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の提案

高橋 研介 †　　高橋 一志 †　　大山 恵弘 †

† 電気通信大学
182-8585 東京都調布市 調布ケ丘 1 丁目 5-1
takaken@ol.inf.uec.ac.jp　　{kazushi, oyama}@inf.uec.ac.jp

あらまし 有害な Web サイトへの対策の一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ため，通信の盗聴の危険性が生じる．本稿では，ブラウザ拡張機能を用いることで，Web ブラウザレベルで全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する．

Proposal of Dynamic Content Filtering System Using Browser Extensions

Kensuke Takahashi†　　Kazushi Tahakashi†　　Yoshihiro Oyama†

† The University of Electro-Communications
1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, JAPAN
takaken@ol.inf.uec.ac.jp　　{kazushi, oyama}@inf.uec.ac.jp

Abstract Filtering systems are one of countermeasures against harmful Web sites. However, the risk of wiretapping occurs, 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, because those systems must decrypt HTTPS traffic for filtering HTTPS sites. In this paper, we propose a content filtering system using browser extensions, which dynamically determines whether a Web page is harmful based on the text on the Web page using Bayesian filtering. All these processes are executed at the browser level.

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一つとして，有害サイトへのアクセス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挙げられる．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 ページに対してアクセスを制限することができない．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 [1] [2] [3].

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 [4] [5]. Web ページの URL や Web ページ上のテキストを使用してアクセス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アクセスを制限することができない．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 [6] が，通信の盗聴の危険性が生じる．

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．本システムは全ての処理を 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する．これにより，Web ページ上の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共に，通信途中で復号することなく HTTPS サイトのフィルタリングを可能にする．

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の拡張機能を紹介する．3 章では提案システムの概要や利点を説明し，4 章では各処理の実装方法を説明する．5 章では提案システムを使用した，フィルタリングの判定精度や処理時間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改善点について述べる．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 [1] 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 ページへのアクセス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 [2] 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，アクセス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 サイトのアクセスを制限することはできない．上田ら [3] 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アクセスは制限されない．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することはできない．

本研究と同様に，ベイジアンフィルタ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．菊池ら [7]，吉村ら [8] 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，フィルタリング手法の提案と判定精度の評価のみが行われており，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行われていない．

Likarish ら [9] 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．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．

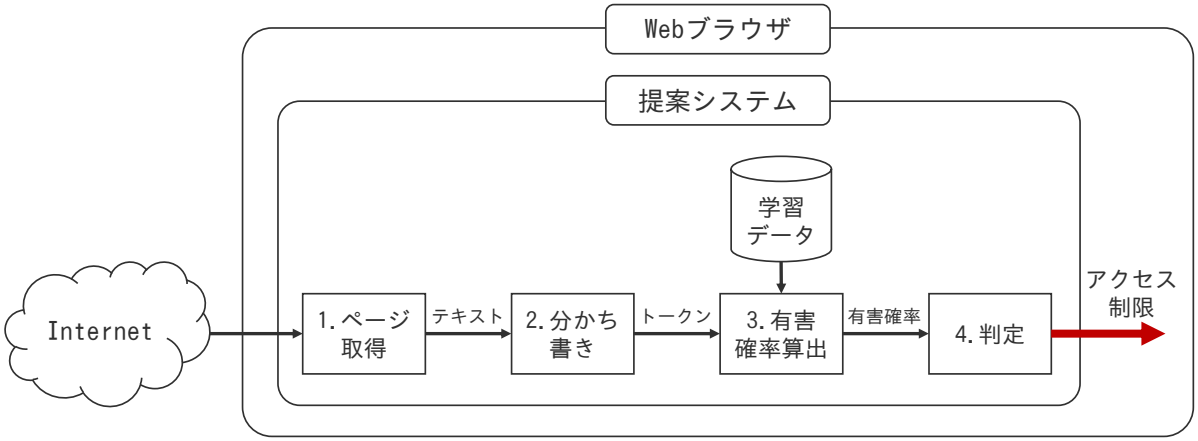


図 1: 提案システムの概略図

Web of Trust [10] 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 ページへのアクセスを制限する．この手法の問題点として，評価が行われていないためアクセス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挙げられる．

