アタックツリー自動生成ツールにおける入力データ作成ミス検知技術の提案

島邉 遼佑† 浅井 健志† 河内 清人†

概要：アタックツリー自動生成ツールの登場により脅威分析作業の自動化が進み，セキュリティ専門家に依存しない，アタックツリーを活用したセキュリティ対策が可能になった．しかし，企画／設計等の早期の開発工程へツールを導入する場合には，分析対象システムに関する情報を定義した入力データをユーザが手動で作成する必要性がある．この時，入力ミスが発生する恐れがある．もし，入力ミスによって生成された誤ったアタックツリーを元にセキュリティ対策を行った場合には，深刻な脆弱性の放置や無駄な対策の実施などの不具合が発生する．従って，アタックツリー自動生成ツールにおける入力ミス検知は重要な課題と言える．我々はこの課題に対し“word2vec”と呼ばれる分散表現技術を応用することで，分析対象のシステム仕様書と，ユーザの作成した入力データ及び，生成されたアタックツリー間における入力単語の意味変化を評価することで入力ミスを検知する方式を提案する．提案方式は，分析対象システム固有の情報に関する入力ミスを検知することが可能である．

Proposal of Input Data Error Detection Method for Attack Tree Automatic Generation Tool

RYOSUKE SHIMABE† TAKESHI ASAI† KIYOTO KAWAUCHI†

1. はじめに

脅威分析とは，システムに内在するセキュリティ的な脅威を洗い出し，脅威が発生する原因を分析する作業である．アタックツリーとは，脅威分析時に利用されるツリー状のグラフであり，発生する可能性のある脅威とその原因である一連のサイバー攻撃活動の内容を表現したものである．（図 1）アタックツリーには，悪用される脆弱性や侵入経路，攻撃手口などの情報が明示されるため，セキュリティ対策の検討に役立つ．

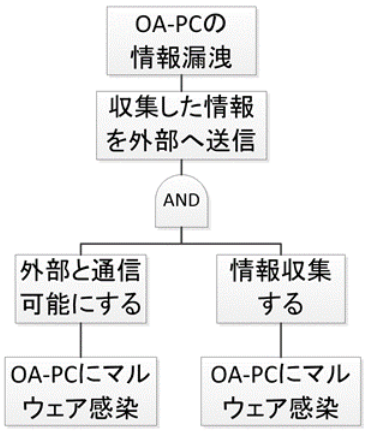

図 1 脅威“OA-PC の情報漏洩”に対するアタックツリー

近年，MulVAL [1]，NetSPA [2]といったアタックツリー自動生成ツールが提案され，分析対象システムの情報を定義した入力データを作成し，ツールへ入力するだけで，アタックツリーを自動生成出来るようになった．これまで，セキュリティ専門家に依存していたアタックツリー作成作業がツールによって自動化されたことにより，セキュリティの専門家ではない開発者であっても，アタックツリー用いた脅威分析と対策の検討が行える環境が整いつつある．

だが，実際のシステム開発にアタックツリー自動生成ツールを導入する際には「アタックツリー自動生成ツールの入力データ作成時に，ユーザによる入力ミスが発生し，誤ったアタックツリーが生成される」という問題の発生が懸念される．この問題は，アタックツリー自動生成ツールが分析を実行するために必要な情報が，装置の設定・OS／APP・ネットワーク構成・情報資産・認証情報・アクセスポリシー等と多岐に渡るためである [3]．（図 2）

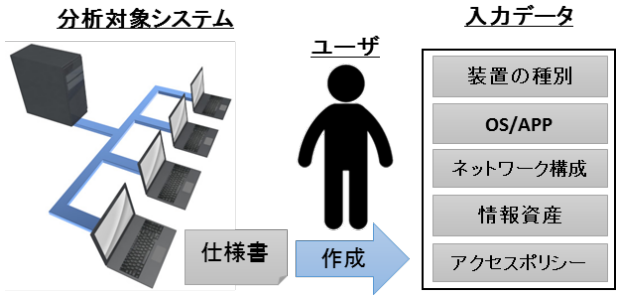

図 2 アタックツリー自動生成ツール入力データイメージ

これに対し，MulVAL ，NetSPA では，分析対象システムをスキャンし，入力データを自動作成する機能を搭載することで入力ミスの発生を防止している．

しかし，企画・設計段階といった早期の開発工程にアタックツリー自動生成ツールを導入する際には，スキャンする対象であるシステムが存在しないため，ユーザによる入力データ作成作業が必要となり，入力ミスを完全に回避す

