

# TV番組のネタバレしを非表示にする Twitter クライアントの提案

三木孝<sup>†1</sup> 山田泰宏<sup>†2</sup> 服部哲<sup>†2</sup> 速水治夫<sup>†2</sup>

Twitter は身の回りの出来事などを手軽に投稿できる Web サービスである。Twitter では投稿することをツイートと称し、利用者は様々な場面でツイートすることができる。本論文では TV 番組を視聴しながらツイートしている実況者といわれるユーザに着目する。近年、実況者によるツイートを付いた TV 番組のネタバレが多い。大抵の実況者は特定のハッシュタグを付加したツイートをするが、中にはハッシュタグを付加せずツイートをする実況者も存在する。実況者をフォローしているユーザは自身のタイムラインが放映時間中に実況者のツイートで埋まってしまう、必要なツイートが見ることができなくなってしまう。また、リアルタイムで観ることができず、後日録画した番組を観る場合、話の筋や結末を知ってしまっているのでは観る楽しさが減少する。現状で特定のキーワードを含むツイートを非表示に設定できるクライアントは存在するが、それだけではネタバレのツイートのすべてを非表示にできない。

本論文では、ハッシュタグ付加ツイートを TV 番組放送時間中に取得と解析を行い、頻繁にツイートされているキーワードを抜き出し、自動的に非表示する。解析した結果から共通するキーワードが含まれるツイートをタイムライン上から非表示にすることでネタバレを防止するシステムを提案する。提案システムはネタバレツイートの非表示を優先することとした。このため、ネタバレしていないツイートも非表示になることも多少あるがそれは致し方ないこととした。

2 種類の評価実験の結果、提案システムは高評価を得た。

## Proposal of Twitter client to hide the spoilers of the TV program

TAKASHI MIKI<sup>†1</sup> YASUHIRO YAMADA<sup>†2</sup>  
AKIRA HATTORI<sup>†2</sup> HARUO HAYAMI<sup>†2</sup>

Twitter is a Web service that can post with ease and events around us. It can be called tweets to be posted on Twitter, user Tweets in various situations. Attention is paid to the user which is said to live who tweeted while watching a TV program in the present study. In recent years, spoilers of the TV program using a live tweet by the person in many cases. Live's most a tweet obtained by adding a hash tag of particular, who live a tweet it without adding the hash tag is also present in. Timeline of itself would be filled with tweets live person to broadcast during the time, a user who follows live person can no longer be seen is required tweet. Further, pleasure to watch is reduced if it can not be viewed in real time, and view the recorded program later, because it has known consequences or story line. Client that can be set to hide the tweet that contains specific keywords in the present circumstances exist, but it can not hide all of the tweets spoilers.

In this paper, we analyze and get to a TV program broadcasting time in the hash-tagged tweet, extracting keywords that are frequently tweet, it is automatically hidden. We propose a system for preventing the spoilers by hide from timeline tweets keywords in common from the results of the analysis and the like. The proposed system was decided to give priority to the non-display of tweet spoilers. For this reason, there is also some that tweet you do not spoilers also be hidden, but it was decided can not be helped.

Results of the evaluation two experiments, the proposed system is to obtain a high evaluation.

### 1. はじめに

現在の日本国内では、Twitter<sup>1)</sup>の利用人口が増加している。利用人口の増加する背景には、スマートフォンユーザの増加と複数の Twitter クライアント<sup>2)</sup>の開発と普及により、手軽に利用することができるためと推測する。また、近年では多くの TV 番組の放送局や制作会社が Twitter との連携により情報を提供している。その一例がハッシュタグ<sup>3)</sup>である。ハッシュタグは誰でも簡単に作れるが、多くの場合は企業側が作ったものを一般ユーザがツイートに付加させ利用している。ハッシュタグを付加させたツイートの多くに

は番組の感想やネタバレが含まれている。

本論文では、ネタバレが含まれるツイート(以後、ネタバレツイート)をタイムライン上から非表示にし、ネタバレを事前に防ぐことを目的とする Twitter クライアントを開発する。

