

contextHashtag による Twitter ユーザ向けイベント推薦システム

海江田隆博[†] 黄 宏軒^{††} 川越 恭二^{††}

[†] 立命館大学大学院 理工学研究科 〒 525-8577 滋賀県草津市野路東 1-1-1

^{††} 立命館大学 情報理工学部 〒 525-8577 滋賀県草津市野路東 1-1-1

E-mail: [†]kaieda@coms.ics.ritsumeai.ac.jp, ^{††}huang@fc.ritsumeai.ac.jp, ^{†††}kawagoe@is.ritsumeai.ac.jp

あらまし 本稿では, contextHashtag による Twitter ユーザ向けイベント推薦システムを提案する. 現在, Twitter の普及により, Twitter ユーザはその機能である Hashtag を用いてイベント情報を得ることが可能である. しかし, Hashtag によるイベント検索では, Hashtag 間に相互関連がなく独立に定義されるために適切なイベントが得にくいという問題がある. そこで, 本研究では, イベント Hashtag を構造化する contextHashtag を提案し, これを用いたイベント推薦システムを提案する. 多次元空間を持つ仮想的 Hashtag として contextHashtag を定義し, contextHashtag の空間内で過去イベント Hashtag を領域として位置付けを行う. さらに, Hashtag 領域間類似性によりユーザが望む過去イベントと類似した新しく生成されたイベントを抽出する.

キーワード Twitter, Hashtag, テキスト分類, 情報推薦

An Event Recommendation System for Twitter Users Using the contextHashtag

Takahiro KAIEDA[†], Hung-Hsuan HUANG^{††}, and Kyoji KAWAGOE^{††}

[†] Graduate School of Science and Engineering, Ritsumeikan University
Nojihigashi 1-1-1, Kusatsu, Shiga, 525-8577 Japan

^{††} Colledge of Information Science and Engineering, Ritsumeikan University
Nojihigashi 1-1-1, Kusatsu, Shiga, 525-8577 Japan

E-mail: [†]kaieda@coms.ics.ritsumeai.ac.jp, ^{††}huang@fc.ritsumeai.ac.jp, ^{†††}kawagoe@is.ritsumeai.ac.jp

Abstract In this paper, we propose an event recommendation system for Twitter users using contextHashtag. Twitter users can get rich event information by Hashtags benefitting from the widespread of Twitter. However, event retrieval merely using Hashtags may not meet the user's expectation due to Hashtags' property where they can be defined independently by different users and lack coincidence. In order to solve this problem, we propose contextHashtag that forms a structured event-hashtag space where an existing event Hashtag can be located as a region. The system then recommends the user new events based on similarity comparison of contextHashtags between new events and ones of the events which the user ever participated.

Key words Twitter, Hashtag, Text Classification, Recommendation System

1. はじめに

現在, eventpage.jp^(注1)などイベント情報が記載されたイベント検索サイトが存在する. イベント検索サイトにより, 利用者が開催場所やキーワードを入力することで関連したイベントを検索することが可能である. 一方で, Twitter^(注2)を代表とするマイクロブログサービスが爆発的に普及し, それに関する研究も盛んに行われている [1][2][3][4]. マイクロブログは従来のブ

ログと比べ文字数に制限があるため, 利用者は短い文章でサービスを提供している Web サイトに投稿する. その結果, アップデートの頻度が多くリアルタイムな情報が多く投稿されるという傾向がある. Twitter ユーザは, その機能である Hashtag からコンサートなどのイベント情報を得ることがある. Hashtag は, Twitter ユーザがメッセージを投稿する際, 「# 英字列 (#hanabi など)」をメッセージ中に付加したものである. Hashtag により, 同一の Hashtag が付加されているメッセージをグループ化することが可能となる. また, Hashtag を検索できる Web サイトも多数存在する. 例えば, hashtags.jp では 20000 件 (2010 年 12 月 26 日現在) を超える Hashtag が登録されている. このよう

(注1): <http://eventpage.jp/>

(注2): <http://twitter.com/>

な Web サイトによって、Hashtag に関連するイベントを検索し、イベント情報を取得することが可能である。Hashtag は Twitter ユーザなら誰でも生成可能である。また、Hashtag は Twitter のようなマイクロブログの特性であるリアルタイム性やアップデートの頻度の多さの影響を受けている。そのため、既存のイベント検索サイトに記載されていないイベントも記載されている。しかし、既存の Hashtag は、Hashtag 間に相互関連がなく独立に宣言されるため、同様のイベントでありながら異なる Hashtag が宣言される。また、イベント検索サイトに管理されているイベント数は膨大であり、Twitter ユーザが参加したいイベントを検索することは困難である。そのため、Hashtag によるイベント検索では Twitter ユーザが望むイベントを得ることが困難という問題がある。

そこで本研究では、Hashtag を構造化する contextHashtag を提案し、これを用いたイベント推薦システムを提案する。contextHashtag は次元空間を持つ仮想的 Hashtag であり、contextHashtag 空間内で過去のイベントに関連する Hashtag を領域として位置付けを行う。その結果、新たに生成されたイベント Hashtag を Hashtag 領域間類似性を用いて Twitter ユーザが参加した過去のイベントと類似した新たに生成されたイベントを推薦することが可能となる。

