CE-002

SNS 間の話題遷移を推定する自己組織化マッピング
Self Organization Map to estimate topic transition between SNS.

浦野 史隆 *
Fumitaka Urano

六井 淳 *
Jun Rokui

1 概要

近年、IT の進化によりインターネットを介することで個人での情報発信が可能な CGM(Consumer Generated Media) の利用が爆発的に増加した。[*1] 使用されるメディアもブログ、twitter[*2]、facebook[*3]、instagram[*4]など豊富に存在し、ユーザが自由に選択することが可能である。発信されている CGM 情報にはユーザの忌憚ない意見が豊富に含まれており、情報源として注目されている。2011 年に行われた NTT コムリサーチ[*5]の調査によると、調査対象約 2000 人のうち、クチコミの影響を受ける消費者は 8 割、クチコミが購入の決め手になる消費者は 4 割と、ユーザのレビューが消費者の購買決定に大きな影響を与えると報告された。これらの情報は web 上に日々蓄積されている。この蓄積された情報を解析して情報の傾向や有用な情報を抜き出す研究が盛んに行われている。

川添らは収集したブログ記事情報を様々な尺度からテキストマイニング技術を用いて統計的に分析し、ビジネスへの有効活用を図っている [1]。南野らは web ページの HTML 情報からブログ判定を行い、ブログの網羅的な自動収集システムを提案している [2]。Sitaram Asur らは twitter 情報から映画の人気度、それに伴う売上の予測モデルを構築し、Box-office[*6]で公開されている実際の売上データから R^2値、P 値 などによって予測モデルの精度を検証した [3]。内田らはブログ記事の Trackback によるネットワーク性に着目し、クラスタリングを適応、クラスタに分割された記事から特徴語を抽出し、クラスタごとの特徴語を視覚化することに成功した [4]。

現在では CGM の中でも各種ブログサービス、twitter,facebook,instagram など様々な SNS(Social Networking Service) が普及している。ICT 総研の調査によると、国内 SNS の登録件数は重複登録を含めて約 2 億 8000 件という報告がなされており、一人で複数の SNS を運営するユーザも珍しくない[*7]。

SNS の普及、種類の増加に伴い、SNS は他の SNS と連携するサービスを提供している。twitter では、投稿したツイートを facebook に反映することができる[*8]。facebook と instagram は両サービスへの同時投稿機能を提供している[*9]。このように、近年では SNS 同士が密接に関わっている。この関係性を抽出し、CGM 情報解析に利用することができれば、正確な解析結果が期待出来る。保住らは複数の web 上のデータから製品の潜在需要、消費動向を分析し、予測を行うシステムを構築した。単一のデータではなく複数のデータを用いることでより精度の高い予測モデルを構築できることを示した [4]。

本研究では twitter を対象に、話題の遷移について解析を行った。解析の際には twitter と比較してリアルタイム性が低いブログ記事情報を取り込み、ブログ記事での話題の遷移を基に twitter 上での話題の遷移を解析し、結果を行列で表現した。作成した行列から関係性を抽出するために行列を SOM(Self Organization Map)[6] に入力して話題の遷移を学習させ、将来的な話題遷移の推定を試みた。解析の際には twitter とブログ記事の特徴に適した方法での話題抽出を行い、データを時間単位で選択することでリアルタイム性に考慮した。

2 解析手法

ツイートとブログ記事の関係性を抽出し、行列に格納する。ブログ記事から話題を象徴する特徴語を抜き出し、ツイートが特徴語に言及しているか分類を行う。特徴語抽出とツイート分類については、本研究の関連研究で行った [7]。手法の概要を示す。

