

AntiSpoiler: ネタバレ防止ブラウザ

中村聡史[†]

近年、様々な分野の新鮮な情報をウェブ経由で即座に手に入れることができるようになった。一方で、こうした情報の氾濫は、ネタバレなどにより視聴や閲覧を楽しみにしているコンテンツの面白さを低減させてしまうという問題がある。本稿では、こうした問題を考慮して、時間ベースでのフィルタリングを行うことにより、面白さを低減させるようなネタバレ情報をフィルタリングする *AntiSpoiler* システムを提案する。本提案システムでは、ユーザの指定したプリファレンスに基づき、コンテンツを動的にフィルタリングする。

AntiSpoiler: An Web Browser to Filter Spoiler

Satoshi Nakamura[†]

Nowadays, we can obtain various kinds of fresh information from the Web. On the other hand, such fresh information sometimes spoils user's enjoyment. This paper proposes a temporal filtering system called the *Anti-Spoiler* system. The system changes filters dynamically based on user-specified preferences and the user's timetable. The system then blocks contents that would spoil the user's enjoyment of a previously unwatched content. The system analyzes a user-requested Web content, and then uses filters to prevent portions of the content being displayed that might spoil user's enjoyment. For example, the system hides the final score of football from the Web content before watching it on TV.

1. はじめに

情報通信技術の発展により、日本に住んでいる人が自宅に居ながらにして、日本から遠く離れた南アフリカで開催されているサッカーワールドカップの日本代表の試合をリアルタイムでテレビ観戦したり、ツール・ド・フランスのレースなどの様子をテレビで観戦したり、アメリカンフットボールの試合結果やチャンピオンズリーグの試合結果、F1のレース結果などをニュースサイトにアクセスして情報を閲覧したりすることが可能となっている。つまり、ある程度大きなイベントであれば、ユーザが望め

ば即座にそうした情報を手に入れることができるようになってきている。また、マイナーなイベントに関する情報であっても、Web2.0 技術の広まりによって誰でも手軽にウェブ上で情報発信できるようになったことで、結果などを収集することが可能となってきている。

一方、そうしたスポーツイベントをテレビ視聴しようと思っても、その中継時間に仕事や用事などの別件がある場合、実際のイベントと同じタイミングではなく、それからしばらく経ってから録画したビデオなどで視聴するということはふつうのコトである。また、チャンネル数の都合等により、生中継されずにある程度遅れて配信されることも多い。ここで、スポーツの試合はシナリオがないものであるため、その試合を視聴する前に結果を知ってしまうと、そのコンテンツ自体の面白みは激減すると言える。そのため、こうした録画した試合を後で視聴することを楽しみにしているユーザの多くは、試合結果に関する情報（ネタバレ情報）に触れないようにするため、積極的に情報遮断を行っている。なお、結果情報に触れてネタバレれることによって、ユーザの面白さを低減させるのはスポーツの試合結果に限った話ではなく、映画や小説、ゲームの結末や、ミステリードラマの犯人の名前など様々である。

ユーザが一般的にネタバレ情報に遭遇するシチュエーションとしては以下のような状況があげられる。

- テレビで別の番組を視聴中に、速報やニュースの形で結果を知ってしまう。
- 新聞や雑誌で対象とは無関係の記事を読んでいる際に、ネタバレ情報が視界に入ってしまう、結果を知ってしまう。
- SNS やブログ、掲示板、ポータルサイトなどウェブページを閲覧中に、ネタバレ情報が視界に入ってしまう、結果を知ってしまう。
- 友人との会話や、赤の他人同士の会話などから不意に結果を知ってしまう。

人の口に戸を立てることは難しく、またテレビ番組のコンテンツはどのようなものが流れてくるか予め予測することができない。また、新聞や雑誌などの場合は、結果が分かってから印刷および配達が行われるため、速報性は高くなく、ある程度コンテンツがカテゴリによって分類されているため、ネタバレの危険性は低いと言える。そこで本研究では、ウェブ上で公開されているコンテンツ内に含まれている情報を、情報遮断の対象とする。

ウェブ上でユーザがネタバレ情報に触れる場としては、報道機関が運営しているニュースサイトや情報を集約するポータルサイト、検索エンジンが返す検索結果、ソーシャルネットワークサービスやレビューサイト、チャットサービスや掲示板サービス、一般ユーザが作成するブログサイトや、より速報性が高い Twitter に代表されるマイクロブログサービスなど様々である。実際にウェブブラウザを利用してウェブコンテン

