(A)タグ付けと(B)構造化を実施した画面の一部を図 5 に示す．(A)では，そのタグから ICS-CERT（Organization_Name）によって発行されたランサムウェア（Malware_Type）の Petya（Malware_Name）に関するレポートであることが推察できる．このように，タグ付けされ

た文章により分析を効率化することが期待できる．また，構造化されたインテリジェンスを機械処理等に活用することが期待される．

5. 評価

5.1 評価項目と評価環境

今回の評価項目は，以下の 3 つである．

(評価1) 固有表現の認識精度

提案手法は，文献[25]の手法をベースラインとして NER を実施し，それに加えて各語の条件付確率を用いて未知語を拾得と認識精度の向上を図るものである．そこで，提案手法がどの程度固有表現を認識できるかの精度を Precision，Recall，および F 値によって検証する．この際，ベースライン手法と提案手法を比較する．

(評価2) 見落としの抑制度

提案手法は，第二候補の条件付確率を用いることにより，従来の NER では見落としていた固有表現の抽出を図る．そこで，本手法によって，従来の NER であれば見落としていた固有表現をどの程度抽出できるか検証する．

(評価3) 処理時間

本手法は，日々の分析業務での活用を想定しているため，処理時間がその範囲内に収まっている必要がある．ここで，本手法の処理としては，情報更新後のモデルの再学習とそのモデルを利用した記事に対する NER がある．そこで，本手法におけるモデルの再学習時間と 1 記事に対する NER の時間を測定し，実用に耐えうるか検証した．

なお，評価用のデータセットとしては，インテリジェンスのひとつである ICS-CERT の Alert に対して，報告者らが IOB2 によりアノテーションしたコーパス（123 記事，4,779 文）を利用した．また，このうち 2011 年 7 月 13 日から 2013 年 10 月 29 日までの 83 記事（全体の約 70%）を訓練用，以降の 40 記事（約 30%）を評価用のデータとして利用した．また，未アノテーションな教師データとしては US-CERT の Alert179 記事と ICS-CERT の Adversary950 記事の合計 1,129 記事を使用した．ハイパパラメータは，ランダムサーチで探索し，文字表現の次元数を 25，文字埋め込み表現の次元数を 100，LSTM の次元数を 100 に設定した．この際，単語埋め込みの学習手法には，CBOW を用いた．また，第二候補から固有表現候補を抽出するか否かの閾値として 0.20 を用いた．なお，評価環境は，表 2 に示す通りである．

表 2 評価環境

CPU	Intel Core i7-7700K 4.20GHz
GPU	GeForce GTX 1080
メモリ	64GB
OS	Ubuntu 16.04.3 LTS

表 3 固有表現の認識精度

	precision	recall	F1-measure
ベースライン	0.75	0.74	0.73
提案手法	**0.77**	**0.80**	**0.78**

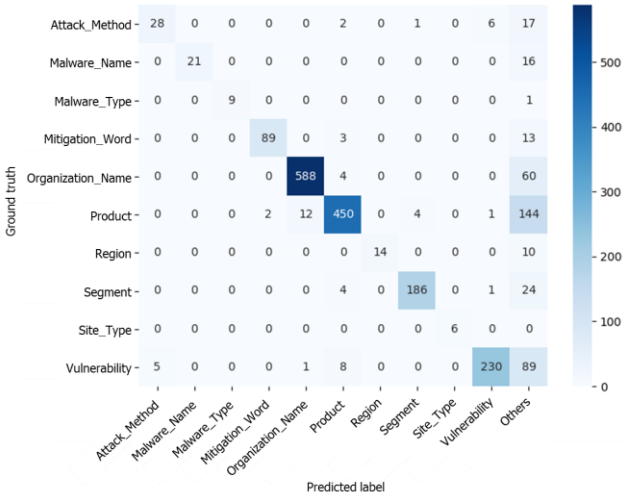


図 6 固有表現ごとの混同行列

表 4 固有表現ごとの認識精度

	precision	recall	F1-measure
Attack_Method	0.30	0.26	0.28
Malware_Name	1.00	0.57	0.72
Malware_Type	1.00	0.88	0.93
Mitigation_Word	0.71	0.85	0.78
Organization_Name	0.92	0.92	0.92
Product	0.67	0.66	0.66
Region	0.67	0.53	0.59
Segment	0.70	0.83	0.76
Site_Type	1.00	1.00	1.00
Vulnerability	0.63	0.62	0.62

