スポーツのネタバレを防止する Twitter クライアントの開発と諸検討

中村 聡史　　川連 一将

明治大学総合数理学部先端メディアサイエンス学科

satoshi@snakamura.org

概要　スポーツの試合を録画試聴することを楽しみにしているユーザにとって，その試合の結果を知ることは，録画視聴の楽しみを低減させてしまうものである．著者らはこれまでウェブブラウザ上でのネタバレ防止手法について提案および実装してきたが，近年情報ソースとして多用されている Twitter などのソーシャルネットワークサービス上でのネタバレには対応できていなかった．そこで本稿では，Twitter を対象とし，広く利用されているクライアントのプラグインとしてネタバレ防止手法を実装し，その可能性や課題についての検討を行った．

キーワード　ネタバレ防止，情報曖昧化，Twitter

1 はじめに

スポーツに興味がある人ほど，仕事や学校などでリアルタイムにスポーツの試合を視聴できず録画視聴しようとしている時に出会ってしまうネタバレ情報を大きな問題として考えている[1]．我々はこれまでウェブブラウザ上でのネタバレを防止するため，ブラウザの拡張としてネタバレ防止する手法を実現してきた[2,3]．

一方，ユーザがインターネットを介して情報を得る方法として，能動的なアクセスが必要なウェブサイトだけでなく，受動的に他者の選別のもとに情報を得ることが出来る Twitter などのソーシャルネットワークサービスがよく利用されている．ここで，Twitter の発信者は悪気があってネタバレ情報を発信しているわけではないが，受け取る側にとってはそれがネタバレ情報となってしまうということは多々ある．実際，Goldbeck らは時差のあるアメリカで Twitter のネタバレ情報が問題となっている点について考察を行っている[4]．また，三木らはウェブサービスとして Twitter のネタバレを防止する仕組みを開発している[5]．これまで我々が実現していたウェブブラウザの拡張では，そうした情報ソースを対象とすることは困難だった．

ネタバレに触れたくないのであれば Twitter を利用しないように心がければ良いが，Twitter は情報コミュニケーションツールとして確立されており，Twitter 上でやりとりされていることを無視することは難しい．また，情報入手のためのツールとして使われていることも少なくなく，Twitter 自体の遮断は相応しくない．

そこで本稿では，Twitter を利用しつつ情報に触れないようにするための仕組みを実現することを目的とする．また，ここでは特に Twitter のクライアントとして広く普及している Janetter[6]のプラグインとしてネタバレ防止のための情報曖昧化手法を適用したシステムを開発し，日常的に利用することで，その可能性や課題などについて検討を行う．

2 提案手法とプロトタイプシステム

プロトタイプシステムは，Twitter クライアントとして広く普及している Janetter のプラグインとして JavaScript を用いて開発を行う．

本プロトタイプシステムは，Janetter が受け取ったユーザがフォローしている他者のツイート集合に対して順次ネタバレに該当する語が含まれていないかを判定する処理を行い，ネタバレに該当する語が含まれていない場合はそのままタイムライン（ユーザが他者のツイートを時間順に確認できるインタフェース）上に情報を提示する．ネタバレに該当する語が含まれているツイートについては情報曖昧化手法を適用することでネタバレ防止を行った後，タイムライン上に提示する．

ネタバレ防止を行うために，我々はこれまでの研究[1]において，4 つの情報曖昧化手法を提案してきた．その中でも墨塗り手法（ネタバレ情報に該当する部分を塗りつぶす手法）と，結果反転手法（ネタバレ情報を任意の確率で反転させる手法）の有効性が認められたため，本稿ではこの墨塗り手法と結果反転手法を適用する．

ツイートにネタバレに関する語が含まれるか否かの判定については，これまでの研究[2]で利用していたユーザ設定による手法に加え，他者のツイートからの自動推定手法を提案及び実装する．ここでは特に，ネタバレ情報が積極的に発信されると考えられる該当ハッシュタグ

(#soccer #japan #nhk など)やキーワードを登録し，そのハッシュタグとセットでつぶやかれるツイート内容を利用する．また，ツイート内容の分析では Brody らの手法[7]を利用してツイートに現れやすい感情表現の表記ゆれのパターンを 1 つにまとめ(「やったあああ」「やったあああああ」などで連続する文字を 1 文字に変換し，「やったあ」として扱う)，TinySegmenter[8]を利用して語の切り出しを行い，その出現頻度に応じてネタバレ対象語を自動生成するというものである．なお，他者のツイートを元にするため，本手法ではシステムで提示するツイートを遅れて表示させる．