BlockSite [11]，LeechBlock [12] 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アクセスを制限する．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アクセスを制限することはできない．

FoxFilter [13]，ProCon Latte Content Filter [14] 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，アクセスを制限する．問題点として，ブラックワードが 1 箇所に含まれているだけでアクセスを制限するため，過度にアクセスを制限する可能性が高いことが挙げられる．

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することで，アクセスを制限する．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．図 1 に提案システムの概略図を示す．

提案システムの利用目的は一般的なフィルタリングシステムと同様に，一般家庭におけるペアレンタルコントロールや，企業などにおけるアクセス制限などを想定している．提案システムは，以下の流れで動作を行う．

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 BODY タグの要素を取得する．
2. 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，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，分かち書きされたトークンを使用して，ベイジアンフィルタにより Web ページの有害確率を算出する．
4. 有害確率が閾値を超えているかどうかを判定し，閾値を超えていた場合アクセスを制限する．

また，フィルタリングシステムをブラウザ拡張機能により実装することによる利点として，以下の 3 点が挙げられる．

- Web ブラウザレベルでフィルタリングが行われることにより，通信途中で復号することなく HTTPS サイトのフィルタリングを行うことができる．
- インストール手順が簡潔であり，導入容易性が高い．

「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

↓ (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 2: 分かち書きの例

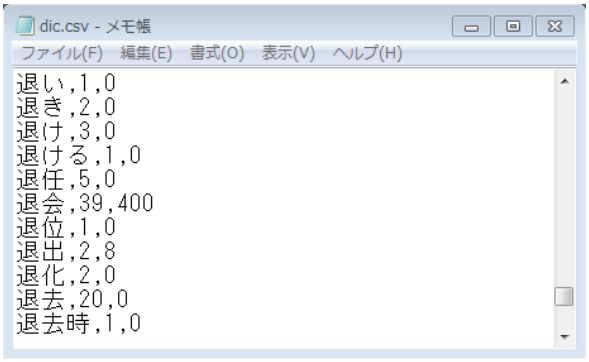


図 3: 学習データファイルの内容

- 同一の Web ブラウザであれば，クロスプラットフォームで使用できる．

なお，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の作成を行う必要がある．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，提案システムと同様の手法でトークンへの分割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全ての 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．

4 実装

4.1 ページの取得

Add-on SDK の tabs モジュールを使用してブラウザのタブの動作を監視することにより，ページの遷移を検知する．ページの遷移を検知すると，getElementsByTagName() メソッドにより BODY タグの要素を取得する．その後，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，BODY タグが省略されており，BODY タグの要素を取得できない場合，HTML タグの要素から HEAD タグの要素を除去したものを代わりに使用する．

4.2 分かち書き

BODY タグの要素に含まれるテキストを，分かち書きによりベイジアンフィルタで使用するためのトークン (形態素) に分割する．図 2 に分かち書きの例を示す．分かち書き後に登場するスラッシュはトークンの境界を表す．ベイジアンフィルタで使用するトークンは，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 文字以下のトークンに関しては，助詞のように単独では意味を成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 [15] を用いた．

4.3 有害確率の算出

有害サイトの判定のため，分かち書きにより分割されたトークンの集合と，拡張機能内に同梱されている学習データファイルを基にして，Robinson 方式のベイジアンフィルタ [16] を用いることで Web ページの有害確率を算出する．学習データファイルには図 3 のように，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 CSV 形式で記述されている．なお，学習データファイルの 1 行目には，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が記述されている．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．

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

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 [16] 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，及び非有害性 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．