1. TF-IDF と文の出現位置による重み付けにより、一日分のブログ記事から特徴語を抽出する [8]。抽出された特徴語を次式とする。

$$W_{bd} = w^1_{bd}, w^2_{bd}, ..., w^i_{bd}, ..., w^n_{bd} \quad (1)$$

W_{bd}は bd 日のブログ記事から抽出された上位 n 件の特徴語群を示す。

2. 収集した一日分のツイートを時間帯ごとに分割する。分割されたツイート群を次式で表す。

$$T_{td} = T^1_{td}, T^2_{td}, ..., T^i_{td}, ..., T^z_{td} \quad (2)$$

td はツイートが投稿された日付を、z はツイートの時間帯を示す。例として z=2 の場合、2 分割であることを示し、T^1_{td} は 0 時から 12 時のツイート群、T^2_{td} は 13 時から 24 時のツイート群となる。

3. 関連研究で提案した圧縮によるツイート分類アルゴリズムを用いて、(1) と (2) の任意の組み合わせに対

* 島根大学大学院総合理工学研究科

*1 http://gaiax-socialmedialab.jp/post-30833/
*2 http://twitter.com/
*3 https://www.facebook.com
*4 https://www.instagram.com
*5 http://research.nttcoms.com/database/data/001436/
*6 www.boxofficemojo.com/
*7 http://ictr.co.jp/
*8 https://www.facebook.com/notes/
*9 https://www.facebook.com/help/instagram/356902681064399

して関連性を次式で計算する [7]。

$$g(x, w_{bd}^i) = \frac{C_{A_{w_{bd}^i}}(x) + \gamma}{C_B(x) + \gamma} \quad (3)$$

x は分類したいツイート、w_{bd}^iは式 (1) の任意の特徴語を表す。$C_{A_{w_{bd}^i}}(x)$、$C_B(x)$ はそれぞれ次式で定義される。

$$C_{A_{w_{bd}^i}}(x) = Z(A_{w_{bd}^i} + x) - Z(A_{w_{bd}^i}) \quad (4)$$

$$C_B(x) = Z(B + x) - Z(B) \quad (5)$$

$A_{w_{bd}^i}$ は特徴語 w_{bd}^iに関するツイートを、B はその他のツイートを数百件文字列連結させたものである。式 (4) 式と (5) は Benedetto らの手法に基づき、分類したいツイート x の圧縮されやすさを計算する[9]。分類したいツイート x が特徴語 w_{bd}^iに関連している場合、式 (4) によってよく圧縮され、式 (3) の値は小さくなる。γ はスムージングパラメータである。

4. 3 の結果を行列に格納する。この時、時系列を配慮して列方向はブログ記事上での時間経過、行方向はツイート上での時間経過に対応させる。

3 提案手法

関連研究にて提案した手法の概略を図 1 左に示す。関連研究では各特徴語とツイートとの関連性値を計算し、その最小値を行列に格納した [7]。この方法では最小値に選ばれた関連性値しか考慮されず、他の特徴語との関連性情報が失われてしまう。そのため本研究では図 1 右のように最小値を計算せず、計算した関連性値情報全てをベクトルとして保持して行列に格納する。

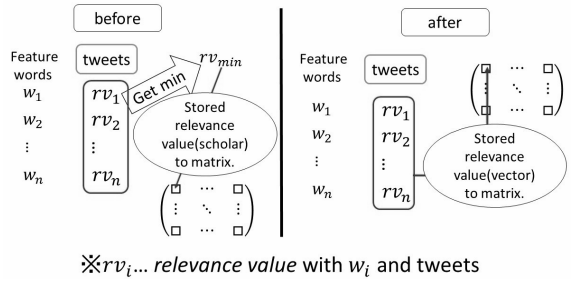

図 1 変更点
difference with former paper.