[†] 京都大学大学院 情報学研究科
Graduate School of Informatics, Kyoto University

ツを閲覧中にネタバレ情報に遭遇してしまうパターンとしては、主に下記のようなものが挙げられる。

- ハイパーリンクのアンカーテキストが、その先のコンテンツがネタバレ情報であることを明記していないため、気づかずにそのコンテンツにアクセスしてしまい、結果を知ってしまう。
- ニュースサイトやポータルサイトなどのトップページを開いた際、すべての記事が短いタイトルで一覧かされているがために、スポーツの結果など速報的な記事（ネタバレ情報）が目に入ってしまう。
- ニュースサイトやブログサイトなどで情報遮断を行っていない無関係のコンテンツを閲覧しているが、そのコンテンツのサイドバーに新着記事として試合結果などのネタバレ情報が表示され、意図せずネタバレに遭遇してしまう。
- 近年、速報性の高いニュースコンテンツも一般検索の検索対象となっている。そのため、無関係のキーワードでウェブ検索を行った際に、検索結果の上位にネタバレ情報が表示されてしまい、試合などの結果を知ってしまう。
- SNS や掲示板、Twitter など対話を楽しむサービスにおいて、無関係な対話を楽しんでいたのだが、他者が良かれと思ってまたは悪意をもってネタバレ情報を漏らしてしまう（実世界での人同士の対話に類似）。

ポータルサイト、検索サービス、コミュニケーションツールなどにおいて、そうしたネタバレ情報は良かれと思って提示されているものであり、情報遮断を行っていないユーザにとっては価値の高いものであることが多い。一方、そうした情報は、その情報遮断を行ったユーザが、情報遮断を行っていないときには積極的に集めたい情報であることも多い。例えば、ワールドカップで日本代表を応援しており、その録画映像を楽しむまで情報遮断をしていたユーザは、そのコンテンツを視聴して日本代表が勝利したことを楽しむと、その楽しみおよび喜びを増幅させるため、ニュースサイトなどを巡回して積極的にそうした情報を収集するであろう。つまり、常時コンテンツをフィルタリングすることが望まれているわけではなく、一時的な情報フィルタリングが必要とされているといえる。

こうした問題を解決するため、我々は過去の研究において、時間ベースでのコンテンツフィルタリング手法を提案した[1]。しかし、過去に提案した手法はローカルプロキシを検閲システムのように利用するものであり、静的なフィルタリングであったため、ユーザ操作によって対話的にコンテンツを変化させることができなかった。また、フィルタリング精度を向上させるために詳細なチェックを行うと、かなりの処理が必要となるなど速度面での問題があり、ユーザにとって不満の多いシステムとなってしまっていた、さらに、Twitterなどのマイクロブログに代表される話し言葉中心のコン

テンツにおいてほとんどのフィルタリングに失敗するなどの問題があった。

そこで本研究では、ネタバレの問題と過去のシステムの問題点を整理し、時間ベースでのネタバレ情報（ここでネタバレ情報とは、試合の結果や映画・小説・ゲームの結末などを意味する）のフィルタリングをブラウザ上で行う手法を提案し、システムを実装する。また、実際にシステムを実運用することにより、手法の可能性について検討を行うとともに、問題点等について考察を行う。なお、本研究で実現するシステムを *AntiSpoiler* と呼ぶ。

2. AntiSpoiler

我々は過去の研究において、ユーザの興味に基づくネタバレ情報の動的なフィルタリングを可能とすることを目的として、ローカルプロキシによってネタバレ情報をユーザから隠す手法を提案してきた[1]。この研究では、ユーザが情報を遮断する対象と設定したコンテンツ（遮断対象コンテンツ）が開始されてから、そのコンテンツ自体をユーザが楽しみ終わるまでをフィルタリング期間とするものである。つまり、スポーツの試合の場合は、試合が開始されてから、録画した試合を視聴し終わるまでがフィルタリング期間の対象となる（図 1-a）。対象が本などの場合は、その本が発売されてから、実際にその本を読み終わるまでがフィルタリング対象となる（図 1-b）。