2. Hashtag の問題点と contextHashtag

2.1 Hashtag の問題点

Twitter ユーザは Hashtag からコンサートなどのイベント情報を取得することがある。これは、既存の Hashtag 検索サイト^(注3)を用いて Hashtag に関連するイベント情報を取得できる。例えば、ある Twitter ユーザがフォローしている Twitter ユーザのメッセージの中に「#ricebowl」が含まれていたとする。それを見た Twitter ユーザは、既存の Hashtag 検索サイトにアクセスし、「#ricebowl」を検索するところにより、その Hashtag に関連するイベント情報を取得することが可能である。既存の Hashtag 検索サイトでは、Hashtag 生成者やそれ以外のユーザが任意で Hashtag と指定した Hashtag に関連するイベント情報を編集することが可能である。しかし、既存の Hashtag は Hashtag 間に相互関連がなく独立に宣言できる。そのため、同様のイベント内容でありながら異なる Hashtag が宣言されるためイベント内容の類似した Hashtag が存在する。例えば、2010 年 12 月 31 日 NHK ホールで開催された紅白歌合戦に関連する Hashtag として、「#nhk_kouhaku61」や「#kouhaku」が存在する。Twitter ユーザが紅白歌合戦の内容に類似したイベントを検索したいとき、これらの Hashtag を用いて紅白歌合戦に関連する Hashtag は検索可能である。しかし、イベントの開催場所や出演者などのイベント情報が類似した Hashtag による検索はできない。これは、Hashtag に関連するイベント情報が整理されていないと考えることができる。このように、既存の Hashtag を扱う環境では、同一のイベントに関連する Hashtag が生成されるが、それらに関連性がないためイベント情報の類似したイベントを検索し

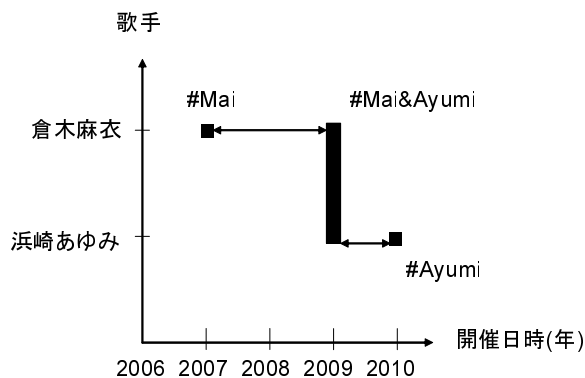


図1 コンサート contextHashtag の概念

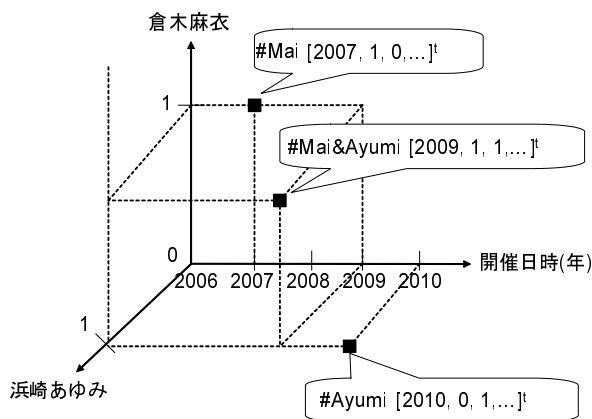


図2 コンサート contextHashtag の空間モデル

づらい。したがって、Hashtag によるイベント検索では Twitter ユーザが望むイベントが得にくい問題がある。

2.2 contextHashtag

contextHashtag とは、Hashtag を基本に生成された次元構造を持つ仮想的 Hashtag である。既存の Hashtag は構造化されていないため、Hashtag 間同士に相互関連を見出すことは困難である。そこで、contextHashtag はイベントのジャンルをもとに Hashtag をグループ化し次元構造を生成する。なお、contextHashtag は Wikipedia のイベント欄^(注4)を用いて生成する。イベントのジャンルがマッチしている Hashtag を次元構造された空間内に位置付ける。その結果、同一ジャンルの Hashtag を同じ空間内に位置付けることにより、Hashtag 間の領域間類似度を算出することが可能になる。例えば、コンサート contextHashtag を用いて説明する。コンサート contextHashtag の概念のイメージを図 1 に示す。図 1 の contextHashtag は、簡単化のために二次元空間内で定義し、領域を点の集合で表現する。また、図 1

表1 コンサート contextHashtag の具体的要素

イベント Hashtag	空間を構成する要素
#Ayumi	浜崎あゆみ, 2010
#Mai	倉木麻衣, 2007
#Mai&Ayumi	倉木麻衣, 浜崎あゆみ, 2009
#AKB	AKB48, 2010

(注3): <http://hashtagsjp.appspot.com/>

(注4): <http://ja.wikipedia.org/wiki/イベント>

内に記述されているイベント Hashtag の具体的要素を表 1 に示す．ここで、コンサート contextHashtag は開催日時(年)と歌手を軸にした空間である．

