DEIM Forum 2015 E2-6

Twitterのトピック変遷の可視化法の提案

北田　剛士† 風間　一洋† 榊　剛史†† 鳥海不二夫†† 栗原　聡†††
篠田　孝祐††† 野田五十樹†††† 斉藤　和己†††††

† 和歌山大学 システム工学部　〒 640–8510 和歌山県和歌山市栄谷 930
†† 東京大学 工学系研究科　〒 113–6654 東京都文京区本郷 7–3–1
††† 電気通信大学 大学院 情報システム学研究科　〒 182–8585 東京都調布市調布ケ丘 1–5–1
†††† 産業技術総合研究所 サービス工学研究センター　〒 305–8568 茨城県つくば市梅園 1–1–1
††††† 静岡県立大学 経営情報学部　〒 422–8002 静岡県静岡市駿河区谷田 52–1
E-mail: †s161068@center.wakayama-u.ac.jp, ††kazama@sys.wakayama-u.ac.jp,
†††sakaki@ipr-ctr.t.u-tokyo.ac.jp, ††††tori@sys.t.u-tokyo.ac.jp, †††††kuri@is.uec.ac.jp,
††††††kosuke.shinoda@uec.ac.jp, †††††††i.noda@aist.go.jp, ††††††††k-saito@u-shizuoka-ken.ac.jp

あらまし　Twitter は現実世界の情報をリアルタイムに知るために重要な手段となりつつある反面，現在どのような話題があるのか，それらが時間とともにどのように変化しているかを把握することは難しい．例えば，潜在的ディリクレ配分法（LDA）のようなトピックモデルを用いることで，トピックを自動的に抽出することができるが，大規模なデータセットではトピック数も大きくなると共に，有用ではないと思われるトピックも多く検出されてしまう問題が存在する．本稿では，PLDA を用いて 1 日ごとに並列抽出したトピックをコサイン類似度に基づいてトピック間を関係づけてトピック系列を作成すると共に，そのトピック系列を選別・順位付けした後に，時間的変化を重視して可視化することで，ツイートアーカイブにおける話題とその変化の理解を容易にする手法を提案する．さらに，実際に東日本大震災前後の約 3 億 6 千万ツイートに適用して，有効性を示す．

キーワード　トピックモデル，LDA，トピック追跡，Twitter，東日本大震災，可視化

1. はじめに

現実社会で起こった出来事を知るための手段の中でも，Twitter はリアルタイムに膨大なユーザの意見を直接知ることができるという点で，特に重要な情報源となりつつある．ただし，Twitter では情報はフォロー関係によって構築されるソーシャルネットワーク上の近傍ユーザ間を伝播するために，タイムラインには個々のユーザに適した情報が選別されて表示される一方，Twitter 空間全体を概観して，現在どのような話題があるのかを把握することはできない．このような Twitter 空間における議論・話題の的確な把握を実現するためには，並行している異なるトピックとその内容の変遷を追跡するだけでなく，さらに異なる側面に関するトピックに分岐したり，逆に拡大したトピックが収束するなどの変化も抽出することも必要となる．例えば，2011 年に起きた東日本大震災では Twitter が活用されたが，地震や原発事故関連のトピックが含まれているだけでなく，地震関連なら被害状況や救援活動，原発事故なら事故への対処や原発の是非などの異なる話題に分岐しているのではないかと予想できる．このような話題の拡大を的確に把握できれば，特にオリンピックや災害時のような大きな出来事において起こった事件や意見を把握して，役立てることができると考えられる．

実際に生成時間に基づいて分割した文書群に潜在的ディリクレ配分法（LDA: Latent Dirichlet Allocation）を適用し，それらを類似度に基づいて連結することでトピックを追跡する研究も行われている [1]．しかし，Twitter のような大規模なデータセットの場合は，最適なトピック数も大きくなると共に，対象としていないトピックやノイズと思われるトピックなども多く検出され，内容を素早く的確に把握することが困難になる．

そこで本稿では，大規模なデータセットから LDA を使って抽出したトピック系列をわかりやすく可視化すると共に，不要なトピックを排除し，探したいトピックに容易に絞り込む手法を提案する．実際には，LDA の並列分散実装である PLDA を用いて，1 日ごとに抽出したトピック群からコサイン類似度に基づいてトピック系列を作成し，そのトピック系列を選別・順位付けした後に，内容の時間的変化を的確に理解できるように可視化する．また，トピック系列の存在期間，トピック系列に含まれる単語を条件として指定することで，目的とするトピック系列に容易に絞り込むことを可能にする．実際に東日本大震災前後の約 3 億 6 千万ツイートに適用し，有用なトピック系列や重要と思われるトピックの分岐が観測されること，絞り込み機能を活用することで目的とするトピック系列の発見が容易になることを示す．

2. 関連研究

2.1 時系列文書からのトピック抽出

ニュース記事や電子メール，Twitter のツイートのように時系列に沿って生成される文書群から，時間と共に変化する議論

や話題などのトピックを追跡する研究は数多く存在する．

まず，文書内の単語群の類似性に基づいて分類した各クラスタをトピックとみなす手法が存在する．水落らは，新聞記事を 1 日ごとに 2 種類の閾値で階層型クラスタリングを行うことで，トピックと複数がトピックが含まれるトピックグループを抽出し，さらに連続する日の関連度が高いトピックを関連づけることで，トピックの時系列を抽出した [2]．菊池らは，電子番組ガイド（EPG）に凝集型の階層型クラスタリングを適用してトピックに分類し，そのトピックの内容を把握するためのキーワードを C-value 法を用いて抽出した [3]．平田らは，ウィンドウと呼ぶ時間的な区間を移動させるスライディングウィンドウ方式を用いて各ウィンドウに含まれるニュース記事に階層型クラスタリングを行ってサブトピックを抽出し，さらに leader-follower 法を用いて異なるウィンドウのサブトピックを統合することでトピックを抽出した [4]．ただし，このようなクラスタリングを用いた手法で得られるのは文書集合であり，その具体的な内容を知るためには，文書の要約や重要なキーワードの提示が必要となるが，Twitter では題名が付与されず，文書長が短いために要約やキーワードの自動抽出も困難である．