3.1 行列作成手順

行列作成手順の具体的な変更点を示す。

式 (1) と、式 (2) の 1 つの要素との関連性値 RV を次式とする。

$$RV = rv_{w_1}^z, rv_{w_2}^z, ..., rv_{w_n}^z \quad (6)$$

w_iは特徴語を、z はツイートの時間帯を表す。従来では、式 (6) の最小値を取得し、行列に格納した。式 (6) の値は各特徴語との関連性値となっているため、最小値以外を使用しない従来法は他の特徴語との関連性情報が欠落してしまう。

そこで、全特徴語との関連性情報を保持するため、式 (6) を次式で表す。

$$vec_{bd}^z = (rv_{w_1}^z, rv_{w_2}^z, ..., rv_{w_n}^z) \quad (7)$$

式 (7) は、時間帯 z に投稿されたツイートと bd 日でのブログ記事の各特徴語 w_iとの関連性をベクトル表記でまとめたものとなる。

式 (7) で得られた vec_{bd}^zを

$$V_{td} = \begin{bmatrix} vec_1^1, vec_1^2, ..., vec_1^z \\ vec_2^1, vec_2^2, ..., vec_2^z \\ ..., \\ vec_{bd}^1, vec_{bd}^2, ..., vec_{bd}^z \end{bmatrix} \quad (8)$$

のように行列に格納する。

ここで、式 (8) の行ベクトルを次式とする。

$$vec_i^1, vec_i^2 ..., vec_i^z \quad (9)$$

式 (9) の z はツイートの時間帯を表している。このため、式 (9) は twitter 上の時間経過による、twitter 上の話題と i 日のブログ記事上の話題との関連性の変化を表す。

また、式 (8) の列ベクトルを次式とする。

$$vec_1^i, vec_2^i, ..., vec_{bd}^i \quad (10)$$

式 (10) の bd はブログ記事の日付を表す。このため、式 (10) はブログ記事上の時間経過による、時間帯 z での twitter 上の話題とブログ記事上の話題との関連性の変化を表す。

3.2 SOM

作成した行列を SOM に入力して話題の遷移を学習させる。SOM とは、コホネンによって提案されたニューラルネットワークの一種である教師なし学習モデルである [6]。特徴として、入力データを任意の次元へ写像し、結果を視覚化できる。SOM による学習過程を Algorithm1 で示す*10。

Algorithm1 のステップ 5 にあるように、学習前半は広範囲のセルの値を更新し、後半になるにつれて更新する範囲を狭める。更新範囲を決定する近傍半径は次式で表される。

$$\theta(t) = \exp(-d^2/\alpha^2) \quad (11)$$

d は更新する周囲のセルと、Algorithm1 のステップ 3 で選択された重みを持つセルとの距離を表す。α は次式となる。

$$\alpha = 1 - t/e \quad (12)$$

t は現在のループ回数、e は学習終了とする回数である。学習前半は式 (12) の値が 1 に近くなり、式 (11) の値は距離を示す d が小さい (近い) ほど大きくなる。学習後半は式 (12) の値が 0 に近くなり、式 (11) の値は小さくなる。

学習終了後、各 SOM のセルには類似したデータが隣接するようになる。この性質を利用して、SOM は分類問題に応用されている。

波多野らは HTML データ群を SOM に入力すること

*10 https://ja.wikipedia.org/wiki/自己組織化写像

Algorithm 1 学習過程

·t=現在のループ回数

·e=学習終了とする回数

·W_v=現在の重みベクトル

·$\theta(t)$=近傍半径

·D=入力ベクトル

start

· 全ての重みを初期化する。

while $t < e$ **do**

1. 入力データを一つ選択する。
2. SOM の全重みと入力データの類似度を計算する。
3. 最も類似した重みを選択する。
4. 選択された重みを次式によって入力データに近づくよう更新する。

$\rightarrow W_v(t+1) = W_v(t) + \theta(t)\alpha(t)(D(t) - W_v(t))$

5. 更新する重みの周囲にある重みも同様に更新する。

·t が大きいほど$\theta(t)$ は小さくなり、更新される周囲の重みの数も減少する。

6. $t = t + 1$

end while

end

で、HTML データ群から抽出した特徴語を視覚的に表現した。これによって、ユーザによる検索の結果を大幅に圧縮し、視覚化するシステムを構築した [10]。西山らは可降水量、風速などの気象情報を入力することで、梅雨期の複雑な関連性と豪雨場を学習させた。その結果、梅雨期の典型的な特徴の学習に成功し、未知のデータに対して豪雨場の抽出を可能にした [11]。

