Vol.2012-IFAT-105 No.14
Vol.2012-NL-205 No.14
2012/1/21

時間的距離に注目したTwitterからの関連単語抽出

白木原 渉[†1] 大石哲也[†1] 長谷川隆三[†2]
藤田博[†2] 越村三幸[†2]

Web 検索では，検索エンジンによって得られた結果がユーザの必要としている情報ではないことがしばしばある．この問題を解決する一つの方法として検索エンジンに与えるクエリを改善するクエリ拡張がある．しかし，これまでの多くのクエリ拡張は，最新の話題を検索するために有効な語を生成できていない．本論文では，最新の話題を検索するために有効な語を生成するための方法として，情報のリアルタイム性を持つマイクロブログである Twitter の投稿（ツイート）を使い，ツイートされた時刻の時間的距離に注目した関連単語抽出アルゴリズムを提案する．このアルゴリズムは重要な語の近くに出現する単語は重要であるという考えに基づいている．更に，このアルゴリズムが実際にリアルタイム性の高い関連語を抽出できることを実験により示す．

RELATED WORD EXTRACTION FROM TWITTER USING TEMPORAL DISTANCE

Wataru Shirakihara,[†1] Tetsuya Oishi,[†1]
Ryuzo Hasegawa,[†2] Hiroshi Fujita[†2]
and Miyuki Koshimura [†2]

In web retrieval, it is often the case that the results given by search engines are not just what we want. To solve this problem there have been many studies on improving queries to be submitted to search engines. However, we cannot get the hot topics with the improved queries. In this paper, we propose the method of related word extraction from twitter using temporal distance. The basic idea of this method is that the words found near important words are also important. We also confirmed that this method generates the proper queries for retrieving the hot topics.

1. はじめに

知識の宝庫である WWW は，研究や仕事の支援のためだけでなく，日々の生活においても広く利用されている．

WWW の利用状況は，Web 検索の他，ブログや SNS による情報の発信や収集，オンラインショッピングなど多岐に渡っている．その中でも，Google[7] や Yahoo! Japan[8] などに代表される検索エンジンの利用に注目した．

検索エンジンに検索要求（クエリ）を与えることにより，クエリを含む Web ページが取得できる．しかし，クエリによっては適切な Web ページが得られないこともある．一般に，クエリが少数の単語しか含まなければ，多量に Web ページが返される．逆に，クエリが多数の単語を含めば，該当 Web ページがまったく得られないこともある．これは検索エンジンが基本的に AND 検索（クエリに含まれる単語をすべて含むような Web ページの検索）を行うためである．

これらの問題を解決する方法として，クエリ拡張により再検索を行う手法がある．クエリ拡張により再検索を行う手法とは，ユーザが与えたクエリ（「初期クエリ」と呼ぶ）に適切な単語を追加して新たなクエリ（「拡張クエリ」と呼ぶ）を生成し，再度検索を行う手法である．初期クエリに含まれる単語数が少ない場合，その検索結果は膨大になることが多いが，拡張クエリを用いた AND 検索により，検索結果の数を減らすことができる．

クエリ拡張においては，新たに追加される単語の選定法が問題となる．この単語が適切に選定されなければ，拡張クエリによる検索結果の改善は望めない．

そこで我々は，クエリに関連する単語を抽出する関連単語抽出アルゴリズム（RWEA：Related Word Extraction Algorithm）を考案した[1]．従来の TFIDF や RSV（Robertson's Selection Value：適合文書中の単語を高く，不適合文書中の単語を低く評価する）のように，単に，文書中の単語の出現頻度に基づく方法とは異なり，RWEA は単語やセンテンス間の距離に着目した重みを単語の出現頻度に乗じる方法を採っている．RWEA は「キーワード A の近くに出現する単語ほど A と関連が強い」という考えに基づき，キーワード群に関連する単語をテキスト中から抽出する．

†1 九州大学大学院システム情報科学府
Graduate School of Information Science and Electrical Engineering, Kyushu University
†2 九州大学大学院システム情報科学研究院
Faculty of Information Science and Electrical Engineering, Kyushu University