また，ある単語が短期間に沢山発言されたような現象をバーストと呼ぶが，バーストした単語をトピックとみなす手法が存在する．例えば，Twitter では，バーストしている単語やハッシュタグをトレンドとして抽出し，さらに位置情報やフォローしているユーザなどの情報を元にカスタマイズする機能を提供している (注1)．Swan らは，ニュース記事から χ^2 検定を用いて，バーストした単語とその期間を抽出した [5]．Kleinberg は，重み付きオートマトンを用いて，電子メールのストリーム中の連続的な時系列イベントと単位時間ごとに発生したイベントに対する離散的な時系列イベントのバーストを検知する 2 種類の手法を提案した [6]．ただし，これらの手法では一つの単語が抽出されるだけで，トピックの詳しい内容や変化はわからない．

これらの手法に対して，データに隠された潜在的なトピックを確率的に推定するトピックモデルを用いて，同一の話題に関連している単語集合をトピックとみなす手法が存在する．この手法ではトピックが複数の単語で表現されることから，そのままでもトピックの変遷などが理解しやすい利点を持つ．代表的なトピックモデルは Blei らが提案した LDA [7] である．ただし，LDA はそのままでは時系列データを扱えないので，Brei らは時系列文書集合中のトピックを追跡できるように拡張した DTM（Dynamic Topic Model）を提案した [8]．例えば，高橋らは，ニュース記事を対象として，DTM を用いて推定したトピックに Kleinberg のバースト検出法を適用することで，トピック単位のバーストを検出した [9]．ただし，DTM はトピックの盛り上がりや衰退，トピック内の単語の変遷には対応しているものの，常に単一のトピックが継続しているものとして追跡するため，トピックが時間の変化と共に異なる側面を持つ複数のトピックへ分岐したり，逆に拡大した複数のトピックが収束するというような現象を把握することはできない．

(注1)：https://blog.twitter.com/2010/trend-or-not-trend

芹沢らは，ニュース記事を 1 日ごとに分割して LDA で抽出したトピックを，さらに連続する日の間で関連づけることで，トピックを追跡する手法を提案した [1]．藤田らは，Streaming API を使って収集したツイートを LDA で解析する際のトピック数を相関係数検定で決定し，さらに隣接する日付で強い相関がある場合のトピックの系列をグラフで図示した [10]．本稿でも同様なアプローチを取るが，トピック間の類似度を term-score [11] の上位 n 件の単語に限定することで，トピックの関連判定に有用でないと思われる単語の影響を除外した．なお，各区間ごとのトピック数の自動調整には対応していない．

2.2 時系列情報の可視化

時系列情報の可視化についても，さまざまな研究が存在する．

例えば，Swan らは，ニュース記事でバーストした単語とその期間を棒グラフを使って可視化した [5]．Leskove らは，ブログから抽出したミーム（meme）を波形状に表示することで，その流行り廃りを可視化した [12]，Brei らは，Science の記事から DTM で抽出したトピックの系列と単語群を可視化した [8]．芹沢らは，抽出したトピックの関係の表示をおこなった [1]．ただし，Swan らや Leskove ら，Brei らの手法は，前述したような分岐や統合のないトピックの可視化である．また，芹澤らは 3 日間・86 件のニュース記事中のトピックの関係を抽出しただけであり，トピックの内容や変遷の表示は目的としておらず，また Twitter のような大規模かつ統制されていない文章群を対象とした場合に想定される重要なトピックの優先的な提示や，ノイズと思われるトピックの除去，注目するトピックへの絞り込みなどは考慮していない．

豊田らは，ハイパーリンク構造解析を用いて抽出した Web コミュニティの時系列的変化を可視化すると共に，ビューアを用いて与えられたキーワードや URL によるコミュニティの検索など，柔軟に表示できるようなコミュニティチャートを作成する手法を提案した [13]．これはトピック追跡の研究ではないが，本稿では同様な柔軟なインタフェースを実現すると共に，トピックの時系列的変化を重視したうえで，トピックやトピック系列が持つ意味や内容が容易に理解できるように可視化する．

3. トピック系列の抽出

本稿では，トピックモデルで抽出されたトピックが時系列順に繋がったトピックグラフをトピック系列と呼ぶ．以下でトピック系列の抽出法について述べる．

3.1 トピックの抽出

まず，時系列文書から単語を抽出して BoW（Bag of Words）表現に変換した後に，一定の時間間隔で複数の集合に分割し，LDA を用いて各期間内の文書からトピックを抽出する．

LDA は文書の生成過程を確率的にモデル化したトピックモデルのひとつであり，一つの文書中に複数のトピックが混合されていると仮定して，単語ごとに潜在的なトピックを決定する．LDA では，文書のトピック分布を確率変数とみて生成するために，単語やトピックの多項分布に対する共役事前分布であるディリクレ分布を使用する．LDA の生成モデルをグラフィカルモデルで描くと，図 1 のようになる．このとき，θ は文書に

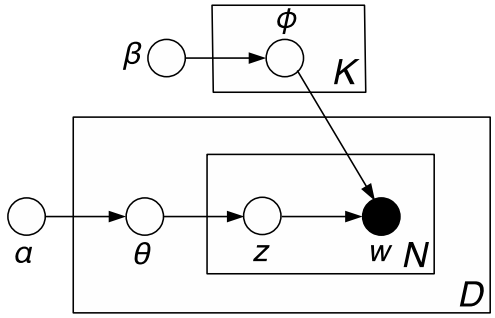


図 1　LDA のグラフィカルモデル

おけるトピックの多項分布，ϕ はトピックにおける単語の多項分布を表し，z は語 w に割り当てられたトピックを表す．また，K はトピック数，D は文書数，N は単語数である．α はトピック分布 θ が従うディリクレ分布のハイパーパラメータ，β は単語分布 ϕ が従うディリクレ分布のハイパーパラメータを表す．各文書はそれぞれトピック分布 θ を持ち，文書中の各単語について，トピック分布 θ に従ってトピック z が選択され，トピック z に対応する単語分布 ϕ に従って，単語 w が生成される．なお，同一の単語であっても，文章中の単語 w の出現位置によって推定されるトピックは異なる．観測変数は w だけであり，他は潜在変数か未知のパラメータである．

