DEIM Forum 2011 F5-5

マイクロブログを対象としたリアルタイムな要約生成システムの試作

坂本　翼† 横山　昌平† 福田　直樹† 石川　博†

† 静岡大学情報学部情報科学科　〒 432–8011 静岡県浜松市中区城北 3–5–1
E-mail: †cs07085@s.inf.shizuoka.ac.jp, ††{yokoyama,fukuta,ishikawa}@inf.shizuoka.ac.jp

あらまし　Twitter を代表とするマイクロブログでは，多数の利用者が自身の気になる話題や考えていることなどを，個別に投稿するため，トピックに関する様々な記事が大量に集まる．これらの記事は，利用者にとって貴重な情報源となるが，膨大な量の記事をすべて読むことは困難である．この課題に対して，トピックの要約を生成することは，利用者が記事を読む負担を軽減する解決方法のひとつである．また，進行中のトピックについて，利用者が情報を得るには，リアルタイムに要約を生成する必要がある．本論文では，ひとつのトピックに関する記事をリアルタイムに取得し，イベントを検出するたびにイベントの要約を生成する．

キーワード　マイクロブログ，バースト検出，複数文書要約

Towards a Real-time Summarizing System for Microblog

Tsubasa SAKAMOTO†, Shohei YOKOYAMA†, Naoki FUKUTA†, and Hiroshi ISHIKAWA†

† Department of Computer Science，Faculty of Informatics，Shizuoka University　Johoku 3–5–1，Naka-ku，Hamamatsu-shi，Shizuoka, 432–8011 Japan
E-mail: †cs07085@s.inf.shizuoka.ac.jp, ††{yokoyama,fukuta,ishikawa}@inf.shizuoka.ac.jp

1. はじめに

近年，Twitter [1] を代表とするマイクロブログは急速に普及し，多くの人々に利用されている．マイクロブログの記事は短いテキストであり，投稿の手軽さから，利用者はそのときの自身の状況や考えていることを投稿することが多い．そのため，マイクロブログでは多くのリアルタイムな記事が投稿され，新しい情報源としてマイクロブログを活用するための研究が多く行われている．

利用者のマイクロブログの利用方法は様々である．特徴的な利用方法のひとつとして，テレビの放送などを見ながらその様子や感想を投稿するといった使われ方がされている．多くの利用者が単一のトピックについて同時に投稿するため，マイクロブログにはそのトピックに関する非常に多くの記事が集まる．これらの記事はそのトピックで起きている出来事や利用者の感想などの情報を含んでおり，共通のトピックの閲覧者に対して貴重な情報源となっている．しかし，膨大な量の記事が短時間に集まるため全ての記事を読むことは利用者にとって困難である．この問題を解決するため，本研究では単一のトピックに関する記事についての要約を生成する．

トピック中でイベントが起こると多くの利用者が発生したイベントについて記事の投稿を行う．しかし，利用者がトピックについて情報を得ようと記事を読む場合，関連記事数の膨大さが問題となる．また，あるトピックの中で，発生しているイベントに関する記事群は多くの利用者が類似した内容を記述するため，情報として非常に冗長である．このような類似内容の関連した記事についての要約では，冗長性を排除することで利用者が記事を読む負担を軽減することができる．トピックの要約を利用者に提示することで，利用者が記事を閲覧する負担を大きく軽減できると考えられる．

また，マイクロブログの記事の特性としてリアルタイム性の高さがあり，マイクロブログは利用者にとってリアルタイムなメディアとして活用されている．そのリアルタイム性の高さから，投稿記事が持つ情報の中には時間の経過とともに情報の価値が低下してしまうものもある．そこで，本研究では利用者が進行中のトピックについてリアルタイムな情報を得られるようにするため，トピックの進行に合わせてリアルタイムに要約を生成することを目的とする．本研究では，トピックの内容はトピック中で断続的に発生するイベントによって構成されるというモデルを考え，トピック中で発生したイベントの要約を時系列順に並べることでトピックの要約を生成する．リアルタイムに要約を生成するため，トピック中でイベントが発生する度にイベントの要約を生成し，利用者に提示する．