5.2 評価結果

5.2.1 （評価 1）固有表現の認識精度

前述のデータセットに対するベースライン手法，提案手法それぞれの固有表現の認識精度を表 3 に示す．いずれの指標においても，提案手法がベースラインの手法を上回っていることから，その差分である未知語拾得の部分が認識精度向上に貢献したといえる．

また，今回のデータセット中には現れなかった「Attacker」と「Campaign_Name」を除く 10 の固有表現について，提案手法による固有表現ごとの認識精度を表 4 に，混同行列を図 6 に示す．表 4 と図 6 から，「Attack_Method」，「Malware_Name」，および「Region」が他の固有表現に比べて見逃しが多く，低精度となっていることが見て取れる．これは，アノテーション済み教師データにおけるこれらの固有表現の出現数が他に比べて少ないことが原因の一つであると推察される．本事象に関しては，今後より大規模なデータセットを用いて評価し，改善が見られるか検証する．

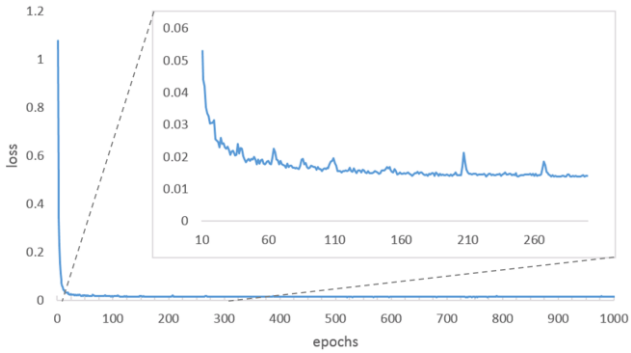


図 7 学習曲線

5.2.2 （評価 2）見落としの抑制度

5.1 節で述べたとおり，第二候補から固有表現候補を抽出するか否かの閾値として 0.20 を用いた．つまり，第一候補のラベルが「いずれの固有表現でもない」かつ第二候補の条件付確率が 0.20 以上の場合に，見落された固有表現として第二候補を抽出する．評価したところ，上述の条件に当てはまる 108 の単語が新たに固有表現として抽出され，そのうち 95 個が真に固有表現である単語だった．なお，データセット中に，第二候補が正解の固有表現は 203 個含まれていた．すなわち，偽陽性を約 12%（13/108）に抑えつつ，従来手法では見落とされていた固有表現のうち，第二候補が正解であったものを約 47%（98/203）抽出できた．この結果が表 3 に示した性能向上に寄与している．以上のように，提案手法によって，固有表現の見落としを抑制可能なことが確認できた．

5.2.3 （評価 3）処理時間

1,000 エポック実施した際の学習曲線を図 7 に示す．この際，1 エポックに 54 秒，合計で約 15 時間を要した．一方で，図 7 の右上部分に示した 10～300 エポック時の拡大図から，120 エポック付近で学習がほぼ収束していることが分かる．このため，Early Stopping[28]を利用し，この時点で学習を打ち切ることにより，同等の精度を確保しつつ，1.8 時間程度でモデル再学習を実施できる．

また，評価用の 40 記事に対する NER には，約 6.37 秒を要した．このことから，1 記事平均は，約 0.16 秒である．仮に，1 日に 2,000 記事のセキュリティブログ（毎月 60,000 件を超えるセキュリティブログが発信されている[9]状況から，1 月を 30 日として試算）を処理するとしても，約 318.5 秒（約 5.3 分）で処理可能である．

以上のように，モデル再学習に約 1.8 時間，NER に 1 記事平均約 0.16 秒，想定される 1 日の記事量 2,000 に対しても約 5.3 分と，実用の範囲内であると言える結果となった．

5.3 考察

(1) 正規表現による固有表現の抽出

3.3 節で述べた通り，フォーマットが定まっている固有表現は，正規表現により抽出する．正規表現による抽出は，ほぼ 100%の精度だったものの，幾つかの誤認識が見られた．例えば，1.1.1.1 のようなソフトウェアのバージョンが IPv4 アドレスとして抽出される現象が見られた．誤認識数は多くなかったため，個別に除外ルールを作成する，人手で修正する等の手法により，対応可能であると推察される．また，URL を抽出した際，IOC として記載されている悪性 URL なのかベンダ等が公開している Web ページへのリンクであるかを正規表現では判定できない．これについては，ホワイトリストや悪性 URL 判定手法の導入が必要である．