Vol.2012-IFAT-105 No.14
Vol.2012-NL-205 No.14
2012/1/21

しかし，これらの手法では，普遍的な関連語は得られるが，最新の話題を反映した関連語は得られにくい．そこで，本研究では RWEA を応用し，クエリ拡張において新たに追加される単語を，よりリアルタイム性の高い単語にする手法を考案した．具体的には，関連単語抽出に用いる文書として，リアルタイム性を持つ Twitter のツイートを利用し，RWEA における単語やセンテンス間の距離を，ツイート間の時間的距離に置き換えることで，RWEA を Twitter の時系列データに適用できるようにした．

第２節で関連研究について述べ，第３節で提案手法について述べる．第４節では，この手法で実験を行って得られた結果を示し，その結果に対する考察を行う．最後に，第５節で結論と今後の課題を述べて本論文をまとめる．

2. 関 連 研 究

現在までに，関連単語の抽出やそれを利用した検索システムに関する研究がいくつか行われてきた．例えば，大塚と喜連川[2] は大規模なアクセスログから関連語を抽出する研究を行った．彼らは，統計的に偏りなく抽出されたユーザの Web 検索の際の検索単語とそれによって閲覧したページを解析することで，関連単語を得た．この研究は，既存の検索エンジンで用いている不特定多数のユーザによって検索された頻度や，クリックされた結果などを組み合わせて関連語を提案する方法と同等またはそれ以上の精度を示している．

岡部と山田[3] は，ユーザからの最小のフィードバックにより検索単語の拡張を行い，検索効率の向上を図ったシステムを提案した．このシステムはトランスダクティブ学習という機械学習法を用いて実現されている．この手法により，トランスダクティブ学習を用いない従来手法と比べ再現率が約 0.07 向上している．

正田ら[4] は，ユーザが与えたクエリでの検索結果の上位 R 件を適合文書，それ以下を不適合文書として RSV を用いてクエリ拡張を行い，新たなクエリの重みにより初期の検索結果をソートする方法を提案している．この研究で，RSV など実験で用いたパラメータの最適な組み合わせを発見し，その組み合わせを用いた結果，精度は向上している．

和泉と西山[5] は，マイクロブログの時系列情報を利用して関連語を抽出した．彼らは，マイクロブログ内で，それぞれの単語がある一定時間内に出現した回数の増減を調べ，その増減に対して相関係数の高い２語を関連語とした．

3. 提 案 手 法

本節では，我々が考案した，時間的距離に基づく関連単語抽出アルゴリズムについて説明する．

3.1 アルゴリズムの概要

本アルゴリズムは，あるキーワード k に関連し，なおかつリアルタイム性の高い単語群を一定期間内の全ツイートの中から抽出し，それらを出力するものである．

単語群の抽出では，k を含むツイートと他のツイートとの時間的距離に着目し，これを基準に各単語を評価し，結果として出力する単語群を形成する．

3.1.1 ツイート間の時間的距離の重要性

上述の通り，本アルゴリズムでは単語が含まれるツイート間の時間的距離を重要視している．具体的には,「ある単語 A があるツイート B に出現した場合，ツイート B 付近のツイートに出現している単語ほど A に関連性が強い単語であろう」という考えに基づいている．

3.1.2 準　　備

アルゴリズムについて説明する前に，準備として以下の記号を定義する．

- K：関連単語を抽出するための基になるキーワード
- T：一定期間内の全ツイート
- $k_i(i = 1, \cdots, m)$：全ツイート T で出現する m 個のキーワード K（T での出現順に $k_1, k_2, k_3, \cdots k_m$）
- $p_j(j = 1, \cdots, n)$:全ツイート T で出現する n 個のツイート（出現順に $p_1, p_2, p_3, \cdots, p_n$）
- $t_j(j = 1, \cdots, n)$：全ツイート T で出現する n 個のツイートの投稿時刻（時刻の早い順に $t_1, t_2, t_3, \cdots, t_n$）
- $w_l(l = 1, \cdots, o)$：全ツイート T で出現する o 個の単語（出現順に $w_1, w_2, w_3, \cdots, w_o$）
- $c_p(p = 1, \cdots, q)$:全ツイート T で出現する q 個の単語（$c_1, c_2, c_3, \cdots, c_q$ は各々異なる）

c_p は，w_l 内に現れる単語の重複を除去したものである．w_l は全ツイート T を高速形態素解析システム MeCab で形態素解析し，その結果得られた名詞のみを抽出し，それらを出現順に並べたものである．ただし，名詞が連続して現れた場合は，それらを連結して１つの名詞とみなす．各ツイート内はそのツイート内に含まれる名詞のみを出現順に並べ，保持している．また，形態素解析した結果，同一単語が複数回出現してもそれらは別のものとみなす．