以下、第 2 章では研究対象の問題と解決、第 3 章では、提案システム概要、第 4 章では実験について記述し、第 5 章でまとめを述べる。

<sup>†1</sup> 神奈川工科大学  
Kanagawa Institute of Technology

<sup>†2</sup> 神奈川工科大学大学院  
Graduate School of Kanagawa Institute of Technology

## 2. 問題と解決

### 2.1 問題

Twitter 上のネタばらしには次のような問題点が存在すると考える。

#### 2.1.1 実況者のツイートによる問題点

実況者はTV番組を視聴しながらTwitter上で番組の感想または内容の詳細や核心部分<sup>4)</sup>を含む文章をツイートする。ネタばらしを嫌うユーザがネタばらしツイートを閲覧することにより、後日録画しておいた番組を観る楽しみが減ってしまう。また、実況者は短時間での投稿数が多い。そのため、他のユーザのツイートが実況者のツイートにより埋もれてしまい見逃すという問題が挙げられる。実況者のフォローを解除すればネタばらしを閲覧することもなくなるが、実況者のツイートすべてがネタばらしを含む内容ではない。よって、フォローを解除すれば解決するとは考えにくい。

#### 2.1.2 類似システムの問題点

一部のユーザやキーワードを指定して非表示にするTwitterクライアントとして「Janetter<sup>5)</sup>」がある。このシステムはTwitter社が開発者向けに提供するTwitterAPI<sup>6)</sup>を用いて開発された。Twitterとしての機能は十分であり、その他にもタイムラインの複数表示やタイムライン昇順降順を選択という機能がある。しかし、Janetterには問題が2つある。1つは、現状Janetterは利用者の上限を超えてしまったため、新規ユーザは利用することができない。もう1つは、キーワードを登録する際にユーザが手動でキーワードを登録することだ。Janetterは非表示にしたいキーワード(以後、NGワード)を登録することにより、NGワードを含むツイートを非表示にする。しかし、実況者はTV番組を視聴しながらツイートするため毎回新しいキーワードを付加させるツイートを投稿する可能性が高い。よって、既存の非表示機能では新しいキーワードに瞬時に対応できない問題が挙げられる。現状、新しいキーワードに対して瞬時に対応するTwitterクライアントはない。

### 2.2 解決

本論文では、前節で述べた問題点に対して、次のような解決策を提案する。

#### 2.2.1 着眼点

ネタばらしツイートには、ハッシュタグを含むネタばらしツイートとハッシュタグを含まないがネタばらしになるツイートの2種類がある。これらには関連するキーワードが存在すると考えられる。このためハッシュタグを含むネタばらしツイートから関連キーワードを抽出し、NGワードに加えることでネタばらしツイートを非表示にできる。関連キーワードの抽出のために、ユーザには非表示にしたい番組のハッシュタグ(公式ハッシュタグ推奨)を入力してもらう必要がある。入力したハッシュタグからそのハッ

シュタグが付加される最新100件のツイートを取得する。取得したツイートに形態素解析を行い形態素化した2文字以上のキーワードをデータベースに登録する。ネタばらしツイートを判断させるためにデータベースから条件に合ったデータを取得しタイムラインと比較する。タイムラインにキーワードと一致したツイートがあれば非表示にすることでネタばらしを防ぐことができる。データベースからキーワードを取得する条件として、自立語、名詞、固有名詞に絞ることでネタばらしツイートとは関係ないツイートを非表示にすることを防ぐことになる。

#### 2.2.2 Janetter との比較

本論文の目的は、ネタばらしツイートまたそれに関連するツイートを非表示にすることである。前節で述べたJanetterと提案システムの比較を表1に示す。

表1 類似システムと提案システム比較

Table 1 The proposed system compared to similar systems.

クライアント名	Janetter	提案システム
◆ 非表示機能		
ユーザ非表示	○	—
キーワード非表示	○	—
ハッシュタグ非表示	—	○
関連キーワード抽出	—	○
自動キーワード非表示	—	○
◆ ツイートに関する機能		
タイムライン表示	○	○
ツイート投稿	○	○
リプライ投稿	○	○
リツイート投稿	○	○
◆ ユーザ情報		
ユーザ情報の変更	○	—
複数アカウント登録/切替	○	—

提案システムのように関連するキーワードを自動で非表示にする機能がJanetterにはないことがわかる。しかし、Janetterには、ユーザやキーワードを指定することによりユーザのタイムライン上から非表示にできる機能やユーザの情報を変更する機能を持つ。

提案システムは、非表示機能の利用とタイムラインを閲覧することを目的にしているTwitterクライアントである。そのため、Janetterで使用できる機能が使用できない部分もあるが、それは致し方ないこととする。