本研究では SOM の類似したデータが隣接する性質を利用して、入力する話題遷移の類似性を学習させる。話題遷移学習のために SOM に入力するデータは式 (9) である。これを行列で表現すると次式のようになる。

$$vec_i^1, vec_i^2 ..., vec_i^z = \begin{bmatrix} rv_{w_1}^1, rv_{w_1}^2, ..., rv_{w_1}^z \\ rv_{w_2}^1, rv_{w_2}^2, ..., rv_{w_2}^z \\ ..., \\ rv_{w_n}^1, rv_{w_n}^2, ..., rv_{w_n}^z \end{bmatrix} \quad (13)$$

式 (13) の要素 $rv_{w_n}^z$ は n 番目の特徴語 w とツイートの時間帯 z との関連性値である。SOM の学習過程において、式 (13) の行ベクトルが選択されるデータである。学習後の SOM は特徴語別の、ツイートの時間経過による類似した話題遷移が周囲に隣接する。

式 (13) の 時間帯 z は、td 日につぶやかれたものであることを示す。これをツイート td 日分のみを入力することで、td 日分のみの話題遷移を学習したSOM を取得できる。この時、td 日と $td+1$ 日の式 (13) を同時に入力することで 2 日分のデータを学習したSOMを取得できる

図 2 のように入力するデータ数を増加させ、複数日分の話題遷移を学習した SOM を作成する。

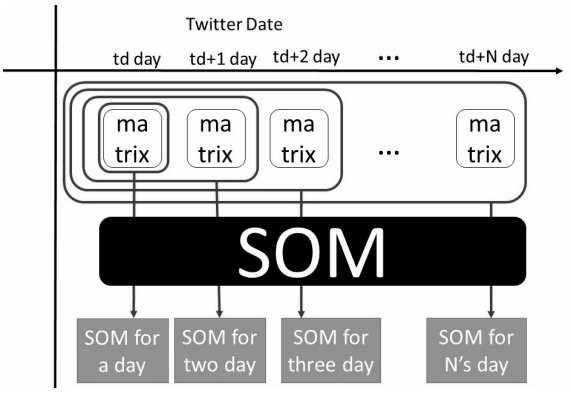

図 2 N 日分の SOM
SOM for N's day.

生成される SOM の例を図 3 で示す。

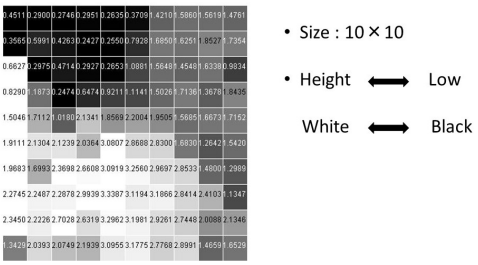

図 3 生成される SOM
created SOM.

大きさは 10 × 10、値は SOM 上の最大値と最小値でスケーリングしており、大きいほど白く、小さいほど黒くなっている。セルには式 (7) のように各特徴語ごとの関連性がベクトルで格納されている。SOM での表現の都合上、ベクトルのスカラ値を計算して表示している。

3.3 差分 SOM

N 日分の SOM を推定するため、作成した SOM から図 4 のようにして変化量を抽出する。はじめに、差分 SOM を取得する。これは N 日分の行列から N-1 日分の行列を減算することによって取得される。この差分 SOM は N-1 日の SOM から N 日の SOM への変化量、つまり一日分のデータを追加した場合の変化量を表している。

取得した差分 SOM から変化量を抽出し、SOM の推定に用いる。

特徴量抽出には pooling を使用した。pooling とは、特徴量を抽出してデータ量を削減する手法である。図 5 は MAX pooling と呼ばれる手法の例である。フィルターサイズ 2 × 2 より、該当部分の最大値を取得する。図 5 では同じ色の部分が該当部分に当たる。