リアルタイムにイベントの要約を生成する場合には，あらか

じめどのような内容のイベントが発生するか知ることができない．そのため，過去に起こったトピックの記事を集めて要約をする場合と比べて，要約の質を向上させるためにイベントの特徴となる単語などによるフィルタリングを行うことが困難であるという課題がある．本研究ではマイクロブログのトピック記事群からリアルタイムに検出したイベントの要約システムの構成を提案し，事前にフィルタリングを用意することなくイベントを要約する手法を提案する．また，提案手法によって生成される要約の適切さを実験によって評価する．

本論文の構成は次のとおりである．2 章では，本研究に関連するマイクロブログの特徴について述べる．3 章では，本研究と関連研究の差分について述べる．4 章では，提案手法について概要を述べ，次に各処理についてそれぞれの節で述べる．5 章では，実装した手法について評価実験を行い，6 章で本研究で得られた成果をまとめる．

2. マイクロブログの記事の特徴

マイクロブログサービスの特徴として，投稿する記事の文に字数制限があることが挙げられる．例えば，Twitter では 140 文字以下という字数制限があり，それぞれの記事は非常に短い文であることが多い．また，その字数制限の短さも手伝い，投稿を手軽に行うことができるため，従来のブログなどの記事投稿サービスと比較して記事内容のリアルタイム性が高い．

リアルタイム性の高さからマイクロブログではテレビやＷｅｂ上で配信される動画コンテンツについて，利用者が視聴をしながら自分が感じていることや番組内で起こっていることをマイクロブログに投稿するということが行われる．このとき，Twitter では番組に関する共通のハッシュタグが投稿される記事に付けられることが多い．ハッシュタグは Twitter において利用者が投稿する際に記事に「#（タグ名）」を入力したタグを付けることで発言がグループ化される機能である．ハッシュタグを使って検索を行うことでそのハッシュタグが付けられている記事のみを一覧することができる．

マイクロブログでは単一のトピックに対して多くの利用者が記事を投稿するため，膨大な量の関連記事が集まることになる．しかし，集まった記事はひとつの共通したトピックについて異なる利用者が投稿したものであるため，同じ内容を表している記事が多い．特にテレビや動画コンテンツの実況が行われているような場合では，番組内でイベントが発生すると多くの視聴している利用者がそのことについて言及した記事を一斉に投稿する傾向がある．そのため，イベントが発生すると投稿記事数が急激に増加する．

図 1 は 2010 年 12 月 31 日の 19 時から 23 時 59 分 59 秒までのにハッシュタグ「#nhk_kouhaku61」が付けられた記事に関するグラフであり，投稿記事数の 1 分間ごとの推移を示している．「#nhk_kouhaku61」のタグが付けられた記事はハッシュタグクラウド [2] から取得した．「#nhk_kouhaku61」は，NHK で放送された「第 61 回紅白歌合戦」に関する記事に付けられていたハッシュタグである．図 1 において，記事数が急激に増

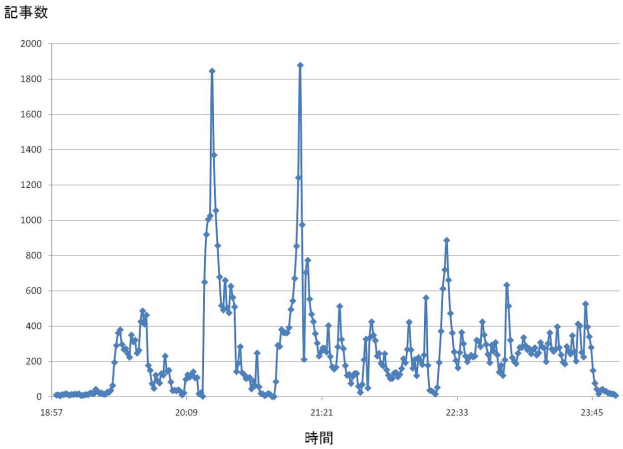

図 1　#nhk_kouhaku61 の投稿記事数の推移

加している時間は「歌手が登場する」,「歌手が歌を歌い始める」などのイベントが発生した時間である．このような投稿記事数の変化を用いてイベントを検出し，それぞれのイベントの要約を生成し時系列順に並べることでトピックの内容を表す要約を生成する．