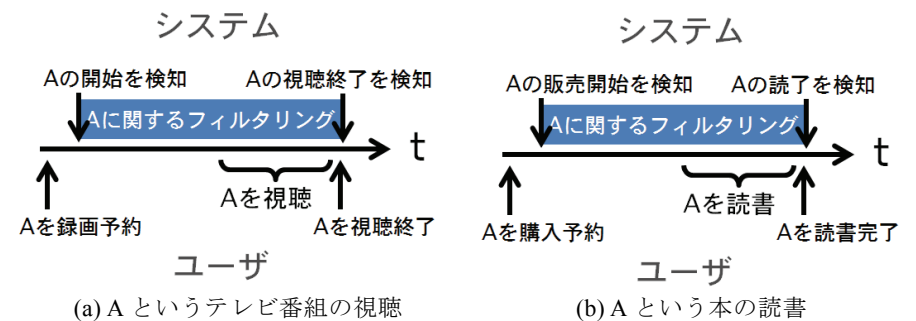
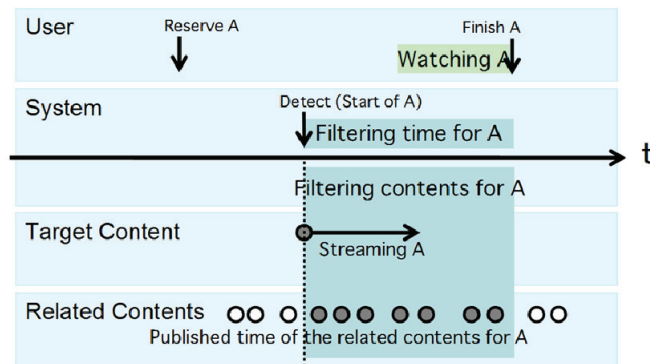


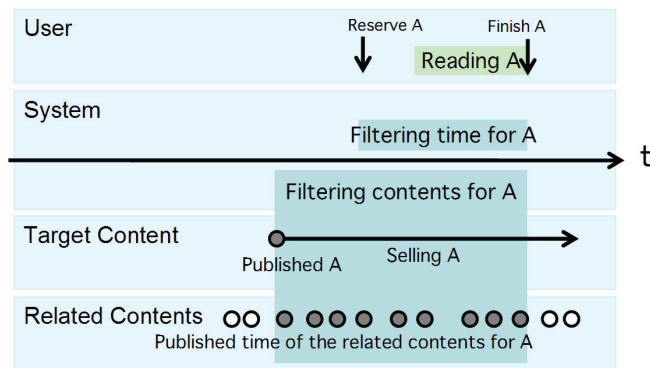
図 1 システムとユーザの振る舞い

ここで、遮断対象コンテンツ関係の情報であっても、発信された日時が遮断対象コンテンツ開始前であれば、それはただ単にその対象に対する基礎情報であったり、内容の予想であったりと、結果を含んでいるということはありません。例えば、筋書きのないスポーツの試合において、試合前に結果を知っているということはありません。

し、本の発売前に一般の読者がその内容を知ることはできない。そのため、ここで遮断対象となる結果情報は、遮断対象コンテンツ開始後のもののみとする。図2の(a)および(b)は、このフィルタリングの関係を図式化したものである。Related Contentsの中で、白丸はフィルタリング対象外のネタバレ情報を含んでいないコンテンツ、黒丸はフィルタリング対象のネタバレ情報を含むと予想されるコンテンツとなる。



(a) スポーツの試合の場合



(b) 本の場合

図2 遮断対象コンテンツと遮断対象となる結果情報

AntiSpoilerでは、ウェブコンテンツをブロックに分割し、ブロック毎のネタバレ度合いを予め用意しておいた辞書に基づき計算し、ネタバレ度合いがある一定のスコア

を超えているときには、そのブロック部分のテキストと背景色を同じ色に設定することで、ネタバレ情報のフィルタリングを行う。フィルタリングされたネタバレ情報はそのままでは閲覧できないが、その部分をマウスカーソルで選択することによりテキスト部分が反転され、テキストとして読むことができるようになるというものである。なお、ブロック部分が他のコンテンツへのリンクとなっている場合は、リンク先のコンテンツを検証し、より精度の高いフィルタリングを行うというものであった。