3.2 単語の評価

全ツイート T 内で出現する単語の評価は以下の手順で行う．

(1) $k_i(i = 1, \cdots, m)$ を基準に $p_j(j = 1, \cdots, n)$ の基礎評価値 $(BasicValue)BV(p_j)$ を計算．

Vol.2012-IFAT-105 No.14
Vol.2012-NL-205 No.14
2012/1/21

(2) $BV(p_j)$ の平滑化.

(3) 単語の出現頻度による最終評価値 (Score)$S(c_p)$ を計算.

3.2.1 $BV(p_j)$ の算出

本アルゴリズムはキーワード K を基に，K と全ツイート T で出現するツイート間の時間的距離を用いて単語の評価を行うことを目的としており，はじめにその基本となる評価値を求める.

まず，キーワード k_i による p_j の評価値を $BV_{k_i}(p_j)$ と表記する．この $BV_{k_i}(p_j)$ を次式により求める．ここで j_0 は，k_i を含むツイート p_{j_0} の添数とする.

$$BV_{k_i}(p_j) = t_n - |t_j - t_{j_0}| \tag{1}$$

それぞれの p_j において $BV_{k_i}(p_j)$ の和を $BV(p_j)$ とする．即ち，$BV(p_j)$ は次式により求める.

$$BV(p_j) = \sum_{i=1}^{m} BV_{k_i}(p_j) \tag{2}$$

表 2 に $BV(p_j)$ を求める例を示す．例で用いる記号は表 1 で示す.

3.2.2 $BV(p_j)$ の平滑化

前節で求められた $BV(p_j)$ は，ツイート p_j が全ツイート T で出現する位置を考えると不公平である．全ツイート T のうち，端に出現するツイート $p_j(j = 1, n)$ については $1 \leq BV_{k_i}(p_j) \leq t_n$ であるのに対し，全ツイート T のうち，中央に出現するツイート $p_{n/2}$ については $t_n - t_{n/2} \leq BV_{k_i}(p_{n/2}) \leq t_n$ である．よって，$BV(p_j)$ のとり得る値の期待値はツイート p_j の投稿時刻 t_j によって異なる．このままでは T の中央に出現するツイートは必然的に与えられる評価値が大きくなってしまい，公平な評価はできない.

この問題を解決するために，求めた $BV(p_j)$ をツイート p_j の投稿時刻 t_j での評価値の期待値を用いて平滑化を行う．ツイート p_j の投稿時刻 t_j での評価値の期待値（Expectation Basic Value）$EBV(t_j)$ は以下の式で求められる.

$$EBV(t_j) = \frac{1}{2t_n}\{t_n(t_n + 2t_j - 1) - 2t_j(t_j - 1)\} \tag{3}$$

ここで，t_n は全ツイート T 内の最後のツイートの投稿時刻である.

本アルゴリズムでは，平滑化後の評価値を $EBV(p_j)$ とし，以下の式で計算する.

$$EBV(p_j) = \frac{BV(p_j)}{EBV(t_j)} \tag{4}$$

表 1 例で用いる各記号の詳細

K : A

T : AGB. EG. AFC. FH. DE.

k_i

k_1	k_2
A	A

k_1 は最初のセンテンスに現れる「A」,
k_2 は 3 番目のセンテンスに現れる「A」である.