図 1 に示すように、表 1 のイベント Hashtag 「#Ayumi」が(歌手, 開催日時) = (浜崎あゆみ, 2010) のように表現できる．さらに、「#Ayumi」の領域を中心に考えると、「#Mai & Ayumi」の方が「#Mai」より類似性が高いことがわかる．また、図 2 のように空間モデルを用いて contextHashtag を構造化する．Hashtag を予め決定された統一された空間内に位置付けることにより、過去の Hashtag との領域間類似度を算出することが可能になる．例えば、利用者が過去に「#Mai & Ayumi」を使用したとする．そこへ、コンサート contextHashtag 空間内に新たに「#AKB」が登録されたとする．しかし、利用者は過去に AKB48 を要素として含む Hashtag を使用したことがないため「#AKB」に関連するイベントは推薦されない．一方で、新たに 2011 年に倉木麻衣のコンサートに関連する Hashtag 「#Mai_2011」が登録されたとする．この場合、利用者は過去にイベント内容の類似した Hashtag を使用したことがあるため推薦される．

このように、contextHashtag は既存の Hashtag には存在しない多次元構造を持つため、空間内に Hashtag を位置づけることが可能になる．contextHashtag の空間内であれば各々の Hashtag 間の類似度を算出することが可能になる．その結果、利用者が過去に使用した Hashtag に関連するイベント情報と新たに生成された Hashtag に関連するイベント情報の類似度を算出し、類似度が高ければ利用者に推薦することが可能になる．

3. contextHashtag によるイベント推薦手法

本研究では、Twitter ユーザに適したイベント情報を推薦するシステムと方式を提案することを目的とする．さらに、contextHashtag によりイベント Hashtag を管理し位置付ける方式をポイントとする．

本研究におけるイベントとは、開催場所と開催時間が決定し、利用者が参加可能である出来事とし、それに対応する Hashtag をイベント Hashtag と定義する．本稿における contextHashtag はイベント Hashtag を基本に生成する．ここで、イベントを特定するためには、開催場所、開催日時、イベント内容を加えた 3 つの要素が重要であると考えられる．開催日時が重要な利用として、開催日時の類似度を取り入れることにより利用者の嗜好の変化を表現することができると考えたためである．例えば、開催場所とイベント内容が同じで、1 年前に参加したイベントと 20 年前に参加したイベントでは、利用者は 1 年前に参加したイベントの方が嗜好が強いと考えられる．また、開催場所とイベント内容が重要な理由として、利用者がイベントに参加する際これらを意識して参加するか判断するためである．イベント内容とは、イベントに関連する名詞の集合で表現し、イベントのタイトルや参考 URL 先の情報から抽出する．例えば、コンサートでは出演者をイベント内容とする．これら 3 つの要素をイベント Hashtag 基本データと呼ぶ．本稿では、イ

ベント Hashtag をハッシュタグクラウド^(注5)や hashtagsjp と呼ばれる Hashtag 検索サイトに属しているイベント Hashtag を対象とする．これら既存の Hashtag 検索サイトは、Hashtag とイベント情報が管理され、イベントに関連する情報を容易に取得できる．さらにイベント Hashtag は、過去イベント Hashtag と未来イベント Hashtag の 2 種類に分けることができる．過去イベント Hashtag とは、利用者がシステムログイン時に既に contextHashtag 空間内に位置付けられているイベント Hashtag である．また、未来イベント Hashtag とは、利用者がシステムログイン時に contextHashtag 空間内に位置付けられていないイベント Hashtag である．

3.1 基本的考え方

本研究では、イベント Hashtag に関連するイベント Hashtag 基本データをもとに、イベントのジャンルが同じイベント Hashtag から構成される contextHashtag を提案する．既存の Hashtag は各々が独立に宣言されるため、Hashtag 間に関連性が存在しないという問題がある．そこで、既存のイベント Hashtag をイベントのジャンルをもとにグループ化する．各々のグループにおいてイベント Hashtag 基本データをもとに多次元構造を生成し、過去イベント Hashtag 全てをこの空間内の領域として位置付ける．その結果、未来イベント Hashtag と過去イベント Hashtag 間の領域間類似度を算出することが可能となる．なお、簡単化のために本稿では既存のイベント Hashtag のグループ化は人手によって行うものとする．

3.2 contextHashtag

contextHashtag を情報検索分野に頻繁に用いられるベクトル空間モデル [5] を用いて構造的に表現する．ベクトル空間モデルを用いることにより、管理している contextHashtag 空間内に過去イベント Hashtag と未来イベント Hashtag を統一的に扱うことができる．なお、各々の contextHashtag 空間内において構成される要素が異なるので、はじめの分類ならびに構造は予め設計する．contextHashtag を以下の構造で表現する．

