情報処理学会論文誌 Vol.58 No.5 1175–1188 (May 2017)

ブラウザ拡張機能を用いた動的コンテンツフィルタリングシステム

高橋 研介[1,a)] 市野 将嗣[1,b)] 大山 恵弘[2,c)]

受付日 2016年7月22日, 採録日 2017年2月9日

概要：有害な Web サイトへの対策の 1 つとしてフィルタリングシステムが存在する．Web ページの URL や Web ページ上のテキストを使用する既存のフィルタリングシステムでは，HTTPS サイトに対してフィルタリングを行う場合，HTTPS の通信内容を通信途中で復号する必要があるうえ，閲覧する Web ページのコンテンツが第三者に渡るという問題が生じる．本稿では，ブラウザ拡張機能を用いることで，Web ブラウザレベルですべてのフィルタリング処理を行う動的コンテンツフィルタリングシステムを提案する．提案システムでは Web ページ上の HTML 要素とテキストに対しベイジアンフィルタを用いることで，その Web ページが有害であるかどうかを判定し，閲覧を制限する．

キーワード：ブラウザ拡張機能，コンテンツフィルタリング，ベイジアンフィルタ

Dynamic Content Filtering System Using Browser Extensions

Kensuke Takahashi[1,a)] Masatsugu Ichino[1,b)] Yoshihiro Oyama[2,c)]

Received: July 22, 2016, Accepted: February 9, 2017

Abstract: Filtering systems are one of countermeasures against harmful Web sites. However, HTTPS traffic must be decrypted for filtering HTTPS sites and the contents of the Web pages must be send to a third party, when filtering HTTPS sites using existing filtering systems that determine whether a Web page is harmful based on the URL or the text on the Web page. In this paper, we propose a content filtering system using browser extensions, which dynamically determines whether a given Web page is harmful based on the HTML elements and the text on the Web page using Bayesian filtering. All these processes are executed at the browser level.

Keywords: browser extension, content filtering, Bayesian filtering

1. はじめに

インターネット上には，青少年にとっての出会い系サイトのように，利用者にとって有害な情報を含む Web ページが存在する．そのような有害サイトへの対策の 1 つとして，有害サイトの閲覧を制限するフィルタリングシステムが存在する．従来のフィルタリングシステムにおける主流の手法として，アクセスした Web ページの URL を用いたブラックリスト方式があげられる [1], [2], [3]．しかし，この手法では内容が変更されたばかりの Web ページや，新たに登場した Web ページに対して閲覧を制限できない．また，制限対象の URL のデータベースを作成・管理するために，多大な時間とコストを要する．これらの問題を解決するため，アクセスした Web ページ上のテキストなどの要素を使用するコンテンツフィルタリングの研究が行われている [4], [5], [6]．

一方，Web では通信の盗聴への対策として，通信の暗号化により安全性を高める動きが生まれており，IETF や W3C は HTTP から HTTPS への移行を進める声明を発表している [7], [8]．近年その動きは加速しており，HTTP

[1] 電気通信大学
The University of Electro-Communications, Chofu, Tokyo 182–8585, Japan
[2] 筑波大学
University of Tsukuba, Tsukuba, Ibaraki 305–8577, Japan
[a)] t1530038@edu.cc.uec.ac.jp
[b)] ichino@inf.uec.ac.jp
[c)] oyama@cc.tsukuba.ac.jp

Archive [9] によると，主要な約 50 万サイトにおける，2016 年 6 月 15 日時点の HTTPS でのリクエストが占める割合は 28%であり，2015 年 6 月 15 日時点の 18%から大幅に上昇している．また，2016 年 4 月には SSL サーバ証明書を無料で取得できる「Let's Encrypt」がベータ版から正式版に移行され [10]，HTTP から HTTPS への移行に対する 1 つの障壁となっていたコストの問題も解消されつつある．Web ページの URL や Web ページ上のテキストを使用して閲覧を制限するフィルタリングシステムでは，Web サーバと Web ブラウザ間の通信からそれらを解析して使用する．そのため，HTTPS サイトに対してフィルタリングを行う場合，通信が暗号化されているため閲覧を制限できない．HTTPS の通信内容を通信途中で復号してフィルタリングを行うシステムが存在する [11] が，閲覧する Web ページのコンテンツが第三者に渡るという問題が生じる．

本稿ではこのような問題を解決するために，ブラウザ拡張機能を用いた動的コンテンツフィルタリングシステムを提案する．本システムはすべての処理を Web ブラウザレベルで行い，アクセスした Web ページが有害であるかどうかをベイジアンフィルタによって判定し，閲覧を制限する．これにより，Web ページ上の HTML 要素とテキストを使用したコンテンツフィルタリングを行うとともに，通信途中で復号せず HTTPS サイトのフィルタリングを可能にする．本研究は Web ページ上のコンテンツを基にした機械学習による判定手法を用いて，有害サイトの閲覧を制限するフィルタリングシステムをブラウザ拡張機能で実装している点で新規性を有し，通信の暗号化の普及にともない発生するフィルタリングシステムの問題解消に貢献する．

本稿は全 7 章で構成されている．2 章では既存研究や既存システムを紹介する．3 章では提案システムの概要を説明し，4 章では各処理の実装方法を説明する．5 章では閾値決定のための予備実験や，フィルタリングの判定精度や処理時間などに関する評価実験の結果について説明する．6 章では提案システムの問題点とその対策について述べる．最後に 7 章では本稿のまとめを述べる．

2. 既存研究と拡張機能

2.1 既存研究

日本語で記述された Web ページを対象に，Web ページ上のテキストなどを使用した動的コンテンツフィルタリングシステムを提案した既存研究が複数存在する．しかし，HTTPS サイトに対するフィルタリングを考慮した研究は存在しない．

井ノ上ら [4] は Web ページ上のテキストに対し形態素解析を行い，その単語を基にベクトル空間モデルを用いることで，有害な Web ページへの閲覧を制限する動的コンテンツフィルタリングシステムを提案している．また，大井ら [5] は Web ページ上の単語の tf-idf を基に，Web ページを複数のカテゴリへ分類し，各カテゴリに設定した閲覧時間を超えた場合，閲覧を制限するシステムを提案している．これらのシステムはプロキシサーバ内に実装されているため，HTTPS サイトの閲覧を制限できない．

上田ら [6] は HTTP パケット内のペイロードからテキストを抽出し，そのテキストを分かち書きした結果に含まれるブラックワードの出現回数に基づき，インターネットの利用状況をメールで通知するシステムを提案している．このシステムは，保護者による子供のインターネット利用の監視のみが目的のため閲覧は制限されない．パケット内のペイロードに含まれるテキストを使用するため，HTTPS サイトに対して解析できない．

本研究で使用している，機械学習を用いた有害サイトの判定を目的とした研究が複数存在する．

池田ら [12] は Web ページ上の HTML 要素と，Web ページ上のテキストに対し形態素解析を行い生成した単語を基に，それぞれ SVM を用いて，それらの結果を組み合わせることで有害サイトを判定する手法を提案している．

菊池ら [13]，吉村ら [14]，中村ら [15] は Web ページ上のテキストに対し形態素解析を行い，単語の共起関係を基にベイジアンフィルタを用いて，有害サイトを判定する手法を提案している．これらの研究では，動的コンテンツフィルタリングシステムとしての実装はされておらず，有害サイトの判定に要する処理時間の評価結果も明示されていない．動的コンテンツフィルタリングシステムとして実装する場合，ユーザがストレスを感じることなくブラウジングできる処理時間に抑える必要があるが，単語の共起関係を用いる場合，単語の組合せ数が大きくなり，実用が困難となる可能性がある．

Likarish ら [16] はブラウザ拡張機能上でベイジアンフィルタを用いることで，フィッシングサイトを検出するシステムを提案している．本研究における検出対象である有害サイトには，作成者の意図により有害と判定されるコンテンツが含められる．一方，Likarish らが検出対象としているフィッシングサイトは，正規の無害なものに見せかけたコンテンツにより構成される．このように有害サイトとフィッシングサイトは特徴が大きく異なるため，それらを検出するための技術も異なるものになる．そして，本研究では有害サイトを検出対象としており，有害サイトを正確に判定するうえで，通常のブラウジングに影響が出ないよう，偽陽性率と偽陰性率を両方低下させる必要がある．一方，フィッシングサイトを検出対象とする場合，1 度でもフィッシングサイトを正規サイトと誤って判定することのないよう，偽陰性率を特に低下させる必要があり，判定に要求される条件が大きく異なる．また，判定手順について，本研究ではブラックリスト/ホワイトリストに追加された Web ページを除く，ほとんどの Web ページに対しベイジアンフィルタによる判定が行われる．一方，Likarish