これまでに実現したシステムはプロキシという実装形態のせいもあり、動作が遅く、また汎用性も低かった。特に、リンク先のコンテンツを検証する場合にかなりの時間がかかってしまうという問題があった。また、当時は存在しなかったTwitterに代表されるマイクロブログなどのより速報性が高く、話し言葉が中心の書き込みが多いサービスに対応するには、従来のフィルタでは対応することができなかった。そこで、本研究ではこれまでに実現した手法を発展させ、ブラウザベースでのフィルタリングを行えるようにする。また、フィルタリングの視覚化について、4つの手法を提案し、実装して比較を行う。

なお、これまでの研究ではスポーツに限らず、映画や本、DVDなども情報遮断対象として扱ってきたが、本研究では単純のためスポーツの試合に特化してフィルタリング手法およびその視覚化について提案及び検討を行う。ここでスポーツの試合の場合、ネタバレ情報として考えられるのは、誰が（どのチームが）勝利したか、誰が（どのチームが）負けたか、誰が（どのチームが）何位に入ったかなどである。

3. 提案手法およびシステム

3.1 情報フィルタリング

情報フィルタリングにおいて、ユーザが閲覧しようとしているコンテンツ自体が情報遮断対象の試合の開始以前に最終更新されている場合、そもそも処理を行わない。コンテンツの最終更新日が不明の場合（CGIなど）や、コンテンツが情報遮断対象の試合の開始以降に更新されている場合、そのコンテンツにネタバレ情報が含まれているかどうかを検証する。

ネタバレ情報が含まれているかどうかの検証の前段階として、まずコンテンツをブロック単位で認識し、そのブロックがネタバレ情報を含んでいるかどうかで判定する。また、そのブロックから他のコンテンツへとリンクが設定されている場合には、そのリンク先のコンテンツの更新日時を取得し、その更新日時が対象イベント開始日時より古いものであればネタバレ情報でないと判定し、新しいものである場合はコンテンツ自体を取得し、そのコンテンツのネタバレ可能性について検証する。なお、ブロックの認識については、視覚的な特性を利用[2]するものなど様々な手法が提案されてい

るが、本研究では単純のため、タグで囲まれた範囲を1つのブロックとする。

ブロック単位でのネタバレの有無の検証においては、情報遮断対象を識別するためのチームおよび選手辞書と、ネタバレ情報を検出するための正規表現辞書を利用する。ここで、チーム及び選手辞書については、情報遮断設定時に手作業でキーワード群(正式名称やニックネームなど)を指定する方法と、そのチームや選手、試合に関する各種の記述がなされたウェブページを複数指定し、システムが自動的にその中から名前情報を抽出して、その語を利用する方法の2つからなる。表1はネタバレ情報を検出するための正規表現辞書の一例であり、予め用意してある。なお、この正規表現辞書については、ユーザが随時追加可能である。また、ここでは、フィルタリングの可視化で利用するため、結果反転辞書(圧勝の反対は完敗、勝利の反対は敗北、優勝の反対は準優勝など)も用意しておく。

表1 正規表現の一例(書き言葉用)

/[人名 固有名詞][助詞]*[0-9]+[*][-:対][]*[0-9]+/
/[0-9]+[*][-:対][]*[0-9]+[助詞]*[人名 固有名詞]/
/[人名 固有名詞][助詞]*([勝利 圧勝 優勝 敗北 敗退 ...])/
/([勝利 優勝 敗北 敗退 ...])[助詞]*[人名 固有名詞][助詞]*/
/[人名 固有名詞][助詞]*V[0-9]+/
/[人名 固有名詞][助詞]*[0-9]+([連勝 連敗 ...])/

3.2 フィルタリングの視覚化

本研究では、フィルタリングの視覚化について、下記の4つの手法を提案する。

- **非表示手法:** 該当ブロック部分を、単純に非表示に設定する手法。ただ単にその部分が抜け落ちるだけであるため、ユーザはそのページを閲覧中に気になることは無いが、その対象がフィルタリングすべきものでなかった場合であっても、ユーザはそのコンテンツを調べることができない。また、何がフィルタリングされているかをユーザは判断することができない。
- **黒塗り手法:** 該当部分の背景およびテキストを同じ色へ設定することで、遮断対象部分が、検閲により黒塗りされたように視覚化される手法。どこがフィルタリングされているかをユーザは把握可能。一方で、ユーザはその部分が気になってしまうと予想される。また、何がフィルタリングされているかをユーザは判断することができない。
- **曖昧記述変換手法:** ネタバレ情報が明確に記述されている部分を、結果反転辞書と疑問形変換を利用して、勝敗をぼかしたような記述に変更する手法(「日本代

