ソーシャルタグの表記と使用傾向に基づく集約

多田　亮平† 　湯本　高行† 　新居　学† 　高橋　豐†

† 兵庫県立大学大学院工学研究科
〒 671-2280 兵庫県姫路市書写 2167
E-mail: †er11e019@steng.u-hyogo.ac.jp, ††{yumoto,nii,takahasi}@eng.u-hyogo.ac.jp

あらまし　近年，ソーシャルブックマークサービスなど，タグ (キーワード) をオブジェクトに与えて管理するサービスが普及している．しかし，タグの表記のゆれは大きく，同義のタグが複数存在するという問題がある．本研究ではこの問題を解決するため，ソーシャルタグの表記と使用傾向に基づく集約手法を提案する．まず表記について注目し，分類のための記号が含まれる場合，タグが複数の単語が連結されいる場合に対応するルールを用意する．このルールに則った加工・集約を行う．次に，使用傾向について，タグを使用したユーザ，付与された URL，同じ URL に付与された共起タグの３つの観点から注目する．各観点からタグ間の類似度を算出し，有効な類似度を定義する．その類似度に対して，適切なしきい値を設定することによって同義のタグを集約する．「はてなブックマーク」から収集したデータセットに対して，表記の特徴による集約手法に対する評価実験を行った．３つのルールを用意し，それぞれに手法を適用した結果，60634 種類あったタグを 37637 種類まで集約することができた．次に，上記の手法を適用したデータセットに対して，使用傾向による集約手法の有効性を調査した．その結果，タグを付与された URL の観点からタグ間の類似度を定義し，類似度が高いタグを集約することが有効なことがわかった．この手法をデータセットに適用した結果，表記の特徴からだけでは判断できない同義のタグを，708 種類から 279 種類に集約することができた．
キーワード　ソーシャルタグ，ソーシャルブックマーク，データベース

Aggregation of Social Tags based on Notation and Tendency of Usage

Ryohei TADA†, Takayuki YUMOTO†, Manabu NII†, and Yutaka TAKAHASHI†

† Graduate School of Engineering, University of Hyogo
2167 Shosha，Himeji，Hyogo，671-2280，Japan
E-mail: †er11e019@steng.u-hyogo.ac.jp, ††{yumoto,nii,takahasi}@eng.u-hyogo.ac.jp

1. はじめに

近年，ソーシャルブックマークサービス (以下 “SBM サービス”) が普及している．これはユーザが気に入った Web コンテンツをブックマークするという行為を，オンライン上で管理するサービスである．このサービスの特徴的な機能の一つに「タグ」がある．ユーザの Web コンテンツに対するブックマークに，複数のキーワード (タグ) を付与することによって，容易にブックマークの管理を行うことができる．しかし，このタグは自由にユーザが記述できるため，非常に多くの表記のゆれが存在する．

多数のユーザが使用しているタグや，多くのブックマーク数を示す Web コンテンツだけをデータの対象とした場合には，この表記のゆれは大きな影響を及ぼさないこともある．しかし，ソーシャルタグを用いて，ニッチな情報などを検索する場合には大きな悪影響を及ぼす．

そこで本研究では同義のタグを集約することを目的とし，ソーシャルタグの表記と使用傾向に基づく集約手法を提案する．まず表記について注目する．タグに分類のための記号が含まれる場合や，複数の単語が連結したタグに対応するルールを用意し，タグの加工を行う．次に使用傾向について注目する．具体的には，タグがどのユーザに使われているか，どのような Web コンテンツに使用されているか，どのようなタグと同時に出現しているか，という傾向にベクトルの観点から注目する．この使用傾向によって類似度を定義する．同義のタグは使用傾向の類似度によって判定できると考え，類似するタグを集約することで同義のタグを集約する．その際，集約するために適切なしきい値を定義する．

タグの辞書を用意して集約した場合では，略語や，言語による表記のゆれしか集約できない．しかし，本手法を用いること

で，略語や言語による差を始め，個人が使用するような独自のタグにも対応できると考える．

ソーシャルタグを集約することによって，タグとして複数の表記を持つ概念を統一する．これによって，サービス上のタグ検索による効率，精度が上がると考えられる．また Web コンテンツから特定の概念を抽出しやすくなると考えられ，ソーシャルタグを用いた協調フィルタリングなど，他の研究の精度の向上が期待できる．

具体的には，本手法を適用することで，既存のサービスのタグ検索の精度向上が期待できる．また協調フィルタリングを用いて，ユーザの嗜好に沿った Web ページを推薦するシステムの場合，推薦されるユーザが独自のタグを使用した場合でも，よく使用されるタグを用いるユーザと同様の恩恵をうけることができると期待できる．

2. 関連研究

ソーシャルタグを用いた研究として，類似するタグを判定する研究が既に存在する．例えば丹羽ら [1] は，ユーザの Web ページ嗜好データを SBM サービスのタグから取得することで，「ユーザのページ嗜好データの不足」を解決し，Web ページの推薦を行う手法を提案している．この手法の一部ではタグの表記のゆれを，類似するタグをクラスタリングすることで解決を試みている．例えば，“apple”クラスタには “itunes”や “ipod” が含まれるようにクラスタリングを行っている．この手法では，タグをクラスタリングすることで表記のゆれの影響を解決しているが，本研究ではタグ自体を加工したり，他の同義のタグに統一することで，直接的にタグの表記のゆれに対する解決を行う．

本研究と同様，SBM サービスのタグの特徴について述べている研究としては，川中 [2] の研究がある．川中は概念を記述するタグを解析し，時系列的な概念の派生関係を抽出する手法を提案している．また川中は，丹羽らの手法中の式，及び結果から以下の様にタグの使用傾向をまとめている．

- ２つのタグが同義の場合，それらが使用された Web コンテンツ集合は，使用しているユーザ集合に比べて類似している傾向がある．
- ２つのタグに関連性がある場合，それらが使用された Web コンテンツ集合，使用しているユーザ集合は共に類似している傾向がある．
- ２つのタグが対立しているような関係の場合，それらを使用したユーザ集合は，使用された Web コンテンツ集合に比べて類似している傾向がある．