らの研究では開発者が提供する大規模なホワイトリストにより多くの主要サイトを除外した後，補助的にベイジアンフィルタによる判定が行われる．なお，日本語で記述されたWebページは対象としていない．

2.2 既存の拡張機能

有害サイトのフィルタリングを目的とした多くの拡張機能が配布されている．しかし，Webページ上のHTML要素やテキストを基に，ベイジアンフィルタのような判定手法を用いてフィルタリングを行う拡張機能は配布されていない．

Web of Trust [17] は，この拡張機能を使用する世界中のユーザがWebページを評価し，評価の低いWebページの閲覧を制限する．この手法の問題点として，評価が行われていないため閲覧が制限されない有害なWebページが多く存在する点や，個人の評価基準が異なる点があげられる．

BlockSite [18]，LeechBlock [19] はユーザが作成した，有害サイトのURLデータベースを使用して閲覧を制限する．ユーザがURLデータベースを作成するため，有害サイト全般に対して閲覧を制限できない．

Blocksi [2]，WebFilter Pro [3] はそれぞれ拡張機能の開発元であるBLOCKSI社，Cloudacl社が保持するURLデータベースを使用して閲覧を制限する．1章で述べたURLを用いたブラックリスト方式であり，同様の問題点をかかえている．

FoxFilter [20]，ProCon Latte Content Filter [21] はユーザが作成したブラックワードデータベースを使用し，Webページにブラックワードが含まれていた場合，閲覧を制限する．問題点として，ブラックワードが1カ所に含まれているだけで閲覧を制限するため，過剰な閲覧制限を引き起こす可能性が高いことがあげられる．

2.3 既存の市販ソフトウェア・フリーウェア

i-FILTER [11]（企業向け製品），i-フィルター [22]（個人向け製品）は市販のフィルタリングソフトウェアであり，デジタルアーツ株式会社が保持しているURLデータベースと動的コンテンツフィルタリングを併用して閲覧を制限する．HTTPSサイトに対するフィルタリングに対応しているが，閲覧するWebページのコンテンツが第三者に渡るという問題が生じる．また，個人向け製品では，HTTPSサイトに対する動的コンテンツフィルタリングはInternet Explorer以外のブラウザに対応していない．

市販のフィルタリングソフトウェアであるInterSafe Personal [1]，フリーウェアであるWindows Liveファミリーセーフティ [23] はそれぞれアルプスシステムインテグレーション株式会社，マイクロソフト社が保持しているURLデータベースを使用して閲覧を制限する．1章で述べたURLを用いたブラックリスト方式であり，同様の問題点をかかえている．

3. 提案システムの概要

提案システムはブラウザ拡張機能を用いた動的コンテンツフィルタリングシステムであり，アクセスしたWebページが有害であるかどうかをベイジアンフィルタにより判定し，閲覧を制限する．なお，提案システムにおける有害サイトの判定手法は，池田ら [12] によるHTML要素とテキストから生成されたトークンを使用する判定手法のアルゴリズムを，菊池ら [13]，吉村ら [14]，中村ら [15] により使用されているベイジアンフィルタに変更したものであり，新規性を有しない．本研究はユーザが提案システムをクライアントサイドで使用することをふまえた，4.5節で説明する有害確率の閾値変更機能や，4.7節で説明するオンライン学習機能の実装に新規性を有し，ベイジアンフィルタによる判定手法を用いたFirefox拡張機能により実装された動的コンテンツフィルタリングシステムが，大部分のWebページに対し，短時間でフィルタリング可能であることを，性能評価により示すことに有用性を有する．

本研究ではFirefox拡張機能により提案システムを実装し，日本語で記述されたWebページをフィルタリングの対象とする．日本語のみを対象としている理由は，著者が日本人であり，日本語で記述されたWebページの収集やコンテンツ内容の目視が容易であることに加え，日本語で記述された有害サイトは多言語で記述された有害サイトに比べ大規模な収集が容易なためである．利用目的は一般的なフィルタリングシステムと同様，一般家庭におけるペアレンタルコントロールなどを想定しており，以降，提案システムにより閲覧を制限する立場を管理者，閲覧を制限される立場を被管理者と呼ぶ．なお，Mozilla公式のFirefox拡張機能の配布サイトであるaddons.mozilla.org上で，提案システムを実験的アドオンとして一般公開している [24]．

図 1 に提案システムの概略図を示す．提案システムは，以下の流れで動作する．

(1) Webブラウザ上のページ遷移を検知し，アクセスされるWebページのHTMLソースを取得する．
(2) HTMLソースからHTML要素を抽出し，HTMLトークンに分割する．
(3) 有害サイトと無害サイトに含まれるHTMLトークンの情報が記述された学習データと，分割されたHTMLトークンを使用して，ベイジアンフィルタによりHTML要素を使用したWebページの有害確率を算出する．
(4) いずれかのカテゴリについて有害確率が上限閾値を超えていた場合閲覧を制限し，下限閾値以上かつ上限閾値以下である場合，テキストによる判定に移行する．
(5) HTMLソースのBODYタグの要素に含まれるテキストに対して分かち書きを行い，テキストトークンに分割する．

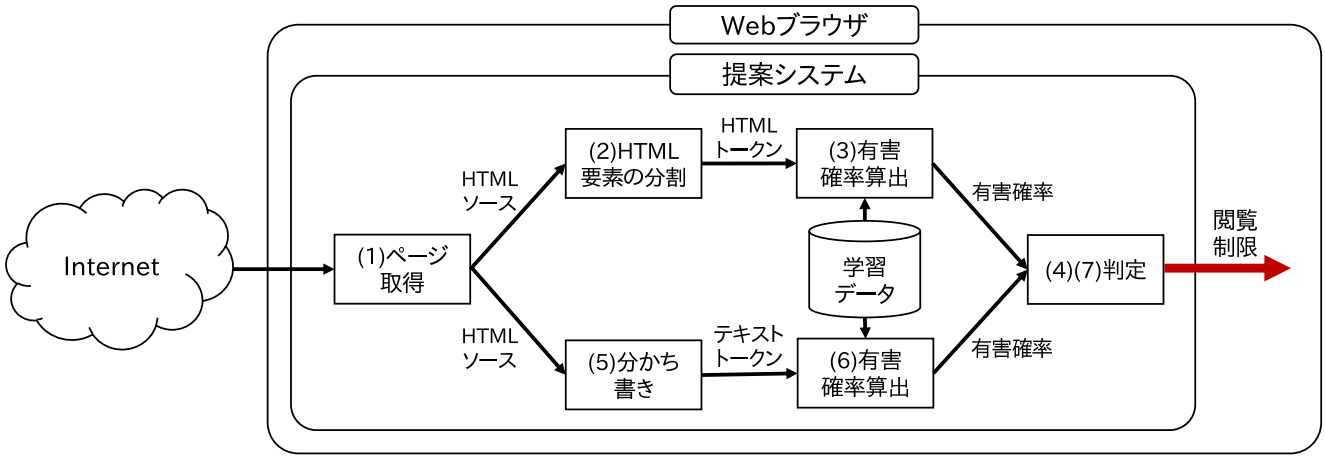


図 1 提案システムの概略図
Fig. 1 Overview of the proposed system.