## 3. 提案システム概要

### 3.1 システム概要

提案システムはTwitterを利用しているユーザなら誰にでも利用することができる。

システムの画面遷移図を図1に示す。

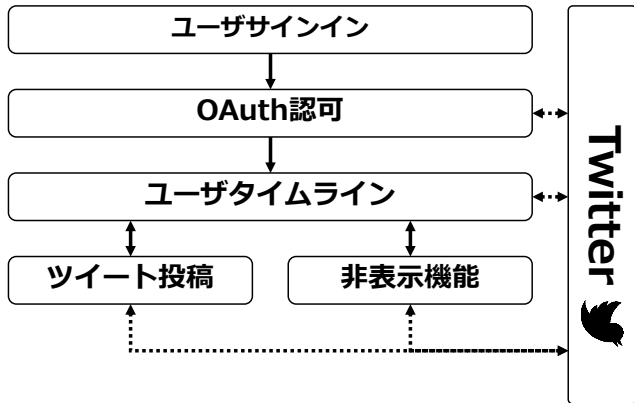


図 1 画面遷移図

Figure 1 Screen transition diagram.

ユーザは提案システムを使う際に、OAuth 認可を行う。認可するためには Twitter 側にログインしている必要がある。

### 3.2 システム構成

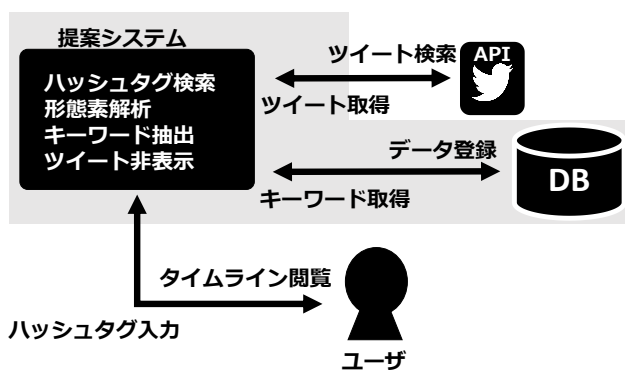


図 2 システム構成 (網掛け部が提案システム)

Figure 2 System configuration.

提案システムは Web アプリケーションとして開発した。また、提案システムは TV 番組の中でもアニメ番組を対象としている。ユーザは提案システムを通してタイムラインの閲覧やツイート投稿、非表示機能を利用する(図 2)。2.2.1 項で述べたようにユーザは非表示にしたい TV 番組のハッシュタグを入力することで、そのハッシュタグが付加している最新 100 件のツイートを取得する。取得したツイートは 1 ツイートごとに形態素解析を行い、形態素化される。形態素化されたデータは 2 文字以上ならばデータベースに登録され、ツイートを非表示にする際に条件指定して抜き出す。抜き出した単語が NG ワードとなる。NG ワードと一致したツイートがある場合、ツイートは非表示になる。

関連キーワードの抽出とツイートを非表示する際の一連を図 3 と図 4 に示す。

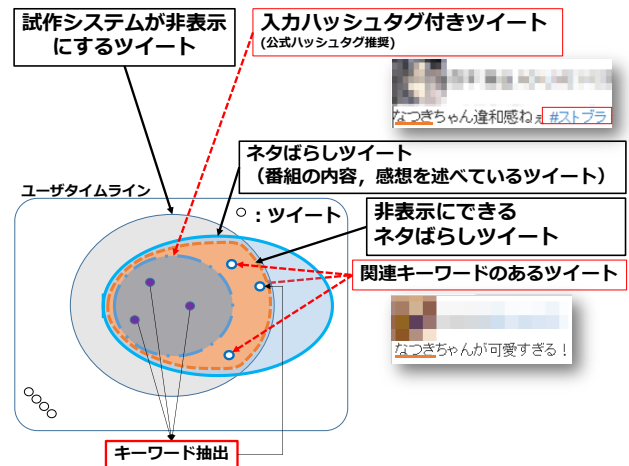


図 3 関連キーワード抽出

Figure 3 Extraction of relevant keywords.