つまり，LDA では，文書は以下のように生成される．$Multi(\cdots)$ は多項分布，$Dir(\cdots)$ はディリクレ分布である．

（1） 各トピック $k = 1, \cdots, K$ について
　（a） 単語分布 ϕ_k を生成: $\phi_k \sim Dir(\beta)$
（2） 各文書 $d = 1, \cdots, D$ について
　（a） トピック分布 θ_d を生成: $\theta_d \sim Dir(\alpha)$
　（b） 各単語 $n = 1, \cdots, N_d$ について
　　i. トピックを生成: $z_{dn} \sim Multi(\theta_d)$
　　ii. 単語を生成: $w_{dn} \sim Multi(\phi_{z_{dn}})$

ここで，ϕ_k はトピック k の単語分布，θ_d は文書 d のトピック分布，z_{dn} は文書 d の n 番目の単語の潜在トピック，w_{dn} は文書 d の n 番目の単語を表す．

この結果，各期間ごとに複数のトピックが抽出される．

3.2　単語の term-score の計算

LDA では，トピックごとに各単語がそのトピックから生成された出現頻度を出力として得ることができる．トピック中で重要な単語を抽出するために，この出現頻度をスコアとみなして単語をランキングすると，どの文書にも多く出現する単語が各トピックのランキング上位にくる傾向がある．しかし，このような単語は一般的であり，トピックの内容を的確に表しているとは言えないことが多い．

この問題を解消するために，単純な出現回数ではなく，term-score [11] と呼ばれる指標を単語のスコアとして用いる．term-score では，多くのトピックに出現する単語のスコアは小さくなり，特定のトピックにのみ出現するようなトピック固有の単語のスコアは大きくなる．term-score は次式で表される．

$$term\text{-}score_{k,v} = \hat{\beta}_{k,v} \log \frac{\hat{\beta}_{k,v}}{(\prod_{j=1}^{K} \hat{\beta}_{j,v})^{\frac{1}{K}}} \tag{1}$$

$\hat{\beta}_{k,v}$ はトピック k における単語 v の出現確率，K はトピック数である．term-score は情報検索における TF-IDF と同様の考えに基づいており，$\hat{\beta}_{k,v}$ が TF に，$log(\cdots)$ が IDF に相当する．

$\hat{\beta}_{k,v}$ は，次式によりトピックの単語分布の推定量として求めることができる．

$$\Phi_{kw} = \frac{n_{kw} + \beta}{n_{k-} + V\beta} \tag{2}$$

n_{kw} はトピック k における単語 w の出現確率，n_{k-} はトピック k における単語の出現確率の総和，β はハイパーパラメータ，V は全文書における語彙数を示す．

3.3　トピックの連続性の判定

まず，トピックごとに，その内容を表す単語と term-score のスコアから成るベクトルを作成する．ただし，本稿では term-score でランキングした時の上位 N 件の単語だけを対象とする．この理由は，トピックの内容との直接的な関係が薄い一般的な単語や，ほとんど使われていない膨大な単語の影響を除去するためであり，これによりトピック間の類似性の有無がより明確に判断できるようになる．

次に，隣接する期間のトピックの間の連続性の有無を，ベクトルの類似性に基づいて判定する．例えば，期間 t と期間 $t+1$ のトピックのベクトルをそれぞれ $\boldsymbol{T_i^t}(1 \leqq i \leqq K)$, $\boldsymbol{T_j^{t+1}}(1 \leqq j \leqq K)$ とした時に，この二つのベクトル間のコサイン類似度 $csim(\boldsymbol{T_i^t}, \boldsymbol{T_j^{t+1}})$ は次式で求められる．

$$csim(\boldsymbol{T_i^t}, \boldsymbol{T_j^{t+1}}) = \frac{\boldsymbol{T_i^t} \cdot \boldsymbol{T_j^{t+1}}}{|\boldsymbol{T_i^t}||\boldsymbol{T_j^{t+1}}|} \tag{3}$$

この値はベクトルのなす角度のコサイン値であるので，$0 \leqq csim(\boldsymbol{T_i^t}, \boldsymbol{T_j^{t+1}}) \leqq 1$ となり，値が 1 に近づくほど二つのベクトルが類似している．そこで，隣接する期間の $\boldsymbol{T_i^t}$ と $\boldsymbol{T_j^{t+1}}$ のすべての組み合わせに対してコサイン類似度を計算し，閾値 S を超える場合に限りトピックが連続していると判定する．

3.4　トピック系列の作成

期間を t_s から t_e，トピックをノード，トピック間の関係をエッジとして，次のようにトピック系列を得る．

（1） 期間 $t = t_s$ とする．
（2） 期間 t と $t+1$ の間の各エッジ e_{ij} の始点ノード n_i を終端ノードとして含むトピック系列があれば，その区間 $[t, t+1]$ にノード e_{ij} を追加する．なければ，新しいトピック系列を生成してノード e_{ij} を追加する．
（3） 各エッジ e_{ej} の終点ノード n_j を含むトピック系列が複数あればマージする．
（4） 期間 $t = t+1$ とし，$t = t_e$ なら終了し，そうでなければ (2) に戻る．

この結果，各区間 $[t, t+1]$ ごとのエッジリストを含むトピック系列が作成できる．

4. トピック系列の可視化

4.1 可視化手法

本稿では，小規模で文体や用語が注意深く選択されているようなニュース記事ではなく，膨大で文章がまったく統制されていない Twitter のツイートのようなデータセットを扱うことを想定しているために，抽出されるトピック数，そしてそれから生成されるトピック系列数は大幅に増大すると考えられる．例えば，渡邊らは $K = 70$，藤野らは $K = 50$ というトピック数を採用した [10], [14]．また，オリンピックや震災などの世の中全体を巻き込む大きな動きがある期間には，適切なトピック数がさらに増大すると考えられる．さらに，トピックの分岐や統合なども扱うことを考えると，画面に収まりきらないことは容易に想像できる．