また，本研究の目的や手法が類似している研究として，藤村ら [3] の研究がある．この研究のデータの対象には blog を用いており，SBM サービスとはタグの付け方に違いがある．blog では，ユーザが自身または blog にタグを付けることで，同じ話題に興味があるユーザを探索できる．本研究では blog と違い，SBM サービスのように複数のユーザがタグを対象に付与する形式を利用している．また，藤村らはタグを付けられた blog の単語を用いて特徴ベクトルを作成し，コサイン類似度を算出する手法を提案している．本研究ではタグを付けられたコンテンツの内容は手法の対象にしていないが，特徴ベクトルを作成し，コサイン類似度を算出する手法を参考にしている．

3. ブックマークデータの収集手法

本研究では提案手法を適用することで，個人が独自のルールで使うタグに対しても集約が行えると考えている．そのため，多数のユーザが使用するタグのみを基点にデータを収集する手法は取らず，多様なタグが使用されているデータを対象とする．多様なタグが使用されている環境には，多数のユーザがサービスを利用している必要がある．このことから，本研究ではデータセットとして国内最大規模の SBM サービスである「はてなブックマーク」を用いる．

また，独自のルールで使用されるタグも収集することを目的とするため，ブックマーク数の多い URL や，最新記事だけに情報を偏らせず，全体からデータを収集する．

まず，はてなブックマークの提供する，最新・注目記事のまとめである HotEntry の記事に注目する．HotEntry は各トピックに対して分野が分かれている．ここでは，トピックの分類が明確な { 社会，政治・経済，スポーツ・芸能・音楽，科学・学問，コンピュータ・IT，ゲーム・アニメ，おもしろ } を用いる．上記の各トピックの RSS 情報を基点として，下の A〜D の工程を行うことでブックマーク情報を収集する．

1） RSS 情報(注1)を用い，各トピックの HotEntry 上位 30 件のタイトル，URL を取得する．

2） 30 件の URL 集合から，それぞれの URL を登録しているブックマーク情報 (ユーザ ID,URL, タグ，ブックマーク日時，コメント) をすべて取得し(注2)，データベースへ格納する．

3） 取得したブックマーク集合からユーザ集合を取得し，各ユーザのブックマークした URL(注3)の最新上位 100 件を取得する．

4） その URL 集合に対して，2) の処理をもう一度繰り返す．

以上のように URL からユーザ，ユーザから URL と検索対象を変えることで，全体から偏りなく BM 情報を取得する．また，タグの集約を目的とした研究であるため，タグのないブックマーク情報は考慮せず，タグ付けされたブックマーク情報のみでデータベースを構築する．

データセットの規模

上記の手法によって，はてなブックマークからブックマーク情報の収集を行った．収集期間は 2011 年 4 月 14 日から，2011 年 10 月 27 日までの期間として行った．作成したデータセットの詳細・規模を表 1 に示す．なおタグの種類数について説明する．ユーザ X がタグ t_A，t_B，ユーザ Y がタグ t_A，t_C を用いてそれぞれブックマークしていた場合，タグの総種類は $\{t_A, t_B, t_C\}$ の 3 種類と数える．

(注1)：http://b.hatena.ne.jp/hotentry/$Topic$.rss

(注2)：http://b.hatena.ne.jp/entry/json/?url=Url

(注3)：http://b.hatena.ne.jp/USER/rss?of$NunberOfItems$

表 1　データセットの規模

データセット情報	件数・種類数
総ブックマーク数	1872176
ユニーク URL 数	18687
ユニークユーザ数	55336
タグ総種類数	60634

4. タグの表記の特徴による集約

4.1 表記の特徴に注目した手法

本節では，SBM サービスを利用しているユーザの「タグの表記方法」に注目し，同義のタグを集約する手法を提案する．以下に述べる手法を順に，データセットへ適用することで，タグの集約を行う．

4.1.1 英語，片仮名表記タグの表記の種類に基づく変換

本項では，英字タグの大文字・小文字，片仮名タグの半角・全角といった表記の種類について注目する．本研究では，はてなブックマーク(注4)をデータセットとして対象にしている．はてなブックマークのように主に日本で利用されるサービスでは，Apple(社) と apple(りんご) が英語で表記される可能性が非常に小さい．このように，英字の大文字と小文字によって意味の異なるタグになる可能性が非常に小さいといえる．そのため，タグに出現する英字に関しては小文字に統一することで英字の表記のゆれを解決する．

また，日本語の片仮名には半角・全角文字が存在する．しかし，同じ概念を示すタグが半角・全角の差によって，別の概念を示す可能性は非常に小さい．よって，タグに出現する片仮名をすべて全角文字に統一ことで半角・全角の差を解決する．

4.1.2 タグのルールベースによる加工

本項では，SBM サービスを利用するユーザのタグの表記パターンに注目し，タグの加工を行うことで表記のゆれを解決する．

はてなブックマークでは，自身の登録したブックマークを検索する際に特徴的な機能が存在する．検索ボックスに文字を打ち込むと,「打ち込んだ段階での文字列と一致するタグ，タイトル，URL」を持つ自身のブックマーク情報を表示する．この機能を利用するユーザは，頻繁に利用するタグを特徴的な表記パターンで登録することによって，自身のブックマーク検索効率を上げようとしていると考えられる．また，ある 1 つのタグの表記だけを見て，そのブックマークが一体何についての Web コンテンツなのかを詳細に把握しようとするユーザがいる．この場合によっても，特徴的な表記パターンでタグ登録を行っているユーザがいる．

以上の 2 点に注目し，下記の 3 つのパターンが見られるタグに対して，それぞれのルールに沿ったタグの加工を順に行う．順に行う理由としては，頭文字パターンと分割可能パターンにルールの重複が含まれるためである．先に，詳細なルールを持つ頭文字パターンを用いる．