Hashtag は任意の文字列である．Hashtag を h_i とすると N 個の Hashtag の全体集合 H を $H = \{h_1, h_2, \dots, h_N\}$ と表現する．同様に、contextHashtag を g^k とすると M 個の contextHashtag の全体集合 G は $G = \{g^1, g^2, \dots, g^M\}$ と表現できる．contextHashtag g^k は、構成する n_k 個の要素 $E^k = \{E_1^k, \dots, E_{n_k}^k\}$ で作られる空間である．ここで、 E_j^k は文字列の集合である．Hashtag の全体集合 H は関連する contextHashtag によって $H = H^1 \cup H^2 \dots H^N$ に分解できる．その結果、 g^k と H^k の関係は次のように表現できる． $\forall h_j^k \in H^k$ について、

$$h_j^k = \left\{ (e_1^k, \dots, e_{n_k}^k) \mid e_i^k \in \tilde{E}_i^k \subset E_i^k (i = 1, \dots, n_k) \right\} \quad (1)$$

$$h_j^k \subset g^k \quad (2)$$

h_j^k は contextHashtag g^k に管理されているひとつの Hashtag である． \tilde{E}_i^k はある条件で絞り込まれた E_i^k の部分集合で文字列の集合である．また、contextHashtag g^k は識別子 Ig^k によって

(注5): <http://hashtagcloud.net/>

名前付けされるとする。 $I g^k$ は任意の文字列である。 イベント Hashtag h_j^k の要素 e_i^k は、管理されている contextHashtag g^k 内の要素 E_i^k に含まれる。

3.3 イベント Hashtag 特徴ベクトルの生成

contextHashtag に管理されているイベント Hashtag 特徴ベクトルを生成する。(1)式において、要素 E^k を開催場所、開催日時、イベント内容の3つの点に分解する。すなわち、contextHashtag の特徴ベクトルは位置特徴ベクトル、時間スカラー、イベント内容特徴ベクトルから構成されるものとする。したがって、contextHashtag g^k に管理されているひとつのイベント Hashtag の特徴ベクトル h_j^k は、そのイベントの開催された位置特徴ベクトル p_j と時間特徴スカラー e_j とイベント内容特徴ベクトル f_j を用いて次のように表現する。なお、開催日時はイベント終了日とする。

$$h_j^k = \begin{pmatrix} p_j^t & e_j & f_j^t \end{pmatrix} \quad (3)$$

t は、ベクトルの転置を示す。

3.3.1 位置特徴ベクトル

開催場所から Geocoding API^(注6)を用いて緯度、経度を取得する。ここで、Geocoding API とは住所やランドマーク名から緯度、経度を検索して XML 形式で返すサービスである。例えば、パラメータとして『東京タワー』で問い合わせると、座標値『35.661913, 139.700943』が返ってくる。

したがって、contextHashtag g^k に管理されているひとつのイベント Hashtag の位置特徴ベクトル p_j は、そのイベントの開催された位置 (p_{jx}, p_{jy}) を用いて次のように表現する。

$$p_j = \begin{pmatrix} p_{jx} & p_{jy} \end{pmatrix} \quad (4)$$

3.3.2 イベント内容特徴ベクトル

イベントのタイトルや内容からイベントに関連するキーワードを抽出し特徴ベクトルを生成する。したがって、contextHashtag g^k に管理されているイベント Hashtag のイベント内容特徴ベクトル f_j は、そのイベント内容 $f_{ji}(i = 1, \dots, n_k)$ を用いて次のように表現する。

$$f_j = \begin{pmatrix} f_{j1}, \dots, f_{jn_k} \end{pmatrix} \quad (5)$$

ここで、 f_{ji} はイベント Hashtag において、イベント内容 i が存在するかどうかを示す値である。

$$f_{ji} = \begin{cases} 1 & (\text{イベント Hashtag にイベント内容 } i \text{ が存在するとき}) \\ 0 & (\text{存在しないとき}) \end{cases}$$

ここで、コンサートに関連する contextHashtag の具体例 \$concert に管理されているイベント Hashtag、#UH_WL を用いて説明する。\$concert では、イベント内容を歌手名とし、予め $f_j = \begin{pmatrix} \text{浜崎あゆみ, AKB48, 宇多田ヒカル} \end{pmatrix}$ という構造が存在するとする。#UH_WL に関連するイベントは、宇多田ヒカルが 2010 年 12 月 8 日に横浜アリーナで行ったコンサートである。これをイベント Hashtag 特徴ベクトルで表現すると次の式になる。(35.512228, 139.620165, 2010/12/8, 0, 0, 1)^t

3.4 イベント Hashtag 間の類似度

利用者が使用した過去イベント Hashtag をもとに、それに類似した未来イベント Hashtag を抽出する。ここで、利用者が使用した過去イベント Hashtag に類似した未来イベント Hashtag に関連するイベントが利用者が望むイベントであるとする。

過去イベント Hashtag と未来イベント Hashtag の領域間類似度を次のように算出する。ここで、領域間類似度とは各々の領域に含まれる点同士が最も類似度の高い値となる点の組み合わせと定義する。このような類似度を用いた理由は、計算量と精度の視点から上記の類似度を採用した。