† 三菱電機株式会社 情報技術総合研究所，神奈川県鎌倉市大船 5-1-1，Mitsubishi Electric Corporation，Information Technology R&D Center，5-1-1 Ofuna Kamakura，Kanagawa，Japan

ることはできない．

もし，入力ミスによって生成された誤ったアタックツリーを活用して，不適切なセキュリティ対策を行った場合には，攻撃される可能性の高い深刻な脆弱性の放置や，無駄な対策の技術の導入，開発の手戻りの発生など，システム開発における様々な不具合を引き起こす可能性がある．

従って，アタックツリー自動生成ツールの入力データ作成作業に伴う入力ミスを検知する技術を開発することは，アタックツリー自動生成ツールを実際のシステム開発に導入する際の重要なテーマである．

しかし，現在のアタックツリー自動生成ツールの研究では，アタックツリーの生成方式や生成速度の高速化，分析性能の向上が中心的なテーマであり，入力ミスの検知技術については十分な議論がなされていない．

よって，本論文では，ニューラルネットワーク・自然言語処理技術を応用したアタックツリー自動生成ツールにおける入力ミス検知方式を提案と考察を行った．今回，提案する検知方式は，“分析対象のシステム仕様書”と“ユーザの作成した入力データ／生成されたアタックツリー”という 2 つのデータ間における単語の意味変化に着目し，入力ミスを検知する技術である．

提案方式のコアとなる技術は，2 つの異なるドメインデータ間における同一名称単語の意味変化を定量的に評価する関数を構築する部分にある．この単語の意味変化評価の主要なアイデアは，Mikolov らが提案した分散表現技術 word2vec [1]から獲得した単語ベクトルと，そのベクトルが持つ「異なるドメインデータから獲得した 2 つの単語ベクトル空間は，線形変換によって対応付けることが出来る」という性質 [4]を応用し，同一名称単語の単語ベクトルを対応付ける最適な線形変換行列を求めた際に生じるフィッティング誤差を意味変化度合いとして評価することにある．よって，仕様書を確認しなければ判別できないような，分析対象システム固有の情報に関する入力ミスを検知することが可能である．

また，本提案方式において構築される単語の意味変化評価関数は，一般的な異常検知技術におけるラベルなし異常データに対する異常度と等しい．

以降，2 章では既存の入力ミス検知技術について検討し，3 章では提案方式に応用する word2vec の導入を行った後に，方式の提案を行う．4 章では提案方式について理論的な考察を行い，5 章にて結論と今後の課題についてまとめる．

2. 関連研究

まず，既存の入力ミス検知技術をアタックツリー自動生成ツールにおける入力ミスに応用することを考え，既存技術について検討を行なう．

既存の入力ミス検知技術の中で最も古典的な方式は，IF-THEN 形式のルールベース方式である．具体例としては，パスワード入力画面における Caps Lock 検出機能や，登録画面における必須入力欄の空白検知機能などが，ルールベースの検知方式に該当する．このルールベース検知方式は，入力データのとり得る状態の範囲が限定されており，入力ミスのパターンが定義しやすい場合において，的確にミスを検知できるという利点がある．しかし，ルールベース方式を採用して，多様な入力ミスパターンに対応するためには，パターンの数に伴う膨大な検知ルールの作成が必要になるため知識獲得のボトルネックが生じる．

このようなルールベース方式の課題を踏まえ，近年では，機械学習・ニューラルネットワーク技術といった統計的な技術を用いて入力ミスを検知する方式も考案され，文法誤り訂正技術（誤字脱字の検知）などに応用されている [5] [6]．統計的な手法を用いて入力ミスを検知する方式は，人手で検知ルールを作成する必要がなく，ルールベースでは対処しきれない，広い範囲の入力ミスを検知することが可能である．

そこで，アタックツリー自動生成ツールにおける入力ミス検知にも，機械学習やニューラルネットワークを応用する統計的な手法に基づく検知方式のアプローチを取る．

機械学習・ニューラルネットワーク技術を応用する入力ミス検知技術の一般的なアプローチとしては，大量の入力データから，事前に学習した正常な入力データのモデル（確率分布）をベースに，新しく入力されたデータの正しさ（入力ミスが存在する不自然な入力ではないか？）を評価することで検知するものがある．従って，アタックツリー自動生成ツールにおいても，大量の入力データ(分析対象システムの情報)から，正常なシステム構成モデルを学習することで，新しく与えられた入力データのシステム構成情報が不自然でないかを評価することで，入力データに存在する入力ミスである単語を検知できる可能性がある．

しかし，アタックツリー自動生成ツールの場合，このような入力データの正常モデルを学習するアプローチでは，分析対象システム固有の情報に関する入力ミスの検知が難しいという課題が残る．