(注4)：http://b.hatena.ne.jp/

先頭記号パターン

上記の説明の通り，はてなブックマークでは入力途中で一致するタグを持つブックマークを検索することが可能である．そのため，本来タグには使用しないような記号を，先頭に追加してタグ登録を行なっているパターンがある．例えば，“!あとでよむ”，“*iPad”などがこれに当たる．このことから，先頭に特定の記号があるタグに対して，その記号がなくなるまで削除する加工を行う．これにより “*あとで読む”や，“*iPad”などのタグを，“あとで読む”，“iPad”の表記へ統一することができる．このとき削除対象とするタグを表 2 に示す．

表 2　先頭記号一覧

*	＊	+	＋	:	：	#	＃	-
＝	ˆ		_	/	／	!	！	?
？	$	＄	’	’	.	．		,
‘	`	&	＆	%	％	@	＠	—
|								

頭文字パターン

先頭記号パターンと目的は変わらないが，ここでは頭文字を追加しているパターンについて述べる．“i_iPhone”，“あ_iPhone”など，ユーザによってはタグに対して頭文字と記号を付与する場合が多く存在する．

ここでは，先頭に 1 文字と特定の記号を順に付与しているタグに注目し，正規表現を用いて解析することで先頭の 1 文字と次の記号を削除する．頭文字パターンを示すタグの表記を統一することで，“i_iPhone”を “iPhone”に，“あ_iPhone”を “iPhone”に集約する．このとき使用する記号を以下の表 3 に示す．

表 3　頭文字記号一覧

_	＿	:	：	·	・	.	．	/	／

分割可能パターン

上の 2 つのパターンと比べて少ない例ではあるが，複数の概念を一つのタグとして登録しているパターンについて述べる．あるトピックもしくは概念を先頭に，それを細分化したトピック，概念を順に表記したタグが存在する．例えば，“Apple-iPhone-App”などのタグが存在する．このようにタグ登録を行うことで，このタグが使われたブックマークが，何について表したコンテンツなのかを詳細に表現している．

このパターンについては，概念ごとに記号を挟んでいる場合が多く，先ほどの例では “-”でトピックを分けている．よって，あるブックマークのタグに特定の記号が出現する場合，その記号の前後でタグを分割し，複数のタグ登録を行ったブックマークとして加工する．この時，分割する記号として用いる記号を以下の表 4 に示す．

表 4　分割記号一覧

/	／	·	・	>	＞	<	＜
&	＆	(	（	)	）	{	｛
}	｝	【	】	「	」		,
.	．	、	。	-	‐	—	:
：	;	；	*	＊	+	＋	―
|							

4.1.3　語幹抽出によるタグの集約

ここでは，ある英語表記の 2 つのタグが同一の語幹を持つ場合について考える．例えば，“manage”と “management”という 2 つのタグが存在する場合，これらは語幹が同一である．また，これらを用いたユーザが示そうとしている内容は同一であるといえる．この語幹が一致するが表記が違うといった問題を，同一語幹のタグを集約することで解決する．

語幹の抽出には，Porter Stemming アルゴリズム(注5)を用い，データセット中に登録されているすべてのタグから語幹を抽出する．そして，抽出した語幹が一致するタグを統一することで，同一の語幹を持つ同義のタグを集約する．また，本研究では語幹が一致するタグ集合を「使用しているユーザ数が多いタグ」に集約することで，意味が把握しやすいと考えられる，多数のユーザが使用するタグで統一する．

4.2　タグの表記の特徴による集約実験

4.1 節の手法を用いて，データベース中のタグを加工・集約した．データベースに手法を適用する前後でのタグの種類数を表 5 に示す．データを収集して，はてなブックマークにある状態のままのデータセットが表 5 中の「手法適用前」である．また，4.1.1 の手法を用いて集約した結果が英語・片仮名の変換，さらに 4.1.2 を用いた結果がルールベースのタグ加工，さらにそれに対して 4.1.3 を用いた結果が同一語幹のタグ集約に当たる．

表 5 から，手法を適用する以前では，データベース中に存在する総タグ種類数は 60634 種類であった．しかし，全ての手法適用後では総タグ種類数は 37637 種類まで減少した．このことから，非常に多くのタグが集約できたといえる．

表 5　各手法による総タグ種類数の変化

データセットの状態	タグ種類数
手法適用前	60634
英語・片仮名の変換	46948
ルールベースのタグ加工	38443
同一語幹のタグ集約	37637

表 6　各パターンによって加工した総タグ種類数

	先頭記号	頭文字	分割可能
加工したタグ種類数	4388	1055	6585

また，4.1.2 のルールベースによる加工で，先頭記号パターン，頭文字パターン，分割可能パターンのそれぞれが，どれだけのタグ種類数に対して加工を行ったのかを表 6 に示す．各パターンでタグの加工を行った結果，先頭記号パターンでは先頭に記号を持つ概念の場合には誤って処理を行ってしまう例 (“.net”など) がわずかに確認できた．また，頭文字パターンと分割可能パターンとして処理したタグについては，いくつか問題があることがわかった．

まず頭文字パターンでは，“e-ラーニング”などのタグが頭文字パターンと判別してしまい，結果として “ラーニング”と誤った加工を行なってしまう場合があった．これは，記号の前の 1 文字が「記号を挟んだ後の文字列の頭文字になりうるか」を判別するルールを追加することで改善できると考える．

次に，分割可能パターンとして処理したタグのパターンが 6585 と他に比べて多くなっているが，多くの誤った加工を行なっているためであると考えられる．分割可能パターンとしては，“(”，“)”を記号に含めているため，顔文字のタグが多く分解されてしまった．手法を適用した例には，“iPhone(app)”のようなタグも存在するが，それらに比べて “(´・ω・`)”などの，“(”，“)”，“.”を用いた顔文字が非常に多くあった．また，“.”(ドット) の前後でタグを分割した場合，“Web2.0”が “Web2”，“0”となってしまうような誤った例が多くあった．このことから，分割可能パターンに関してはルールが的確に設定できていないため，効果的な手法ではないことがわかった．単純な文字の一致ではなく，ユーザごとに複数タグをどのように連結して使用しているかを考慮する必要がある．