(6) 有害サイトと無害サイトに含まれるテキストトークンの情報が記述された学習データと，分かち書きされたテキストトークンを使用して，ベイジアンフィルタによりテキストを使用した Web ページの有害確率を算出する．

(7) 有害確率が閾値を超えているかどうかを判定し，いずれかのカテゴリについて閾値を超えていた場合閲覧を制限する．

4. 実装

4.1 HTML ソースの取得

Add-on SDK の tabs モジュールを用いてブラウザのタブの動作を監視し，ページの遷移を検知する．ページの遷移を検知すると，window/utils モジュールを使用して，DOMContentLoaded イベントが実行されるタイミングで，ドキュメントの HTML ソースである document.documentElement.outerHTML を取得する．この HTML ソースがページとして判定に使用されるコンテンツとなる．

4.2 HTML トークンの生成

池田ら [12] と同様の手法を使用しており，取得した HTML ソースから，< と > で囲まれた HTML 要素を抽出する．抽出した HTML 要素について空白や記号を区切り文字として分割し，HTML トークンを生成する．**図 2** に HTML トークン生成の例を示す．分割後に登場するスラッシュは HTML トークンの境界を表す．

4.3 テキストトークンの生成

取得した HTML ソースから，BODY タグの要素を取得する．BODY タグが省略されている場合，HTML ソースから HEAD タグの要素を除去したものを代わりに使用する．BODY タグの要素からコメント・スクリプト・スタイルシート・HTML タグを除去することで，Web ページ上に表示されるテキストのみを抽出し，分かち書きを行う．なお，半角カタカナについては，分かち書きの前に全角カタカナに置換する．ベイジアンフィルタで使用するテキストトークンは，漢字・ひらがな・カタカナのいずれかが含まれるものに限定する．加えて，ひらがな 2 文字以下のテキストトークンに関しては，助詞のように単独では意味をなさない可能性が高いため除外する．分かち書きには Firefox 拡張機能と同様に JavaScript で実装されている TinySegmenter [25] を用いた．**図 3** にテキストトークン生成の例を示す．分かち書き後に登場するスラッシュはテキストトークンの境界を表す．

料金のご案内
↓ (HTML 要素の抽出)

↓ (HTML トークンへ分割)
a / href / guide / price / html / span / span / a

図 2 HTML トークン生成の例
Fig. 2 Example of HTML token generation.

↓ (テキストの抽出)
「価値観」「趣味趣向」の近い男女を
マッチングさせることを可能にしました
↓ (分かち書き)
「/価値観/」/「/趣味/趣向/」/の/近い/男女/を/
マッチング/さ/せる/こと/を/可能/に/し/まし/た
↓ (テキストトークンの限定)
価値観/趣味/趣向/近い/男女/マッチング/可能

図 3 テキストトークン生成の例
Fig. 3 Example of text token generation.

4.4 有害確率の算出

有害サイトの判定のため，生成された各トークンの集合

と，学習データを基にして，Robinson 方式のベイジアンフィルタ [26] を用いることで Web ページの有害確率を登録されたカテゴリごとに算出する．Web ページの有害確率は，以下の Robinson 方式のベイジアンフィルタのアルゴリズムに従い算出される．

(1) トークン w が有害サイトに登場する確率 $p(w)$ を算出する．

$$p(w) = \frac{\frac{b}{n_{bad}}}{\frac{g}{n_{good}} + \frac{b}{n_{bad}}} \tag{1}$$

b は全有害サイトにおけるトークンの登場回数の合計，g は全無害サイトにおけるトークンの登場回数の合計，n_{bad} は有害サイトのページ数，n_{good} は無害サイトのページ数である．

(2) トークン w の有害確率 $f(w)$ を算出する．

$$f(w) = \frac{s \cdot x + n \cdot p(w)}{s + n} \tag{2}$$

n は有害サイトと無害サイトのページ数の合計である．x は学習データファイルに登場しないトークンが Web ページ上に登場する予測確率であり，s はその予測に与える強さである．既存手法 [26] では，$x = 0.5$，$s = 1$ が妥当とされているため，本研究においても同様の値を使用する．

(3) Web ページの有害性 S，および非有害性 H を算出する．

$$S = 1 - \left\{ \prod_{i=1}^{n} (1 - f(w_i)) \right\}^{\frac{1}{n}} \tag{3}$$

$$H = 1 - \left\{ \prod_{i=1}^{n} f(w_i) \right\}^{\frac{1}{n}} \tag{4}$$

n は Web ページ上に登場するトークンの異なり数である．異なり数とは，同一のトークンが何度登場してもこれを 1 トークンとし，全体で異なるトークンがいくつあるかを数えた数である．

(4) Web ページの有害確率 P を算出する．

$$P = \frac{1 + \frac{S-H}{S+H}}{2} \tag{5}$$

有害確率は 0 から 1 の範囲の値をとる．

4.5 閲覧の制限

算出された有害確率を基に，**図 4** のフローチャートに従って閲覧を制限するか判定する．まず，判定精度は低いが高速な HTML 要素による判定によって，有害確率が極度に低い/高い Web ページの閲覧許可/制限を決定し，HTML 要素による判定における有害確率が中程度のものについては，比較的低速だが判定精度の高いテキストによる判定を行う．なお，HTML トークンの異なり数が小さい場合は判定精度が低下するため，HTML 要素による判定

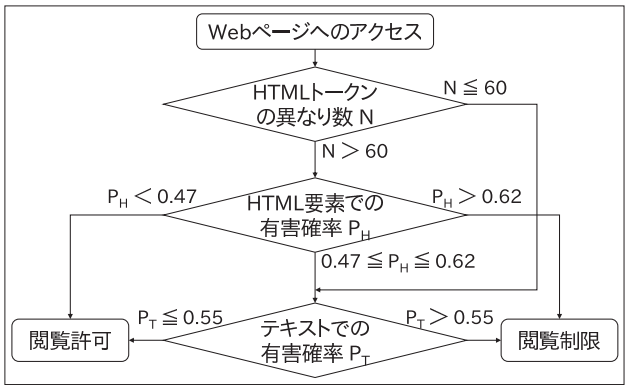


図 4 判定の流れ
Fig. 4 Judgment workflow.

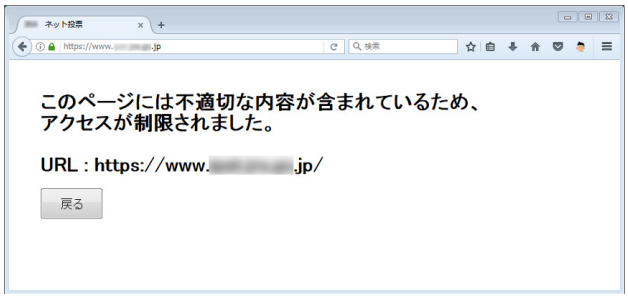


図 5 閲覧制限画面
Fig. 5 Viewing restrinciton page.

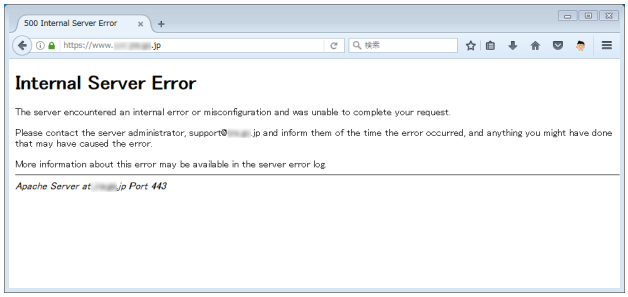


図 6 ダミーのエラー画面
Fig. 6 Dummy error page.

が飛ばされ，テキストによる判定から行われる．トークンの異なり数と有害確率の閾値は，5.2 節の予備実験の結果に基づき設定した．分かち書きの結果テキストトークンが 0 個だった場合，有害確率を 0.55 とし，無害サイトとして判定する．なお，提案システムでは設定により有害確率の閾値を 0.9 倍，0.95 倍，1.05 倍，1.1 倍（HTML 要素による判定における有害確率の上限閾値は逆の値をとる）に変更可能であり，管理者の意向により判定の厳格さをコントロールできる．

いずれかのカテゴリで有害サイトとして判定された場合は，対応を以下の 4 つから選択する．1 つ目は，Web ページの BODY タグを持つ DOM ノードを，**図 5** のように，閲覧が制限されたことを伝える内容，URL および「戻る」ボタンを配置した DOM ノードに置換し，閲覧を制限する．

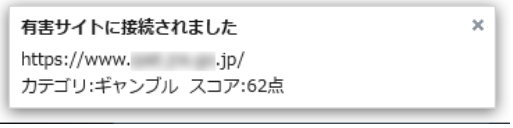


図 7 警告の表示
Fig. 7 Warning display.