同一 contextHashtag 内に管理されている過去イベント Hashtag を h_i と未来イベント Hashtag を w_j とすると、2つの h_i, w_j 間の類似度 $Sim(h_i, w_j)$ を(6)式で定義する。ただし、 $0 \leq Sim(h_i, w_j) \leq 1$ である。また、過去イベント Hashtag h_i は位置特徴ベクトル p_i 、時間スカラー e_i 、イベント内容特徴ベクトル f_i から構成される。同様に、未来イベント Hashtag w_j は位置特徴ベクトル p_j 、時間スカラー e_j 、イベント内容特徴ベクトル f_j から構成される。

$$Sim(h_i, w_j) = Sim_p(p_i, p_j) \times Sim_f(f_i, f_j) \times Sim_e(e_i, e_j) \quad (6)$$

ここで、 $Sim_p(p_i, p_j)$ を(7)式で $Sim_f(f_i, f_j)$ を(8)式で定義する。 $Sim_p(p_i, p_j)$ は位置特徴ベクトル間の距離に1を足し、その逆数により求める。 $Sim_f(f_i, f_j)$ は、イベント内容特徴ベクトル間のなす角の余弦値により求める。

$$Sim_p(p_i, p_j) = \frac{1}{1 + d(p_i, p_j)} \quad (7)$$

$$Sim_f(f_i, f_j) = \frac{f_i \cdot f_j}{\|f_i\| \|f_j\|} \quad (8)$$

ただし、 $d(p_i, p_j)$ はヒュベニの公式^(注7)(基準値を 100km とする)である。さらに、 $Sim_e(e_i, e_j)$ を(9)式で定義する。 α を任意の自然数とする。また、 $F_x(e_i, e_j)$ は、 x (年, 月, 日などの単位を示し contextHashtag ごとに予め設定される)に基づく e_i, e_j との間の時間を算出する関数である。 $Sim_e(e_i, e_j)$ は $F_x(e_i, e_j)$ に1を足し、その逆数により求める。

$$Sim_e(e_i, e_j) = \left(\frac{1}{F_x(e_i, e_j) + 1} \right)^{\frac{1}{\alpha}} \quad (9)$$

ここで α は 1 以上の整数であり、極端に時間差がある場合にのみ時間スカラーの類似度の影響を考慮するために用いる定数である。以降で述べるシステムでは $\alpha = 4$ を使用する。

4. contextHashtag による Twitter ユーザ向けイベント推薦システム

4.1 イベント推薦システムの流れ

本稿では、利用者の過去参加したイベントに類似した新たな

(注6): <http://www.geocoding.jp/>

(注7): <http://yamadarake.web.fc2.com/trdi/2009/report000001.html>

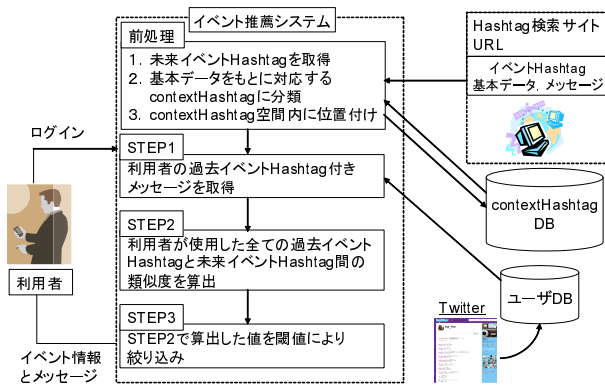


図3 提案システムの処理手順

緯度	経度	A	B	C	D	E	F	G	H	I	開催日時
35.689506	139.691701	0	1	0	0	0	0	0	0	0	2010
35.689506	139.691701	1	0	0	0	0	0	0	0	0	2006
35.689506	139.691701	0	0	1	1	0	0	0	0	0	1999
35.689506	139.691701	0	0	0	0	0	0	0	1	0	2003
35.689506	139.691701	0	0	0	0	0	1	1	0	0	1998
35.689506	139.691701	0	0	0	1	0	0	0	0	1	1992
35.689506	139.691701	0	0	1	1	1	0	0	1	0	2000
35.689506	139.691701	0	0	0	0	1	0	0	0	0	2010
35.689506	139.691701	1	0	0	0	1	0	0	0	0	1996
35.689506	139.691701	1	0	0	1	1	0	0	0	0	2000
35.689506	139.691701	0	0	0	0	0	0	1	1	1	1998
35.689506	139.691701	0	0	1	0	1	0	0	1	0	1991
35.689506	139.691701	1	1	0	0	0	0	0	1	1	2000
35.689506	139.691701	0	0	0	0	1	0	1	0	1	2005
35.689506	139.691701	0	0	1	0	0	0	0	0	0	1990
35.443708	139.638026	0	0	1	1	0	0	0	0	0	1995
35.443708	139.638026	0	0	0	0	0	0	1	1	0	1998
35.443708	139.638026	0	0	0	0	0	0	0	0	1	1998

図4 テストデータ (一部抜粋)