5.　使用傾向のベクトルによる表現

ここでは，SBM サービス内での「タグの使われ方」に注目し，タグ間の類似度を定義することでタグの集約を図る．

SBM サービス内のタグには，表記の特徴だけでは対応できない表記のゆれが存在する．例えば “Web サービス”と “Web ツール”など，ほぼ同じ意味を示すが，使う語の差によって別の表記になる同義のタグがある．このような同義のタグの表記のゆれに対して，タグがどのように使われているかの傾向に着目して手法を提案する．本研究では特に，タグを使用されたリソース（URL），共起タグ，使用したユーザの 3 点に注目して手法を提案する．

以降に示す手法，式を説明するために，ブックマークをユーザ u，リソース（URL）r，タグ集合 T の 3 つの要素で構成する対象と考え，(u, r, T) と表す．

5.1　類似度の算出方法

本手法では，タグを使用したユーザ，タグが使用されたリソース，タグと同時に使用された共起タグ，の 3 点からタグの使用傾向に着目する．

具体的には，タグとユーザ集合との関係，タグとリソース集合との関係，タグと共起タグ集合との関係をそれぞれ重み付けすることで，ベクトルとして表現する．このとき，任意の 2 つのタグ t_A，t_B から得られる各ベクトル (ユーザ，リソース，共起タグ) のコサイン類似度を算出することによって，t_A，t_B の類似度を定義する．t_A，t_B から得られるベクトルをそれぞれ，

(注5)：http://tartarus.org/ martin/PorterStemmer/index.html

$$V(t_A) = \{V(t_A)_1, V(t_A)_2, ..., V(t_A)_n\} \quad (1)$$

$$V(t_B) = \{V(t_B)_1, V(t_B)_2, ..., V(t_B)_n\} \quad (2)$$

としたとき，t_A，t_B のコサイン類似度は以下のように算出する．

$$Sim(t_A, t_B) = \frac{V(t_A) \cdot V(t_B)}{|V(t_A)||V(t_B)|} \quad (3)$$

2 つのタグ t_A，t_B が存在した場合，以下の 5.2.1 項，5.2.2 項，5.2.3 項に示す方法でベクトルを作成する．各ベクトルから得られた類似度の傾向を調査することで，類似度を用いた適切なタグの集約手法を考案する．

5.2 タグの特徴ベクトル作成手法

本節では，タグに対してリソース，共起タグ，ユーザのそれぞれに基づいてタグから特徴ベクトルを作成する．作成したタグの特徴ベクトルを（3）式を用いることで，2 つのタグ間の類似度を算出する．

5.2.1 タグに対するリソースベクトルの作成手法

本項目ではタグ間の類似度を，リソース（URL）によって定義する．データベース中に存在するユニーク URL が n 個存在したとき，図 1 のようにタグ t_A，t_B と，それぞれが使用されたリソース集合の関係を重み付けすることでグラフとして表現する．図 1 では，リソース r_1 にはタグ t_A が使用されたブックマークが 0 個，タグ t_B が使用されたブックマークが 3 個．リソース r_2 にはタグ t_A が使用されたブックマークが 1 個，タグ t_B が使用されたブックマークが 10 個存在するということを示している．

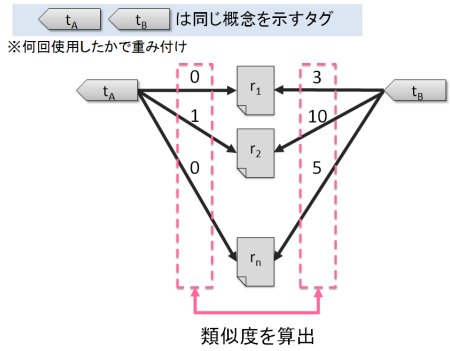

図 1　タグに対するリソースベクトルの作成

図 1 中の，それぞれ赤線で囲った数の配列を，タグ t_A，t_B から得られるリソースベクトル $V_{res}(t)$ として定義する．ここで，$V_{res}(t)$ を式で表すために，リソース r_i にはタグ t が使用されたブックマークが何個あるかを (4) 式に示す．

$$Num_{res}(r_i, t) = |\{(u, r_i, T)|t \in T, (u, r_i, T) \in B)\}| \quad (4)$$

以上を踏まえ，タグ t から得られるベクトルを (5) 式に示す．

$$V_{res}(t) = \{Num_{res}(r_1, t), Num_{res}(r_2, t), \\ ..., Num_{res}(r_n, t)\} \quad (5)$$

この t_A，t_B が持つリソースベクトル $V(t)$ から，(3) 式，(5) 式を用いてコサイン類似度を算出する．リソースベクトル $V_{res}(t)$ を用いて算出した類似度を**「タグ t_A，t_B のリソース類似度 $Sim_{res}(t_A, t_B)$」**と定義する．タグ t_A，t_B が同義である場合，それぞれは同じリソースに対するブックマークに使用される傾向があるため，このリソース類似度 $Sim_{res}(t_A, t_B)$ は高くなると考えられる．

5.2.2 タグに対するユーザベクトルの作成手法

本項ではタグ間の類似度を，使用しているユーザによって定義する．5.2.1 と同様に，データベース中に存在するユニークユーザが m 人存在する場合，図 2 のようにタグ t_A，t_B と，それぞれを使用したユーザ集合との関係を重み付けすることでグラフとして表現する．図 2 の数字は，ユーザ u_1 がタグ t_A，t_B を使用したブックマークが何個あるかを示している．図 2 では，ユーザ u_1 が t_A を使用したブックマークが 0 個，t_B を使用したタグが 3 個あることを示している．