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 の範囲の値を取り，有害確率が閾値を超えた場合，その Web ページを有害サイトとして判定する．本研究では既存手法 [16] に準じて閾値を 0.5 に設定した．なお，分かち書きの結果トークンが 0 個だった場合，有害確率を 0.5 とし，無害サイトとして判定する．

図 4: アクセス制限画面

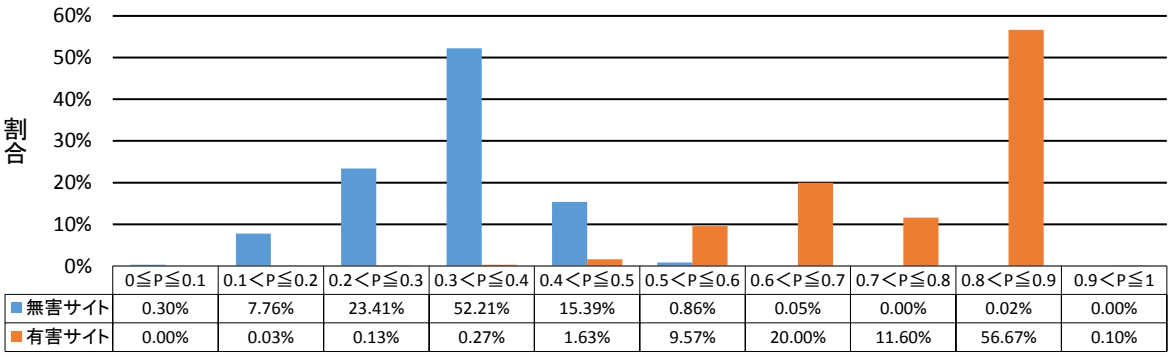
有害サイトとして判定された場合は，Web ページの BODY の innerHTML プロパティを，図 4 のように，アクセスが制限されたことを伝える内容，URL 及び「戻る」ボタンが配置されたページに書き換えることでアクセスを制限する．

5 評価実験

5.1 実験方法

評価に用いるデータセットとして，有害サイト 3000 ページ，無害サイト 10000 ページを用意した．有害サイトには，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 ページを使用し，クローリング及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 [17] において，出会いカテゴリに分類されている．無害サイトには，goo カテゴリー検索 [18] に登録されている Web ページの中で取得することができた約 29 万ページに対しランダムサンプリングを行い，10000 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 600 ページと無害サイト 2000 ページをテストデータ，残りの有害サイト 2400 ページと無害サイト 8000 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 39.0，OS が Microsoft Windows 7 Professional SP1 (64bit)，RAM が 8GB，CPU が Intel Core i7-2600 の環境で行われたが，ブラウザが Android 版 Firefox 40.0，OS が Android 4.4.4 及び，ブラウザが Firefox ESR 38.2.0，OS が CentOS 6.7 (64bit) の環境においても正常に動作することを確認した．



	0≦P≦0.1	0.1<P≦0.2	0.2<P≦0.3	0.3<P≦0.4	0.4<P≦0.5	0.5<P≦0.6	0.6<P≦0.7	0.7<P≦0.8	0.8<P≦0.9	0.9<P≦1
無害サイト	0.30%	7.76%	23.41%	52.21%	15.39%	0.86%	0.05%	0.00%	0.02%	0.00%
有害サイト	0.00%	0.03%	0.13%	0.27%	1.63%	9.57%	20.00%	11.60%	56.67%	0.10%

図 5: 有害確率の分布

表 1: 判定精度の評価結果

真陽性率	97.93%
真陰性率	99.07%
偽陽性率	0.93%
偽陰性率	2.07%
正解率	98.50%
適合率	99.06%
再現率	97.93%
F 値	0.985