そこで，以下のように，抽出されたトピック系列に対して，フィルタリングとランキングを施した後で，ユーザの意図に合ったトピック系列を優先して表示したり，ユーザが見る視点を対話的に変更することで Twitter のトピック空間を探索できるような機能を提供する．

（1） トピック系列のフィルタリング
（2） トピック系列のランキング
（3） トピック系列の描画

以下で詳しい手順について述べる．

4.2 トピック系列のフィルタリング

まず，以下のような視点に基づいて，ユーザの意図に合わない不必要なトピック系列を除外する．例えば，不必要なトピック系列とは，LDA のような確率に基づいた教師なし学習を用いた時に生じがちな，ユーザがその意味を理解することができないようなトピック系列や，内容があまりに一般的なトピック系列，挨拶など特定のトピックを意味しない日常的な行動を表すトピック系列である．

- 対象期間…オリンピックなどの特定のイベントなどを手掛かりに，その直前又は直後のトピックだけに絞り込むことができる．
- コサイン類似度の閾値…類似性の判定条件を強化あるいは緩和することで，抽出結果を調節できる．
- トピック系列の継続期間…一般に，継続期間が長いほど一般的又は日常的な行動のトピックであることが多く，逆に継続期間が非常に短い場合は短期的なトピックであることが多い．そこで，継続期間を手掛かりに閲覧したいトピック系列の特性を選択できる．
- トピックの特徴語…「地震」など，閲覧したいトピック系列のトピックを表す単語を複数指定できる．
- 除外対象とするトピック系列…条件指定の組み合わせで対処できない場合には，ユーザが明示的に除外するトピック系列を指定できる．

4.3 トピック系列のランキング

閲覧直後やうまくフィルタリングできなかった時は，描画対象となるトピック系列が膨大になり，画面内に描画しきれないことがある．また，ユーザにとって重要なトピック系列ほど上位に配置した方が理解しやすいと考えられる．

そこで，以下のような指標でトピック系列をランキングし，ランキング上位から画面に収まる順位まで描画する．

- トピック系列サイズ…長期間存続するトピック系列ほど多くのユーザに注目されているという観点に基づく．
- エッジ数…トピックの分岐・収束が多いほど議論が盛り上がっているという観点に基づく

4.4 トピック系列の描画

最後に，ランキング上位から，トピック系列を上から下に積み重ねるように順に描画する．描画には，HTML5(注2) の Canvas 要素と JavaScript を用いた．Canvas 要素は，動的な 2 次元ビットマップ画像のための HTML 要素であり，JavaScript と併用することで，GIF や JPEG のような静的な画像を用意せずに，自由度の高い描画が可能である．

なお，時系列変化の理解を容易にするために，次のことを行った．

- 先頭トピックの強調表示…画面内に多くのトピック系列を表示するために積み重ね方式を採用したが，そのために生じた画面下部のトピック系列ほど理解しにくくなるという問題を解決するために，トピック系列の先頭トピックをピンク色で強調表示した．
- 新出語の強調表示…トピック系列の内容が徐々に変わっていくことと，それが変更されるタイミングを示すために，そのトピック系列でトピックの単語の term-score の上位 N 件に新たに現れた時点でピンク色で強調表示した．
- ピークの強調表示…トピック系列において，いつの時点が話題のピークであるかを示すために，そのトピック系列でトピックの単語の出現頻度の上位 N 件の総和が最も高いトピックを二重線で強調表示した．
- 特徴語の強調表示…トピックの特徴語を指定するフィルタリング機能と合わせて，指定した特徴語を太字で強調表示した．

5. 実　　験

5.1 データセット

Twitter API(注3) を用いて，3 月 5 日〜24 日の間に 200 件以上日本語でツイートしたアクティブなユーザのツイートを収集し，後日収集漏れを減らすために各ユーザに対して再収集して，それをデータセットとした．200 件は，Twitter API の呼び出し 1 回で取得できる最大ツイート数である．

データセットの規模は，ツイート数が 362,435,649 件，ユーザ数が 2,711,473 人である．データセットには，ツイート ID（64 ビット整数），ツイートしたユーザのスクリーン名，本文，

(注2)：http://www.html5.jp
(注3)：http://apiwiki.twitter.com/

ツイート元．ツイート時間，リプライ先のツイート ID，リプライ先のスクリーン名などの情報が含まれる．本稿では，ツイートの本文からキーワードを抽出して用いた．

5.2 単語の抽出

LDA は文書を BoW として扱うので，Mecab [15] で日本語形態素解析して得られる単語（形態素）から BoW を作成した．

ただし，LDA では潜在的なトピックを確率的に推定するために，BoW にどのような単語を含めるかが，抽出されるトピックの質に大きく影響する．まず，抽出された形態素のうち，一般・自立・固有名詞・サ変接続・形容動詞語幹の名詞だけを対象とした．ただし，標準の IPA 辞書では複合語はうまく処理できない．例えば，名詞を連結して自動的に複合語とする手法もあるが，ある名詞を含むさまざまなバリエーションができてしまい，確率的な手法を用いた場合の結果の質が下がることがある．そこで，新語や流行語，専門用語も正しく解析でき，妥当な複合語が抽出されるように，はてなキーワード (注4) や原子力百科事典 ATOMICA(注5) の用語を辞書に追加して用いた．また，一般的にツイートは口語で校正せずに書かれるためにうまく形態素解析できないだけでなく，Twitter 上で流行している独特の言い回しなどがあり，それらがノイズの原因になることが多い．そこで，以下のような単語をストップワードとした．

- 記号のみで構成されている単語
- アルファベットのみで構成されている単語
- ひらがなのみで構成された一般名詞（固有名詞などは含まない）
- 長音符（ー）や「笑」などの 1 文字の単語