p_j

p_1	p_2	p_3	p_4	p_5
AGB	EG	AFC	FH	DE

t_j

t_1	t_2	t_3	t_4	t_5
1.0	1.5	3.0	5.0	6.0

w_l

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}
A	G	B	E	G	A	F	C	F	H	D	E

c_p

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
A	B	C	D	E	F	G	H

表 2 時間的距離による単語の評価例

	k_1	k_2			
	A	A			
	p_1	p_2	p_3	p_4	p_5
	A G B	E G	A F C	F H	D E
t_j	1.0	1.5	2.0	5.0	6.0
$BV_{k_1}(p_j)$	6.0	5.5	5.0	2.0	1.0
$BV_{k_2}(p_j)$	5.0	5.5	6.0	3.0	2.0
$BV(p_j)$	11.0	11.0	11.0	5.0	3.0
$EBV(t_j)$	3.50	3.79	4.17	4.17	3.50
$EBV(p_j)$	3.14	2.90	2.64	1.20	0.86

Vol.2012-IFAT-105 No.14
Vol.2012-NL-205 No.14
2012/1/21

表 3 $S(c_p)$ 算出までの各値の例

	p_1			p_2		p_3			p_4		p_5	
	w_1 A	w_2 G	w_3 B	w_4 E	w_5 G	w_6 A	w_7 F	w_8 C	w_9 F	w_{10} H	w_{11} D	w_{12} E
$EBV(p_j)$	3.14			2.90		2.64			1.20		0.86	
$EBV(w_l)$	3.14	3.14	3.14	2.90	2.90	2.64	2.64	2.64	1.20	1.20	0.86	0.86

	c_1 A		c_2 B	c_3 C	c_4 D	c_5 E		c_6 F		c_7 G		c_8 H
	w_1	w_6	w_3	w_8	w_{11}	w_4	w_{12}	w_7	w_9	w_2	w_5	w_{10}
$EBV(w_l)$	3.14	2.64	3.14	2.64	0.86	2.90	0.86	2.64	1.20	3.14	2.90	1.20
$tf(c_p)$	2		1	1	1	2		2		2		1
$AveEBV(c_p)$	2.89		3.14	2.64	0.86	1.88		1.92		3.02		1.20
$V(c_p)$	0.19		0.54	0.04	0.00	0.00		0.01		0.44		0.00
$S(c_p)$	0.55		1.70	0.11	0.00	0.00		0.02		1.33		0.00
順位	3		1	4	-	-		5		2		-

表 2 に $EBV(t_j)$ を計算し，$EBV(p_j)$ を求めるまでの例を示す．$EBV(t_j)$ は投稿時刻 t_j での評価値の期待値であるので，中心付近 ($t_j = t_n/2$ 付近) の値が大きくなっている．これにより $BV(p_j)$ では出現位置により不公平な評価を得ていたものが平滑化されている．

3.2.3 $S(c_p)$ の算出

本アルゴリズムはツイート間の時間的距離を最重要視して評価値計算を行っているが，それに加えて TF/IDF 法などで用いられるテキスト内での単語の出現頻度 $TF(TermFrequency)$ の概念も考慮する．つまり，全ツイート T 内で複数回出現した単語はある程度重要視すべきである，ということである．

ここで以下のように定義する．

$$\forall w_l \in p_j (EBV(w_l) = EBV(p_j)) \tag{5}$$

まず，全ツイート T 内で複数回出現した単語について，その単語の $EBV(w_l)$ の平均値 $AveEBV(c_p)$ を以下の式を用いて計算する．

$$AveEBV(c_p) = \frac{\sum_{w_l = c_p} EBV(w_l)}{tf(c_p)} \tag{6}$$

ここで，$tf(c_p)$ は，全ツイート T 内での単語 c_p の出現回数である．例えば，単語 w_a と単語 $w_b (a \neq b)$ が同じ単語 (c_c) だったとすると，$AveEBV(c_c) = (EBV(w_a) + EBV(w_b))/2$ となる．無論，T 内で出現が 1 回のみの単語 $w_l (= c_p)$ に関しては $AveEBV(c_p) = EBV(w_l)$ である．