また，Twitter において RT(Retweet) や QT(Quote Tweet) を記述した記事が投稿されることがある．RT，QT は Twitter において，他人の記事を転送したい場合や引用したい場合に記述される．注目度の高い記事は多くの人に RT されるため，同一内容の記事が多くの人によって何度も投稿されることになる．こうした記事は他人の記事を引用するという特性上，その記事内容が示す時間は実際のトピックが進行している時間と異なる場合がある．例えば，あるイベントについて言及した記事が数分後に多く RT された場合，実際のトピックの進行とは少し遅れてその内容の記事が多く集まることになり，検出したいイベント発生時間と異なる時間に投稿記事数が急増する．そのため，RT や QT を含む記事については，通常の記事とは異なる処理を行う必要があると考えられる．

3. 関 連 研 究

様々な新聞やブログ記事，テレビ番組の報道記事など，構造化されていない複数の関連文書を要約する手法 [3] はインターネットの普及による情報の肥大化に伴い注目され，マイクロブログもその適用対称のひとつである．次に，マイクロブログの要約を行った研究について示す．

3.1 クエリに着目した要約

Sharifi ら [4] はクエリに着目してマイクロブログの要約を行った．Sharifi らの手法では，関連記事から与えられたクエリを含む単語列の頻出パターンを発見することで要約を生成する．Sharifi らの要約手法はトピックの時間的変化を捉えるものではなく，本研究とは目的が異なる．

3.2 トピックのに動的変化に着目した要約

単一のトピックの時間的な変化に着目してマイクロブログの要約を行った研究として，高村らの研究 [5] がある．高村らの手法では，トピックに関する記事の中から他の記事との被覆度

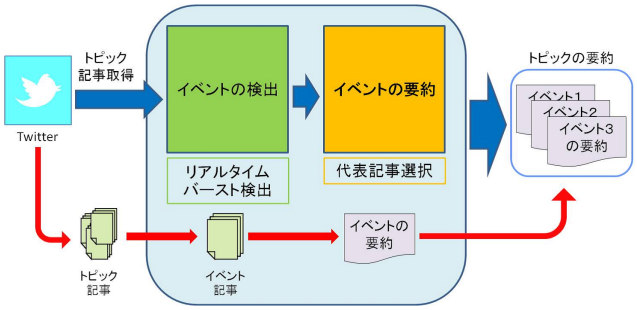

図 2　要約システムの概要.

が大きい記事を代表記事として少数選択し，時系列順に並べることでトピックの要約を行っている．代表記事を決定するため，各記事間の被覆度を表す記事間係数(注1)を指標として求めている．トピックの要約を生成するため，トピックに関する記事の中から他の記事との記事間係数の合計が最大となるような記事を代表記事として選択する．高村らは，マイクロブログにこれを適用し，マイクロブログ記事の持つ時間情報に着目した手法を提案した．この代表記事選択による要約手法は，高村らの先行研究 [6] で行われていたものであり，選択された代表記事によって他の全ての記事の内容ができる限り表現される，というモデルによってトピックは要約される．また，多様な記事が含まれるマイクロブログから代表記事を選択し，要約を生成する場合，生成したい内容を適切に表さないノイズとなる記事が多く含まれる. [5] では，トピックに関連する記事から要約を構成する記事をフィルタリングによって抽出している．例えば高村らは，サッカーの試合で起こった出来事を要約として生成することを目的とし，選手名もしくはチーム名とサッカー用語の両方を含む記事だけをフィルタリングによって抽出し，要約の要素として用いた．

本研究ではトピックで発生したイベントの要約を行うために，高村らの用いた代表記事を選択するという手法を用いる．イベントが発生している期間のトピックに関する記事はそのイベントの内容を表している可能性が高いと考えられる．ただし，本研究ではリアルタイムにイベントの要約を生成することを目的とするため，事前に適切なフィルタを用意するのは困難である．そのため，本研究ではフィルタリングを用いず，マイクロブログのトピックの記事群から適切な要約を生成する手法を提案する．