2 つ目は，**図 6** のように，ダミーの Apache の内部エラー画面に DOM ノードを置換し，閲覧を制限する．3 つ目は，管理者が指定した URL にリダイレクトする．4 つ目は閲覧を制限せず，notifications モジュールを用いて，**図 7** のように，有害サイトに接続されたことを伝える内容を画面の右下に警告として表示する．

4.6 DOM ノードの非表示化

Web ページ全体ではなく，ユーザにとって好ましくない Web ページの一部を非表示にする拡張機能が複数存在する [27], [28]．提案システムにはこれを動的コンテンツフィルタリングに適用した機能を備えている．この機能は，検索エンジンにおける検索結果のフィルタリングに使用できる点や，一部のコンテンツのみ有害な Web ページに対して過剰な閲覧制限を防止できる点が利点としてあげられる．動作の流れは以下のとおりである．ページの遷移を検知した後，BODY タグの要素を取得する．その後，DOM ノードの clientHeight と clientWidth の積が 89000 px 以上になるよう，特定のブロックレベル要素（div，p，address，form，table，ul，li，ol，dl，h1，h2，header，footer，nav，main，aside，article，section）ごとに DOM ツリーを展開し，各 DOM ノードごとに Web ページ全体と同様の手順で有害確率の算出や判定を行い，有害と判定された場合は DOM ノードを非表示にする．面積の閾値は，大手検索サイトにおいて，検索結果を 1 件ずつフィルタリングできることを目安に設定した．なお，Google の検索結果ページにおいては，clientWidth が 0 の DOM ノードが，全検索結果を含んだ子ノードを保持しているため，その DOM ノードに関しては例外的に DOM ツリーを展開する．

4.7 学習データ

提案システムの使用にあたり，拡張機能の開発者は学習データを作成する必要がある．拡張機能の開発者は有害サイトと無害サイトを一定数用意し，4.2 節と 4.3 節の手法によりトークンの生成を行う学習用ツールを使用して学習データを作成する．URL を用いたブラックリスト方式では制限対象となるすべての URL を収集する必要があるのに対し，提案システムでは一定数の有害サイト，無害サイトを収集するだけでよい．学習データファイルは，HTML トークンの情報が記述されたファイルと，テキストトークンの情報が記述されたファイルに分かれており，それぞれ学習データとして使用した全有害サイトにおけるトークンの登場回数の合計と，全無害サイトにおけるトークンの登場回数の合計が JSON 形式で記述されている．なお，各学習データファイルには，学習データとして使用した有害サイトのページ数，無害サイトのページ数，および有害サイトのカテゴリに関するメタデータが記述されている．

拡張機能にはデフォルトで使用される学習データファイルが含まれているが，管理者による追加での学習が可能である．ブラウザで閲覧している Web ページに対して，ui/button/action モジュールを用いたアクションボタンか context-menu モジュールを用いたコンテンツメニューから容易にオンライン学習ができるようになっているほか，保存された HTML ファイルをバッチ学習する機能が存在する．また，管理者によって任意のカテゴリを追加することが可能である．これらにより，各々に応じた学習データを使用してのフィルタリングを行うことが可能となる．なお，基本的には追加学習により正しい判定に修正できるが，例外的に有害サイト/無害サイトと判定したい場合は，URL もしくはドメインをブラックリスト/ホワイトリストへ追加することにより，閲覧を制限/許可できる．

5. 評価実験

5.1 評価方法

評価で使用するデータセットとして，有害サイトの例には SafetyOnline3.1 [29] で 18 歳未満利用制限とレイティングされている，アダルト，出会い，ギャンブルの 3 カテゴリについてそれぞれ 3,000 ページ，合計 9,000 ページ，無害サイト 10,000 ページを用意した．これらの全 Web ページは日本語を中心として記述されている．有害サイトには，アダルトサイト，出会い系サイトと出会い系サイトに関する情報を扱う Web ページ，ギャンブルに関する情報を扱う Web ページを使用し，クローリングおよび検索サイトを通じて，独自に収集した．なお，これらの全 Web ページは Trend Micro Site Safety Center [30] において，アダルト/成人向けカテゴリもしくはポルノカテゴリ，出会いカテゴリ，ギャンブルカテゴリにそれぞれ分類されている．無害サイトには，goo カテゴリ検索 [31] に登録されている Web ページの中で取得できた約 29 万ページに対しランダムサンプリングを行い，10,000 ページを使用した．

実験では 5-分割交差検証により評価を行った．データセットを 5 等分し，有害サイト 3 カテゴリについてそれぞれ 600 ページ，合計 1,800 ページと無害サイト 2,000 ページをテストデータに，残りの有害サイト 7,200 ページと無害サイト 8,000 ページを学習データとして評価を行い，これを順に 5 回繰り返し平均を算出した．本実験はブラウザが Mozilla Firefox 47.0，OS が Microsoft Windows 7 Professional SP1（64 bit），RAM が 8 GB，CPU が Intel Core i7-2600 の環境で行われたが，ブラウザが Firefox ESR

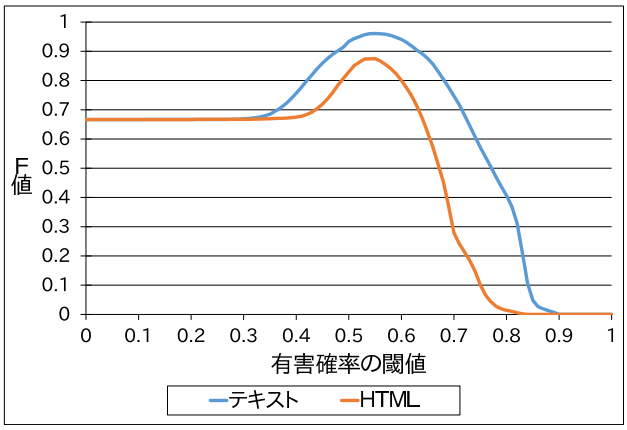


図 8 有害確率の閾値の変化による F 値の変化
Fig. 8 Change of F value due to the change in the harmful probability threshold.

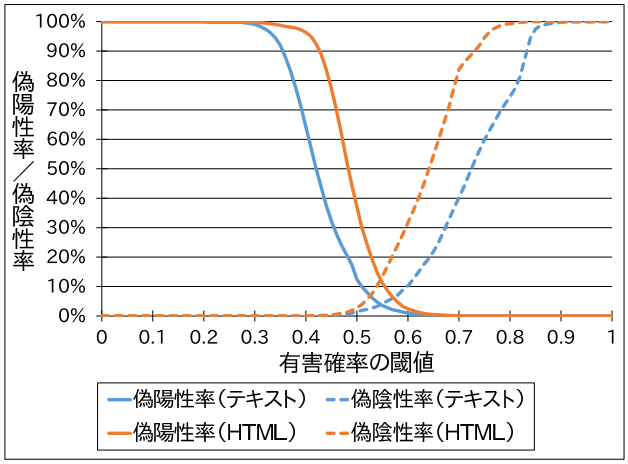


図 9 有害確率の閾値の変化による偽陽性率・偽陰性率の変化
Fig. 9 Change of the false-positive rate and the false-negative rate due to the change in the harmful probability threshold.

38.2.0，OS が CentOS 6.7（64 bit）の環境においても，提案システムの正常な動作を確認した．

5.2 予備実験

判定における有害確率の閾値とトークンの異なり数の閾値を決定するため，それらの変化より判定精度がどのように変化するか実験する．また，学習データ数の変化により判定精度がどのように変化するか示し，考察を行う．

5.2.1 有害確率の閾値

フィルタリングにおける有害確率の閾値の変化による，判定精度の F 値の変化を**図 8** に示す．F 値が最大となったのは，テキストによる判定，HTML 要素による判定のどちらも有害確率の閾値が 0.55 の場合となった．これを基に，以降の評価実験では，テキストによる判定，HTML 要素による判定の際，有害確率の閾値として 0.55 を使用する．なお，この際の F 値はテキストによる判定では 0.961，HTML 要素による判定では 0.875 となり，テキストによる判定の方が，高精度となった．続いて，フィルタリングにおける有害確率の閾値の変化による，偽陽性率と偽陰性率の変化を**図 9** に示す．無害サイトを有害サイトと誤って判定する偽陽性率，有害サイトを無害サイトと誤って判定する偽陰性率ともにテキストによる判定では，HTML 要素による判定に比べ，つねに小さくなった．提案システムで採用しているテキストと HTML 要素による判定を複合させた判定手法では，HTML 要素による判定における誤判定を極力減少させるため，HTML 要素による判定において偽陽性率と偽陰性率が約 1%ずつに収まるように，HTML 要素による判定における有害確率の下限閾値を 0.47，上限閾値を 0.62 とした．

5.2.2 トークンの異なり数

ベイジアンフィルタでの判定に使用するトークンの異なり数の変化による，判定精度の F 値の変化を**図 10** に示す．使用するトークンの異なり数は 10 から 200 まで 10 ずつ増

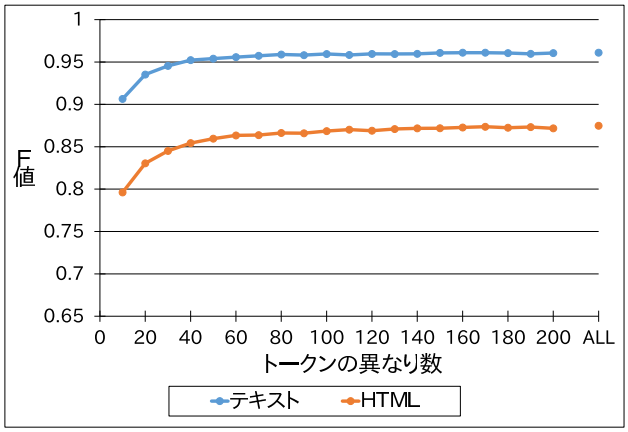


図 10 トークンの異なり数の変化による F 値の変化
Fig. 10 Change of F value due to the change in the number varies of tokens.