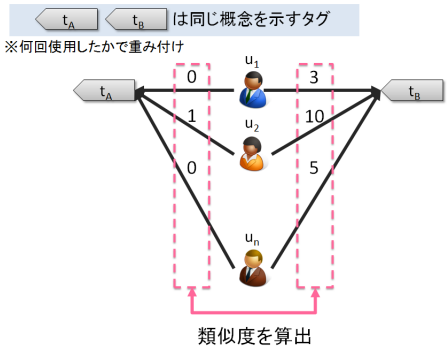

図 2　タグに対するユーザベクトルの作成

図 2 中の，それぞれ赤線で囲った数の配列を t_A，t_B から得られるユーザベクトルとして定義し，タグ t から得られるユーザベクトル $V_{user}(t)$ として定義する．ここで，$V_{user}(t)$ を式で表すために，ユーザ u_i がタグ t を使用したブックマークが何個あるかを，式 (7) 式に示す．

$$Num_{user}(u_i, t) = |\{(u_i, r, T)|t \in T, (u, r_i, T) \in B)\}| \quad (6)$$

以上を踏まえ，タグ t から得られるユーザベクトル $V_{user}(t)$ を (5) 式に示す．

$$V_{user}(t) = \{Num_{user}(u_1, t), Num_{user}(u_2, t), \\ ..., Num_{user}(u_m, t)\} \quad (7)$$

タグ t_A，t_B から得られるユーザベクトル $V_{user}(t)$ と，(3) 式，(7) 式を用いてコサイン類似度を算出する．この値を**「タグ t_A，t_B のユーザ類似度 $Sim_{user}(t_A, t_B)$」**と定義する．タグ t_A，t_B が同義である場合，使用するユーザの傾向は類似していないため，ユーザ類似度は低くなると考えられる．

5.2.3 タグに対する共起タグベクトルの作成手法

本項目ではタグ間の類似度を，共起タグによって定義する．そのためにまず，共起タグについて説明する．特定のタグ t から得られる共起タグ集合 T_c とは，タグ t が使用されたリソースに付与された，それ以外のタグの集合とする．以下の 4 つ

のブックマークから得られる場合について，共起タグの例を述べる．

$(u_1, r_1, \{t_1, t_2\})$
$(u_2, r_1, \{t_2, t_3\})$
$(u_3, r_1, \{t_2, t_4\})$
$(u_3, r_2, \{t_1, t_4\})$

ユーザ u_1，u_2，u_3 は，それぞれタグを使用して同じリソース r_1 をブックマークしている．このとき，タグ t_1 の共起タグは，リソース r_1 に付けられた他のタグ $\{t_2, t_3, t_4\}$ となる．また，タグ t_1 と t_4 は 2 つのリソース r_1，r_2 で共起している．このことから，タグ t_1，t_4 共起回数は 2 とする．

5.2.1，5.2.2 と同様に，データベースに存在するタグの種類が l 個あった場合，図 3 のようにタグと，それを使用したユーザ集合との関係を重み付けすることでグラフとして表現する．図 3 では，タグ t_A は t_1 と 0 回共起し，t_B は t_1 と 3 回共起したことを示している．このときタグ t_1〜t_l は総タグ種類数あるので，タグ t_A，t_B も存在する．そのため，タグの共起回数については**自身の共起回数は 0 とする**．

図 3 中の，それぞれ赤線で囲った数の配列を，タグ t_A，t_B から得られる共起タグベクトル $V_{ctag}(t)$ として定義する．このとき共起タグベクトル $V_{ctag}(t)$ を式で表すために，(8) 式に共起タグの数を示す．なお，B はデータベース中に存在する全ブックマークの集合とする．

$$Num_{ctag}(t', t) = \begin{cases} 0 & (t = t') \\ |\{r|(u_i, r, T_i), (u_j, r, T_j) \in B, \\ t \in T_i, t' \in T_j\}| & (t \neq t') \end{cases} \quad (8)$$

以上を踏まえ，タグ t から得られる共起タグベクトルを (9) 式に示す．

$$V_{ctag}(t) = \{Num_{ctag}(t_1, t), Num_{ctag}(t_2, t), \ldots, Num_{ctag}(t_l, t)\} \quad (9)$$

(3)，(9) 式を用いてコサイン類似度を算出し，その値を**「タグ t_A，t_B の共起タグ類似度 $Sim_{ctag}(t_A, t_B)$」**と定義する．タグ t_A，t_B が同義である場合，共起するタグは似ている傾向があるため，この共起タグ類似度は高くなると考えられる．

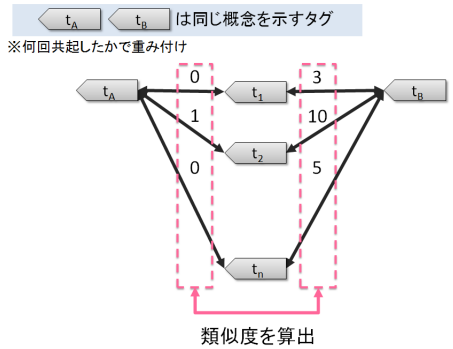

図 3　タグに対する共起タグベクトルの作成

5.3　有効な手法の調査

まず，5.2 の手法において，各ベクトルによる類似度がどのような傾向にあるか，またどの手法が最も有効かを調査した．あるタグに対して，残りの全タグを類似度の高い順にリストとして取り出す．そのリストの上位を被験者に評価をしてもらい，スコア付けを行った．このとき，累計のスコアが高い方がより同義のタグ，もしくは関連のタグを取り出していることを示す．このスコアを見ることで，各ベクトルから算出した類似度が高い値域において，どのような傾向があるか把握できると考える．

しかし，データベース中の全てのタグからペアを作成し，類似度を算出すると処理時間が非常に大きくなってしまう．そのため，表記の特徴による集約を行ったデータベースを用いる．しかし，それでも 60634 種類もタグがあるため，タグの対象を「使用するユーザが 5 人以上のタグ」に絞り，類似度の算出処理を行った．この結果，対象のタグは全てで 6712 種類となった．

表 7　調査に使用したタグ

c	ipod	windows	音楽	政治

表 7 に示すタグに対して，データベース中の残りのタグとの類似度を，リソースベクトル，共起タグベクトル，ユーザベクトルのそれぞれを用いて算出した．各ベクトルを用いて算出した類似度が 0.6 以上のタグのペアを，最大 5 件類似度が大きい順に抽出し，被験者 N 人に評価してもらった．このとき，タグのペアをランダムに並び替えて提示することで，「類似度が高い順」といった先入観を与えずに評価を得た．評価方法については，以下の基準でスコア付けをしてもらった．