さらに，ツイートには,「おはよう」のような挨拶の交換や「地震だ！」,「バルス」などの現実世界のイベントに対する叫び声など，文章としての意味は持たない短いツイートもかなり含まれている．これらは LDA のトピック抽出結果の質を下げてしまうことから，例えばエントロピーの低いツイートを除去したり，ユーザ単位でツイートをまとめて 1 文書にするなどの手法がおこなわれている．本稿では，しっかり意味を持つ文章が記述されているツイートのみを取り出すために，8 語以上抽出されたツイートのみを対象とした．

5.3 トピック系列の抽出

現実の巨大なツイートデータを対象にできるように，本稿では LDA を並列分散化した PLDA [16] を用いた．

PLDA におけるトピック抽出のパラメータは，トピック数 K を 10 から 50 まで変化させて実行し，それぞれ $\alpha = 50/K$，$\beta = 0.01$，イテレーション回数は 500 回とした．また，トピック間の類似度を比較する際の単語ベクトルを，トピックの単語の term-score 上位 $N = 10$ 件とし，閾値 $S = 0.7$ とした．単語ベクトルを上位 10 件として比較しているため，閾値を低く設定してしまうと，いくつかの単語が重なるだけで異なる内容のトピック同士を関係付けてしまう恐れがあるため，高めの閾値を設定した．

(注4)：http://d.hatena.ne.jp/keyword/

(注5)：http://www.rist.or.jp/atomica/

表 1　使用機器環境

OS	Cent0S 6.6 (64bit)
CPU	Intel Core i7 3930K (3.20Ghz/12MB/6C/12T)
メモリ	64GB DDR3-1600

表 2　PLDA における処理時間

コア数	トピック数	処理時間
1	10	24m06s
1	50	57m57s
6	10	06m30s
6	50	14m28s

表 3　描画されるトピック数

トピック数	トピック系列数	トピック総数	分岐数	収束数
10	22	79	0	0
20	44	145	0	0
30	70	244	5	3
40	85	346	11	12
50	113	461	14	9

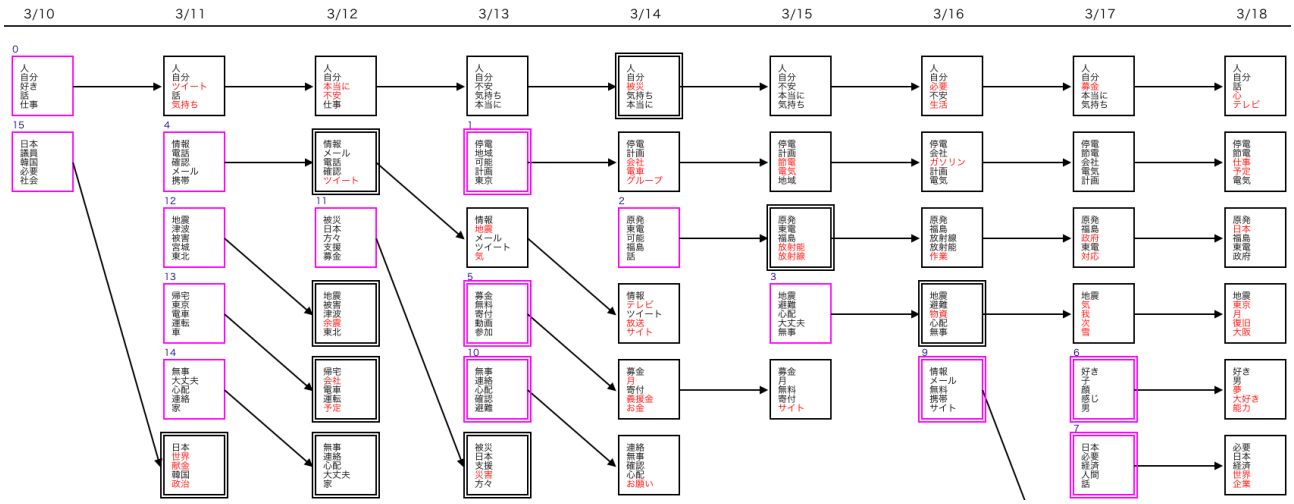
表 1 に示す環境で，2011 年 3 月 11 日の 1 日分の 29,710,098 件のツイートデータから PLDA を使ってトピック抽出をした時の処理時間を表 2 に示す．一般に，LDA ではデータサイズやトピック数 K，イテレーション回数に応じて処理時間が増加するが，コア数 1 とコア数 6 の場合の実行時間を比較すると，$K = 10$ で 0.27 倍，$K = 50$ で 0.25 倍になっており，並列化により実行時間が大幅に減少していることが分かる．