さらに $tf(c_p)$ を用いて以下の式で表される重み $V(c_p)$ を計算する．

$$V(c_p) = e^{-(rank)^2/t_n} \times (1 - e^{-tf(c_p)}) \tag{7}$$

ここで，$rank$ は，c_p を $AveEBV$ の値で降順に並べた時のランキングである．重みの式 (7) は，$AveEBV$ の値が大きく，かつ複数回現れた単語を重要視したため，このような形となった．

そして，得られた $AveEBV(c_p)$ と $V(c_p)$ を用いて以下の式で全ツイート T で出現する単語 c_p の評価値 $S(c_p)$ を計算する．

$$S(c_p) = AveEBV(c_p) \times V(c_p) \tag{8}$$

こうして求められた $S(c_p)$ を，本アルゴリズムでの全ツイート T 内で出現する単語 c_p の評価値とする．

表 3 に $S(c_p)$ 算出までの例を示す．最終的な単語の評価値 $S(c_p)$ が求められ，本アルゴリズムでは，全ツイート T 内の単語は，評価値の大きい B，G，A，C，F の順でキーワード K へ関連度が高いとみなす．値が非常に小さくなり，ほぼ 0 に近い値になった D，E，H については，関連度が無いか，あっても非常に小さいとみなす．

4. 実　　験

4.1 実験方法

本アルゴリズムが実際に機能するかどうかを確かめる実験を以下の要領で行った．

- ツイートの取得には Twitter の streamingAPI を用いた
- 2011 年 7 月 13 日 13 時〜7 月 20 日 13 時のツイートを取得
- 取得した総ツイート数は 604999 件
- 全てのツイートに対して形態素解析を行い，名詞のみを抽出
- 全ツイートを通して 1 件しか現れなかった名詞は除外
- 得られた名詞の中から 4 つ選び，それぞれをキーワードとして関連語の抽出を行った

名詞を選ぶ際に，Andrei らの調査[6] をもとに名詞の分類を行い，ナビゲーショナルクエリ（目的のページがはっきりしているクエリ）を 2 件，インフォメーショナルクエリ（ユーザが知識を得ようと検索するときに用いるクエリ）を 2 件選んで実験を行った．本アルゴリズムでは，インフォメーショナルクエリに対して有効に働くことを想定している．

4.2 結　　果

4.2.1 インフォメーショナルクエリ

インフォメーショナルクエリとして,「地震」と「アメリカ」を選んだ．これらをキーワー

Vol.2012-IFAT-105 No.14
Vol.2012-NL-205 No.14
2012/1/21

表 4 結果１：キーワード「地震」に対する関連語候補

評価値	名詞	出現回数
0.945546173852338	15 日 21 時 01 分頃	4.0
0.859962529611439	横浜-阪神 3-08 回表	2.0
0.859783921078855	代打関本	2.0
0.853721651479708	カーリナさん	2.0
0.841605770515833	震度 3 茨城県北部栃木県北部群馬県北部群馬県南部千葉県北東部東京都多摩東部神奈川県東部新潟県中越	2.0
0.841218654490826	震度 4 茨城県南部栃木県南部埼玉県北部埼玉県南部千葉県北西部東京都 23 区	2.0
0.827406012007911	栃木南部５弱	3.0
0.824765650307296	震度 4 茨城県北部茨城県南部埼玉県北部埼玉県南部千葉県北西部東京都 23 区	2.0
0.824264662717685	震度 3 福島県中通り栃木県北部群馬県北部群馬県南部千葉県北東部千葉県南...	2.0
0.823760781166999	震度 5 弱栃木県南部	2.0
0.799441151328177	栃木 5 弱	2.0
0.783730700462727	マ室長	2.0
0.777470822907518	方々大丈夫	2.0
0.752132622731553	東京付近	2.0
0.700796647016718	２３区	2.0
0.661919979960237	地震 TL	2.0
0.638178790989944	震度２	4.0
0.587129569315478	震度いくつ	3.0
0.58055092761769	真岡 5 弱	2.0
0.516704316279543	2011 年 7 月 15 日 21:04 地震情報	2.0

ドとしたときの本アルゴリズムの出力のうち，上位 20 単語とそれらの評価値，出現回数を表 4，表 5 に示す．

4.2.2 ナビゲーショナルクエリ

ナビゲーショナルクエリとして,「なでしこ」と「祇園祭」を選んだ．これらをキーワードとしたときの本アルゴリズムの出力のうち，上位 20 単語とそれらの評価値，出現回数を表 6，表 7 に示す．