4. 提案手法

図 2 に要約システムの概要を示す．まず，要約を行う対象となるトピックに関する記事を Twitter からハッシュタグを用いてリアルタイムに取得する．次に，取得する記事の時間あたりの投稿数を用いて，リアルタイムにバーストを解析することでトピック中のイベントを検出し，バーストが発生している区間の記事群をイベント関連記事群として抽出する．抽出したイベント関連記事群から代表記事を選択することでイベントの内容の要約とし，生成した要約を随時時系列順に並べていくことでトピック全体の要約として利用者に提示する．

4.1 トピックの記事の取得

本研究では，リアルタイムに記事を取得するために，Twitter より提供されている Streaming API [7] を用いる．要約対象とするトピックに関連した記事を取得するために，Streaming API の status/filter というキーワードフィルタを用いる．ここでは，単一のトピックに関する記事のみを集めるために Twitter のハッシュタグをキーワードとしてフィルタリングした記事を取得する．対象とするトピックに関するハッシュタグをキーワードとしてフィルタリングした記事を取得することでそのタグが付けられたトピックに関連する記事のみを取得できる．一部の利用者はハッシュタグを付けないため，関連する記事をすべて取得することはできないが，本研究では要約が目的であるので，一定以上の記事数があればトピックの内容を表すために十分であると考えられる．

4.2 イベントの検出

イベントの検出は時間当たりの投稿記事数を用いてバースト解析を行う．リアルタイムにバーストを検出する方法として蝦名らの用いたリアルタイムバースト検出法 [8] を用いる．蝦名らの手法は従来のリアルタイムバーストの解析手法のように一定期間毎にバーストを解析するのではなく，ドキュメントの発生ごとにバーストを解析する．また，短時間に大量のドキュメントが発生した場合でも高速性を保つアルゴリズムを用いている．そのため，短期間に大量のドキュメントが発生するマイクロブログの解析に適していると考えられる．また，バーストの閾値を一定値に定めず，ドキュメントの発生頻度が低い場合でも直前の状態と比較して発生頻度が急激に高くなっていればバーストが検出可能である．マイクロブログでの記事の発生頻度はトピックごとに異なり，同じトピック内でも時間と共に変化するため，一定の閾値を用いる手法と比較して，直前の状態との比較によってバーストを解析することは，より多くのイベントを発見する場合に効果的である．

本論文では，バースト解析によってバースト検出の開始時間からバースト検出の終了時間までのバースト発生区間をひとつのイベントとして要約を行う．バーストが発生している間に新たに到着した記事を随時イベント関連記事として抽出する．このとき，抽出する各記事に対して，イベントの要約のための前処理を行う．記事の前処理の流れを示す．

(1)　記事のフィルタリング
(2)　記事の形態素解析

これらの記事のフィルタリングと記事の形態素解析を，記事を抽出するたびに行う．

まず，要約の対象とする記事群を日本語で書かれた記事のみを使用するようにフィルタリングする．また，記事のテキスト中に含まれるハッシュタグや Web ページへのリンク URL，など本文の内容を表さないものは除外する．さらに，リプライに含まれるユーザ名についても除外する．リプライとは，Twitter において他の利用者のユーザ名を記述することで，その利用者

(注1)：高村らの研究ではひとつの記事はエントリと呼ばれており，記事間係数はエントリ間係数と呼ばれている．

に返信できる機能である．また，記事に含まれる全角英数字は全て半角英数字に変換した．

記事の形態素解析には MeCab [9] を用いる．各記事を MeCab の形態素解析によって解析した結果の単語群から内容語 (名詞・動詞・形容詞) を抽出して記事情報として登録する．このとき，登録する単語の中から，非自立語，接尾語，代名詞はストップワードとして除く．また,「する」,「なる」といった文書の特徴となりづらい単語や，記号のみで構成されている単語もストップワードとする．また，平仮名やカタカナ一字で抽出された単語もストップワードとする．

これらの前処理をイベントの要約の前に，リアルタイムに記事が到着するたびに行うことで，イベントの要約を開始してからの処理にかかるコストを軽減する．バーストの検出が終了したときに，それまでのバースト期間中の記事をイベント関連記事とし，イベントの要約を開始する．