イベントを推薦するシステムを提案する。本システムの処理手順を図3に示す。本研究における利用者とは、Twitter ユーザーでありイベント Hashtag 付きのメッセージを頻りに Twitter へ投稿する者とする。前提条件として、システム管理者が予め contextHashtag を生成し、過去イベント Hashtag 全てを該当する contextHashtag 内で位置付けする。contextHashtag データベースには位置付けされたイベント Hashtag を格納する。ユーザデータベースには利用者が Twitter に投稿したメッセージを格納する。前処理として、まず、システム管理者が Hashtag 検索サイトから未来イベント Hashtag を取得する。次に、システム管理者が Hashtag 検索サイトや参考 URL から未来イベント Hashtag の基本データを取得し、そのデータをもとに対応する contextHashtag に分類する。そして、システム管理者が未来イベント Hashtag を contextHashtag 空間内に位置付ける。

【STEP1】 利用者はシステムにログインすると、システムはユーザデータベースから利用者が使用した過去イベント Hashtag 付きメッセージを取得する。

【STEP2】 システムは、STEP1 で取得した利用者が使用した contextHashtag 空間内で管理されている過去イベント Hashtag と未来イベント Hashtag との類似度を算出する。

【STEP3】 システムは STEP2 で算出された値を閾値により絞り込みを行い、閾値以上であれば利用者に未来イベント Hashtag に関連するイベント情報とメッセージを出力する。

4.2 contextHashtag によるイベント推薦の具体例

4.2.1 前 提

3章で説明した contextHashtag により利用者が望む未来イベントの検出が可能であることを示すために、図4に示すようなテストデータをもとにイベント推薦を行う。

具体例として、表2に示すように利用者は過去に3度コンサートに関連するイベント Hashtag を使用したとする。それらのイベント Hashtag と他の100個のイベント Hashtag を対象に類似度を算出し、類似度の高い上位3件のイベントを推薦する。なお、予めコンサートに関連するイベント Hashtag を基に contextHashtag を生成しておく。イベント内容特徴ベクトルを歌手9名より構成し、位置特徴ベクトルを日本の地名9ヶ所に定める。表3に地名と対応する緯度経度を示す。また時間スカラーは年単位であり、1990年から2010年に限定する。

4.2.2 イベント推薦の具体例

具体例の実行結果を表4、表5、表6に示す。表4は利用者が過去に使用したイベント Hashtag を #1996MaiAyu とした場合の他のイベント Hashtag との類似度の高い上位3件の計算結果を示す表であり、表5は利用者が過去に使用したイベント Hashtag を #1992Sazan とした場合の他のイベント Hashtag との類似度の高い上位3件の計算結果を示す表である。さらに、表6は利用者が過去に使用したイベント Hashtag を #1991Smap とした場合の他のイベント Hashtag との類似度の高い上位3件の計算結果を示す表である。

表4では、コンサートの開催場所が東京で歌手に浜崎あゆみや倉木麻衣に関連するイベントを出力する。表5では、コンサートの開催場所が京都付近で歌手がサザンオールスターズに関連するイベントを出力する。表6では、コンサートの開催場所が札幌で歌手が SMAP に関連するイベントを出力する。このように、利用者が過去に使用したイベント Hashtag 基本データの関連性の近いイベントを出力することが可能である。また、表4、表5、表6から、過去に使用した Hashtag のイベント内容に関連のないイベントは出力されていない。また、表5からイベント内容が類似していれば京都に比較的距離の近い大阪のイベントも出力されたが、札幌や福岡などさらに離れているイベントは出力されていない。実行結果から開催場所が著しく離れているイベントやイベント内容が異なるイベントなど、利用

表2 過去に使用したイベント Hashtag

Hashtag	開催場所	歌手	開催日時(年)
#1996MaiAyu	東京	浜崎あゆみ, 倉木麻衣	1996
#1992Sazan	京都	サザンオールスターズ	1992
#1991Smap	札幌	SMAP	1991

表3 地名と対応する緯度経度

地名	緯度, 経度
東京	35.689506, 139.691701
横浜	35.443708, 139.638026
京都	35.011636, 135.768029
大阪	34.693738, 135.502165
札幌	43.062096, 141.354376
仙台	38.268215, 140.869356
名古屋	35.181446, 136.906398
福岡	33.590355, 130.401716

表4 具体例の実行結果 1

開催場所	歌手	開催日時(年)	類似度
東京	浜崎あゆみ, 倉木麻衣, EXILE	2000	0.546
東京	浜崎あゆみ	2006	0.388
東京	倉木麻衣	2010	0.359

表5 具体例の実行結果 2

開催場所	歌手	開催日時(年)	類似度
京都	サザンオールスターズ	1998	0.615
大阪	サザンオールスターズ	1997	0.468
大阪	サザンオールスターズ, 倉木麻衣	1993	0.286

表6 具体例の実行結果 3

開催場所	歌手	開催日時(年)	類似度
札幌	SMAP	2001	0.549
札幌	SMAP, 倉木麻衣, 浜崎あゆみ	1993	0.439
札幌	SMAP, 倉木麻衣	1999	0.408