全セルの変化量を取得するため、本研究では変化量の標準的な指標である標準偏差を抽出した。抽出の例を図 6 に示す。SOM の サイズ 10 × 10 に対し、窓幅 2 × 2、スライド幅 2 で pooling を行った。

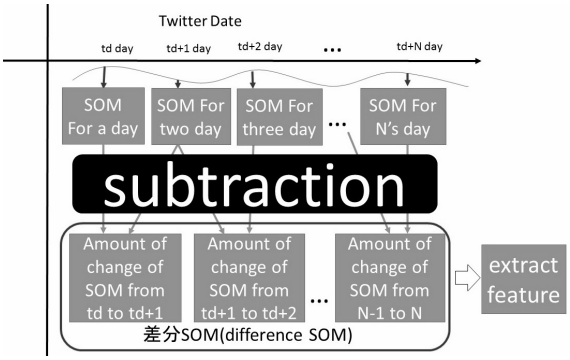

図 4　差分 SOM と特徴抽出
difference SOM and feature extraction.

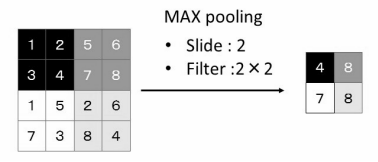

図 5　pooling

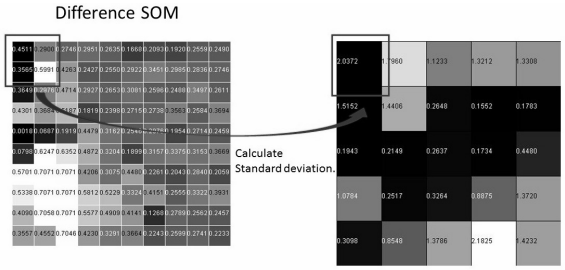

図 6　特徴抽出
feature extraction.

4　推定手法

3.3 章で取得した特徴量を用いて SOM の推定を行う。例を図 7 で示す。

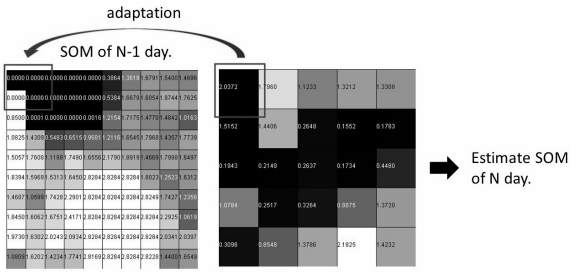

図 7　SOM の推定
estimate SOM.

図 7 右の特徴量のうち黒枠の部分を、図 7 左にある N-1 日の SOM の黒枠部分に適応させる。

適応例を図 8 に示す。各グラフの横軸は i 日分の SOM を、縦軸は SOM の特定のセルの値を示す。差分 SOM を計算する際に、セル毎に差分 SOM の値が正だった回数、負だった回数を記録する。正の値が多いセルでは特徴量を減算、負の値が多い部分では加算を行い、特徴量を適応させる。

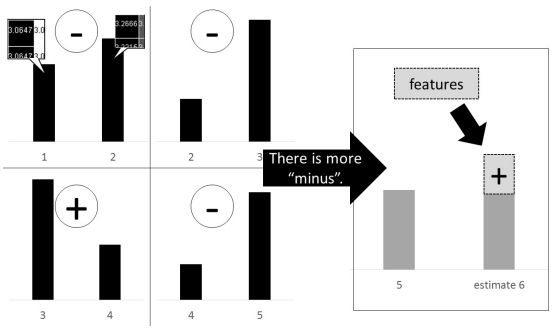

図 8　適応例
Example of adaptation.