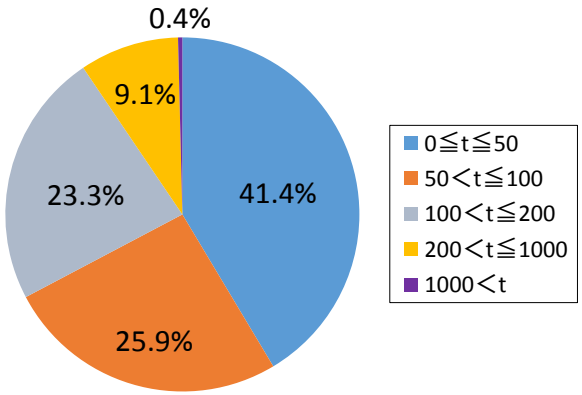


図 6: 処理時間の分布 (t[ms])

5.2 判定精度

フィルタリングにおける判定精度の評価結果を表 1 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は，それぞれ約 1%，約 2%となった．偽陽性率，偽陰性率の上昇に繋がった要因として，Flash や画像をメインに作成されている Web ページや，入口ページのように，テキスト量が少ない Web ページでは含まれるトークンが少ないため，正確に判定できなかったことが挙げられる．また，偽陽性率の上昇に繋がった要因として，出会い系サイトにコンテンツがやや類似した，恋愛や結婚の話題を扱う Web ページが無害サイトの中に存在したことが挙げられる．

本実験と同様に，有害サイトに出会い系サイトを用いて，Robinson 方式のベイジアンフィルタを使用した菊池らが行った評価実験 [7] では，偽陽性率が 0%，偽陰性率が 0.80%となっている．菊池らが行った評価実験に比べ判定精度が低下した要因として，使用したデータセットが異なることや，トークンの分割手法の違いが考えられる．

図 5 は判定精度の評価における無害サイトと有害サイトの有害確率の分布を示したものである．無害サイトの有害確率は主に 0.3 から 0.4 の間に集中しており，有害サイトの有害確率は主に 0.8 から 0.9 の間に集中していることが確認できる．

5.3 判定における処理時間

BODY タグの要素の取得から有害確率の算出までを処理時間とし，有害サイトと無害サイトの合計 13000 ページに対して計測を行った．計測結果である判定における処理時間の分布を図 6 に示す．結果として，全体の 90.5%の Web ページに対して 0.2 秒以内で処理が終了した．この処理時間であれば，ユーザ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使用できると考えられる．また，1 秒以内で処理が終了するものまで拡大すると，全体の 99.6%の Web ページとなった．しかし，残りの 0.4%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．これはユーザがフィルタリングシステムを使用する上で，ストレス

を感じる処理時間だと考えられる．処理時間が増加した要因として，Webページ上のテキスト量が多く，分かち書き及び有害確率の算出に要する時間が増加したことが挙げられる．なお，全体の平均処理時間は0.10秒，最大処理時間は7.90秒であった．

6　提案システムの改善点

6.1　拡張機能特有の改善点

ブラウザ拡張機能として実装されたフィルタリングシステムには，特有の改善点が存在する．

一つは，フィルタリングを行う拡張機能を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．改善策として，パスワード認証により拡張機能の管理をコントロールすることが考えられる．Webブラウザによっては，拡張機能自身が無効化やアンインストールをキャンセルすることが可能なため，その過程においてパスワード認証を加える手法である．もしくは，過去に提案した拡張機能のステルス化 [19] を行うことで，拡張機能のインストール自体を気付かれないようにする手法が考えられる．

もう一つは，フィルタリングシステムがブラウザ依存になることである．改善策として，より多くのブラウザ拡張機能で実装することにより，ブラウザ依存を実質的に解消することが考えられる．今後の課題として，Webブラウザの中でもシェアの高いGoogle ChromeやInternet Explorerの拡張機能による提案システムの実装が挙げられる．