4.3 イベントの要約

要約手法には，高村らの要約対象とする記事群から代表記事を選択するという手法を用いる．本研究では，抽出したイベント関連記事群から代表記事を選択する．代表記事の選択には，選択した記事が他の記事の内容をどれだけ被覆しているかの度合いを測る記事間係数を用いる．記事間係数の計算には，記事内の単語集合の被覆度を用いる．

代表記事を決定するために，各記事の 記事間係数を計算する．まず，高村らの定義した記事間係数を用いた．高村らは記事間係数 sim_{ij} を記事 i が記事 j を被覆する度合いとして式 (1) のように定義している．

$$sim_{ij} = \frac{|d_i \cap d_j|}{|d_j|} \tag{1}$$

d_i, d_j はそれが含む内容語の集合である．選択記事と他の記事との記事間係数の合計が最大となるものを代表記事として決定する．

しかし，式 (1) の記事間係数を用いる場合，ノイズとなる記事が含まれないことが前提となり，選択した記事の冗長さについて考慮できない．多くの単語を含むほど記事間係数に有利に働くため，余分な単語を多く含む記事が選択される可能性がある．本研究ではリアルタイムに要約を行うため，このような冗長な記事をフィルタリングによって除外することが困難である．イベント関連記事にはイベント関係のない，ノイズとなる記事が含まれている場合があり，代表記事として適切でない記事が選択されてしまう可能性がある．ここでの目的は，イベントの内容をよく表す代表記事を選択することであるため，イベントの内容を表す重要単語をより多く含む記事かつ余分な単語を含まない記事を代表記事として選択したい．

そこで，本手法ではイベントにおける重要単語を考慮した記事間係数を用いる．イベント内で出現頻度の高い単語はそのイベントについて重要な単語であると考えられるため，重要単語はイベント内での出現回数が閾値以上である単語とした．イベント関連記事に含まれる単語について，単語の総出現数と単語の種類の商 (小数点以下を切り捨て) を閾値とし，出現数が閾値以上の単語を重要単語とした．ただし，出現回数が 1 回の単語の多くはノイズとなるため計算時に除いた．

決定した重要単語を用いて，本手法では重要単語集合との被覆度を考慮した記事間係数の式 (2) を用いる．

$$sim_{ij} = \frac{|d_i \cap d_j \cap I|}{|d_i \cup I|} \tag{2}$$

I は重要単語の集合である．重要単語集合に含まれない単語はイベントを表すために余分な単語として扱われる．式 (2) により要約に必要な単語とそうでない単語を判定することですることで，重要単語を含み，他の記事との関連が強い記事が代表記事として選択される．また，選択記事が重要単語ではない余分な単語を含んでいる場合には分母が大きくなるため，記事間係数が小さくなり，代表記事として選択されない．

しかし，頻出単語の中には特定のイベントだけではなく，トピック全体を通して頻出する単語があり，このような単語はイベントに関して重要ではないと考えられる．そこで次に，イベントに着目して TF-IDF を計算し，単語に idf による重み付けを行い，重要単語を決定した．現在のイベント内におけるある単語 $term$ の重要度 $tfidf(term)$ を以下のように定めた．

$$tfidf(term) = tf * log(\frac{N}{df}) \tag{3}$$

ここで，tf は現在のイベントにおける $term$ の出現数，N はトピックについて発生している総イベント数，df は $term$ が頻出したイベント数とする．各イベントにおいて出現回数が閾値以上の単語を頻出単語として登録し，それぞれの単語が頻出したイベント数を記録する．ここでの頻出単語を決める閾値は，ひとつ前に述べた重要単語決定法と同様に，単語の総出現数と単語の種類の商 (小数点以下を切り捨て) とした．各単語の $tfidf$ を計算し，$tfidf$ の値が閾値以上の単語を重要単語とした．ここでの閾値は単語の $tfidf$ の合計と単語の種類の商 (小数点以下を切り捨て) とした．ただし，出現回数が 1 回の単語の多くはノイズとなるため計算時に除いた．