以上の操作を、図 7 の黒枠部分が重複しないように行う。適応後に得られる SOM を N 日の推定 SOM として扱う。

5　検証実験

実験で使用したデータの詳細を表 1 に示す。

	期間 (2016 年)	件数	収集 API
twitter	6/17〜6/22	829,328 件	twitter4j[*11]
yahoo ブログ	6/18〜6/28	214,181 件	自作

表 1　データ表
data table

使用した yahoo ブログ記事のジャンル一覧を表 2 に示す。

エンターテインメント	ビジネスと経済
家庭と住まい	学校と教育
芸術と人文	健康と医学
コンピュータとインターネット	科学

表 2　ジャンル表
genre table

SOM の推定精度は図 9 に示すように、N 日の推定 SOM と、N 日分の行列を入力して実際に作成された SOM の差分から評価する。得られる SOM(以下誤差 SOM) は、正解値と推定値の誤差を示している。

得られた誤差 SOM は

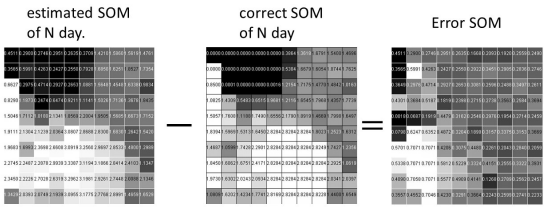

図 9 推定精度の検証
testing of estimate precision.

1. 誤差 SOM 単体の詳細
2. 誤差の累積値による全体の誤差

の 2 点から評価する。

5.1 誤差 SOM 単体の評価

得られた誤差 SOM の各セルを昇順にソートし、棒グラフで表現する。図 10 で例を示す。グラフ上部の数値は誤差 SOM を取得するのに入力した行列数を表す。縦軸は誤差の値、横軸はセルの数を示す。今回は SOM のサイズが 10× 10 なので横軸は 1 から 100 までの値をとる。

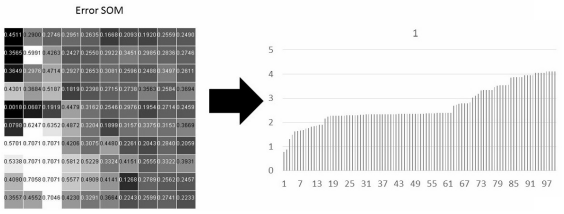

図 10 例:結果の表現
example:expression of result.

5.2 全体の評価

誤差 SOM 各セルの値を合計した累積誤差を取得する。図 11 で示すように、各データ数ごとに累積誤差を算出し、データを追加した場合の累積誤差の変化を検証する。

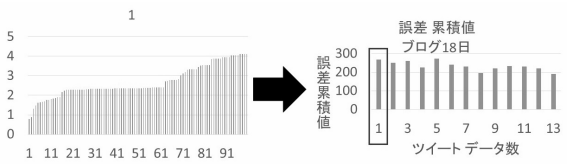

図 11 例:全体の評価
example:overall evaluation.

5.3 検証結果

結果をジャンルごとに示す。全体の評価結果にある直線は最小二乗直線であり、これが右下がりだと全体的に誤差が減少していることを示している。

図 12 では、入力データ数を追加するたびに左側が小さくなっている。この棒グラフは予め昇順にソートしているので、データを追加するたびに推定精度が上昇していることを示している。図 13 を見ると、4 つの結果全てで最小二乗直線が右下がりになっており、全体の誤差も小さくなっている。「芸術と人文」、「科学」、「家庭と住まい」、「ビジネスと経済」のジャンルで同様の傾向が確認できた。

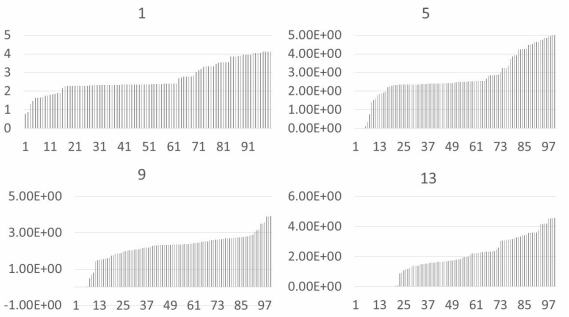

図 12 単体 ジャンル:エンターテインメント
single genre:entertainment.