4.2.3 考　　察

結果１（「地震」に対する関連語候補 ）では，地震の起きた時刻や場所，震度を表す名詞が多数上位にあがった．実際に，実験期間中に起こった地震の情報であることも確認した．第２位，第３位の名詞は，プロ野球に関する名詞と思われるが，地震との関係を調べたところ，プロ野球の試合中に地震が起こったていたことがわかった．また，結果２（「アメリカ」に対する関連語候補 ）では，女子サッカー決勝（日本対アメリカ）の話題が多くあがった．

表 5 結果２：キーワード「アメリカ」に対する関連語候補

評価値	名詞	出現回数
0.996213066568626	さわー	6.0
0.995285222174862	[〓*]	6.0
0.991164481624255	pk	5.0
0.987860532650204	日本最高	5.0
0.984644052151846	澤 MVP	5.0
0.978486090769689	ー世界一	4.0
0.976245001363717	なでしこ最高	9.0
0.97438609220087	#なでしこ	4.0
0.971831579566214	延長戦突入	5.0
0.967690460056579	なでしこ延長戦	4.0
0.967214802228432	ナイスキー	5.0
0.965428961846955	バンザーイ	5.0
0.961568961093987	ナイスセーブ	10.0
0.95327220238717	アウトサイド	4.0
0.952637540702252	#nadeshiko	6.0
0.952457811635824	フェアプレー賞	5.0
0.950122549962178	ー PK	3.0
0.950063529838545	神堀	3.0
0.949320558904349	岩淵	3.0
0.949060748485336	北ア	3.0

インフォメーショナルクエリ（地震，アメリカ）で得られた関連語は，通常の関連語よりもリアルタイム性が強く，我々が目標としている関連単語の抽出が可能であることがわかった．

一方，ナビゲーショナルクエリ（結果３：なでしこ，結果４：祇園祭）で得られた関連語は，どちらの場合も，関連性が認められるものがほとんど無かった．そもそも関連度の評価値自体が全体的に低く，関連度の高い名詞が抽出できなかったことを示している．これらの原因としては，API を用いてツイートを取得する期間全体に渡って，まんべんなく投稿されているような名詞に対しては，基礎評価値 BV の値が集中して加算されるような時間帯が存在せず，結果として関連度が高くならないということが考えられる．

また，そもそも本アルゴリズムは，インフォメーショナルクエリに対する関連語を抽出することを想定しているため，これらのことはさほど問題ではないと言える．

5. 結論と今後の課題

本論文では，クエリ拡張を行うための一手法である関連単語抽出アルゴリズムを応用し，Twitter を用いてリアルタイム性の高い関連語を抽出する手法を提案した．いくつかのキー

Vol.2012-IFAT-105 No.14
Vol.2012-NL-205 No.14
2012/1/21

表 6 結果３：キーワード「なでしこ」に対する関連語候補

評価値	名詞	出現回数
0.845306217372373	息子ら	2.0
0.842748555617537	見積り依頼	2.0
0.774626529031441	茶渋軍団	2.0
0.559020974476374	ことゲーム	2.0
0.479609842176935	★。、::。.::	2.0
0.31255461752826	グッモ	2.0
0.211977139314853	これさ	2.0
0.203740712223628	打倒つね	4.0
0.104218663790196	無事定時	2.0
0.0625094206548284	今日休日ダイヤ	3.0
0.0205408683986801	TNX	2.0
0.0159652976225829	ZATSU	2.0
0.00860605013551808	隣県	2.0
0.0014439613371559	おなかwww	2.0
0.00133594754064234	女子W杯オバマ大統領	2.0
0.000674405382626386	Vol.01	2.0
0.000124137676240824	] 全英オープン	2.0
7.82269100478678e-05	アンパンマン特集	2.0
2.56261320422396e-05	ポマード	3.0
6.64383686097055e-06	サッカー女子 W 杯なでしこジャパン優勝キタ――――（	2.0