加させており，折れ線グラフ右の ALL は使用するトークンの異なり数に制限を設けない場合である．判定精度はつねにテキストによる判定が HTML 要素による判定を上回り，どちらもトークンの異なり数が増えるにつれて判定精度が上昇している．テキストによる判定の場合，トークンの異なり数が小さくても F 値は 0.9 を超え，高精度となった．一方，HTML 要素による判定の場合，トークンの異なり数が小さいと F 値が 0.8 を下回った．F 値の上昇幅が縮小し始めるトークンの異なり数が 60 以上であれば安定した判定精度が出せると判断し，HTML 要素による判定に突入するためのトークンの異なり数の閾値を 60 とした．

5.2.3 学習データ数

フィルタリングにおける学習データ数の変化による，判定精度の F 値の変化を**図 11** に示す．有害サイト 7,200 ページ，無害サイト 8,000 ページを 100%として，5%（有害サイト 450 ページ，無害サイト 500 ページ）刻みで学習データ数を増やしていき計測を行った．なお，有害サイト

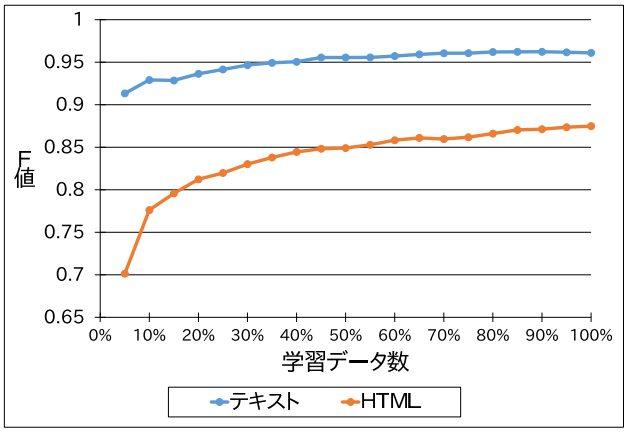


図 11 学習データ数の変化による F 値の変化
Fig. 11 Change of F value due to the change in the number of training data.

と無害サイトの比率は一定であり，テストデータ数は有害サイト 1,800 ページ，無害サイト 2,000 ページで一定である．テキストによる判定の場合，学習データ数が小さくても F 値は 0.9 を超え，学習データ数が増加するにつれ，さらに判定精度が徐々に向上した．一方，HTML 要素による判定の場合，学習データ数が小さいと F 値は約 0.7 まで低下する結果となった．ただ，学習データ数が増加するにつれ，判定精度は大きく向上した．この結果から，拡張機能の学習データファイル内に搭載されている 3 カテゴリについては，本稿におけるデータセットをすべて学習データとして使用しているため十分な判定精度を見込めるが，管理者によって新たに追加されたカテゴリについて判定する場合，学習データ数が小さいと判定精度の低下が見込まれる．学習の自動化などにより，学習プロセスをより容易にすることが今後の課題である．

5.3 本実験

予備実験で決定した閾値を使用したテキストと HTML 要素による判定を複合させた判定手法，およびテキストによる判定手法，HTML 要素による判定手法について，判定精度や処理時間に関する比較を行う．また，DOM ノード非表示化について適切に非表示化が行われるかを確認し，その結果について考察を行う．最後に，既存のフィルタリングシステムとの比較を行い，提案システムがどのような場合に優位性を持てるのかを示す．

5.3.1 判定精度と平均処理時間

フィルタリングにおける判定精度と平均処理時間の計測結果を**表 1** に示す．テキストと HTML 要素による判定を複合させた判定手法の場合，偽陽性率，偽陰性率は，それぞれ約 4%，約 4.5%となった．テキストによる判定と比較すると，偽陽性率 0.48%，偽陰性率 0.25%の低下が見られたものの，平均処理時間についてはおよそ半分に減少している．テキストによる判定と HTML による判定を複合させることにより，判定精度と処理時間という両手法の優れた部分が活用されている．

表 1 各判定手法における判定精度と平均処理時間
Table 1 Accuracy and average execution time of each judgement method.

	テキスト	HTML	複合
真陽性率 [%]	95.77	86.48	95.52
真陰性率 [%]	96.47	88.78	95.99
偽陽性率 [%]	3.53	11.22	4.01
偽陰性率 [%]	4.23	13.52	4.48
正解率 [%]	96.12	87.63	95.76
適合率 [%]	96.45	88.52	95.97
再現率 [%]	95.77	86.48	95.52
F 値	0.961	0.875	0.957
平均処理時間 [ms]	111.8	12.6	58.3

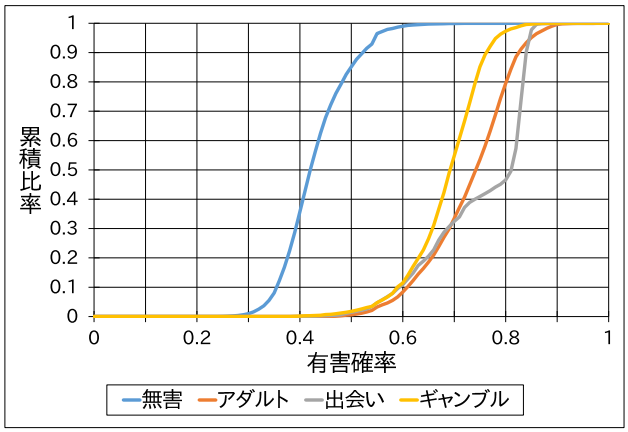


図 12 テキストによる判定における有害確率の累積比率
Fig. 12 The cumulative percentage of harmful probability in the judgement by the text on the webpage.

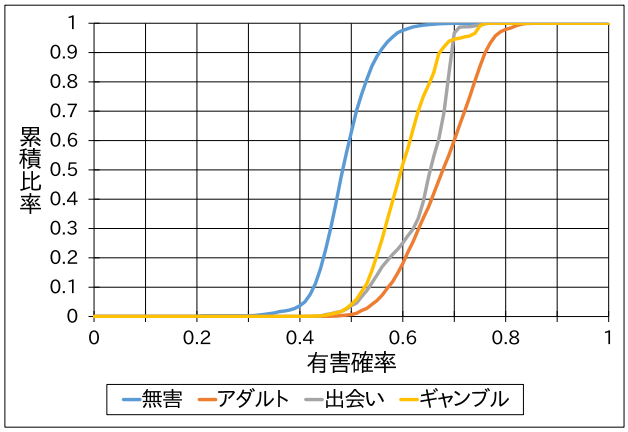


図 13 HTML 要素による判定における有害確率の累積比率
Fig. 13 The cumulative percentage of harmful probability in the judgement by the HTML elements on the webpage.

カテゴリごとのテキストによる判定における有害確率の累積比率を**図 12** に，HTML 要素による判定における有害確率の累積比率を**図 13** に示す．これらはグラフの傾きが大きい箇所ほど，その有害確率と算出された Web ページ

表 2 関連研究との判定精度の比較
Table 2 Comparison of accuracy with relevant studies.