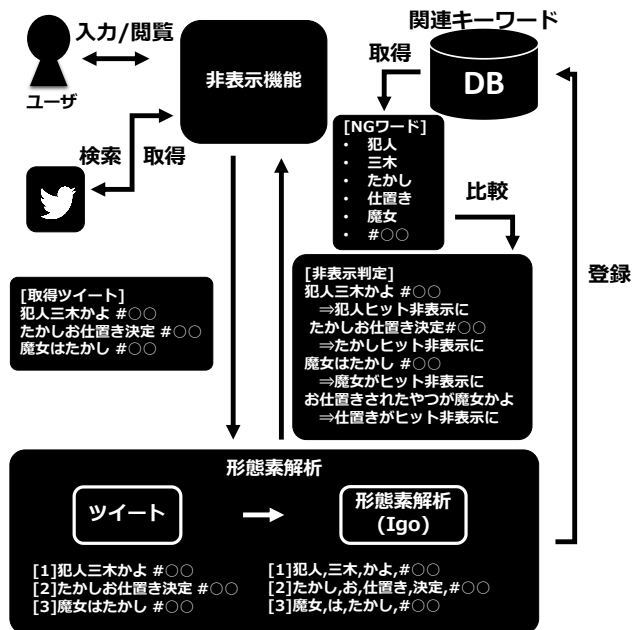


図 4 ツイート非表示の流れ

Figure 4 Flow of Tweets hidden.

これらの動作は、ユーザが非表示にしたい番組のハッシュタグを入力してから 30 分間毎分行われる。

同時に複数のハッシュタグを入力することは出来ない。複数入力する場合は別々にハッシュタグを入力する必要がある。

### 3.3 Igo(形態素解析器)

本論文では、形態素解析を行う際には、「Igo<sup>7)</sup>」という形態素解析器を利用している。Igo は、Java で実装された形態素解析器であり、解析結果の互換性は「MeCab<sup>8)</sup>」と大差がない、比較的高速で処理を行う特徴を持つ。

## 4. 評価実験

本論文では、2種類の評価実験を行う。

### 4.1 評価実験 1

評価実験は対象番組に対し、どの程度ネタばらしツイートを非表示に出来たかの検証。

対象番組については関連キーワードを抽出した際に、使用したハッシュタグの一覧を表2に示す。

表2 対象番組と使用ハッシュタグ一覧

Table 2 Use hash tags and lists the target program.

対象番組	ハッシュタグ
夜桜四重奏 ～ハナノウタ～(A)	#yzq
機巧少女は傷つかない(B)	#mdoll
魔法少女まどか☆マギカ(C)	#madoka_magica
のんのんびより(D)	#なのん

### 4.2 実験結果 1

実験は、前節の表2で上げた対象の番組放送中に提案システムを用いて行った。

#### 4.2.1 非表示率の視覚化

対象番組を非表示にしたい際にどの程度ネタばらしを非表示に出来たかを視覚的に判断するのは難しい。視覚化するにあたり、情報検索における再現率と適合率またF尺度という指標を用いて非表示率の視覚化を行った<sup>9)</sup>。

#### 4.2.2 再現率

再現率とは、完全性を評価するための尺度であり、検索対象となる文章集合の中の検索質問に適合する文章のうち、実際に検索された文章の割合を示す。検索漏れの少なさを示す尺度である。

以下に、再現率の計算式を表す。

$$\text{再現率}(recall) = \frac{\text{検索された文章の適合文書の数}(R)}{\text{全文書中の適合文書の数}(C)}$$

図5 再現率の計算式

Figure 5 Formula recall.

#### 4.2.3 適合率

適合率とは、正確性を評価するための尺度であり、検索された文章集合の中で、検索質問に適合する文章の割合を示す。検索ノイズの少なさを示す尺度である。

以下に、適合率の計算式を表す。

$$\text{適合率}(precision) = \frac{\text{検索された文章の適合文書の数}(R)}{\text{検索された文章の数}(N)}$$

図6 適合率の計算式

Figure 6 Formula precision.

#### 4.2.4 F尺度

F尺度とは、再現率と適合率の調和平均である。F尺度の値が高ければ、性能が良いことを意味する。

以下に、F尺度の計算式を表す。

$$\text{F尺度}(F\text{-measure}) = \frac{2 \times recall \times precision}{recall + precision}$$

図7 F尺度の計算式

Figure 7 Formula F-measure.

本論文では、F尺度の値を重視して実験の評価を求める。

#### 4.2.5 非表示結果

本論文での再現率と適合率の計算式を図8と図9に示す。

$$\text{再現率}(recall) = \frac{\text{実際に非表示したネタばらしツイート数}(R)}{\text{非表示すべきネタばらしツイート数}(C)}$$

図8 本論文での再現率の計算式

Figure 8 Formula recall in this paper.

$$\text{適合率}(precision) = \frac{\text{実際に非表示したネタばらしツイート数}(R)}{\text{非表示したツイート数}(N)}$$

図9 本論文での適合率の計算式

Figure 9 Formula precision in this paper.