(2) 見落とした固有表現を抽出する際の閾値

今回は，閾値に 0.20 を用いて評価を行い，NER の性能向上に寄与することを示した．本閾値は，値を増加することにより，固有表現として拾得するか否かの判定が厳しくなるため，recall が低下する代わりに，precision が向上する．実際に本手法を運用する際には，どちらの値を重視するかによって閾値を調整するのが望ましいと推察される．

(3) 処理時間

提案手法におけるモデル再学習や NER の処理時間は，全体の記事量や記事ごとの内容量に依存する．今回のデータセットに対しては，実用の範囲内であることを示したが，運用の中でデータ数は単純増加していくと思われるため，その中でも実用に耐えうるか，より大規模なデータセットを用いて今後検証していく．

6. 関連研究

2.2 節で述べたとおり，提案手法と同様に，インテリジェンスを構築・構造化し，分析効率化や機械処理への活用を図る研究は，辞書・オントロジによるもの[6][7][8]と機械学習手法によるもの[10][11][12][13][14][15]の 2 つに大別できる．提案手法は，これらの手法におけるインテリジェンス構築におけるコストが大きい（課題 1），未知語を見落としてしまう可能性がある（課題 2）という両課題を緩和しつつ，脅威情報や脆弱性情報から固有表現を抽出することにより，インテリジェンス分析業務の効率化を図る．

より速報性の高い情報を収集することを目的に，SNS からベンダのパッチリリース情報[29]や脅威情報・脆弱性情報の抽出を試みている研究もある[30][31][32]．これらの研究は，速報性の高い情報を抽出可能という点で有用である一方で，情報抽出に留まっているものが多く，構造化に取り組んでいるものも 2.2 節で挙げた課題を持つ．そこで，これらの研究で抽出した速報性の高い情報に対して，提案手法を適用することで，上記課題の緩和と分析効率化が期待できる．

7. おわりに

本稿では，未知語を考慮した NER によるインテリジェンス構造化手法について述べた．提案手法は，ベースラインとして，文字の CNN による表現と単語の埋め込み表現

を結合したベクトルを入力とし，Bi-LSTM 層，結合層，および CRF 層によって NER を行うモデルを用いることにより，インテリジェンスとして着目すべき固有表現を抽出する．この際，単語埋め込み層を教師なしで学習することにより，アノテーションコストを抑制しつつ，認識精度の向上を図る．また，CRF 層で算出した各語の条件付確率に着目することにより，セキュリティ分野では特に頻出な未知語に対する見逃し率を抑制し，認識精度の向上を図る．これらの手法により，インテリジェンスの構築・構造化を支援し，分析の効率化を狙う．

評価では，提案手法のプロトタイプを実装し，ベースラインの手法よりも高精度で固有表現を認識できることと従来手法では見逃していた未知語を認識できていることを確認した．さらに，モデルの再学習時間や 1 記事に対する NER に要する時間を計測し，実運用にも耐えうることを示した．

今後の課題としては，より大規模なデータセットによる評価がある．また，提案手法で抽出した固有表現を関係抽出タスクや情報検索タスクに応用し，より高精度なインテリジェンス構築に取り組むことがある．

参考文献

[1] National Institute of Standards and Technology: SP 800-61 Rev. 2 Computer Security Incident Handling Guide, available from < https://csrc.nist.gov/publications/detail/sp/800-61/rev-2/final> (2018-07-25 accessed).

[2] National Institute of Standards and Technology: National Vulnerability Database, available from <https://nvd.nist.gov/> (2018-07-25 accessed).

[3] Japan Vulnerability Notes，入手先<https://jvndb.jvn.jp/>（2018-07-25 アクセス）．

[4] R., A., Bridges, C., L., Jones, M., D., Iannacone, K., M., Testa, and J., R., Goodall: Automatic labeling for entity extraction in cyber security, ASE Third International Conference on Cyber Security, Academy of Science and Engineering (ASE), (2014).

[5] N., McNeil, R., A., Bridges, M., D., Iannacone, B., Czejdo, N., Perez, and J., R., Goodall: 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).

[6] L., Obrst, P., Chase, and R., Markeloff: Developing an Ontology of the Cyber Security Domain, CEUR Workshop Proceedings, Vol. 96, pp.49－56 (2012).