	判定アルゴリズム	共起	有害サイトの定義	偽陽性率 [%]	偽陰性率 [%]	適合率 [%]	再現率 [%]	F 値
提案	BF (Robinson)	無	アダルト・出会い・ギャンブル	4.01	4.48	95.97	95.52	0.957
[4]	ベクトル空間モデル	無	アダルト	6	29	-	-	-
[12]	SVM	無	不法・主張・アダルト・グロテスク・未承諾広告	-	-	78.1	70.0	0.738
[13]	BF (Robinson)	無	アダルト・出会い・風俗	0.00	2.93	-	-	-
	BF (Robinson)	有		1.20	2.27	-	-	-
	BF (Robinson-Fisher)	無		0.40	0.07	-	-	-
	BF (Robinson-Fisher)	有		1.40	0.00	-	-	-
[14]	BF (Robinson)	無	不明	-	-	95.1	82.6	0.884
	BF (Robinson)	有		-	-	90.3	86.4	0.883
[15]	BF (Robinson-Fisher)	無	出会い	-	-	-	-	0.8503
	BF (Robinson-Fisher)	有		-	-	-	-	0.9575

が多いことを表す．判定精度の良いテキストによる判定の方が，HTML 要素による判定に比べ，無害サイトと有害サイトの各カテゴリのグラフの距離が離れていることが確認できる．また，ギャンブルカテゴリは 3 カテゴリの中で，最も無害サイトに近いカテゴリであることが確認できる．

フィルタリングにおける判定精度について，本研究と関連研究との間で比較した結果を**表 2** に示す．数値の記載がない項目は論文中において公開されていないものであり，判定アルゴリズムにおける BF はベイジアンフィルタの略である．各研究における有害サイトの定義や公開されている項目が様々であり，単純な数値の比較には注意を要するが，公開されている項目の範囲で比較を行うと，井ノ上ら [4]，池田ら [12]，吉村ら [14] に対しては本研究の判定精度が上回り，中村ら [15] に対しては両研究で同程度の判定精度となり，菊池ら [13] に対しては本研究の判定精度が下回る結果となった．菊池らに対して判定精度が下回った理由として，図 12 ・図 13 の有害確率の累積比率において他のカテゴリに比べ有害確率の低い傾向にあるギャンブルカテゴリを菊池らは有害サイトとして定義していないことがあげられる．本研究において菊池らと同様の有害サイトの定義を使用した場合，判定精度が向上すると考えられる．

5.3.2 処理時間の分布

HTML ソースの取得から閲覧制限の決定までを処理時間とし，有害サイトと無害サイトの合計 19,000 ページに対し計測を行った．計測結果である処理時間の分布を**図 14** に示す．テキストと HTML 要素による判定を複合させた手法の場合，全体の 71.7%の Web ページに対して 50 ms 以内で処理が終了し，全体の 94.8%の Web ページに対して 0.2 s 以内で処理が終了した．この処理時間であれば，被管理者はストレスをほとんど感じることなくフィルタリングシステムを通じてブラウジングを行えると考えられる．また，1 秒以内で処理が終了するものまで拡大すると，全体

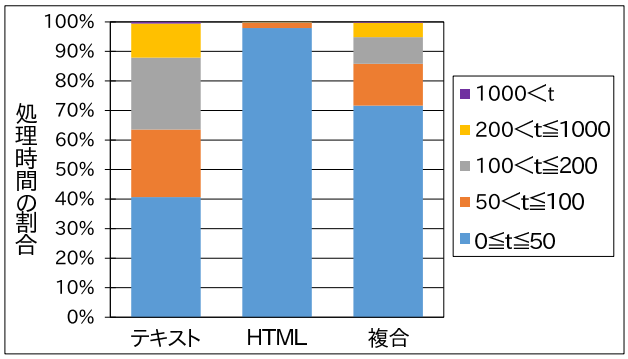


図 14 各判定手法における処理時間の分布（t[ms]）
Fig. 14 Distribution of execution time of each judgement method (t[ms]).

の 99.7%の Web ページとなった．しかし，残りの 0.3%の Web ページに対しては，1 秒を超える時間を処理に要する結果となった．これは被管理者がフィルタリングシステムを通じてブラウジングを行ううえで，ストレスを感じる処理時間だと考えられる．なお，19,000 ページ中 9,474 ページが HTML 要素による判定のみで閲覧制限を決定しており，テキストによる判定と比較すると，50 ms 以内で処理が終了する割合は 1.76 倍となり，処理時間の短縮を確認できた．

5.3.3 DOM ノード非表示化の実証

Web ページの一部に有害と判定されるコンテンツが含まれている場合，DOM ノード非表示化を用いることで，そのようなコンテンツの閲覧を制限できるかを実験した．**図 15** は検索サイトにおける「地方」の検索結果ページであり，ギャンブルカテゴリに属する複数の Web サイトが検索結果に表示されている．これに対し DOM ノード非表示化を行った結果が**図 16** であり，図中の赤枠で囲まれていたギャンブルカテゴリに属するコンテンツの閲覧が適切に制限されることを確認した．また，**図 17** はスポーツ

図 15 検索結果ページに含まれるギャンブルカテゴリに属する DOM ノード

Fig. 15 DOM nodes of the gambling category which are included in the search results page.

図 16 検索結果ページにおけるギャンブルカテゴリに属する DOM ノードの非表示化

Fig. 16 A result of hiding DOM nodes of the gambling category which are included in the search results page.

図 17 無害サイト上に含まれるギャンブルカテゴリに属する DOM ノード

Fig. 17 DOM nodes of the gambling category which are included in the harmless website.

ニュースサイトであり，Web ページ上にギャンブルカテゴリの要素を含んだコンテンツが表示されている．これに対し DOM ノード非表示化を行った結果が**図 18** であり，図中の赤枠で囲まれた公営競技に関するニュースとアクセスランキングの閲覧が制限されることを確認した．

図 18 無害サイトにおけるギャンブルカテゴリに属する DOM ノードの非表示化

Fig. 18 A result of hiding DOM nodes of the gambling category which are included in the harmless website.

アクセスランキングの非表示化については，一部含まれている無害なコンテンツを過剰に閲覧制限している点と，順位部分のみ非表示化されたため外観がやや不自然になっている点が課題としてあげられる．DOM ノードに含まれるトークンの異なり数，DOM ノードの面積に加え，Web ページ内の有害確率の高いトークンの偏りなどに注目し，DOM ノード非表示化の手法を改善していきたい．

5.3.4 既存システムとの比較

既存システムの URL ブラックリストに掲載されていない新規に作成された有害サイトや，被管理者によって試行される可能性のある既存システムに対する回避手法を用いて作成された有害サイトを提案システムにより閲覧が制限されるかを実験した．新規有害サイトとして，データセットとは別に 4 ページを独自に収集した．これらの Web ページは Trend Micro Site Safety Center では有害なカテゴリに分類されておらず，目視により有害サイトと判断した．また，通常アクセスした場合は閲覧の制限が可能な有害な HTTP サイトと HTTPS サイトに対し，既存システムに対する回避手法としてあげられる Web アーカイブ（Wayback Machine [32] を使用）経由によるアクセス，検索キャッシュ（Google キャッシュを使用）経由によるアクセス，Web 翻訳（Google 翻訳を使用）経由によるアクセスにより有害サイトを作成した．これらの有害サイトに対して提案システム，市販のフィルタリングソフトである i-フィルター 6.0 [22]，InterSafe Personal Ver.2.3 [1]，ブラウザ拡張機能を用いたフィルタリングシステムである Blocksi [2] を用いて，閲覧の制限を行った結果を**表 3** に示す．表中の「○」は閲覧制限の成功，「△」はカテゴリ指定による閲覧制限の成功，「×」は閲覧制限の失敗を表す．

提案システムの場合，新規サイトに対して全ページの閲覧が制限された．また，アーカイブ，キャッシュ経由の場合，経由前のプロトコルにかかわらず閲覧が制限された．しかし，Web 翻訳経由に関しては閲覧が制限されなかった．これは iframe 内に翻訳後のコンテンツが表示されてい

表 4 日本語と英語で記述された有害サイトの比較
Table 4 Comparison of harmful Web sites written in Japanese and English.

	日本語有害サイト				英語有害サイト			
	アダルト	出会い	ギャンブル	平均	アダルト	出会い	ギャンブル	平均
HTML ソースのバイト数 [KB]	72.46	23.37	42.04	45.96	104.8	43.09	72.76	73.55
テキストトークンの異なり数	420	305	293	339	446	330	333	370
HTML トークンの異なり数	411	192	311	305	941	454	638	678

表 3 既存システムとの比較
Table 3 Comparison with existing systems.