表が勝利」を「日本代表が勝利したかもしれないし敗北したかも」「日本代表が勝利？」などに置換)。ユーザは、その曖昧表現を発見することで、そこに知りたい結果が書かれているかどうかを判断可能となる。一方、不自然な文章となってしまい、勝利なのか敗北なのか、どちらであるかということ、不自然度合いで判断可能となる可能性がある。

- **木を隠すなら森の中手法:** 結果反転辞書を利用して、ネタバレとして予想される情報と結果が反転したものや、類似した結果などをページ中に挿入していく手法。ユーザはどれが本当か分からなくなる。ネタバレの可能性は低くなるが、ゴミだらけになってしまい鬱陶しく感じる可能性がある。

なお、フィルタリングにおいてリンク先のコンテンツを検証する場合、その検証に時間を要するため、そうしたブロックについては一旦黒塗り手法でフィルタリングを行い、問題がないと判定されたらフィルタリングを解除し、問題があると判定された場合は指定された視覚化手法でフィルタリングを行う。

3.3 実装

提案手法に基づき、Mozilla Firefoxの拡張としてJavaScriptおよびXULを用いてシステムを実装した。図3は、Yahoo!スポーツ上で、非表示手法を用いて2010年サッカーワールドカップの日本代表とデンマーク代表の試合についてフィルタリングを行っている様子である。

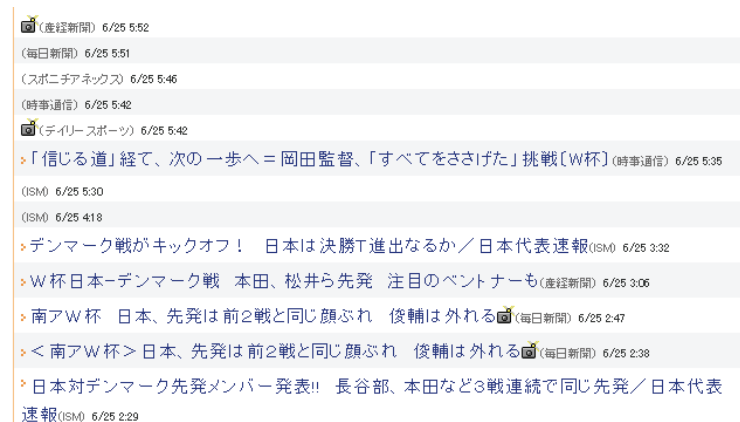


図3 非表示手法によるYahoo!スポーツでのフィルタリング

また、図 4 および図 5 は、同じくサッカーワールドカップに関する試合について、mixi および Twitter において黒塗り手法でフィルタリングを行っている様子である。さらに、図 6 は、Yahoo!スポーツにおいて、曖昧記述変換手法を利用してフィルタリングを行っている様子である。



図 4 mixi におけるフィルタリングの例



図 5 Twitter におけるフィルタリングの例

- ▽ 岡田ジャパン、オランダに敗れるも一歩前進 (ISM) 6/20 7:19
- ▽ 日本がオランダに惜敗、デンマークが勝利しカメルーンは敗退決定 = W杯 (スポーツナビ) 6/20 6:09

(a) 曖昧記述変換手法でのフィルタリング前

- ▽ 岡田ジャパン、オランダに勝つも敗れるも一歩前進 (ISM) 6/20 7:19
- ▽ 日本がオランダに勝利したかも、デンマークが勝利か敗北かカメルーンは進出か敗退決定 = W杯 (スポーツナビ) 6/20 6:09

(b) 曖昧記述変換手法でのフィルタリング後

図 6 Yahoo!スポーツにおけるフィルタリングの例

ネタバレ防止のためのフィルタリング対象コンテンツの指定については、図 7 のような ToDo 管理インターフェースで、イベントの開始日時や情報遮断のためのキーワード群またはウェブページ群を指定する。また、イベントを楽しんだのちには、このインターフェースにチェックを入れることで、そのイベントの終了を伝達する。

