時系列ニュースの話題性の可視化

中山　　豪†　　細部　博史††

† 法政大学情報科学研究科　〒 184–8584 東京都小金井市梶野町 3 丁目 7 番 2 号
†† 法政大学情報科学部　〒 184–8584 東京都小金井市梶野町 3 丁目 7 番 2 号
E-mail: †go.nakayama.5v@stu.hosei.ac.jp, ††hosobe@hosei.ac.jp

あらまし　数多くのオンラインニュースの中から本当に読みたい記事を見つけることは難しい．本研究では，ニュースの話題性を可視化することによってこの問題を解決する手法を提案する．本研究で開発したシステムでは，動的トピックモデルとバースト解析を用いてニュースの話題性を算出し，ThemeRiver と動的検索を組み合わせた可視化結果を示す．ThemeRiver は時系列データを可視化する手法だが，トピックが多くなればなるほど離れたトピック同士の比較が難しくなる．提案手法の特徴は，ユーザが動的な可視化結果を見てトピックを比較できるようにするために，スライダを用いた動的検索をおこなえる点である．本研究では，提案手法により得られる可視化結果を例示するとともに，ユーザは話題性の高いトピックを視覚的に把握できることを実験によって示す．

キーワード　ニュース，トピックモデル，ThemeRiver，バースト解析

1. はじめに

近年，新聞記事や雑誌の電子化がますます進む状況において，新聞社などが Web 上で配信する数多くの記事を PC やスマートフォンで閲覧できるようになった．無料で読めるニュースが多いことは，ニュースを読む人にとってメリットである．一方でこの現状は情報過多であると言え，ニュースを選ぶ手間や時間がかかり，読者の負担が増えているととらえることもできる．本研究は，この負担を減らすことを目的としている．

先行研究で開発されたシステムの 1 つとして T-Scroll [1] が挙げられる．詳細については関連研究の節で説明するが，有用ではあるが使いやすさや分かりやすさに欠けるインタフェースであるという実験結果が出ている．したがって本研究では，ユーザにとって分かりやすい可視化結果が得られることに重点を置く．

本研究の提案手法は，動的トピックモデル [2] とバースト解析 [3] を組み合わせてニュースを解析し，ThemeRiver [4] と動的検索 (dynamic query) [5] を用いて可視化するというものである．ThemeRiver は要素（本研究ではトピック）の時間軸に沿った変化を示すのに適しているが，離れた要素の比較は難しい．動的検索を用いて可視化結果を動的に変化させることでこの問題を解決する．ニュースの解析において，この可視化の方法を提案することが本研究における新規性である．また，ユーザが必要とする情報を与えることができるという有用性や，ニュース読者の負担を減らすという目的の達成に繋がる．

2. 関連研究

2.1 T-Scroll

ニュースの話題性の可視化システム T-Scroll [1] では，トピックを代表するキーワードをノードとし，キーワード同士の関係性の深さをエッジとして表現している．なお本研究とは違い，クラスタリングによってトピックを分類している．ノードの大きさはトピックの規模，色はクラスタの質を表している．トピックは時系列に並べられており，左から右へ見ていくことで，ユーザは時系列文書集合のトレンドやトピックの詳細を把握することができる．評価実験の結果，インタフェースの使いやすさや分かりやすさの改善が必要とされた．

2.2 ThemeRiver

ThemeRiver [4] は，分類された Theme（本研究ではトピックに該当）に関する情報の時間的推移を可視化するシステムである．ThemeRiver は可視化手法としては，積み上げグラフの 1 種と見なすことができる．各 Theme は情報の増減に応じて太さの変化する 1 つの流れとして表現され，Theme 全体はこのような流れの束として，左から右へ流れる 1 つの川のように表現される．これによって，データの時間経過による増減が一目で分かり，時系列データの全体像を掴むことに長けている．本研究では，このような川の流れを表すような可視化手法を取り入れたが，本来の ThemeRiver にある Theme を代表するキーワードは表示していない．また ThemeRiver では難しい，離れた Theme 同士の比較に対する解決策として，スライダによる動的検索を導入した．