	経由前	経由後	提案	[22]	[1]	[2]
新規 1	HTTP	-	○	○	×	×
新規 2	HTTP	-	○	○	×	×
新規 3	HTTP	-	○	×	×	×
新規 4	HTTP	-	○	×	×	×
アーカイブ	HTTP	HTTPS	○	△	×	×
アーカイブ	HTTPS	HTTPS	○	△	×	×
キャッシュ	HTTP	HTTP	○	○	△	×
キャッシュ	HTTPS	HTTPS	○	×	×	×
Web 翻訳	HTTP	HTTPS	×	△	×	×
Web 翻訳	HTTPS	HTTPS	×	△	×	×

るため，HTML ソースの取得によって iframe 内の有害なコンテンツを取得できないことが原因であり，この問題に対する考察を 6.2 節で述べる．i-フィルターの場合，新規サイトに対してはコンテンツフィルタリングにより閲覧が制限されたページとされないページが混在した．アーカイブ，Web 翻訳経由に関しては，それぞれアーカイブ，Web 翻訳全体の閲覧をカテゴリ指定により制限するように設定できるが，無害サイトに対するアーカイブや Web 翻訳の過剰な閲覧制限につながっている．なお，キャッシュ経由に関しては，HTTPS サイトの場合，閲覧が制限されなかった．InterSafe Personal の場合，URL を用いたブラックリストによる判定のみのため，新規サイトに対してはいずれも閲覧が制限されなかった．キャッシュ経由に関しては，HTTP サイトの場合であれば，キャッシュ全体の閲覧をカテゴリ指定により制限するように設定できるが，無害サイトに対するキャッシュの過剰な閲覧制限につながっている．また，経由後が HTTPS サイトの場合はいずれも閲覧が制限されなかった．Blocksi の場合，URL を用いたブラックリストによる判定のみであり，回避手法への対策も講じられていないため，すべての有害サイトに対して閲覧が制限されなかった．

5.4 英語で記述された Web ページへの適用可能性

提案システムにおける有害サイトの判定手法が英語で記述された Web ページへ適用可能であるか調査するため，日本語で記述された有害サイトと，英語で記述された有害サイトのコンテンツを比較する．比較に使用する英語有害サイトは，アダルト，出会い，ギャンブルの 3 カテゴリについて，検索サイトを通じて，それぞれ 300 ページを独自に収集した．なお，これらの全 Web ページは日本語有害サイトと同様，Trend Micro Site Safety Center において各当該カテゴリに分類されている．英語サイトにおけるテキストトークンについては，Web ページ上に表示されるテキストに対し，大文字を小文字にすべて置換した後，空白や記号を区切り文字として分割し，テキストトークンを生成する．なお，2 字以下のテキストトークンは冠詞や前置詞のように単独では意味をなさない可能性が高いため除外する．

比較結果を**表 4** に示す．日本語有害サイトと比較すると，HTML ソースのバイト数と HTML トークンの異なり数については英語有害サイトが大きく上回り，テキストトークンについては同程度の異なり数が含まれる結果となった．5.2.2 項におけるトークンの異なり数に関する予備実験の結果をふまえると，英語有害サイト上には HTML トークン，テキストトークンともに，判定に十分な異なり数のトークンが含まれている．そのため，英語で記述された Web ページに対しても，テキストトークンの生成手法を変更することで，同様の判定手法を適用し，判定精度を確保できる見込みがある．

6. 問題点と対策

6.1 ブラウザ拡張機能を用いた実装における問題

ブラウザ拡張機能としてフィルタリングシステムを実装することで発生する問題が 2 つ存在する．1 つは，誰でも容易に無効化，もしくはアンインストールできることである．そこで，提案システムでは，スタイルシートを適用しアドオンマネージャ上から提案システムの表示を消し，拡張機能のインストールを気づかれないようにしている．また，提案システムの各機能を使用するにあたりパスワード認証を実装している．もう 1 つは，ブラウザ依存となることである．現時点では，Firefox 拡張機能の開発に使用される専用モジュールの代替により，Google Chrome で使用される拡張機能として同様の拡張機能を開発することは可能だと考えられる．その一方，Firefox 拡張機能と Chrome 拡張機能は将来的に互換性を持つ予定である [33]．また，2016 年夏に利用可能となる予定の Microsoft Edge の拡張機能は Firefox 拡張機能を利用できる予定であり [34]，将

来的には拡張機能のブラウザ依存は解消される見込みにある．

6.2 iframe を利用して表示されるコンテンツ

iframe を利用して表示されるブラウジングコンテキストの中には，有害なコンテンツが含まれる場合がある．しかし，提案システムではトップレベルブラウジングコンテキストのみを有害サイトの判定に使用しており，iframe 内のコンテンツを判定に使用していない．これは Firefox 拡張機能にクロスドメイン制約が存在し [35]，クロスドメインの場合，Firefox 拡張機能から iframe 内のコンテンツを取得できないためである．

この問題の対策として，iframe の src 属性で設定された URL へ Firefox 拡張機能からリクエストを再送することにより，iframe 内のコンテンツを取得する手法が考えられる．この手法を導入することで，提案システムにおいて iframe 内のコンテンツも有害サイトの判定に使用することができる．ただし，この手法では問題を解決できない事例が 2 つ考えられる．1 つ目は，毎回表示内容が変更される広告のように，同じ URL へのリクエストに対して，サーバがそれぞれ異なるレスポンスを返す場合．2 つ目は，翻訳中のページを経由してから翻訳後のコンテンツへ動的に変更される Web 翻訳サイトのように，iframe 内のコンテンツが動的に変化する場合である．これらの場合，ブラウザ上に表示されているコンテンツと，リクエストの再送によって取得されるコンテンツに違いが生じ，誤判定につながる可能性がある．上記の対策により取得可能な iframe 内のコンテンツを有害サイトの判定に使用するように実装すること，および上記の対策により取得できないコンテンツの新たな取得手法を考案することは今後の課題である．

6.3 攻撃の可能性

提案システムがかかえる別の問題として，クライアントサイドで動作する機械学習を用いた Web ページの分類器に対する攻撃の可能性を Liang らは指摘している [36]．その手法は学習データや分類アルゴリズムを分析し，有害確率を低下させる素性を Web ページに加えるというものであり，ソースコードの難読化により分類アルゴリズムを特定されないようにすることを暫定的な対策としてあげている．しかし，Firefox 拡張機能はオープンソースソフトウェアであるうえ，addons.mozilla.org 上に公開する場合はソースコードを難読化できないため，学習データや分類アルゴリズムを公開せざるをえず，この対策を適用できない．また，Liang らは別の対策として，素性をハッシュ化することで特定されにくくすることをあげている．この対策については，学習データのファイルサイズ増加にともなう読み込み時間の増加や，ハッシュ化に要する処理時間が小さければ導入を検討したい．

7. おわりに

本稿では，ブラウザ拡張機能を用いることで，Web ブラウザレベルですべての処理を行う動的コンテンツフィルタリングシステムを提案した．評価実験の結果，Web ページ上のテキストを使用した判定と HTML 要素を使用した判定を組み合わせることにより，判定精度として偽陽性率 4.01%，偽陰性率 4.48%，処理時間として全体の 71.7%の Web ページに対して 50 ms 以内，全体の 94.8%の Web ページに対して 0.2 s 以内，全体の 99.7%の Web ページに対して 1 s 以内でフィルタリングが行われることを確認した．また，DOM ノードの非表示化により，Web ページ上の有害なコンテンツのみを閲覧制限できることを示した．そして，既存システムでは閲覧を制限できない可能性のある，新規有害サイトやアーカイブ，キャッシュ経由により作成された有害サイトに対する閲覧制限において優位性を示した．今後は各章であげた課題を解消することで，より実用的なフィルタリングシステムを目指していきたい．

参考文献

[1] アルプスシステムインテグレーション株式会社：InterSafe Personal，アルプスシステムインテグレーション株式会社（オンライン），入手先 ⟨http://www.alsi.co.jp/security/isp/⟩（参照 2016-07-01）．

[2] Blocksi: Blocksi, addons.mozilla.org (online), available from ⟨https://addons.mozilla.org/en-US/firefox/addon/blocksi/⟩ (accessed 2016-07-01).

[3] webfiltering: WebFilter Pro – The web content filtering addon!, addons.mozilla.org (online), available from ⟨https://addons.mozilla.org/en-US/firefox/addon/webfilter/⟩ (accessed 2016-07-01).

[4] 井ノ上直己，帆足啓一郎，橋本和夫：文書自動分類手法を用いた有害情報フィルタリングソフトの開発，電子情報通信学会論文誌，Vol.J84-D2, No.6, pp.1158–1166 (2001).