者に適していないイベントは出力されていない。

従って、これらの表4, 表5, 表6に示すように, contextHashtag を用いることで利用者に適したイベントを提供することが可能である。

5. 予備評価実験

5.1 実験条件

本稿で提案した contextHashtag によるイベント推薦システムの有効性を確認するため, 実データを用いて予備評価実験を行った。本システムは Java 言語を用いて実装した。また, Twitter API の Java ラッパである Twitter4J を利用してメッセージを取得した。まず, コンサートに関連する contextHashtag を予め作成し, 既存の Hashtag 検索サイトから取得したイベント情報を基に 2010 年 5 月から 2010 年 8 月中頃の間に生成されたイベント Hashtag を過去イベント Hashtag とし, contextHashtag 空間内に位置付けを行った。この際, イベント内容を出演者とし, 680 名を対象としたが出現回数が少ないという理由から 150 名にまで絞り込みを行った。そして, contextHashtag データベースに過去イベント Hashtag として 20 個を格納した。ここで, 2010 年 8 月中頃に降に開催されるイベント Hashtag 25 個を未来イベント Hashtag とする。未来イベント Hashtag と過去イベント Hashtag と類似度を算出し, 閾値以上であれば推薦する。なお, 本実験では閾値を 0.05 以上とした。

ここで, 使用した contextHashtag の具体例を $\#halcali$ を用いて説明する。コンサート contextHashtag では, $f_j = (\text{緯度}, \text{経度}, \text{halcali}, \text{Perfume}, \text{加藤ミリヤ}, \dots, \text{開催日時(年)})^j$ という構造を持つ。 $\#halcali$ に関連するイベントは, HALCARI が 2011 年 2 月 13 日に渋谷で行われたコンサートである。これをイベント Hashtag 特徴ベクトルで表現すると次の式になる。(35.655289, 139.704536, 1, 0, 0, \dots , 2011)^j

評価実験の被験者は学生 6 名を対象とした。被験者には予め過去イベント Hashtag のイベント情報を提示し, 「参加した

かったイベント」を選択する。同様に, 未来イベント Hashtag に対して, 「参加したいイベント」を選択し, これを正解集合と定める。評価尺度には, 情報検索の評価に頻繁に使用される再現率, 適合率, F 値を用いて評価する。比較手法として, contextHashtag を使用せず, 一般的なベクトル空間モデルを用いた手法を採用する。これは, イベントに出演している出演者のみの情報からイベントの特徴ベクトルを生成する。同様に過去イベント Hashtag と未来イベント Hashtag 間の特徴ベクトル間の類似度を算出し, 閾値以上であれば推薦する。

また, 提示されるイベント情報に対して主観的評価を行った。評価尺度として以下の 3 つの項目があり, これを 1(悪い)~4(良い)点で評価した。

- (1) 自分の嗜好に適したイベントが推薦されたか
- (2) 本システムを利用したいか
- (3) 見やすいか

5.2 実験結果ならびに考察

本システムの出力例を図 5 に示す。また, 情報検索の評価方法による実験結果を表 7, 主観的評価による実験結果を表 8 に示す。

表 7 より, 再現率と F 値は高いことがわかる。再現率と F 値が高い理由として, 比較手法よりシステムが利用者の望むイベントを抽出したと考えられる。その結果, 利用者の望むイベントの見逃しを防ぐことができると考えられる。なお, 対象とする実データ数が少ないため適合率が低くなったと考えられる。これは, 本システムが普及することにより, 新しい Hashtag が頻繁に生成することにより解決可能である。本稿ではイベント内容, 開催場所, 開催日時により類似度を算出したが, 遠くても大規模なイベントなら参加するという意見があった。そのため, 今後, イベント内容, 開催場所, 開催日時以外の情報も類似度算式に取り入れるか検討する。さらに, 利用者ごとに何を重視しているかシステムが自動的に判断し, 利用者ごとに適した類似度算式を導くようにする必要がある。

表 8 より, 主観的評価により本システムを使用したいという評価が高いことがわかる。しかし, 推薦されたイベントが利用者の嗜好に適しているかという評価ではあまり高くないことがわかる。その理由として, 利用者の嗜好にあまり適していなくても, システムとしてあったら便利と思う利用者が多いと考えられる。さらに, 本システムによる出力は推薦されるイベント情報と, そのイベント対応する Hashtag が付加されている最新

表7 情報検索における評価結果

	比較手法	提案手法
適合率	0.269	0.247
再現率	0.431	0.611
F 値	0.309	0.351

表8 主観的評価結果

	嗜好に合うか	利用したいか	見やすいか
平均値	2.67	3.5	2.67
標準偏差	0.47	0.5	0.47

イベントHashtag	イベント名	開催場所	開催日時	イベント内容(出演者)	URL
#nhk_kouhaku61	第61回 NHK紅白歌合戦	NHKホール	2010/12/31-2010/12/31	aiko, アンジェラ・アキ, いきものがかり, 石川さゆり, 植村花菜, AKB48, 川中美幸, クミコ, 倅田来未, 伍代夏子, 小林幸子, 坂本冬美, 大童よしみ, DREAM COMES TRUE, 中村美津子, 西野カナ, 浜崎あゆみ, Perfume, 平原綾香, 水樹奈々, 水森かおり, 和田アキ子, 嵐, 五木ひろし, HY, EXILE, NYC, 加山雄三, 北島三郎, 郷ひろみ, コブクロ, SMAP, TOKIO, 徳永英明, AAA, 水川きよ子, FUNKY MONKEY BABYS, 福山雅治, flumpool, 細川たかし, ポルノグラフィティ, 森進一, 遊助, L'Arc"en"Ciel	http://www8.nhk.or.jp/kouhaku/