具体的な例としては，固有名詞にあたる単語の単語カテゴリを間違える（表 1），同じカテゴリの入力単語同士を入れ替えて入力する（表 2）といった例である．

表 1 ネットワーク構成情報入力欄イメージ

ネットワーク名称	アドレス種別	回線	…
業務用ネットワーク_1	IPv4	有線	…
業務用ネットワーク_2	IPv4	無線	…
業務用 PC_1	IPv4	有線	…
…	…	…	…

ネットワーク名称を入力する箇所に，“業務用 PC_1”と

いう装置名称が入力されているミス．

表 2 装置情報入力欄イメージ

装置名称	利用者名	装置種別	…
業務用サーバ_1	従業員_1	OA-PC	…
業務用 PC_2	従業員_2	OA-PC	…
業務用 PC_1	サーバ管理人_1	サーバ	…
業務用サーバ_2	サーバ管理人_2	サーバ	…
…	…	…	…

システム仕様書では「“業務用 PC_1”は従業員_1 が利用し，“業務用サーバ_1”はサーバ管理人_1 が利用している」と記述されていた場合，成分(1,2)と成分(1,4)の入力単語は逆である．

上記のような 2 つの入力ミスパターンは，装置名称やネットワーク名称など，開発しているシステム固有の情報を表現する入力単語に関する入力ミスパターンであるため，この場合，他のシステムの分析用の入力データを学習するだけでは入力ミスなのか判定できない．

3. 提案方式

今回，アタックツリー自動生成ツールにおける入力ミスを検知するために，我々は，入力ミスか否かは，ユーザがツールの入力データを作成する際の情報源である“分析対象システムの仕様書”を確認することで判別ができると考えた．つまり，入力単語の意味がシステム仕様書で定義されている意味と異なっていた場合には，それは入力ミスであると判別できる．

よって，本提案方式は，1. 単語カテゴリにおける入力ミス，2. 単語同士の関係性における入力ミスといった，分析対象システム固有の情報に関する入力ミスを検知するために，入力データという単一のドメインでなく，“分析対象のシステム仕様書”と“ユーザの作成した入力データ／生成されたアタックツリー”という 2 つの異なるドメインデータを活用し，2 つのドメインデータにおける入力単語（ユーザが入力した単語）の意味変化を評価することで入力ミスを検知する方式である．提案方式における入力単語の意味変化評価は，分散表現技術 word2vec を応用することで行う．

本章では，まず，セクション 3.1 で word2vec の概要と既存研究において報告されている word2vec の単語ベクトルに関する性質に触れ，続くセクション 3.2 では，提案方式のコアとなる word2vec をベースとした 2 つのドメインデータにおける単語の意味変化を評価する関数”意味変化評価関数 a(w)”を構築するフローを説明する．

そして，最後のセクション 3.3 では，意味変化評価関数 a(w)を用いることで，“分析対象システムの仕様書”と，“ユーザが作成した入力データ及び，ツールが生成したアタックツリー”間での入力単語の意味変化度合いを評価し，入力ミスと思われる，意味が大きく変化した単語を検知する判定式を提示し，アタックツリー自動生成ツールにおける入力ミス検知方式が異常ラベルなしデータに対する異常検知問題として帰着できることを示す．

3.1 分布仮説に基づく分散表現技術“word2vec”

分散表現とは，「単語の意味は，その単語が出現する文脈（周囲に出現する単語群）によって決定される」という分布仮説 [7]に基づいて，単語が文章中に出現する位置や共起する単語の傾向を統計的に解析することで，単語の意味を表現する多次元ベクトルを与える技術である．分散表現によって獲得した単語ベクトルは，類似した意味を持つ単語同士の単語ベクトルが近くなる性質を持っている．

そして近年，最も応用が進んでいる分散表現技術が，Mikolov らによって提案された“word2vec”である [8]. word2vec は，文脈中の特定の単語の周辺に出現する単語を予測する能力を持つニューラルネットワーク（Skip-Gram モデル）を，大量のテキストデータから学習させることで分散表現を獲得する技術である．

word2vec の応用が盛んになっている理由の 1 つに，獲得した単語ベクトルが線形演算を行うことで，単語の同士の関係性を表現した意味の遷移を表現することが出来るという性質があるためである．この性質は“加法構成性”と呼ばれ，V("king")-V("man")+V("woman")≅V("queen")といった演算が可能となる．(図 3)