5.4 可視化結果の分析

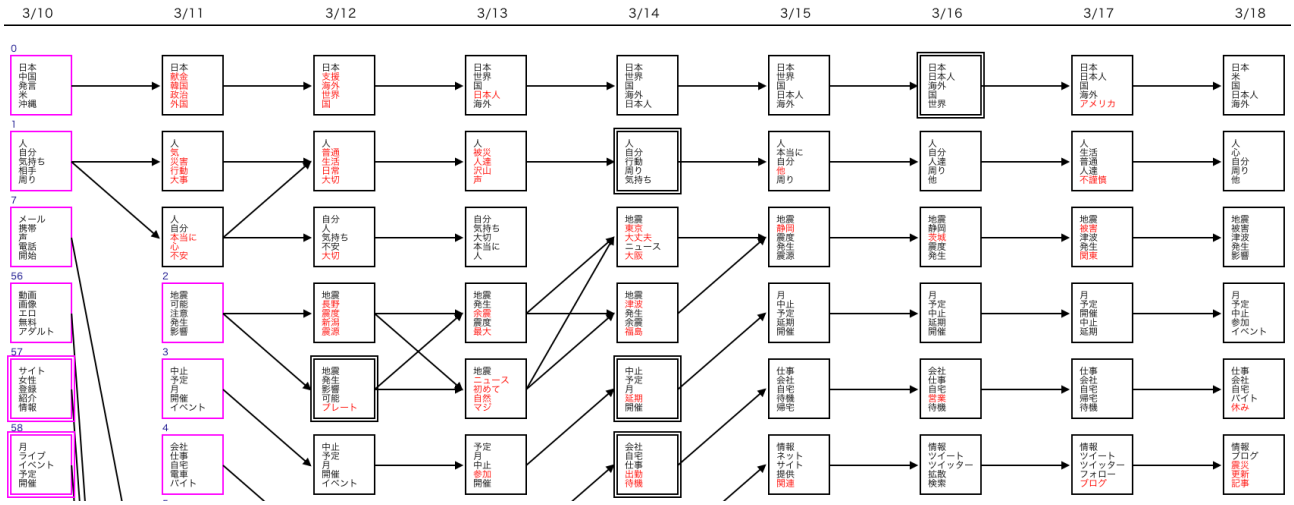
東日本大震災による Twitter への影響を調べるため，対象期間を 3 月 10 日から 3 月 18 日，トピック数を $K = 10$ と $K = 50$ で可視化した結果を，それぞれ図 2 に示す．一つの矩形が一つのトピックを表し，矩形内部に term-score の上位 5 単語を表示する．エッジで結ばれた一連のトピックがトピック系列であり，縦方向に同一日付のトピックを配置し，左から右に時系列が進行するように描画する．

図 2 から，どちらのトピック数の場合も単語群が類似したトピックが系列化されていると共に，その単語群も変化しており，各トピック系列の時系列的な内容変化を追跡できることが分かる．しかし，$K = 10$ の図 2(a) では，目的としていたトピックの分岐や収束が見られない．この原因として，複数の側面を持つようなトピックが一つにまとめられて抽出されていることが考えられる．これに対して，$K = 50$ の図 2(a) では，複数のトピックの分岐や収束が確認できた．

さらに，使用するデータセットの期間を 3 月 7 日から 3 月 23 日とし，PLDA により抽出するトピック数 K を 10 から 50 まで変化させたときに作成されるトピック系列数，トピック総数，分岐・収束数を表 3 に示し，継続期間当たりのトピック系列数の分布を図 3 に示す．トピック総数は，全トピック系列におけるトピックの合計である． これらの結果から，LDA の抽出トピック数 K が増大すると，抽出されるトピック系列数が



(a) トピック数 $K = 10$ の場合



(b) トピック数 $K = 50$ の場合（一部）

図 2　東日本大震災直後のトピック系列の可視化結果

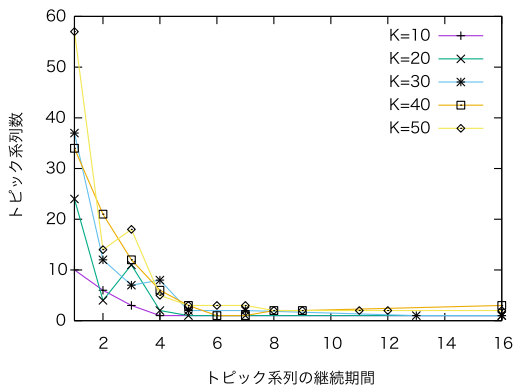


図 3　トピック系列の継続期間の分布

それ以上に増大し，同時にトピック系列のグラフ構造が複雑になることがわかる．

今回は $K = 50$ で実行したが，既存研究と比べてデータセットサイズが大きいことを考えると，適切なトピック数はより大きくなり，それに伴いトピック系列はさらに複雑になると考えられることから，本稿で提案するような柔軟なユーザインタフェースが必要になる．

5.5　フィルタリング・ランキングの分析

さらに，本稿で提案するフィルタリング・ランキングが，抽出トピック数 K が大きい場合のトピック系列の閲覧に有効かどうかを評価した．期間を 3 月 7 日から 23 日，抽出トピック数を $K = 50$ とし，「地震」，「原発」という 2 つの単語のどちらかを含むトピック系列のみに絞り込んだ結果を，図 4 に示す．抽出されたトピック系列数は 7，トピック総数は 39 であった．表 3 の $K = 50$ の結果と比較すると，9 割以上のトピックが「地震」，「原発」とは直接関係がないトピックだった．

さらに，得られたトピック系列を，特に原発関係に着目して，Wikipedia の「福島第一原子力発電所事故の経緯」に関する記述 (注6) と照合した．まず，トピック系列の期間に着目すると，地震関連トピック系列の開始は東日本大震災発生の 11 日と合致しているものの，原発関連トピック系列の開始は 12 日と遅延があった．次にトピック系列 1 の内容に注目すると，12 日のトピックには 11 日 20 時 50 分の避難指示を示すと思われる

(注6)：http://ja.wikipedia.org/wiki/福島第一原子力発電所事故の経緯

「避難」，13 日のトピックには 12 日 15 時 36 分の 1 号機の水素爆発を表すと思われる「爆発」，18 日には 17 日 19 時 35 分から開始した自衛隊による大型破壊機救難消防車による注水を表すと思われる「自衛隊」が抽出された．若干の遅れはあるが，原発事故の状況変化に応じた単語が抽出されていると思われる．さらに，トピック系列 1 の発言のピークは 15 日であり，これはマスコミで映像が報道された 14 日 11 時 1 分の 3 号機爆発に対応していると思われる．なお遅延は，次の 2 種類の遅延が加算されていると推測している．1 番目は，現場の状況を直接知ることができないために，東京電力や官邸が事実を確認してから記者会見やプレスリリースで発表された後でないとわからないことによる情報伝達遅延である．2 番目は，8 語以上抽出されたツイートのみを対象にするために，「爆発した！」というようなリアルタイムに行われる短いツイートは除外され，その後にさらに情報を抽出した上に行われた議論に相当する長いツイートのみが選ばれたからである．なお，本稿では現実世界のイベント検出ではなく，Twitter 上の議論や話題の変遷の観測を対象とするために，これらの遅延に関しては問題はない．