表 7 結果４：キーワード「祇園祭」に対する関連語候補

評価値	名詞	出現回数
0.840828451986951	デコムービー	2.0
0.666672033028377	DM31	2.0
0.663824454775018	戦国武闘会	2.0
0.616943737336945	皇族	2.0
0.502249653312921	やまだひさし	2.0
0.475508522242177	フロブ	2.0
0.474484078490907	クロフ	2.0
0.111590644464597	カテコマ	2.0
0.11026181696014	さん死去	2.0
0.109821547426752	テレビリポーター	2.0
0.109382611184758	人気 http://bit.ly/qSnur2	2.0
0.0326764794658905	小説 UP	2.0
0.0325120307756402	http://amba.to/puy6eY	2.0
0.032348284101912	美肌一族	2.0
0.0184278139188265	ヤマナカコ	2.0
0.0106203771365212	在庫数 No,1	4.0
0.00952209493575402	スピングル専門店	2.0
0.00946590102058766	スピングルムーブ SPM-612 取扱店:【SPINGLE 限定ノベルティ GET スピングルムーブ取扱	2.0
0.00941000219489747	楽天市場ジャンル別月間 MVP 受賞店舗	2.0
0.00929908457702135	SPIN...http://bit.ly/oECrBc	2.0

ワードに対して，実際に関連単語抽出を試みたところ，リアルタイム性の高い関連語が得られるキーワードと，関連度の高い関連語がほとんど得られないキーワードが存在することがわかった．

ツイートを取得する期間が短い場合，ある時間帯に集中して投稿されるような名詞が出現しにくくなると考えられる．今回の実験で使った「なでしこ」という名詞に関しては，女子サッカーの試合が行われている期間よりも，ツイートの取得期間がより長ければ，ある期間に集中して投稿される名詞となっていた可能性が高い．しかし，ツイートの取得期間が長くなった場合，結果１の「地震」に関して言えば，期間中に起こったいくつかの地震の情報が混在する可能性が高いため，１つの地震の情報だけを取り出したい場合，それにマッチした関連語を抽出するのは困難だと言える．これらのことを考えた場合，各キーワードが話題性を保つ期間（「なでしこ」は長く,「地震」は短い）を突き止めることができれば，各キーワードに対して適切なツイートの取得期間を設定できるはずである．そのために，あるキーワードについて，ツイートの取得期間を変えながら関連度を計算し，関連度の高い名詞が多く抽出できた取得期間をそのキーワードの話題性を保つ期間として設定することを今後の課題とする．

また，新たに取得したツイートに対して客観的に納得できるような評価実験を行うことも，今後の課題とする．

謝辞 本研究は科研費（21500102）の助成を受けたものである．

参 考 文 献

1) 大石 哲也，倉元 俊介，峯 恒憲，長谷川 隆三，藤田 博，越村 三幸, “ 関連単語抽出アルゴリズムを用いた Web 検索クエリの生成, ”電子情報通信学会 論文誌，VOL.J92-D, NO.3，pp.281-292，2009.
2) 大塚 真吾，喜連川 優, “ 大規模アクセスログを用いた検索支援システムの提案, ”日本データベース学会 Letters，vol.5，no.1，pp.13-16，2006.
3) 岡部 正幸，山田 誠二, “ トランスダクティブ学習による最小文書判定からのクエリ拡張, ”人工知能誌，vol.21，no.4-I，pp.398-405，2006.
4) T. Masada, T. Kanazawa, A. Takasu, and J. Adachi, “ Improving Web search by query expansion with a small number of terms, ”NTCIR-5 Workshop Meeting, 2005.

Vol.2012-IFAT-105 No.14
Vol.2012-NL-205 No.14
2012/1/21

5) 和泉 諒，西山 裕之，“ マイクロブログの時系列情報を利用した関連語発見手法に関する研究，”全国大会講演論文集，no.1，pp.681-683，2011.
6) Andrei Broder，“ A taxonomy of web search，”SIGIR Forum，vol.36，no.2，2002，SIGIR.
7) Google，http://www.google.co.jp/
8) Yahoo! JAPAN，http://www.yahoo.co.jp/