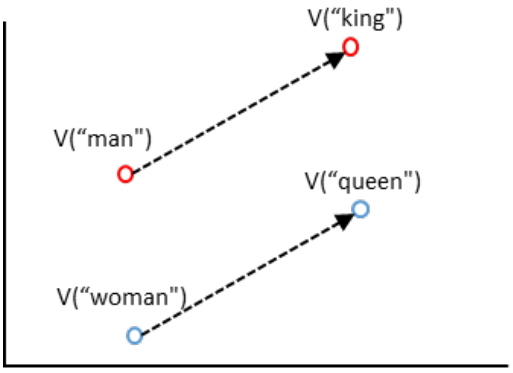

図 3 V("king")-V("man")+V("woman")≅V("queen")という加法構成性を持つベクトル空間

更に，Mikolov らは，word2vec が「異なるドメインデータから獲得した 2 つの単語ベクトル空間は，線形変換によって対応付けることが出来る(空間に幾何的な類似性がある)」という性質を持っていることを実験的に示し(図 4)，この性質を利用して，類似する異言語コーパスから獲得した単語ベクトルと少数の訳語ペア$\{x_i, z_i\}_{i=1}^{n}$を元に（x_i, z_iは共に単語ベクトルである)，両言語のベクトル空間に対する最適な線形変換行列を計算することで，対訳ペアにはなかった未知の単語をもう一方の言語の単語へ高い精度で翻訳

する技術を考案した [4].

異言語のベクトル空間に対する最適な線形変換行列W^*は“翻訳行列”と呼ばれており，下式の最適化問題を解くことで求められる行列として定義される．

$$W^* \coloneqq argmin_W || \sum_{i=1}^{n} W \cdot x_i - z_i ||$$

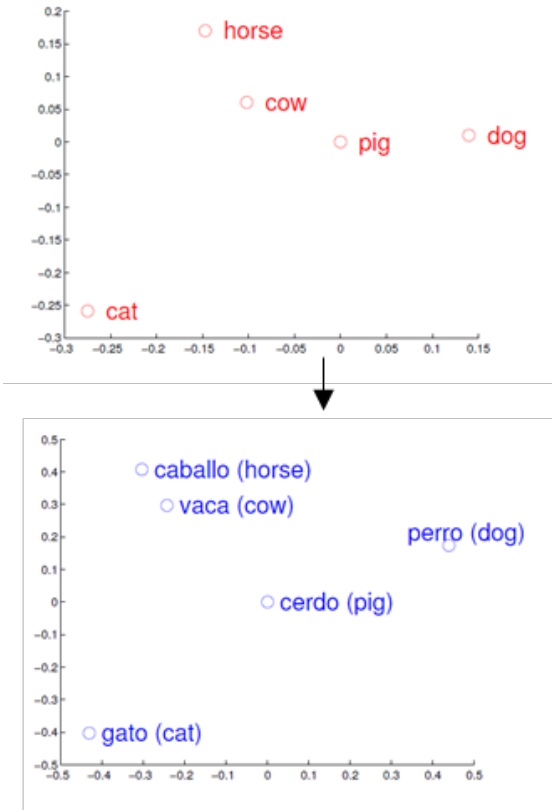

図 5 英語とスペイン語におけるベクトル空間の類似性

本図は [4]における p2-Figure1 を著者らで加工したものである

3.2 word2vec に基づく意味変化評価関数の構築

ここでは，同一言語の異なるドメインデータX, Zと，2 つのドメインデータに共通して出現する同一名称単語$\{w_i\}_{i=1}^n$が与えられた場合に，両ドメイン間での word2vec に基づく意味の変化を評価する関数$\mathrm{a}(w_i)$を構築する方法を示す．

「意味変化評価関数$\mathrm{a}(w_i)$の構築手順」

i. それぞれのテキストデータX, Zを学習データとして，意味変化を評価する任意の単語w_iに対して，2 種類の単語ベクトル$\{V_X(w_i)\}_{i=1}^n, \{V_Z(w_i)\}_{i=1}^n$を獲得する．

ii. 同一名称の単語ベクトルペア$\{V_X(w_i), V_Z(w_i)\}_{i=1}^n$に対して，以下の最小化問題を確率的勾配降下法などで解き，行列M^*を計算する．

$$M^* \coloneqq argmin_M || \sum_{i=1}^{n} M \cdot V_X(w_i) - V_Z(w_i) || + \lambda ||M||$$

iii. 求めた行列M^*を使って，任意の単語w_iの単語ベクトルを変換した際のフィッティング誤差を集めたデータ E を用意する．

$$E \stackrel{\text{def}}{=} \{err(w_i) | err(w_i) \coloneqq ||M \cdot V_X(w_i) - V_Z(w_i)|| \}_{i=1}^n$$

iv. 最後に，意味変化評価関数$\mathrm{a}(w_i)$をデータ E において，$err(w_i)$という値が出現する事象のエントロピーとして定義する．