以上の計算式で求められた値が表3の再現率と適合率である。得られた2つの値により、対象番組のF尺度を求めることができる。この値により、どの程度ネタばらしツイートを非表示に出来たかを評価することができる。

対象番組のネタばらしツイートをどの程度非表示に出来たかの結果を表3に示す。

表3 対象番組の再現率・適合率・F尺度

Table 3 recall-precision rate F-measure of the target program.

対象番組	再現率	適合率	F尺度
A	97.10%	95.80%	96.45%
B	94.20%	85.91%	89.86%
C	96.00%	87.01%	91.28%
D	84.30%	84.35%	84.32%

### 4.3 評価実験 2

前節では、対象番組のネタばらしツイートをどの程度非表示に出来たかを検証した。しかし、他のユーザに利用してもらい受け入れられるかという実験は行っていないためWebアンケートによる評価実験を行った。

#### 4.4 実験結果 2

アンケートは、Webアンケートにて集計を行った。6名

の解答を得ることに成功した。表 4 にアンケートの解答を示す。

表 4 アンケート解答 1  
Table 4 Questionnaire Answers 1.

項目	YES	NO
タイムライン上にネタばらしがあることに困るか	2 人	4 人
提案システムによりネタばらしツイートを非表示にすることは出来たか	4 人	2 人

表 5 アンケート解答 2  
Table 5 Questionnaire Answers 2.

項目	1	2	3	4	5	平均
レイアウト・操作性	0	1	3	2	0	3.2
総合評価	0	0	1	3	2	4.2

#### 4.5 実験考察

2 種類の評価実験において、どの程度ネタばらしツイートを非表示に出来たかの実験では、F 尺度の値が高いもので 96% を超え、低い値でも 84% という高い評価を得ることが出来たことにより、当初の目的であるネタばらしツイートの非表示については改善できたのではないかと考えられる。しかし、Web アンケートでの評価は不満を訴える解答がいくつかあった。以上を総じて提案システムは改善しなければならぬが、将来的に有用性はあるのではと考えられる。

#### 5. おわりに

Twitter では TV 番組のネタばらしツイートが増加し、閲覧者は自分の好きな番組のネタばらしを見たくなくとも見ってしまうといったようなことがある。現状のクライアントに備わっている機能を使えばユーザやキーワードを登録するだけで非表示にできるが、ユーザの入力による登録のため入力漏れや手間がありすべてのネタばらしツイートを非表示にすることは困難である。そこで、本論文ではユーザが非表示にしたい番組のハッシュタグを入力し、その情報を基にタイムライン上からネタばらしツイートを非表示にするシステムを提案・開発した。ネタばらしツイートを非表示するため、入力したハッシュタグが付加するツイートから共通するキーワードを抜き出し、抜き出したキーワードをタイムラインと比較し、一致した場合非表示にした。評価実験より、本システムは入力したハッシュタグに共通するツイートの非表示に成功した。また、ユーザからのアンケートも高評価を得た。

現状のキーワード抽出では、再現率を重視するため適合率は低い。従って、関係ないツイートまで非表示にしてしまう可能性もある。今後は、再現率と適合率両方を重視するアルゴリズムを考え、より非表示機能の機能性を高める

システムを目指す。

#### 参考文献

- 1) Twitter,  
<https://twitter.com/>
- 2) Twitter クライアント-Twitter まとめ Wiki  
<http://usy.jp/twitter/index.php?Twitter%E3%82%AF%E3%83%A9%E3%82%A4%E3%82%A2%E3%83%B3%E3%83%88>
- 3) Twitter ヘルプセンター  
[https://support.twitter.com/articles/243963-](https://support.twitter.com/articles/243963-4)
- 4) ネタバレ  
<http://zokugo-dict.com/24ne/netabare.htm>
- 5) Janetter,  
<http://janetter.net/jp/>
- 6) Twitter Developers,  
<https://dev.twitter.com/>
- 7) Igo – a morphological analyzer,  
<http://igo.sourceforge.jp/>
- 8) MeCab: Yet Another Part-of-Speech and Morphological Analyzer,  
<http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html>
- 9) 北研二, 津田和彦, 獅々堀正幹: 情報検索アルゴリズム 第 5 版, p18, 共立出版株式会社(2005)