- 抽出したペアのタグが同義であるならば 2
- ペアのタグが関連する語に当たるならば 1
- 関係のないタグである場合は 0

被験者と各ベクトルを用いて得られたスコアの関係を表 8 に示す．表 7 のタグ，ベクトルごとに得られた 5 件のタグのスコア合計を，10(スコアの最大値 2 と件数 5 の積) で割ることで正規化する．さらに，これを被験者ごとにまとめるために，得られた正規化したスコア合計を，調査した 5 つのタグで平均を算出し表 8 に示している．

また，表 7 中のタグとスコアの関係を表 9 に示す．ユーザ，ベクトルごとに得られた 5 件のタグのスコア合計を，同様に 10 で割ることで正規化する．これを被験者で平均を算出し表 9 に示している．上記 2 つの平均値は大きければ，同義もしくは関連するタグを取り出せていることを示す．また，1 に限りなく近ければ，5 件のタグは全て同義のタグであることを示している．

表 8，9 から，共起タグ類似度，ユーザ類似度に比べて，リソース類似度が大きいタグのペアはスコアが大きく，同義もしくは関連のタグが現れやすい傾向がわかった．また，リソース類似度によるタグのペアのスコアは，1 に限りなく近いというわけではないため，関連が強いが同義でないタグが多く現れていることが分かる．

また，スコア 2(同義のタグ)，1(関連するタグ) の割合を表 10

表 8　被験者ごとのスコア平均

	正規化したスコア		
被験者	リソースベクトル	共起タグベクトル	ユーザベクトル
A	0.58	0.32	0.12
B	0.64	0.36	0.16
C	0.44	0.08	0.02
D	0.42	0.16	0.08
E	0.50	0.28	0.12
全体平均	0.52	0.24	0.10

表 9　タグごとのスコア平均

	正規化したスコア		
	リソースベクトル	共起タグベクトル	ユーザベクトル
c	0.42	0.20	0.06
音楽	0.54	0.22	0.08
windows	0.58	0.30	0.00
ipod	0.46	0.28	0.12
政治	0.58	0.20	0.24
平均	0.52	0.24	0.10

〜13 に示す．被験者がスコアを 2 と付けた回数を，被験者，タグごとに平均を算出した結果をそれぞれ表 10，11 に示す．次に，被験者がスコアを 1 と付けた回数を，被験者，タグごとに平均を算出した結果をそれぞれ表 12，13 に示す．表 10，11 から，リソース類似度の高いタグに同義のタグが一番多く現れていることが分かる．また，表 12，13 から，リソース類似度の高いタグには関連するタグが一番多く現れていることも分かる．すべての被験者が，表 7 のすべてのタグでリソース類似度の高いタグにスコア 1,2 を多く付けている．このことから，被験者やサンプルによる評価の差は小さく，リソース類似度を用いることが一番有効であることが分かる．

さらに，表 7 中の各タグから取り出したタグを 5 件を各被験者に評価をしてもらったが，リソースベクトルを用いた場合は 5 件中の 4 件が同義のタグ，もしくは関連タグとなっている．このことから，他のベクトルを用いた評価との比較だけではなく，絶対的な評価としても有効であることが分かる．

表 10　被験者ごとのスコア 2 の割合

	スコア 2 の割合		
被験者	リソースベクトル	共起タグベクトル	ユーザベクトル
A	1.4	0.20	0.20
B	2.0	0.20	0.20
C	1.2	0.00	0.00
D	0.60	0.00	0.00
E	1.0	0.00	0.00
平均	1.2	0.08	0.08

実際にタグ “windows” と他のタグ，“ipod” と他のタグの 3 つの類似度を算出し，各類似度が高い上位 5 件のタグの例を表 14，15 に示す．表 14，15 から，リソース類似度を用いた場合，関連するタグではあるが同義でないタグや，上位・下位に相当するタグも現れていることが分かる．一方，共起タグ類似度，ユーザ類似度からは，同義のタグが得づらいことが分かった．

表 11　タグごとのスコア 2 の割合

	スコア 1 の割合		
タグ	リソースベクトル	共起タグベクトル	ユーザベクトル
c	2.4	1.2	1.0
音楽	2.6	2.2	0.80
windows	2.2	3.0	0.00
ipod	2.2	2.0	0.40
政治	3.8	2.0	2.4
平均	2.6	2.1	0.92

表 12　被験者ごとのスコア 1 の割合

	スコア 1 の分布		
被験者	リソースベクトル	共起タグベクトル	ユーザベクトル
A	2.8	2.0	1.2
B	2.4	3.2	1.2
C	2.0	0.80	0.20
D	3.0	1.6	0.80
E	3.0	2.8	1.2
平均	2.6	2.1	0.92

表 13　タグごとのスコア 1 の割合

	スコア 1 の割合		
タグ	リソースベクトル	共起タグベクトル	ユーザベクトル
c	1.8	0.80	0.80
音楽	2.0	1.4	0.60
windows	1.8	2.2	0.00
ipod	1.6	1.4	0.20
政治	3.0	1.8	1.8
平均	2.0	1.5	0.68

共起タグベクトルによる類似度が高いタグには同義のタグより，関連語が多く現れる傾向が見られる．また，ユーザ類似度が高いタグのペアには，使用しているユーザが近いだけのペアが現れるため，直接的な関連が分からないペアが多く見られる．また関連語も僅かに得られるが，その場合は「下位に相当する概念」，および下位の概念に関連する概念といった,「直接的な関連性は小さい概念」を示すタグが得られている．このことから，上位，下位関係に相当するタグ間において，ユーザ類似度は大きくなると考えられる．

表 14　タグ “windows” との類似度の高いタグの例

リソースベクトル	共起タグベクトル	ユーザベクトル
xp	ソフトウェア	google
windowsxp	ソフト	iphone
win	パソコン	firefox
update	excel	chrome
高速化	mac	android

6. 使用傾向による集約手法の実験

6.1 手法の有効性の確認

前章において，適合率の観点から，各ベクトルによる類似度を用いて比較実験を行った．その結果，リソース類似度が高いタグには同義のタグが出現しやすい傾向が分かった．そのため