6.2　実装上の改善点

現状の提案システムおける実装上の改善点が複数存在する．

一つ目は，評価実験において見られた，Webページ上のテキスト量が少ないため，判定精度が低下することである．テキストを使用しないで有害サイトを判定する手法として，池田ら [20] はhtml要素を用いる手法を提案している．テキスト量が一定を下回っている場合，テキストを用いる代わりにこのような手法により判定することが改善策として考えられる．

二つ目は，評価実験において見られた，Webページ上のテキスト量が多いため，処理時間が増加することである．提案システムではRobinson方式に従い，全てのトークンを使用して有害確率を算出している．一方，Graham方式のベイジアンフィルタ [21] では，有害確率が0.5から最も離れている15個のトークンを使用して有害確率を算出する．テキスト量が一定を上回る場合，限定した数のトークンを使用して有害確率を算出することが改善策として考えられる．

三つ目は，Webページによって，メニューやフッタなどの部分にWebページのメインコンテンツとは関係性の低いトークンが含まれ，判定精度を低下させている可能性があることである．これは提案システムにおいて，WebページのBODYタグの要素に含まれるテキスト全体を判定に用いることに起因する．中村ら [22] は共起関係に用いるWebページの範囲を制限した上で，単語の共起関係を基にフィルタリングを行う手法を提案している．このように判定に用いるテキストの範囲を制限することで，判定精度が向上すると考えられる．

7　まとめ

本稿では，ブラウザ拡張機能を用いることで，Webブラウザレベルで全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．提案システムを使用した評価実験の結果，判定精度として偽陽性率0.93%，偽陰性率2.07%，処理時間として全体の90.5%のWebページに対して0.2秒以内，全体の99.6%のWebページに対して1秒以内でフィルタリングが行われることを確認した．今後は6章で述べた改善点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くと共に，さらなる実験を行うことで，ベイジアンフィルタに用いるトークンの限定条件や学習データ数の変更，有害サイトのジャンル追加・変更による，判定精度と処理時間の変化を明らかにしたい．

参考文献

[1] 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.

[2] 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.

[3] 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.

[4] S. Farrell, H. Tschofenig. RFC 7258 - Pervasive Monitoring Is an Attack, 2014. https://tools.ietf.org/html/rfc7258

[5] Mark Nottingham. Securing the Web, 2015. https://w3ctag.github.io/web-https/

[6] Web プロキシ機能 | i-FILTER http://www.daj.jp/bs/i-filter/proxy/

[7] 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.

[8] 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.

[9] Peter Likarish, Eunjin Jung, Don Dunbar, Thomas E Hansen, and Juan Pablo Hourcade. B-APT: Bayesian Anti-Phishing Toolbar. *In Proceedings of the 2008 IEEE International Conference on Communications*, 2008.

[10] Web of Trust, WOT https://addons.mozilla.org/en-US/firefox/addon/wot-safe-browsing-tool/

[11] BlockSite https://addons.mozilla.org/en-US/firefox/addon/blocksite/

[12] LeechBlock https://addons.mozilla.org/en-US/firefox/addon/leechblock/

[13] FoxFilter https://addons.mozilla.org/en-US/firefox/addon/foxfilter/

[14] ProCon Latte Content Filter https://addons.mozilla.org/en-US/firefox/addon/procon-latte/

[15] 工藤拓. TinySegmenter. http://chasen.org/~taku/software/TinySegmenter/

[16] Robinson Gray. Spam detection, 2002. http://radio.weblogs.com/0101454/stories/2002/09/16/spamDetection.html

[17] Trend Micro Site Safety Center http://global.sitesafety.trendmicro.com/

[18] goo カテゴリー検索 http://category.goo.ne.jp/

[19] 高橋研介, 高橋一志, 大山恵弘. 悪意を持つ Firefox 拡張機能による攻撃手法とステルス化手法の検討. コンピュータセキュリティシンポジウム 2014 論文集, No.2, pp.362-369, 2014.

[20] 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.

[21] Paul Graham. Hackers and Painters : Big Ideas from the Computer Age, pp.121-129, 2004.

[22] 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.