図 4 では，地震と原発と直接関係あるトピック系列に絞り込んだが，大地震は避難，救助，停電，物資の共有など，様々な問題を引き起こすが，それらは観測できなかった．そこで，そのような間接的に関連があるトピック系列を閲覧するために，単語ではなくトピック系列の期間で絞り込んだ．期間を 3 月 7 日から 23 日，抽出トピック数を $K = 50$，トピック系列の期間を 7 日〜13 日と指定した結果を，図 5 に示す．短期間で消滅する一時的な盛り上がりを排除するために最小継続期間を 7 日．定常的に議論されている一般的な話題を除外するために最大継続期間を地震発生後からデータセットの最終日までの 13 日間とした．描画されたトピック系列数は 14，トピック総数は 145 であった．この結果，トピック系列 1 のような仕事やイベントなどの地震による影響についての議論，トピック系列 2 のような支援活動についての議論，トピック系列 6 のような輪番停電に関する議論なども抽出できた．

以上の結果から，本システムを用いることで Twitter の話題の変遷が観測できること，さらに絞り込み手段を変更することで，注目した実世界のイベントに直接関係する話題だけを観測したり，逆にそれから派生するような話題まで広範囲に観測するなど，柔軟な閲覧ができることがわかる．

6. おわりに

本稿では，PLDA を用いてツイートデータからトピックを抽出し，コサイン類似度に基づいて生成されたトピック系列を Web ブラウザ上に可視化した．ツイートデータのような大規模なデータにおいて，トピック数 K は大きな値をとることが予想されるが，大きなトピック数で抽出されたトピック系列においては，抽出数が増加するだけでなく，複雑な構造をもつ．そこで，ランキングやフィルタリング機能により柔軟な絞り込みを行うことで，必要とする情報や重要な情報を追跡するような可視化法を提案した．

実際に，東日本大震災前後のツイートデータを用いて「地震」,「原発」という単語で実際に絞り込みを行った結果，それらの単語と直接関係のあるトピック系列のみを抽出することができた．また，トピック系列を期間によって絞り込むことで，「避難」,「停電」などの大地震によって間接的に引き起こされるトピック系列も合わせて抽出することができた．

トピックの変遷の追跡に関しては，原発関連トピック系列に着目してその有効性を示した．その際，トピックの内容に若干の遅延が確認されたが，これに関しては次の 2 種類の遅延が加算されたためと推測した．原発事故のように直接ユーザが情報を得ることができず，マスメディア等の情報公開を待つために起こる遅延と，リアルタイムにつぶやかれるような短いツイートを除外したため，その後にさらに情報を抽出した上に行われた議論に相当する長いツイートのみが選ばれたことによる遅延である．本稿の目的は現実世界のイベント検出ではなく，Twitter 上の議論や話題の変遷の観測であるために，これらの遅延に関しては問題はなく，むしろ意味のある重要なトピックを選別し抽出できたといえる．

今後の課題は，以下の通りである．一つは最適トピック数 K の決定である．今回は $K = 50$ としたが，議論や話題の変遷を詳細に観測するために最適な K はより大きいと思われるだけではなく，日によって最適なトピック数は異なるのではないかと推測される．これに関しては既存研究 [1], [10] を含めて，本稿の目的に合致し，かつ大規模データセット時の処理時間の増加が許容できる手法を検討する予定である．もう一つは，可視化の改良である．表示されていない下位の単語の確認機能やトピック系列のよりわかりやすい描画法に加えて，単語ではなく具体的な議論内容を知るためのトピック系列中の各トピックに対応したツイートの表示や，議論の盛り上がりの変化を知るためのトピック系列中のバースト検出と表示などを予定している．

謝　辞

本研究を行なうにあたり，ツイートデータの収集に協力していただいたクックパッド株式会社の兼山元太氏に感謝する．また，本研究は JSPS 科研費 24300064，26330345 の助成を受けた．

文　献

[1] 芹澤翠, 小林一郎. 文書内のトピック数を考慮したトピック追跡の試み. 第 18 回年次大会発表論文集, pp. 1196–1199. 言語処理学会, 2012.

[2] 水落大史, 井上悦子, 吉廣卓哉, 村川猛彦, 中川優. 新聞記事集合に対する時系列のトピック抽出. *DEIM Forum 2010 D6-3*, 2010.

[3] 菊池匡晃, 岡本昌之, 山崎智弘. 階層型クラスタリングを用いた時系列テキスト集合からの話題推移抽出. 日本データベース学会論文誌, Vol. 7, No. 1, pp. 85–90, 2008.

[4] 平田紀史, 児玉政幸, 伊藤正都, 大圃忠親, 新谷虎松. ニュース記事閲覧のための複数ウィンドウ方式を用いた特定トピック追跡システムの試作. 第 70 回全国大会講演論文集, pp. “1–633”–“1–634”. 情報処理学会, 2008.

[5] Russell Swan and James Allan. Automatic generation of overview timelines. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 49–56, 2000.

[6] Jon Kleinberg. Bursty and hierarchical structure in streams. *Proceedings of the 8th ACM SIGKDD international confer-*