表 15　タグ “ipod” との類似度の高いタグの例

リソースベクトル	共起タグベクトル	ユーザベクトル
touch	touch	エネルギー産業
ip 電話	ios5	touch
ipodtouch	モバイル業界	標準化
apri	あとで見ない	リスク管理
移行	benri	990

本章では，リソース類似度を用いた場合の有効性を確認するために，適合率，再現率，F 値を調査することでリソース類似度を用いることの有効性を確認する．

まず，前章と同様に，使用するユーザ数が 5 人以上のタグに対してリソース類似度の算出を絞り，対象のタグは全てで 6712 種類とする．これらのタグから全ての組み合わせで「リソース類似度を持つタグペアのデータセット」を作成する．そのため，ペアの数は 6712 の ${}_{6712}C_2$ 個となる．

次に，被験者に評価してもらうデータの取得方法について詳しく述べる．上述のデータセットからタグのペアを取得するが，このとき，以下の範囲からタグのペアを取得する．

- $0 \leqq Sim_{res} \leqq 0.2$
- $0.2 < Sim_{res} \leqq 0.4$
- $0.4 < Sim_{res} \leqq 0.6$
- $0.6 < Sim_{res} \leqq 0.8$
- $0.8 < Sim_{res} \leqq 1.0$

各範囲から 20 件ずつランダムにタグのペアを取得し，タグのペアが全 100 個となるように取得する．このようにペアを取得することで，どの類似度の範囲からどのようなタグのペアが得られるのかを調査しやすくする．取得した 100 個のペアを，被験者 5 人に分担して評価してもらう．前章での実験同様に，以下の基準でスコア付けをしてもらった．このとき，被験者に提示するタグのペアはランダムに並び替えており，先入観を持たせることなく評価を得た．

- 抽出したペアのタグが同義であるならば 2
- ペアのタグが関連する語に当たるならば 1
- 関係のないタグである場合は 0

この評価とは別に，システムによってリソース類似度 Sim_{res} がしきい値 θ 以上のタグを集約する．取得した 100 個のペアに対して，類似度が θ 以上のタグを集約する場合，集約するペアの集合を P_{sys} とする．被験者が 2(同義) と評価したタグのペアは集約するべきタグのペアと考え，被験者の過半数が 2 としたタグのペア集合を P_{user} とする．このとき，しきい値 θ によって集約するタグのペアが正解となる適合率，再現率，および F 値は以下の式 10，11，12 で算出する．

$$\text{適合率} = \frac{|P_{sys} \cap P_{user}|}{|P_{sys}|} \tag{10}$$

$$\text{再現率} = \frac{|P_{sys} \cap P_{user}|}{|P_{user}|} \tag{11}$$

$$F \text{ 値} = \frac{2 \cdot \text{適合率} \cdot \text{再現率}}{(\text{適合率} + \text{再現率})} \tag{12}$$

横軸にしきい値 θ，縦軸に適合率，再現率および F 値をとったグラフを図 4 に示す．

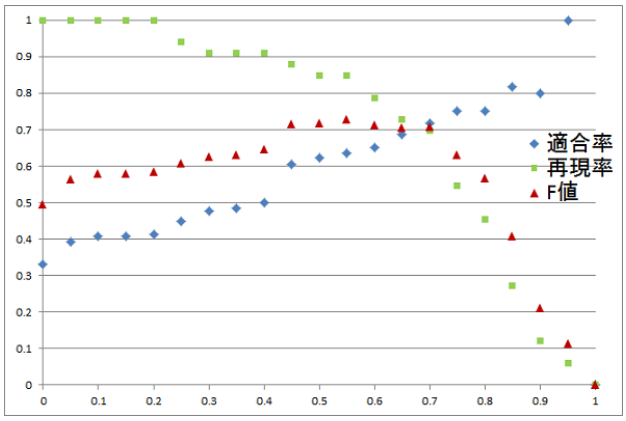

図 4　タグに対するリソースベクトルの作成

図 4 から，しきい値を大きくすると適合率は大きくなり，同義のタグを取得しやすくなることが分かる．一方，しきい値を大きくすると再現率は小さくなり，同義のタグ全体に対して，システムが同義と判定するタグが少なくなるが分かる．また，F 値はしきい値 θ が 0.55 で最大値をとり，0.45 から 0.70 の区間でおよそ最大値をとり続けている．このことから，θ が 0.45 から 0.70 までの閉区間では，システムの性能はほぼ変わっていないといえる．

本研究では集約を行うことが目的であるため，集約する数が多いほうが良いが，集約する適合率が低いことには問題があると考える．そのため，再現率より適合率が高いことを優先する必要がある．よって，F 値が横ばいの 0.45 から 0.70 の区間で，適合率が最も高いしきい値 θ が 0.70 を集約の基準として用いる．このことから，**しきい値 θ が 0.70 以上のペアを集約する条件とする**．

6.2　使用傾向による集約結果

前節の条件を用いて，データベース中のタグに対して集約する処理を行う．このとき，リソース類似度が大きいタグのペア (t_A, t_B),(t_A, t_C) が存在する場合，リソース類似度 $Sim_{res}(t_B, t_C)$ に注目する必要がある．$Sim_{res}(t_B, t_C)$ がしきい値より大きい場合は，t_A, t_B, t_C の中で，使用ユーザ数が最も大きいタグで統一できる．一方，$Sim_{res}(t_B, t_C)$ がしきい値より小さい場合，t_A, t_B と t_A, t_C に比べて，t_B, t_C は意味的に遠いといえる．

しかし，仮に t_A, t_B を統一した後のリソースベクトルと，t_C のリソースベクトルを比較した場合，t_A, t_B を統一した後のリソースベクトルには t_C と類似する t_A のリソースの要素・傾向を含んでいるといえる．また，逆に t_A, t_C を統一した後のリソースベクトルと，t_B のリソースベクトルを比較した場合も同様のことが言える．このことから，タグ t_B, t_C を使用しているユーザの傾向が違ったためリソース類似度は小さいが，タグ t_B, t_C は本質的に同じ意味を指していると考える．そのため，本研究ではリソース類似度が大きいタグを全て，使用ユーザ数が大きい方のタグへ集約する手法をとる．