以上で決定した重要単語集合を用いて各記事の記事間係数を計算し代表記事を決定する．

5. 評価実験

5.1 実験設定

提案した手法について，比較実験を行い評価する．イベントの要約について，比較手法として式 (1) を用いたもの (手法 1)，提案手法として式 (2) を用いたもの (手法 2) と式 (3) を用いて単語の TF-IDF を考慮したもの (手法 3) を用いる．これら 3 つの手法による要約を生成し，比較実験を行った．

要約するトピックとして,「第 61 回 NHK 紅白歌合戦」に関する Twitter の記事を対象とした．紅白歌合戦の番組は 19 時 30 分から 23 時 45 分頃まで放送されており，関連するハッシュタグ「#nhk_kouhaku61」によって番組放送前の 19 時 00 分 00 秒から番組終了後 23 時 59 分 59 秒までの時間に投稿された記事 75787 件を対象とした．ただし，記事中に RT，QT を含む記事は今回は処理の対象とはしなかった．

まず，収集した記事を投稿時間順にシステムに入力してイベ

表 1　正解要約と各手法の実験データ

	抽出単語数	正解要約との単語の一致数
正解要約	218	—
手法 1	356	115
手法 2	264	119
手法 3	252	110

表 2　評 価 結 果

	recall	L	調和平均
手法 1	0.528	0.323	0.401
手法 2	0.546	0.451	0.494
手法 3	0.505	0.437	0.468

ント検出を行った．その結果，全部で 49 イベントが検出され，それぞれの手法によるイベントの要約を生成した．次に，実験協力者 7 人によって，検出された 49 イベントについて，それぞれのイベント関連記事群から人手で代表記事をひとつ選択することで，正解となる要約を作成した．この際，実験協力者にはイベントの内容を最もよく表していると思う記事を選択するように指示をした．

作成した正解要約と各手法によってシステムが生成した要約に含まれる単語を比較することで評価を行った．通常の文書要約においてよく用いられる自動評価手法として ROUGE [10] がある．ROUGE は，正解要約に含まれる単語のうち，システムが生成した要約が含む単語の割合を表す．ここでは，ROUGE をイベントの要約結果に適用し，再現率 (recall) として用いる．しかし，ROUGE は要約結果の網羅率について評価をすることができるが，冗長性については評価をすることができない．本論文では余分な単語を含まないような記事を代表記事として選択することを目的としているため，生成した要約の冗長性についても評価指標が必要である．そこで，システムの要約結果の長さに着目した評価指標 (L) を用いる．L はシステムの要約結果が含む単語のうち，正解要約が含む単語の割合を表す．また，総合評価として recall と L の調和平均を計算した．以下に各評価指標の計算式を示す．

$$recall = \frac{\sum_{i=1}^{n} |R_i \cap S_i|}{\sum_{i=1}^{n} |R_i|} \tag{4}$$

$$L = \frac{\sum_{i=1}^{n} |R_i \cap S_i|}{\sum_{i=1}^{n} |S_i|} \tag{5}$$

$$\text{調和平均} = \frac{2 \cdot recall \cdot L}{recall + L} \tag{6}$$

ここで，n はイベントの個数，R_i は i 番目のイベントにおいて正解要約が含む内容語の集合，S_i は i 番目のイベントにおいてシステムの要約結果が含む内容語の集合を表す．

5.2 結　　果

表 1 に正解要約と各手法のイベントごとの抽出単語数の合計と正解要約との単語の一致数の合計の実験データを，表 2 に各手法の評価結果を示す．表 2 より，L は手法 1 と比較して単語の重要度を考慮した手法 2，手法 3 の方が評価が良い．その結果，調和平均による評価が向上している．また，recall についてもわずかに評価の向上が見られた．表 1 より，手法 1 と比較して手法 2 ではより少ない単語数で，より多くの正解要約と一致する単語を含んでいる．このことから，提案手法である手法 2，手法 3 では手法 1 と比較して，必要な単語を含み，余分な単語を含まない記事を選択できたと考えられる．