[5] 大井彩香，寺田 実，丸山一貴：Web ページの分類と閲覧時間を利用したコンテンツフィルタリング，第 10 回情報科学技術フォーラム講演論文集，No.4, pp.137–140 (2011).

[6] 上田達巳，高井昌彰：子どもの保護を目的とした Web アクセス監視支援システム，情報処理学会論文誌，Vol.49, No.3, pp.1155–1162 (2008).

[7] Farrell, S. and Tschofenig, H.: RFC 7258 – Pervasive Monitoring Is an Attack, Internet Engineering Task Force (online), available from ⟨https://tools.ietf.org/html/rfc7258⟩ (accessed 2016-07-01).

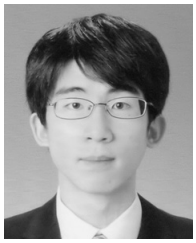
[8] Nottingham, M.: Securing the Web, W3C Technical Architecture Group (online), available from ⟨https://w3ctag.github.io/web-https/⟩ (accessed 2016-07-01).

[9] Souders, S.: HTTP Archive – Trends, httparchive.org (online), available from ⟨http://httparchive.org/trends.php⟩ (accessed 2016-07-01).

[10] Aas, J.: Leaving Beta, New Sponsors – Let's Encrypt – Free SSL/TLS Certificates, Internet Security Research Group (online), available from ⟨https://letsencrypt.org/2016/04/12/leaving-beta-new-sponsors.html⟩ (accessed 2016-07-01).

[11] デジタルアーツ株式会社：「i-FILTER」URL フィルタリ

ング・Web サービス制御，デジタルアーツ株式会社（オンライン），入手先 ⟨http://www.daj.jp/bs/i-filter/⟩（参照 2016-07-01）．
[12] 池田和史，柳原　正，服部　元，松本一則，小野智弘，滝嶋康弘：HTML 要素に基づく有害サイト検出手法，情報処理学会論文誌，Vol.52, No.8, pp.2474–2483 (2011).
[13] 菊池琢弥，内海　彰：語の共起情報に基づく有害サイトフィルタリング手法，第 9 回情報科学技術フォーラム講演論文集，No.2, pp.1–6 (2010).
[14] 吉村卓也，藤井雄太郎，伊藤孝行：Robinson 型判定手法を用いた単語共起フィルタの検証，第 10 回情報科学技術フォーラム講演論文集，No.2, pp.85–90 (2011).
[15] 中村健二，田中成典，山本雄平，安彦智史：共起関係の抽出範囲を考慮した有害情報フィルタリング手法，情報処理学会論文誌，Vol.54, No.2, pp.571–584 (2013).
[16] Likarish, P., Jung, E., Dunbar, D., Hansen, T.E. and Hourcade, J.P.: B-APT: Bayesian Anti-Phishing Toolbar, *Proc. 2008 IEEE International Conference on Communications* (2008).
[17] WOT Services: Web of Trust, WOT: Website Reputation Ratings, addons.mozilla.org (online), available from ⟨https://addons.mozilla.org/en-US/firefox/addon/wot-safe-browsing-tool/⟩ (accessed 2016-07-01).
[18] Wips.com s.r.o.: BlockSite, addons.mozilla.org (online), available from ⟨https://addons.mozilla.org/en-US/firefox/addon/blocksite/⟩ (accessed 2016-07-01).
[19] Anderson, J.: LeechBlock, addons.mozilla.org (online), available from ⟨https://addons.mozilla.org/en-US/firefox/addon/leechblock/⟩ (accessed 2016-07-01).
[20] Inspired Effect: FoxFilter, addons.mozilla.org (online), available from ⟨https://addons.mozilla.org/en-US/firefox/addon/foxfilter/⟩ (accessed 2016-07-01).
[21] Paolini, H.: ProCon Latte Content Filter, addons.mozilla.org (online), available from ⟨https://addons.mozilla.org/en-US/firefox/addon/procon-latte/⟩ (accessed 2016-07-01).
[22] デジタルアーツ株式会社：有害サイトフィルタリングソフト「i-フィルター」，デジタルアーツ株式会社（オンライン），入手先 ⟨http://www.daj.jp/cs/⟩（参照 2016-07-01）．
[23] Microsoft: Microsoft アカウント | ファミリー，Microsoft（オンライン），入手先 ⟨https://account.microsoft.com/family/about⟩（参照 2016-07-01）．
[24] Takahashi, K.: Harmful Web Contents Cleaner, addons.mozilla.org (online), available from ⟨https://addons.mozilla.org/en-us/firefox/addon/harmful-web-contents-cleaner1/⟩ (accessed 2016-07-01).
[25] 工藤　拓：TinySegmenter:Javascript だけで書かれたコンパクトな分かち書きソフトウェア，ChaSen.org（オンライン），入手先 ⟨http://chasen.org/~taku/software/TinySegmenter/⟩（参照 2016-07-01）．
[26] Robinson, G.: Spam Detection, Gary Robinson's Rants (online), available from ⟨http://radio-weblogs.com/0101454/stories/2002/09/16/spamDetection.html⟩ (accessed 2016-07-01).
[27] ごうだまりぽ：CustomBlocker, Chrome ウェブストア（オンライン），入手先 ⟨https://chrome.google.com/webstore/detail/customblocker/elnfhbjabfcepfnaeoehffgmifcfjlha⟩（参照 2016-07-01）．
[28] Palant, W.: Adblock Plus, addons.mozilla.org (online), available from ⟨https://addons.mozilla.org/en-us/firefox/addon/adblock-plus/⟩ (accessed 2016-07-01).
[29] 一般財団法人インターネット協会：レイティング基準 SafetyOnline3.1（オンライン），入手先 ⟨http://www.iajapan.org/filtering/press/20081017-SafetyOnline3_1.pdf⟩（参照 2016-07-01）．
[30] トレンドマイクロ株式会社：Trend Micro Site Safety Center，トレンドマイクロ株式会社（オンライン），入手先 ⟨https://global.sitesafety.trendmicro.com/⟩（参照 2016-07-01）．
[31] エヌ・ティ・ティレゾナント株式会社：goo カテゴリー検索，エヌ・ティ・ティレゾナント株式会社（オンライン），入手先 ⟨http://category.goo.ne.jp/⟩（参照 2016-07-01）．
[32] Kahle, B.: Internet Archive: Wayback Machine, The Internet Archive (online), available from ⟨https://archive.org/web/⟩ (accessed 2016-07-01).
[33] Needham, K.: The Future of Developing Firefox Add-ons, Mozilla (online), available from ⟨https://blog.mozilla.org/addons/2015/08/21/the-future-of-developing-firefox-add-ons/⟩ (accessed 2016-07-01).
[34] Verma, A.: Microsoft Edge Can "Steal" Extensions from Firefox and Chrome, Fossbytes (online), available from ⟨http://fossbytes.com/microsoft-edge-can-steal-extensions-from-firefox-and-chrome/⟩ (accessed 2016-07-01).
[35] Mozilla Developer Network: Cross-domain Content Scripts – Mozilla – MDN (online), available from ⟨https://developer.mozilla.org/en-US/Add-ons/SDK/Guides/Content_Scripts/Cross_Domain_Content_Scripts⟩ (accessed 2016-07-01).
[36] Liang, B., Su, M., You, W., Shi, W. and Yang, G.: Cracking Classifiers for Evasion: A Case Study on the Google's Phishing Pages Filter, *Proc. 25th International Conference on World Wide Web*, pp.345–356 (2016).



高橋 研介

2015 年電気通信大学情報理工学部総合情報学科卒業．現在，同大学大学院情報理工学研究科総合情報学専攻修士課程在学中．

市野 将嗣　（正会員）

2003 年早稲田大学理工学部電子・情報通信学科卒業．2008 年同大学大学院理工学研究科博士課程修了．2007 年日本学術振興会特別研究員．2009 年早稲田大学大学院基幹理工学研究科研究助手．2010 年同大学メディアネットワークセンター助手．2011 年電気通信大学大学院情報理工学研究科助教．2016 年同大学院情報理工学研究科准教授．バイオメトリクス，ネットワークセキュリティに関する研究に従事．博士（工学）．電子情報通信学会会員．

大山 恵弘 （正会員）

2001 年東京大学大学院理学系研究科情報科学専攻修了．科学技術振興事業団研究員，東京大学大学院情報理工学系研究科助手，電気通信大学大学院情報理工学研究科准教授を経て，2016 年より筑波大学システム情報系准教授．博士（理学）．システムソフトウェア，ソフトウェアセキュリティに関する研究に従事．