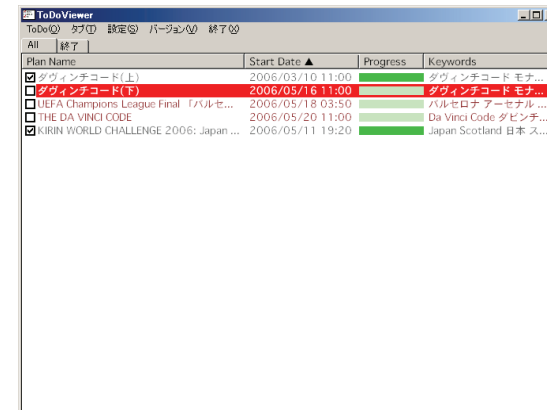


図 7 イベント管理インターフェース

4. 考察

提案及び実装したシステムをワールドカップ期間中に、ワールドカップの各試合を対象として著者および実験協力者1名で運用を行った。

まず、コンテンツのフィルタリング速度について、コンテンツ提示からフィルタリングまでが瞬時に行われるため、負荷は全く感じない。また、怪しい部分についてはコンテンツを一旦黒塗り手法でフィルタリングを行ない、その後、検証して問題ない場合はその黒塗り手法を解除するという実装にしているため、多少検証に時間がかかるコンテンツであっても、ストレスは感じなかった。

なお、リンクされたコンテンツの更新日時を取得するため、現状では HTTP の HEAD リクエストと、そのリクエストに対する応答で返される Last-Modified というエンティティを利用している。静的コンテンツの場合は、この Last-Modified を取得することによりコンテンツの最終更新日を問題なく取得できるが、CGI などを利用して動的に生成されるコンテンツの場合、最終更新日には意味が無いことが多い。そこで、今後の拡張として、前回そのサイトをアクセスしてから、次にアクセスするまでのコンテンツの差分の有無をチェックし、その差分情報に基づきコンテンツの発信日時を推定する手法を実現する予定である。また、ニュース記事などの場合、コンテンツ自体に発信日時が記述されていることもあるため、コンテンツを分析することにより、そうした情報を取得していくことも考えている。

対象ブロックコンテンツがネタバレ対象であるかどうかについての判定について、今回の手法は辞書に登録されているキーワードがブロックに含まれていればフィルタリング対象とするという簡易的なものであった。そのため、全体のネタバレ情報のうち、どれだけ多くのネタバレ情報をフィルタリングできたかという、フィルタリングの再現率は高く、Yahoo!スポーツ(a)と mixi ニュース(b)、朝日新聞(c)において、用意した 108 のネタバレ情報のうち、102 のネタバレ情報を検出出来ていた(ブロックをタグ単位にしているため、テーブルタグなどで結果が表示されている場合に検出できないなどの問題もあった)。一方、108 のネタバレ情報しかないのに、142 のブロックをネタバレ情報として検出してしまうなど、余計な物をフィルタリングしてしまう適合率の面で問題があった。これは、今回の手法が再現率を重視したものであることが原因の一つである。今後は、0 または 1 で判断するのではなく、段階的に判断し、閾値で切り替えることによって、ある程度再現率を保ったまま、適合率を上昇させる予定である。

なお、ユーザの興味の度合いは対象イベントが終了してから時間が経過すれば経過

するほど薄れていくものであると考えられる。長期のフィルタリングはフィルタリングの精度を低減させるものであり、ユーザにとって必ずしも有用なものであるとは言えない。そこで、対象イベントが終了してからの経過時間に基づくフィルタリングの閾値変更についても、今後考えていく予定である。

新聞社などの報道機関が発信する結果情報についてはある程度テキスト長が短くてもフィタリングはある程度精度よくできていた。一方、Twitter や掲示板サービスなどにおいて一般ユーザから発信される結果情報のフィルタリング精度は期待した程ではなかった。これは、新聞社などの報道機関が発信する情報の場合、コンテンツは書き言葉で記述されていることが多いが、Twitter などのサービスでは話し言葉で記述されることが多いためである。この問題を解決するには、フィルタリングを書き言葉用、話し言葉用と2つ用意し、サイトによってフィルタリング辞書を切り替えることが考えられる。なお、話し言葉らしさ、書き言葉らしさを計測する研究もなされている[3]ため、そうした技術を応用してコンテンツを自動判定することも考えられる。