表 3　イベント A の要約結果

	要約
正解要約	紅白始まったー (*^^*) 奈々ちゃん奈々ちゃん！
手法 1	はじまりましたね紅白。この観客席のジャニ追っかけと AKB 追っかけと奈々様追っかけの比率はどんなものかしらと。高比率ならそれこそ CD 業界の消費モデルそのものでは
手法 2	紅白始まったー (*^^*) 奈々ちゃん奈々ちゃん！
手法 3	紅白始まったー (*^^*) 奈々ちゃん奈々ちゃん！

表 3 に手法 1 と手法 2 及び手法 3 の間で特に大きな変化が見られたイベント A の正解要約とシステムの要約結果を示す．イベント A は紅白歌合戦の開始時のイベントであり，歌手である水樹奈々に関する記事が多く見られた．手法 1 では単語を多く含む記事が有利に働き，結果として正解要約とは意味的に内容の異なる記事が選択されている．このことから，提案手法によって，重要な単語とそうでない単語を判定することが要約の質の向上について有効に働いていると考えられる．

手法 2 と手法 3 について，手法 2 と手法 3 では 49 イベント中 14 イベントで生成した要約が異なった．表 2 より，手法 2 と比較して TF-IDF による単語の重み付けを行った手法 3 の評価が下がっている．この原因として TF-IDF が有効に働いた場合とそうでなかった場合があったことが考えられる．表 4 に TF-IDF が有効に働いたイベント B の例を，表 5 に TF-IDF が有効に働かなかったイベント C の例を示す．

イベント B は歌手グループである Perfume が登場して歌唱していたときのものである．イベント B において,「紅白」という単語の出現回数が多いため，単純に出現回数を指標として重要単語を決定した手法 2 では「紅白」を含む記事を選択した．しかし,「紅白」という単語自体は他の多くのイベントでも頻出しており，イベント B 自体に関連に強いものではないため正解要約には含まれなかった．一方，手法 3 では「紅白」を含まない記事が選択され，TF-IDF による単語の重み付けがが有効に働いたと考えられる．

また，イベント C は歌手の植村花菜が楽曲「トイレの神様」を歌唱する直前である．イベント C において，まもなく楽曲が始まるという意味で「くる」や「来る」などの単語が多く見られ，正解要約でも「くる」を含む記事が選択されている．しかし,「くる」という単語は他の多くのイベントでも頻出しているため，手法 3 では「くる」の単語の重みが小さくなり,「くる」という単語を含まない記事が選択されている．本実験では，多くのイベントで TF-IDF による単語の重み付けは有効に働かず，結果として，手法 3 の recall の評価が下がってしまったと考えられる．

手法 3 では，IDF の D をイベント関連記事群とし，各イベントの中でも頻出している単語のみを重み付けの対象とした．

表 4　イベント B の要約結果

	要約
正解要約	Perfume のダンスには感心。良く、踊れるわぁ
手法 2	PERFUME と聞いて、ちょっこし紅白へ (笑)。これがエヴァ風衣装？
手法 3	PERFUME の衣装も、可愛い！着てみたい !! .

表 5　イベント C の要約結果

	要約
正解要約	トイレの神様くるー。
手法 2	トイレの神様くる？
手法 3	トイレの神様♪

しかし，IDF による重み付けが有効な場合とそうでない場合があることから，対象とする IDF の D の範囲を検討する必要があると考えられる．例えば，今回の実験では「紅白」のようなトピックに関連して普遍的に表れる名詞に対して TF-IDF の重み付けは有効に働いていたが,「くる」のような一般語に対しては有効に働いていない．そのため，TF-IDF による重み付けについて，一般語とトピックに特有な単語の場合で処理を分けることなどが考えられる．

6. おわりに

本論文では，Twitter の記事から単一のトピックに関して抽出された記事を用い，リアルタイムにイベントを検出して要約対象とし，随時イベントの要約を生成して並べることでリアルタイムにトピックの要約を生成するシステムの構成を提案した．また，重要単語を考慮することで事前に適切なフィルタを用意することなく，要約を生成する手法を提案した．また，提案手法と重要単語を考慮しない場合による要約結果を実験によって比較し，提案手法によって要約結果の質が向上することを確認した．