$$\mathrm{a}(w_i) \stackrel{\text{def}}{=} -\ln p(err(w_i)|E)$$

ここで構築した意味変化評価関数$\mathrm{a}(w_i)$は，Mikolov らが示した「異なるドメインデータから獲得した 2 つの単語ベクトル空間は, 線形変換によって対応付けることが出来る」という性質のサブセットに相当する「同一言語の異なるドメインデータから獲得した 2 つの単語ベクトル空間は，同一名称の単語が両ドメイン上で同じ意味を持っている場合，線形変換によって対応付けることが出来る」という仮説をベースとしている．付け加えれば，意味変化評価関数$\mathrm{a}(w_i)$は，この仮説の対偶である「同一言語の異なるドメインデータにおいて，同一名称の単語であるにも関わらず線形変換によって対応付けることが出来ない単語は，両ドメイン上で同一の意味を持っていない」という命題を，行列によるフィッティング誤差という視点から，意味の差異（変化度合い）を定量化する関数である．

ii において，λ ||M||というL_1正則化項を導入しているのは，入力ミスが正しい入力に比べ低頻度であるという仮定の元で，入力ミスによって大きな意味変化が発生した少数の単語ベクトルが，行列M^*にフィッティングしない特徴を持つスパースでコンパクトな行列M^*が計算される確率を向上させるためである．よって，L_1正則化項を導入することで，意味変化が大きい少数の単語であるほど$err(w_i)$の値が大きくなることが期待できる．

最後に，意味変化評価関数$\mathrm{a}(w_i)$を，フィッティング誤差を集めたデータ E における大きな err(w)が観測できる事象のシャノン情報量として定義することで，確率分布という視点で定量化する．

3.3 意味変化評価関数 a(w)を応用した入力ミス検知方式

我々は，ある入力データ上の入力単語w_iが与えられた際に，その単語が入力ミスか否かを判定するために，同一言語の異なるドメインデータとして“分析対象のシステム仕様書”と“ユーザの作成した入力データ／生成されたアタックツリー”を対象に，意味変化評価関数a(w)を構築し，設定した許容値閾値τを逸脱するかを判定する方式を提案する．

従って，提案するアタックツリー自動生成ツールにおける入力ミス検知方式は，下式の成立判定問題であり，閾値τを超えるような単語w_iを入力ミスとして判定する．

$$\mathrm{a}(w_i) \stackrel{\text{def}}{=} -\ln p(err(w_i)|E) > \tau$$

これは異常ラベルなしデータ中に存在する異常値を検知する際に，広く利用されている異常度の評価式と同一である．

4. 考察

以下からは提案方式について，次の観点について考察を行う．

- 提案方式で有効に検知できる入力ミスについて（セクション 4.1）
- 提案方式では検知が難しい入力ミス（セクション 4.2）
- 許容値閾値 τ の設定（セクション 4.3）
- 入力データとアタックツリーの前処理（セクション 4.4）

4.1 提案方式で有効に検知できる入力ミス

ここでは，入力データを学習するだけでは検知できないような，対象システム固有の情報に関する入力ミスパターンである，

パターン 1.　単語カテゴリにおける入力ミス

パターン 2.　単語同士の関係性における入力ミス

に対して，本提案方式が有効であることを 2 章で示した入力ミス例に適応することで示す．

パターン 1 は，装置名称／ネットワーク名称／情報資産名称／…等の固有名詞である単語を入力する際の入力ミスであり，具体的には，“業務用 PC_1”といった装置名称が，ネットワーク名称を入力される欄に入力されているようなパターンを示していた．(表 3)

表 3 入力データにおけるネットワーク構成情報入力欄

ネットワーク名称	アドレス種別	回線	…
業務用ネットワーク_1	IPv4	有線	…
業務用ネットワーク_2	IPv4	無線	…
業務用 PC_1	IPv4	有線	…
…	…	…	…

表 1 の再掲

この入力データを元に，アタックツリーを生成した場合，図 6 のように分析結果である攻撃活動内容において，“業務用 PC_1”という単語が，あたかもネットワーク名称のように使用されている文脈の出現が想定できる．