今回実現した4つの手法のうち、非表示手法および黒塗り手法については、その表示自体が気になってしまううえ、何がフィルタリングされているかわからないという不安感があったが、ある程度の再現率でフィルタリング出来ていたためあまり大きな問題にはならなかった。一方、曖昧記述変換手法や木を隠すなら森の中手法は、結果反転辞書による結果反転をあまり効率的に作成できていなかったため(図6参照)、結局ネタバレしてしまうという問題があった。しかし、この両手法共に、うまく変換できた場合にはネタバレ情報の存在を示してくれるとともに、それを見た際に対象イベントへの興味が増幅されることがあり、十分可能性を秘めていると考えられる。そこで、今後は変換および挿入の精度を向上させていく予定である。

ワールドカップの試合のように、世界中の人々が興味するようなイベントの場合、試合中に Twitter にアクセスすると、大半の書き込みはその試合に関するものになっている。そのため、本システムを利用するとほとんどの発信内容が遮断対象の結果情報となってしまう、ページの大半が何らかの形でフィルタリングされるなど、あまり意味のないものになってしまう。また、Twitter などのように膨大なユーザが高頻度で気軽に発信するようなサービスの場合、ユーザごとによって表記ゆれがあるうえ、コンテンツを記述せずに結果に関するキーワードのみが発信されることが多いなどの問題があり、ネタバレを完全に遮断することは難しい。そこで、そもそもそうしたサービスにアクセスする際、これまでのネタバレに関するフィルタリング実績に基づき、ある程度のネタバレ危険性がそのサイトにあることを警告として表示し、アクセスするかどうかを確認するシステムを今後実装する予定である。

また、今回実装したシステムでは、動的にコンテンツがロードされる Ajax を利用したページや、Flash などを利用したコンテンツを処理することが出来ない。これは、本システムが Mozilla Firefox の拡張として JavaScript と XUL を用いて実装されている

a <http://sports.yahoo.co.jp/>
b http://news.mixi.jp/list_news.pl
c <http://www.asahi.com/>

ためであり、本手法の限界である。この問題を解決するにも、そもそもそうした情報を提示する可能性のある Flash や Ajax がある場合、そのサイトは情報遮断を行っているユーザにとって危険なサイトであるとして、警告を出すことが有効であると考えている。

今回実現したシステムには画像処理機能を搭載していないため、情報遮断対象イベントに登場する選手が喜んでいる画像（つまりその選手が所属しているチームが勝利したことがわかるなど）がコンテンツとして提示されている場合に、そのコンテンツをフィルタリングすることが出来ない。また、仮に画像処理技術を導入したとしても、そのシーンの認識は非常に難しいものであり、精度は期待できない。そこで、今後はその画像の周辺テキスト（テキストキャプションなど）を利用し、画像がその情報遮断対象イベントに深く関係すると予想される場合はそのコンテンツをフィルタリングするなどの仕組みを取り入れる予定である。

5. 関連研究

情報の量が爆発している現代において、膨大な情報を如何にして管理するか、対象とする情報を如何にして探すかなどの点が大きな問題となっている。

膨大な情報の中から適切な情報に出会うための方法として、情報推薦や情報フィルタリング[4][5]に関する研究が行われており、こうした研究ではユーザがローカルディスクに蓄積しているコンテンツや、ユーザがブックマークしているコンテンツ、ユーザのこれまでのアクセス履歴に基づいてユーザの興味を推定し、その興味の度合いに応じてユーザが興味を示すと期待される情報を高く評価してランキングしたり、推薦したりすることが一般的である。つまり、サッカー好きなユーザに対してはサッカーに関する結果を積極的に届け、アメリカンフットボールが好きなユーザに対してはアメリカンフットボールの結果を積極的に届けるようになる。これは、ただ単に情報を手に入れたいと思うユーザにとって是有用なシステムであるが、情報遮断を行なおうとしているユーザにとっては余計なお世話であり、ネタバレ情報に触れる危険性を上昇させるものである。