今後の課題として，イベントの要約内容の補完が挙げられる．本論文ではひとつのイベントに対して，代表記事をひとつだけ選択することで要約を生成したが，ひとつのイベント内で複数の内容を持つ場合など，ひとつの代表記事だけではイベントの内容を表すのに不十分な場合がある．このような場合に，ひとつではなく複数の代表記事を選択することで，イベントの内容を補完することができると考えられる．そのため，ひとつのイベントに対して，必要であれば複数の代表記事を選択することで，イベントの要約の質を向上させることを検討している．

システムとして実際に利用する場合には，提示する要約内容が利用者にとって有益であるかを評価する必要がある．提案手法ではできる限りトピック中のイベントを検出することを目的としたため，トピックの進行とともに提示される要約が長くなってしまう．利用者によって詳細な要約が求められる場合や短い要約が求められる場合があることが考えられるため，システムとして提示する場合に記事数の総量を指標としたイベントの重要度などを用いて，利用者の目的に沿った長さの要約を提示することなどを検討している．また，本論文では記事間係数による単語の類似性に基づいた手法を用いて，イベントの主要な内容のみを要約結果として出力している．しかし，実際にマイクロブログで投稿される記事には，イベントの内容に関して利用者の様々な意見がある．このような記事の相違性に着目して要約をする手法についても今後の課題である．

また，マイクロブログにおける記事内容の表記揺れへの対処が必要である．マイクロブログの記事は一般の人が自由に記述できるため表記揺れが多く発生する．例えば,「言う」という単語の表記揺れとして「いう」や「云う」などがある．表記揺れが発生することによってひとつのイベントの関連記事から複数の代表記事を選択する場合に，単語の表記揺れのために，同じ内容を示す記事が異なる選択されてしまう可能性がある．このような表記揺れに対して単語の代表表記を用いて表記揺れを緩和することが考えられる．代表表記を付与する日本語の形態素解析機として Juman [11] などがあり，一般的な語句の表記揺れは緩和できる．しかし，人物の呼び名などについては限界があるため，それぞれの記事で異なる表記であっても同じ意味を表す単語を判定する必要がある．

また，本論文ではシステムや要約結果のリアルタイム性について検討を行っていない．実際にリステムを利用するに当たって，システムの要約生成のリアルタイム性が実用的なものであるかどうかを検討する必要がある．リアルタイムに要約を生成する場合とトピック終了後に要約を生成する場合の要約結果にどのような差異があるかなどについても今後の検討課題である．また，紅白歌合戦について要約を行ったが，スポーツの試合中継に関する記事など，他のトピックについても提案手法が有効に働くかどうかを今後検討する必要がある．

謝辞 本研究の一部は科学研究費補助金（課題番号 22650012）の助成，および日立 uCSDP アカデミック支援プログラムの支援による．

文　献

[1] Twitter，http://twitter.com/.
[2] ハッシュタグクラウド，http://hashtagcloud.net/.
[3] Inderjeet Mani 著, 奥村学・難波英嗣・植田禎子訳．自動要約（共立出版，2003），pp. 165–204.
[4] Beaux Sharifi，Mark-Anthony Hutton，and Jugal Kalita. Summarizing microblogs automatically. In Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'2010)，pp.685–688，2010.
[5] Hiroya Takamura，Hikaru Yokono and Manabu Okumura. Summarizing microblog stream．人工知能学会研究会資料 SIG-SWO-A1001-03.
[6] 高村大也，奥村学．施設配置問題による文書要約のモデル化．人工知能学会論文誌，Vol. 25，No. 1，pp.174-182，2010.
[7] Twitter Streaming API, http://dev.twitter.com/pages/streaming_api.
[8] 蝦名亮平，中村健二，小柳滋．リアルタイムバースト検出手法の提案．日本データベース学会論文誌，Vol.9，No.2 November 2010.
[9] MeCab，http://mecab.sourceforge.net/.
[10] Chin-Yew Lin. ROUGE: a package for automatic evaluation of summaries. In Proceedings of the Workshop on Text Summarization Branches Out，pages 74–81，2004.
[11] 日本語形態素解析システム Juman, http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman.html.