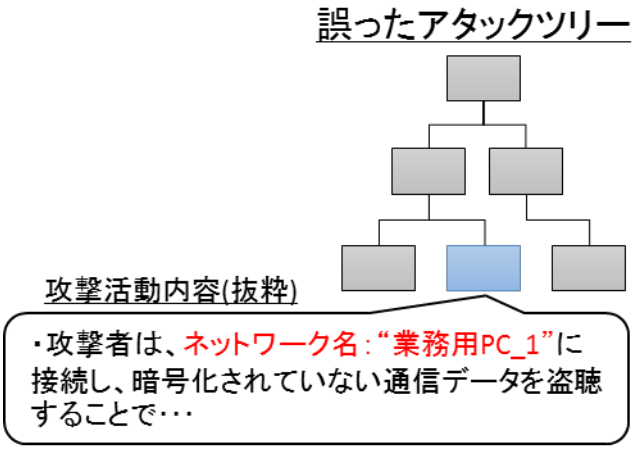

図 6 表 3 の入力を元にアタックツリー自動生成ツールが作成したアタックツリー

この時，word2vec で分散表現を獲得した場合，システム仕様書では“業務用 PC_1”は，装置名称が出現する文脈に登場するため，他の装置名称である“業務用 PC_2”，“業務用サーバ_1”といった単語と似た単語ベクトルを持つことが予測される．（図 7）

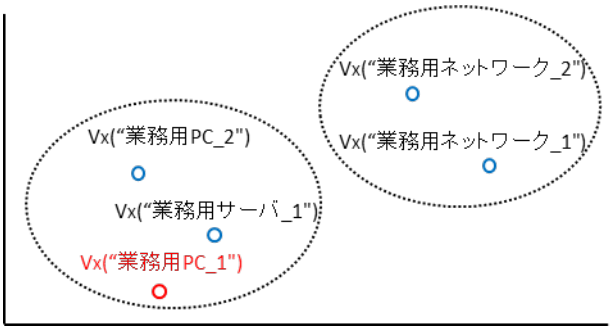

図 7 システム仕様書のベクトル空間 X

一方，入力データ上では“業務用 PC_1”という単語は，ネットワーク名称として入力されているため，入力データやアタックツリーの攻撃活動内容では，ネットワーク名称が出現する文脈で出現することになるため，他のネットワーク名称である“業務ネットワーク_1”，“業務ネットワーク_2”といった単語と似た単語ベクトルを持つことが予測される．（図 8）

以上の議論を踏まえると，装置名称というカテゴリの単語ベクトルと，ネットワーク名称という単語ベクトルは，システム仕様書におけるベクトル空間と，入力データ／アタックツリーにおけるベクトル空間に対し，図 8 の用にマッピングされるだろう．

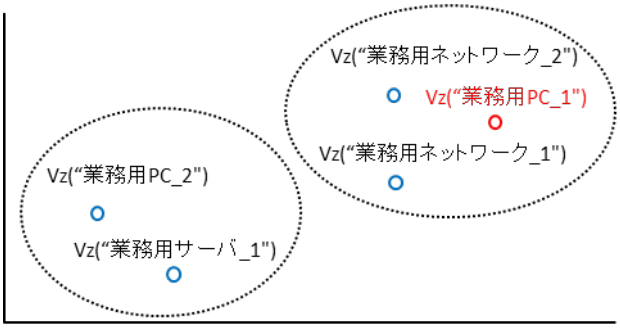

図 8 入力データ／アタックツリーのベクトル空間 Z

この時，ベクトル空間 X とベクトル空間 Z に対する線形変換を考えた場合，入力ミスである“業務用 PC_1”という単語ベクトルだけが，線形変換（拡大／縮小，平行移動，回転）で対応付けることが出来ないことが分かる．

従って，装置名称とネットワーク名称を対応付ける最適な行列M^*による“業務用 PC_1”のフィッティング誤差である$err($“業務用 PC_1”$)$は大きな値となり，入力ミスでない多数の入力単語wのerr(w)は小さな値となる．このため，a(“業務用 PC_1”$) = -\ln p(err($“業務用 PC_1”$)|E)$は，他の正常な入力単語wのa(w)より大きな値になることが導かれるため，提案方式で検知することが出来る．

パターン 2 は，パターン 1.と同様に，固有名詞である単語に関する入力ミスであるが，同じ要素カテゴリ内に属する入力単語間で発生する入力ミスである．具体的には，ある 2 つの装置名称”業務用 PC_1”と”業務用サーバ_1”が入れ替わった状態で入力され，結果として，それらの装置の属性値であるような入力単語（利用者名／装置種別／搭載 OS 名称／APP 名称…等）との関係性が損なわれるような入力ミスであった．（表 4）

表 4 入力データにおける装置情報入力欄

装置名称	利用者名	装置種別	…
業務用サーバ_1	従業員_1	OA-PC	…
業務用 PC_2	従業員_2	OA-PC	…
業務用 PC_1	サーバ管理人_1	サーバ	…
業務用サーバ_2	サーバ管理人_2	サーバ	…
…	…	…	…