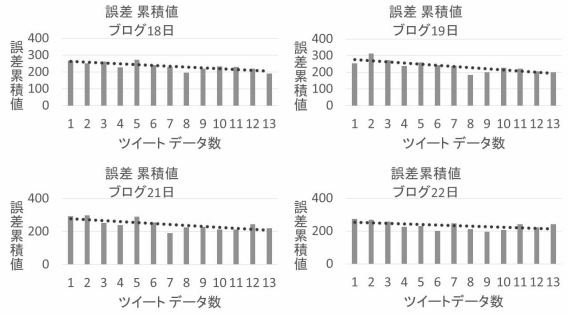

図 13 全体 ジャンル:エンターテインメント
overall genre:entertainment.

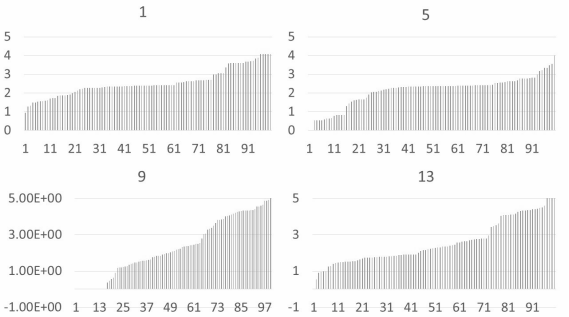

図 14 単体 ジャンル:コンピュータとインターネット
single genre:computer and internet.

図 14 では 9 日分までは順調に誤差が減少しているが、13 日分では誤差が大きくなっている。図 15 をみると最小二乗直線は右下がりであり、全体的に誤差が小さくなっているが、徐々に横ばいになってきている。「学校と教育」のジャンルで同様の傾向が確認できた。

図 16 では最小二乗直線が右上がりになっているのが確

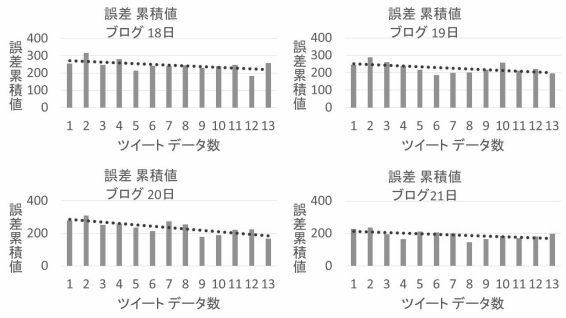

図 15 全体 ジャンル:コンピュータとインターネット
overall genre:computer and internet.

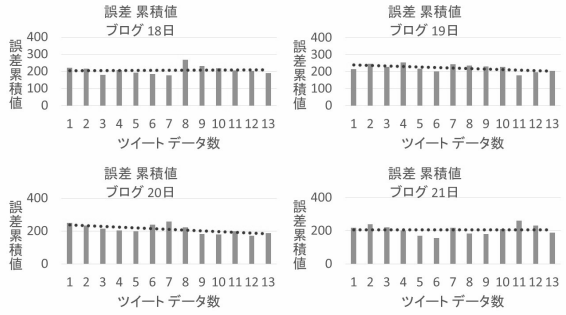

図 16 全体 健康と医学
overall genre:health and medical science.