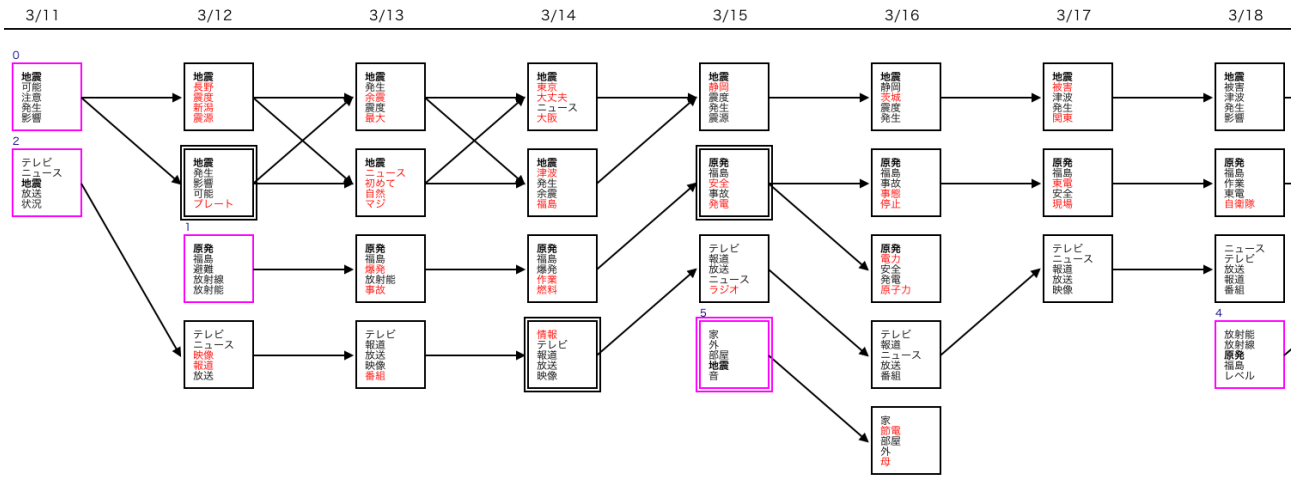


図 4　地震，原発という単語での絞り込み結果（一部）

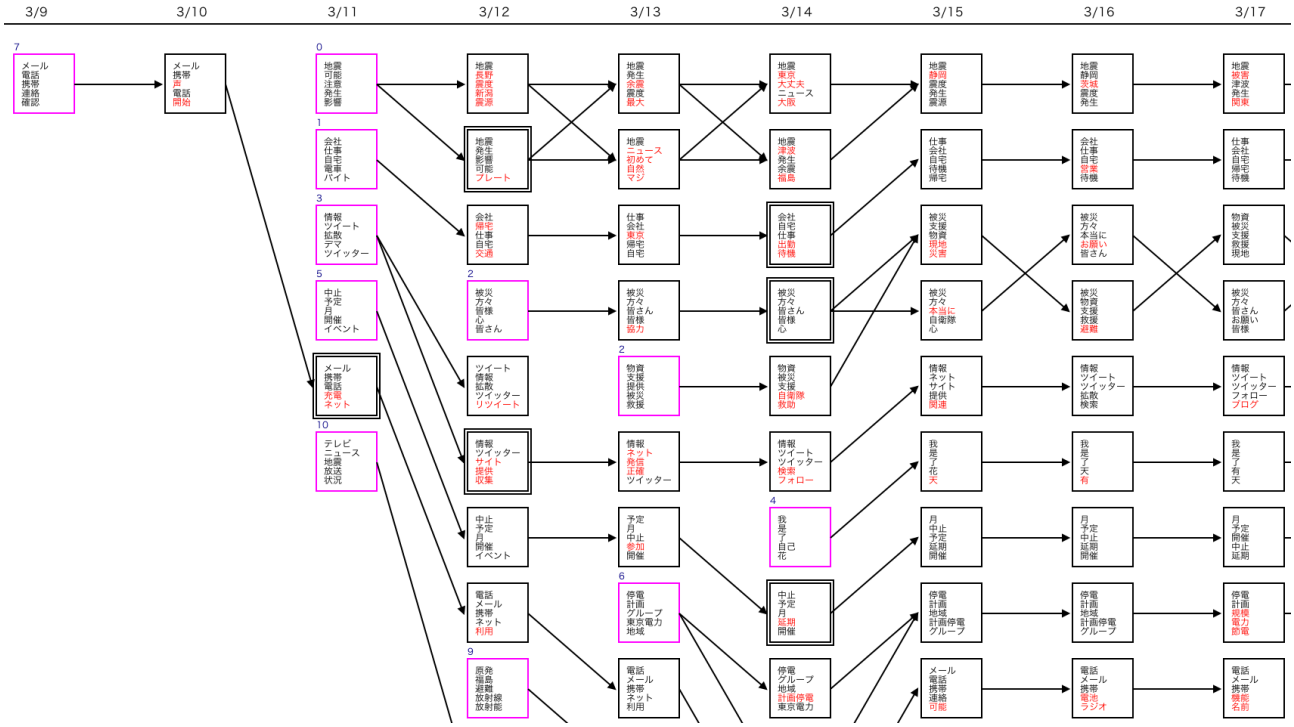


図 5　トピック系列の継続期間による絞り込み結果（一部）

ence on Knowledge discovery and data mining, pp. 91–101, 2002.

[7] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. In John Lafferty, editor, *Journal of Machine Learning Research*, Vol. 3, pp. 993–1022, 2003.

[8] David M. Blei and John D. Lafferty. Dynamic topic model. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 113–120, 2006.

[9] 高橋佑介, 横本大輔, 宇津呂武仁, 吉岡真治, 河田容英, 神門典子, 福原知宏, 中川裕志, 清田陽司. 時系列トピックモデルにおけるバーストの同定. *DEIM Forum 2012 F5-5*, 2012.

[10] 藤野巌, 星野祐子. Twitter におけるトピックの同定手法の提案とそれを用いたトピックの変遷解析. *DEIM 2014 C4-2*, 2014.

[11] David M.Blei and John D. Lafferty. *Text Mining: Theory and Applications*, chapter TOPIC MODELS. Taylor and Francis, 2009.

[12] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 497–506, 2009.

[13] 豊田正史, 喜連川優. 日本のウェブアーカイブにおけるウェブコミュニティ発展過程の詳細分析. *DEWS 2003*, 2003.

[14] 渡邊恵太, 加藤昇平. ユーザ興味を反映した情報推薦のための潜在的ディリクレ配分法を用いた協調フィルタリング. 信学技報, Vol. 113, No. 429, pp. 15–20, 2014.

[15] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)*, pp. 230–237, 2004.

[16] Yi Wang, Hongjie Bai, Matt Stanton, Wen-Yen Chen, and Edward Y. Chang. PLDA: Parallel latent dirichlet allocation for large-scale applications. In *Proceedings of the 5th International Conference on Algorithmic Aspects in Information and Management*, pp. 301–314, 2009.