[7] M., Iannacone, S., Bohn, G., Nakamura, J., Gerth, K., Huffer, R., Bridges, E., Ferragut, and J., Goodall: Developing an Ontology for Cyber Security Knowledge Graphs, In Proceedings of the 10th Annual Cyber and Information Security Research Conference (CISR '15), pp. 1－4 (2015).

[8] L., S., Kiat, M., A., Obaja, L., Wei, and O., C., Hui: MalwareTextDB: A Database for Annotated Malware Articles, Proceedings of the 55th Annual Meeting of the Association for Computational, Vol. 1, pp.1557－1567 (2017).

[9] IBM: IBM Watson to Tackle Cybercrime, available from <https://www-03.ibm.com/press/us/en/pressrelease/49683.wss> (2018-07-25 accessed).

[10] V., Mulwad, W., Li, A., Joshi, T., Finin, and K., Viswanathan: 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 '11), Vol. 3, pp. 257－260 (2011).

[11] A., Joshi, R., Lal, T., Finin, and A., Joshi: Extracting Cybersecurity Related Linked Data from Text, 2013 IEEE Seventh International Conference on Semantic Computing, pp. 252－259 (2013).

[12] N., McNeil, R., A., Bridges, M., D., Iannacone, B., Czejdo, N., Perez, and J., R., Goodall: 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).

[13] C., L., Jones, R., A., Bridges, K., M., T. Huffer, and J., R., Goodall: 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).

[14] R., R., Ramnani, K., Shivaram, S., Sengupta, and Annervaz K. M.: Semi-Automated Information Extraction from Unstructured Threat Advisories, In Proceedings of the 10th Innovations in Software Engineering Conference (ISEC '17), pp.181－187 (2017).

[15] 山嵜麿与，神谷造：文書からの脅威と脆弱性知識の自動抽出，コンピュータセキュリティシンポジウム 2017（CSS2017）論文集，pp.393－398（2017）．

[16] Tang, S., Zhang, N., Zhang, J., Wu, F. and Zhuang, Y.: NITE: A Neural Inductive Teaching Framework for Domain Specific NER, Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2642－2647 (2017).

[17] Burr, S. and Mark, C.: Active Learning with Real Annotation Costs, NIPS Workshop on Cost-Sensitive Learning (2008).

[18] 齋藤邦子，鈴木潤，今村賢治：CRF を用いたブログからの固有表現抽出，言語処理学会第 13 回年次大会論文集(2007).

[19] P., Jeffrey, R., Socher, and C., D., Manning: Glove: Global Vectors for Word Representation, EMNLP 2014, 1532--1543 (2014).

[20] STIX Project Documentation: About STIX, available from <https://stixproject.github.io/> (2018-07-25 accessed).

[21] US-CERT, available from <https://www.us-cert.gov//> (2018-07-25 accessed).

[22] ICS-CERT, available from <https://ics-cert.us-cert.gov/> (2018-07-25 accessed).

[23] Feedspot: Top 50 Cyber Security Blogs and Websites in 2018 For IT Security Pros, available from <https://blog.feedspot.com/cyber_security_blogs/> (2018-07-25 accessed).

[24] Natural Language Toolkit, available from <https://www.nltk.org/> (2018-07-25 accessed).

[25] Ma, X. and Hovy, E.: End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016), pp. 1064－1074 (2016).

[26] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K. and Dyer, C.: Neural Architectures for Named Entity Recognition, Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 260－270 (2016).

[27] ICS-CERT: Alert (ICS-ALERT-17-181-01C) Petya Malware Variant (Update C), available from <https://ics-cert.us-cert.gov/alerts/ICS-ALERT-17-181-01C> (2018-07-25 accessed).

[28] 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)

[29] R., Syed: Analyzing Software Vendors' Patch Release Behavior in the Age of Social Media, Proceedings of the International Conference on Information Systems Transforming Society with Digital Innovation 2017 (ICIS '17) (2017).

[30] S., Mittal, P., K., Das, V., Mulwad, A., Joshi, and T., Finin: CyberTwitter Using Twitter to generate alerts for Cybersecurity Threats and Vulnerabilities, International Symposium on Foundations of Open Source Intelligence and Security Informatics", IEEE Computer Society (2016).

[31] A., Ritter, E., Wright, W., Casey, and T., Mitchell: Weakly Supervised Extraction of Computer Security Events from Twitter, In Proceedings of the 24th International Conference on World Wide Web (WWW '15), pp. 896－905 (2015).

[32] A., Javed, P., Burnap, and O., Rana: Prediction of drive-by download attacks on Twitter, Information Processing & Management (2018).