表 2 の再掲

この場合，”業務用 PC_1”と”業務用サーバ_1”という単語ベクトルは，システム仕様書上でも，入力データ／アタックツリー上でも，装置名称というカテゴリに属する類似した単語であるため，両ベクトル空間で近傍にプロットされることが予測されるため，線形変換によるフィッティング誤差が小さいように思われる．

しかし，word2vec には加法構成性と呼ばれる，線形演算を行うことで意味を遷移させることが出来る性質が報告されていた(セクション 3.1)．そこで，図 3 のように，装置名称_X と X の利用者名という単語のペアに対して，V("A の利用者名")-V("装置名称_A")+V("装置名称_B")≒V("B の利用者名")という加法構成性が，システム仕様書と入力データ／アタックツリーの両ドメインのベクトル空間に構成されると仮定する．このとき，装置名称と利用者名というカテゴリの単語ベクトルをマッピングすると，2 つのベクトル空間の間で，“従業員_1”と”サーバ管理人_1”という単語ベクトルが入れ替わった空間が現れることになる（図 9，図 10）．

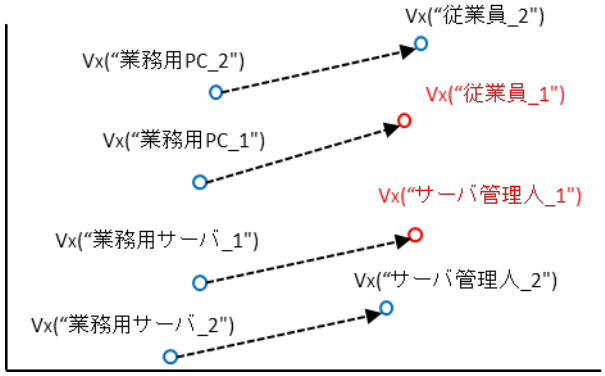

図 9 システム仕様書のベクトル空間 X

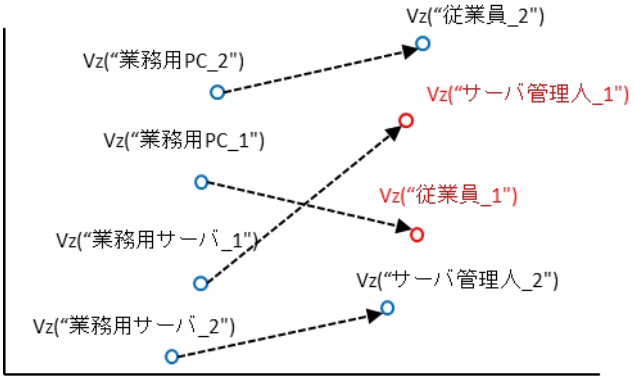

図 10 入力データ／アタックツリーのベクトル空間 Z

この時，ベクトル空間 X からベクトル空間 Z に対する“従業員_1”と”サーバ管理人_1”という単語ベクトルを対応付ける写像は線形ではないので，装置名称と利用者名というカテゴリの単語ベクトルを対応付ける最適な行列M^*によるフィッティング誤差$err($“従業員_1”$)$，$err($”サーバ管理人_1”$)$は大きい値となるため，パターン 1 と同様に提案手法で検知できることが分かる．

4.2 提案方式では検知が難しい入力ミス

しかし，本提案手法では，次のような入力ミスパターンに効果的ではないと考えられる．

- 単純なスペルミスや表記の揺れといった入力ミスパターン

⇒単語名で対応が取れなくなるため，対応する線形変換を構築できないことからa(w)が計測できない．しかし，このような入力ミスは，予測変換機能などの既存技術で対応可能だと考えられる．

- 加法構成性を持たない，類似する文脈で出現する 2 値的な入力単語

⇒本提案手法は，ベクトル空間の線形対応性の崩れを活用して検知を行うため，加法構成性を他の単語と持たず，類似する文脈で出現する 2 値的な入力単語である“YES/NO”属性の単語（接続している／いない）は検知が難しい．

また，解く対象の入力ミスパターンによらず，システム仕様書・入力データ・アタックツリー自動生成ツールが生成したアタックツリーのデータの規模が小さい場合には，単語ベクトルの獲得自体が上手くいかないため，検知精度が低くなると考えられる．

4.3 許容値閾値 τ の設定

また，本提案方式を実装する際には，入力ミスとして判定する閾値τを適切に設定する必要がある．

これに対しては，意味変化評価関数a(w)は，異常ラベルなしデータが与えられた際の異常度であるため，既存の異常検知技術における閾値τの決定方法を活用することが出来る．例えば，入力ミスについて，以下のような条件を仮定すれば，古典的なアルゴリズムであるホテリングのT^2法が適用できると考えられる．

- 検知対象の入力単語数が word2vec で獲得した単語ベクトルの次元数に対して十分大きい
- フィッティング誤差err(w)は互いに独立な正規分布に従う
- 入力ミスは，正しい入力よりも圧倒的に少数

ホテリングのT^2法では，入力ミスが発生する確率値を設定することで閾値τを具体的に求めることが可能である．例えば，入力ミスが発生する確率値の範囲の候補としては，単純な入力ミスが発生する確率を調査したWeisbergらの結果から，0.01~0.16 程度を設定することが考えられるだろう[9] [10].