#nhk_kouhaku61に関するメッセージ

投稿者	メッセージ	送信日時
manachan86	RT @nhk_kouhaku: 2月13日(日)の「MUSIC JAPAN」は“紅白トーク特別企画”。AKB48,倅田来未,AAA,西野カナ,Perfumeのみなさんが、去年の「第61回紅白歌合戦」の思い出・舞台裏秘話を公開します。http://www.nhk.or.jp/mj/#nhk_kouhaku61	Sat Feb 12 09:27:55 JST 2011
144338	RT @AAANEWS_: RT @nhk_kouhaku: 2月13日(日)の「MUSIC JAPAN」は“紅白トーク特別企画”。AKB48,倅田来未,AAAのみなさんが、「第61回紅白歌合戦」の思い出・舞台裏秘話を公開 http://www.nhk.or.jp/mj/#nhk_kouhaku61	Fri Feb 11 20:35:01 JST 2011
kt24halobe	RT @nhk_kouhaku: 2月13日(日)の「MUSIC JAPAN」は“紅白トーク特別企画”。AKB48,倅田来未,AAA,西野カナ,Perfumeのみなさんが、去年の「第61回紅白歌合戦」の思い出・舞台裏秘話を公開します。http://www.nhk.or.jp/mj/#nhk_kouhaku61	Fri Feb 11 17:50:36 JST 2011
kametome	RT @nhk_kouhaku: 2月13日(日)の「MUSIC JAPAN」は“紅白トーク特別企画”。AKB48,倅田来未,AAA,西野カナ,Perfumeのみなさんが、去年の「第61回紅白歌合戦」の思い出・舞台裏秘話を公開します。http://www.nhk.or.jp/mj/#nhk_kouhaku61	Fri Feb 11 16:06:45 JST 2011
amass.jp	RT @nhk_kouhaku: 2月13日(日)の「MUSIC JAPAN」は“紅白トーク特別企画”。AKB48,倅田来未,AAA,西野カナ,Perfumeのみなさんが、去年の「第61回紅白歌合戦」の思い出・舞台裏秘話を公開します。http://www.nhk.or.jp/mj/#nhk_kouhaku61	Fri Feb 11 14:15:07 JST 2011

図5 出力例

のメッセージだけなので非常に見やすいという意見がある一方、あまり色や大きさによる変化がないため、あまり見栄えが良いとは言い難いという意見がある。そのため、今後は見やすさをよりよくするためにインタフェースにも力を入れる必要がある。以上より、本システムは以下のことが言える。

- (1) 情報検索における評価より、contextHashtag を用いたイベント推薦は有効である
- (2) 主観的評価より、イベント推薦システムとして有用性がある

6. おわりに

本研究では、contextHashtag による Twitter ユーザ向けイベント推薦システムを提案した。contextHashtag により、過去イベント Hashtag と未来イベント Hashtag との領域間類似度が算出可能となった。その結果、Hashtag という Twitter の機能を使用することによりリアルタイムなイベントや規模に関係なくイベントを推薦することが可能になった。

今後は、contextHashtag を自動的に生成し、未来イベント Hashtag を空間内に自動的に位置付けを行うよう改良する。また、利用者ごとに適した類似度算出式方法を考える。さらに、システムをインターネット上に公開することで実際に Twitter ユーザに利用してもらい、継続的な評価、改善を行いながら精度を上げていく。

謝辞 本論文では hashtagsjp, ハッシュタグクラウドからイベント情報を取得している。ここに記して感謝の意を表す。

文献

- [1] Akshay Java, Xiaodan Song, Tim Finin, Belle Tseng“Why we twitter: understanding microblogging usage and communities”WebKDD/SNA-KDD '07 Proceedings of the 9th WebKDD and 1st SNA-KDD 2007

- workshop on Web mining and social network analysis, pp. 56-65, 2007.
- [2] Owen Phelan, Kevin McCarthy, Barry Smyth “Using Twitter to Recommend Real-Time Topical News”RecSys '09 Proceedings of the third ACM conference on Recommender systems, pp. 385-388, 2009.
- [3] Miles Efron“Hashtag Retrieval in a Microblogging Environment”SIGIR '10 Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pp. 787-788, 2010.
- [4] 田中克明, 堀浩一 “Twitter ハッシュタグに基づく Tweet 群からの変化抽出”, 電子情報通信学会 Web インテリジェンスとインタラクション研究会, 2010, 7, pp. 31-32, 2010.
- [5] 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, Vol.19, No.3, pp.365-372, 2004.