認できる。個々の値を見てみると、左上のグラフではデータ数 8 を境に誤差が大きくなっている。右下のグラフではデータ数 7 以降、誤差が徐々に大きくなっている。今回の実験で唯一誤差が大きくなった。

6 考察

図 13、図 15 では最小二乗直線が右下がりになっており、データ数を増加させることで推定の精度が向上したと考えられる。

図 14 では 9 日分よりも 13 日分の方が誤差が大きかった。図 16 では全体の誤差が大きくなっていることが確認された。これは、入力した過去のデータが推定の精度を低下させていることが考えられる。twitter 情報は話題の移り変わりが激しいためこのような結果になったと推測される。図 13 のようにジャンルによっては精度が上がり続けているため、ジャンルごとの話題が移る速度が異なることも考えられる。

7 まとめと今後

各特徴語ごとの関連性情報を保持した新たな行列表現手法を提案した。SOM を使用し、twitter とブログ記事間の話題遷移を学習させ、N 日分の SOM や差分 SOM による SOM の推定手法を提案した。提案した手法による推定の精度検証を行い、その結果を示した。

データ数を追加するたびに誤差が小さくなり、推定精度の上昇が確認できた。単体に加え、累積誤差による全体評価から、数日の期間で誤差が小さくなり続け、推定精度の上昇が確認できた。一部ではデータ数が一定以上増加すると推定精度が減少するということが確認された。以上のことから、データ数が多いほど推定精度の上昇が期待できるが、多すぎると推定に悪影響を及ぼすと考えられる。

今後は誤差が大きくなる原因の究明や他の特徴抽出手法から、推定精度の向上を図る。更に、推定された SOM から正しく話題が予測できるか検証を行う。現在はクローズな環境で推定を行っているため、オープンな環境で推定を行い、リアルタイムで話題を推定できるか検証を行う。

また、現在使用しているブログ記事は yahoo ブログ記事のみであるため、他のブログ記事でどのような傾向が見られるのか、ブログ記事と twitter に加えて他の SNS を加えて傾向を確認する。

8 参考文献

参考文献

[1] 川添 恭平, 木村 義紀著 ” 消費者ニーズを発見・獲得するブログ解析技術の研究開発” 技術情報誌 第 8 号 pp.36-41 2008.6
[2] 南野 朋之, 鈴木 泰裕, 藤木 稔明, 奥村 学著 ”blog の自動収集と監視” 人工知能学会論文誌 19 巻 6 号 pp.511-520 2004
[3] Sitaram Asur,Bernardo A. Huberman ”Predicting the Future with Social Media.” IEEE Agent Technology (WI-IAT) 2010
[4] 内田誠 , 柴田尚樹著 “ブログ記事ネットワークからの emerging topic の抽出と可視化,” 人工知能学会第 20 回全 6 国大会論文集, 2006.
[5] 保住 純, 飯塚 修平, 中山 浩太郎, 高須 正和, 嶋田 絵理子, 須賀 千鶴, 西山 圭太, 松尾 豊著 ”Web マイニングを用いたコンテンツ消費トレンド予測システム” 人工知能学会論文誌 vol.29 pp.449-459 2014
[6] Kohonen T. “Self-organizing formation of topologically correct feature maps. “Biol Cybern 43:59-69, 1982.
[7] 浦野 史隆, 六井 淳 著 ”SNS 情報間の時差相関解析” 信学技報 vol.116,no.213,NLC2016-27,pp.79-84,2016-09
[8] 新谷 研, 角田 達彦, 大石 巧, 長尾 眞著 ” 単語の共起頻度と出現位置による新聞の関連記事の検索手法” 情報処理学会論文誌 vol.38 No.4 1997
[9] D. Benedetto, E. Caglioti, and V. Loreto, ”Language trees and zipping,” Physical Review Letters, vol.88, no.4 pp.28 Jan. 2002.
[10] 波多野 賢治, 佐野 綾一, 段一為, 田中 克己著” 自己組織化マップと検索エンジンを用いた Web 文書の分類ビュー機構” 情報処理学会論文誌 vol.40 No.SIG(TOD 1) pp.47-59 1999 年 2 月
[11] 西山 浩司. 遠藤 伸一, 神野 健二, 河村 明著” 自己組織化マップを利用した梅雨期特有の気象場の分類” 水工学論文集 vol.49 pp.241-246 2005 年 2 月