しかし，提案方式におけるフィッティング誤差が真にどの様な分布を持つかは，実験的に分布を推定する必要がある．

4.4 入力データとアタックツリーの前処理

システム仕様書と異なり，アタックツリー自動生成ツールの入力データ，及びアタックツリーは自然言語文ではない．本質的には，word2vec は学習用のデータ形式が自然言語文章でなくとも，規則的に要素が並んだデータであれば，単語ベクトルを獲得することができるため，入力データとアタックツリーであっても単語ベクトルを獲得できる．しかし，意味変化を評価するドメインの傾向が大きく異なることになるため，ベクトル空間の類似性が低くなる可能性がある．従って，入力データとアタックツリーの内容を，何らかのパーサを用意し，自然言語文章に変換する前処理が必要になると考えられる．

5. 結論と今後

本論文では，入力単語の意味変化という視点から，アタックツリー自動生成ツールにおける入力ミスを検知する方式の理論提案と考察を行った．

本提案方式は，入力データのみを学習させる手法とは異なり，仕様書を確認しなければ発見できないような，分析対象システム固有の単語の意味を考慮した検知が行える．加えて，アタックツリー自動生成ツールというアプリケーションによらず，特定の情報源となるデータを元に人間がデータを作成するというタスクであれば，広く適応できる技術であると考えている．

今後の課題としては，本提案方式の対する考察が理論的な段階に留まっている為，有効性を確かめるために実験を行い，性能を評価する必要がある．

6. 参照文献

[1] X. Ou, S. Govindavajhala, and A. Appel., “MulVAL: A logicbased network security analyzer.,” 14th USENIX Security Symposium, pages 113‑128, 2005.

[2] R. L. a. K. P. Kyle Ingols, “"Practical attack graph generation for network defense",” IEEE, 2006.

[3] Shengwei Yi, Yong Peng, Qi Xiong, Ting Wang, Zhonghua Dai, Haihui Gao, Junfeng Xu, Jiteng Wang, Lijuan Xu , “Overview on attack graph generation and visualization technology., ” International Conference on Anti-Counterfeiting, Security and Identification (ASID), 2013.

[4] Mikolov, Tomas, Le, Quoc V., and Sutskever, Ilya. , “Exploiting Similarities among Languages for Machine Translation.,” arXiv:1309.4168 [cs], 2013.

[5] Z. Xie, A. Avati, N. Arivazhagan, D. Jurafsky, A. Y. Ng., “ Neural language correction with characterbased attention.,” arXiv:1603.09727, 2016.

[6] 小山田創哲，兼村厚範，石井信，“根拠を明示するニューラル文法誤り訂正,” DEIM Forum, G4-3, 2017.

[7] Z. S. Harris, “Distribuitonal structure,” Word, Vol. 10, No. 2-3 pp, 1954.

[8] Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg,and Dean, Jeffrey., “Distributed Representations of Words and Phrases and Their Compositionality.,” In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS’13, pp.3111-3119, USA, 2013c. Curran Associates Inc., 2013.

[9] 吉村治正, 小久保温, 澁谷泰秀, 渡部諭, “社会調査の入力ミスの発生率について,” 青森大学付属総合研究所紀要 Vol. 15, No. 1, 1-5, 2014.

[10] W. H, “The Total Survey Error Approach.,” University of Chicago Press, 2005.

[11] 浅井健志、島邉遼佑、河内清人, “サイバー攻撃対策の選定に向けたアタックツリーの自動生成,” SCIS, 2018.

[12] 石川雅弘, “分布仮説に基づく語の意味変化分析の試み,” つくば国際大学 研究紀要, No.22, 2016.

[13] 石渡祥之佑, 鍜治伸裕, 吉永直樹, 豊田正史, 喜連川優, “文脈語間の対訳関係を用いた単語の意味ベクトルの翻訳,” 人工知能学会論文, 31 巻 6 号, p. AI30-A_1-10, 2016.

[14] 井手剛, 杉山将, “異常検知と変化検知,” 講談社, , 2015.