実際に，リソース類似度が 0.70 以上のタグのペアに対して，

上記の方法で集約を行った．リソース類似度が 0.70 以上のタグのペアは 708 件取得できた．それらのタグのペアを集約した結果，708 個のタグを 279 個まで集約できた．集約したタグの例をいくつか以下の表 16 に示す．表の被集約の列が集約されるタグで，集約先の列が集約後に残るタグである．

表 16　集約したタグのペア例

被集約	集約先
データベース	db
im@s	アイドルマスター
movie	映画
cinema	映画
aprilfool	エイプリルフール
4 月 1 日	エイプリルフール
眼鏡	メガネ
wi	fi
e	learning
e ラーニング	learning
ハァハァ	´д‘
^o^	＼
ω	顔文字
パブリックドメイン	mp3
楽天	shopping
イイハナシダナー	これは泣ける
中東	エジプト
日本代表	サッカー

表 16 の例から，“データベース”と “db”，“im@s”と “アイドルマスター”のような略語の関係は集約できている．また “movie”，“cinema”といった違う単語の同義のタグも “映画”に集約できている．英語と日本語を統一するだけでなく，“エイプリルフール”に関しては，英語 “aprilfool”の場合も，日にち表記である “4 月 1 日”も集約できている．また，“眼鏡”と “メガネ”のような漢字や片仮名の差も吸収できている．

しかし，主な問題点として，表記の特徴に注目した手法で分割したタグが，意味を成さない形になってしまっている例が多くあった．例えば “wi-fi”の場合は，表に示すように “wi”と “fi”に分割され，さらにタグ “wi”が “fi”に集約されてしまっている．また，タグ “e-learning”の場合は，“e”と “learning”に分割されたため，個別のタグと扱われてしまった．さらに “e”に比べて，一般的に使用する機会が多い “learning”の使用ユーザ数が大きいため，“e”は “learning”に集約されてしまったと考えられる．また 4.2 節で述べたように，指定した分割記号に顔文字を構成する記号が多くあったため，記号 1 文字のタグがあったり，顔文字を分割したものが集約されているような例が確認できた．これらは，表記の特徴に注目した手法を改良することで，改善できると考えられる．

次に，少数のタグが，多くの URL に使用されるタグに集約されてしまっている例も確認できた．例えば，著作権に関するタグ “パブリックドメイン”が，主にクラシックミュージックの話題に使用されるため，音楽系のタグである “mp3”に集約されてしまった．また，Web 商品サイトである “楽天”が，その上位語にあたる “shopping”に集約された例もあった．このことから，使用傾向に注目した手法については，上位語を判定する手法が必要であると考えている．リソース類似度が高いタグのペアに対して，上位語を判定して取り除くか，上位語のスコアを小さくする手法を適用することで適合率が増加すると考えられる．

データセットの問題点として，データ収集期間の主な出来事，ニュースの情報に偏りがあったと考えられる．“中東”と “エジプト”は，地域と国の関係であるため，同義とはいえない．しかし，データ収集期間中に中東，エジプト関連のニュースが同時に発信されたため，リソース類似度が大きくなってしまい，同義であると判定してしまった．また，サッカー日本代表の話題が多い時期にデータ収集を行ったため，“日本代表”と “サッカー”は同義であると判定してしまった．これは，長期間収集し続けたデータセットを用いることで解決できると考えられる．

7. ま と め

本稿では，ソーシャルタグの表記と使用傾向に基づく集約手法を提案し，手法の評価実験を行った．まず，表記の特徴による集約について集約実験を行った結果，タグを 60634 種類から 37637 種類に集約したことから，非常に多くのタグを集約することができたといえる．しかし，頭文字パターン，分割可能パターンにおけるルールが不適切であるため，タグが単体では意味の分からない文字列に分割してしまった例が多く確認できた．

使用傾向に基づく集約手法については，リソース，共起タグ，ユーザに着目して特徴ベクトルを作成し，特徴ベクトルの類似度を基準にすることで，同義のタグを集約できるか確認実験を行った．確認実験から，リソース類似度を用いる手法が最も有効だと結論を得た．

次に，リソースベクトルを用いてリソース類似度を算出ししきい値以上のタグを集約する手法を提案し，実験を行った．実験からしきい値は 0.70 と定め，その時の F 値は 0.7 程度となり，有効な手法であると確認できた．しかし，この手法の問題点として，リソース類似度が大きいタグには上位語，下位語の関係にあるタグのペアも同義であると判定してしまうことが分かった．

最後に，表記の特徴による集約を行ったブックマーク情報に対して，使用傾向に基づくタグの集約を行った．集約を行った結果，手法によって 709 件のタグのペアが，しきい値以上の類似度を示したため取得できた．これらを集約した結果 709 種類のタグを，279 種類にまで集約することができた．

今後の課題について順に述べる．まず，表記の特徴による集約については，ルールを適切に設定する必要がある．本手法で提案しているルールでは単純すぎるため，頭文字の判定などを追加することで改善できると考えられる．次に，使用傾向による集約手法については，F 値が 0.7 と大きい結果が得られているが，上位・下位の語を判定する手法を考案，適用する必要がある．上位・下位の語を集約する対象から除くことで，更に適合率が増加すると考えられる．

文　　献

[1] 丹羽 智史, 土肥 拓生, 本位田 真一: Folksonomy マイニングに基づく Web ページ推薦システム, 情報処理学会論文誌, Vol.47, No.5, 1382-1392, 2006.

[2] 川中 翔: ソーシャルブックマークにおけるタグの派生関係の抽出, 東京大学大学院 基盤情報学専攻修士論文, 2009.

[3] 藤村 滋, 藤村 孝, 片岡 良治, 奥 雅博: Blog のタグ間類似度のスコアリング, DBSJ Letters Vol.5, No.4, http://www.dbsj.org/journal/vol5/no4/fujimura.pdf