プロトタイプシステムを利用している様子は図 1 の通りである．spoilers_test というテストアカウントが「日本代表が勝利」とつぶやいている様子である．墨塗り手法の場合，「勝利」の部分が「クリックで表示」という文字に置き換えられているため，ネタバレ情報を遮断することに成功している．なお，その結果に興味が有る場合は，「クリックで表示」という部分をクリックすることで，元の文章を表示することが可能となっている．一方，結果反転手法の場合は，結果反転辞書[3]を利用して任意の確率で結果を反転されており，この場合は「勝利」が「完敗」に書き換えられている．ユーザは結果が任意の確率で反転されることを知っているため，ネタバレ情報に触れたとしてもそれが本当に正しい情報なのか疑うため，結果としてネタバレを防止することが可能となっている．図 2 はシステムの設定画面であり，ここで手法やネタバレ対象語などを指定することが可能となっている．

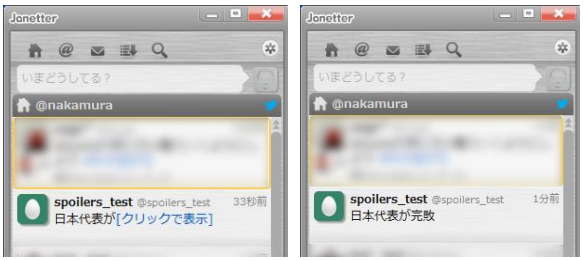

図1　プロトタイプシステムの動作例．左図が墨塗り手法を適用している様子，右図が結果反転手法を利用している様子．

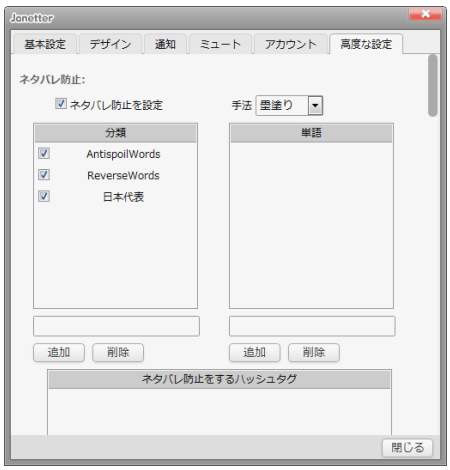

図2　設定画面

3 考察とまとめ

開発したシステムを日常的に利用し，墨塗り手法である程度のネタバレ防止はできたが，Twitter では主語がないことが多いため，ネタバレがそのまま提示されてしまうことがわかった．結果反転手法では，ユーザがシステムが嘘を付いているかもしれないと考えるため，ネタバレを提示してしまっていても気づかないこともあったが，主語が無い場合には気づいてしまうため，本質的にはネタバレを防ぐことはできていなかった．ハッシュタグなどからネタバレの可能性がある語を自動推定する手法については，今回は単純な手法を用いたため，無関係の語が抽出されることも多く，あまり効果的ではなかった．こうした問題については，試合中は主語がないものであっても勝敗にまつわる語が含まれていればネタバレ対象とし，再現率を高める事が考えられる．また，その試合に関連するツイートの内容と，過去の試合におけるツイートの内容などを効果的に利用しつつ，辞書ベースのアプローチと組み合わせることが考えられる．今後はこうした点に取り組む予定である．

本稿で実現した手法は，プラグインとして実装可能なものであるため，プラグイン開発が可能な他のツールでも同様に実現可能となる．また，今回は Twitter のみを対象としているが，同様のことは他のソーシャルネットワークサービスで可能となるであろう．

謝辞

本研究の一部は，文部省科学研究費補助金 挑戦的萌芽研究(#25540080)の支援によるものです．ここに記して謝意を表すものとします．

参考文献

[1] 中村聡史，小松孝徳:スポーツの勝敗にまつわるネタバレ防止手法:情報曖昧化の可能性，情報処理学会論文誌 Vol.54, No.4 pp.1402-1412, 2013.

[2] Nakamura S. and Tanaka K.: Temporal filtering system for reducing the risk of spoiling a user's enjoyment, Proc. of IUI 2007, pp. 345-348, 2007.

[3] Nakamura S. and Komatsu T.: Study of information clouding methods to prevent spoilers of sports match, Proc. of AVI 2012, pp. 661-664, 2012.

[4] Golbeck J., The Twitter mute button: A Web filtering challenge, Proc. of ACM CHI 2012, pp. 2755-2758, 2012.

[5] 三木孝，山田泰宏，服部哲，速水治:TV 番組のネタばらしを非表示にする Twitter クライアントの提案，グループウェアとネットワークサービス研究会, 2014.

[6] Janetter: http://janetter.net/jp/.

[7] Brody S, Diakopoulos N.: Coooooooooooooollllllllllll-l!!!!!!!!!!!!!!: Using word lengthening to detect sentiment in microblogs, Proc. of EMNLP, pp. 562-570, 2011.

[8] TinySegmenter, http://chasen.org/~taku/software/TinySegmenter/.