フィルタリングに関する研究やシステム開発は多数なされており、ソフトウェアやサービスとして実装されているが、フィルタリングの対象はコンピュータに害をなすウイルス[6][7]や、膨大に送られてくる SPAM メールなどのように、すべてのユーザにとって忌むべき不要なものである。また、子どもにとって有害であるとの判断により暴力的コンテンツや性的コンテンツをフィルタリングするサービス[8]もある。こうした研究やシステムでは、単純ベイズ分類器を応用したベイジアンフィルタを使うもの[9]や、問題のあるコンテンツをブラックリストとして用意してアクセスできないよう

にするもの、問題のないコンテンツをホワイトリストとして用意してそのみにアクセスできるようにするものなど様々である。しかし、本研究で対象とするようなユーザの興味が高ければ高いほど、逆にそのコンテンツをフィルタリングするといったパーソナルなフィルタリングには適していない。

6. おわりに

本研究では、ユーザのある対象に対する情報遮断をしたいという欲求に注目し、その対象自体が開始されてから、ユーザがその対象を楽しみ終わるまでの間、その対象に関するネタバレ情報を時間ベースでのフィルタリングする手法を提案し、Mozilla Firefox の拡張として実装した。また、実装したシステムの運用により有効性、可能性、問題点などを明らかにした。一般ユーザでの実験についてはまだ十分ではない。そのため、今後は一般ユーザも巻き込んでユーザテストを実施する予定である。

隠された情報はユーザの興味をひきつけるものである。そのため、あまりにこうした情報フィルタリングを行ってしまうと、ユーザの負荷を上昇させてしまう可能性がある。そこで、今後の研究において、ユーザの興味をそこまでひきつけることなく、フィルタリングを行う視覚化手法について、アイトラッキングシステムなどを利用して検証予定である。また、コンテンツのブロック単位での古さを検知する研究もなされている[10]。そこで、今後はこうしたブロック単位でのコンテンツの古さを検知することで、より正確なブロック単位での情報フィルタリングを行う予定である。一方、今回提案した手法は、対象イベントに対する辞書を作る手間があった。また、現時点では日本語しか対応しておらず、英語のコンテンツを閲覧した際にネタバレしてしまうという問題がある。辞書作成の容易化や多言語化などについては、今後取り組む予定である。

謝辞 本研究は文部科学省科学研究費補助金 挑戦的萌芽「モバイル協調検索に関する研究」（研究代表者：中村聡史、課題番号 22650018）によるものです。また、システム開発において慶応義塾大学の赤塚大典氏に支援を頂きました。ここに記して謝意を表します。

参考文献

- 1) Satoshi Nakamura and Katsumi Tanaka: Temporal Filtering System for Reducing the Risk of Spoiling a User's Enjoyment, 2007 International Conference on Intelligent User Interfaces (IUI2007), pp.345-348, Hawaii (Jan. 2007).

- 2) Deng, C., Shipeng, Y., Ji-Rong, and Wen. Wei-Ying, Ma., "VIPS: a Vision-based Page Segmentation Algorithm", Microsoft Technical Report (2003-79), (2003).
- 3) 佐野大樹:「話し言葉らしさ・書き言葉らしさ」の計測－語彙密度の日本語への適用性の検証－. 機能言語研究, 5, 89-102 (2009).
- 4) Riecken, D., guest ed. Personalized views of personalization (special edition). Communications of the ACM, Vol. 43, No. 8 (2008).
- 4) 土方 嘉徳, 情報推薦・情報フィルタリングのためのユーザプロファイリング技術, 人工知能学会誌 19(3), 365-372 (2004)
- 5) Symantec Corp., Norton Internet Security, <http://www.symantec.com/>.
- 6) McAfee Inc., Virus Scan, <http://www.mcafee.com/>.
- 7) DigitalArts Inc., i-FILTER, <http://www.daj.co.jp/en/ir/>.
- 8) Sahami, M., Dumais, S., Heckerman, D. and Horvitz EM., A Bayesian Approach to Filtering Junk Email. AAAI Workshop on Learning for Text Categorization, July 1998, AAAI Technical Report WS-98-05.
- 10) Adam Jatowt, Yukiko Kawai and Katsumi Tanaka: Detecting Age of Page Content, Proceedings of the 9th ACM International Workshop on Web Information and Data Management (WIDM 2007), ACM Press, Lisbon, Portugal, pp. 137-144 (2007).