2.3 動的検索

動的検索 [5] は，ユーザが検索ワードの入力などをおこなうのではなく，検索条件をスライダやボタンを使用して設定することで可視化結果を動的に変化させる手法である．Shneiderman は，動的検索では 100 ミリ秒以内で検索結果の更新をおこなうと説明している．本研究で提案するシステムでは，ユーザがスライダで検索条件を設定する．また，検索結果の更新速度は 100 ミリ秒を基準として，これを満たすような実装をおこなう．

3. 準備：ニュースの話題性の解析

3.1 動的トピックモデル

トピックモデルは，ビッグデータから有用な情報を抽出する手法として注目を集めている．本研究では，Blei らが考案

したトピックモデルの拡張の 1 つである動的トピックモデル (dynamic topic model) [2] を用いる．動的トピックモデルは文書データの時系列情報を扱うことができる．本研究でおこなうような新聞記事の解析をはじめとして，論文やブログの解析等に適している．

本研究における動的トピックモデルの役割は，取得したニュースを任意の数のトピックに分類し，トピックごとに関連性の高いキーワードを算出することである．具体的には，それぞれのニュースのタイトルに含まれるキーワードの集合を w，出力させたいトピックの数を K とすると，それぞれのトピック z_i $(i = 1, ..., K)$ の w の確率分布は $p(w|z_i)$ と推定できる．また，文書 d におけるトピック z_i の確率分布は $p(z_i|d)$ と推定できる．これら $p(w|z_i)$ と $p(z_i|d)$ の推定には Blei らが作成したライブラリを使用した．

3.2 バースト解析

バースト解析 [3] とは，Kleinberg によって考案されたバースト検知アルゴリズムを用いて，時系列データにおいて特定の情報の急激な増加を検知することができる解析手法である．ニュースを解析する場合を例に説明すると，社会的に重大な事件 A が発生した後に，事件 A に関するニュースは一定期間多くなる．この期間を含むニュース記事の集合をバースト解析にかけると，事件 A に関するトピックのバースト度が高いと算出される．このトピックの状態をバースト状態と呼ぶ．逆に，あるトピックに関する報道が普段通りの数しかなされていない場合，そのトピックは非バースト状態である．本研究ではバースト状態のトピックを話題性の高いトピックとする．

本研究では，ニュースのデータを 1 時間に 1 回という離散時間で集めるため，enumerating バーストという手法を用いる．便宜上，引き続きニュースの解析と事件 A の例を使って説明する．まず n 時間で集められるニュースデータの集合の内，ある 1 時間分の集合を B_i $(i = 1, ..., n)$ とし，B_i における記事の総数を d_i，事件 A に関係する記事の総数を r_i とする．非バースト状態を表す式は，ニュース収集期間における期待値である

$$p_0 = \frac{\sum_{i=1}^{n} r_i}{\sum_{i=1}^{n} d_i}$$

とする．またバースト状態を表す式は，p_0 に任意のパラメータ s をかけた

$$p_1 = p_0 s$$

とする．なお，このパラメータ s は $s > 1$ かつ $p_1 \leqq 1$ となる値でなければならない．パラメータ s は，バースト状態と判定されやすいかどうかを決めるものである．s が 1 に近ければ近いほど，事件 A に関する記事の割合が少なくてもバースト状態であると判定されやすいという特性を持つ．本来 Kleinberg のバースト検知アルゴリズムは，トピックではなく文書内のキーワードのバースト度 $bw(i, w)$ を算出するもので，

$$bw(i, w) = -\log\left[\binom{d_i}{r_i} p_0^{r_i}(1-p_0)^{d_i - r_i}\right] - \left(-\log\left[\binom{d_i}{r_i} p_1^{r_i}(1-p_1)^{d_i - r_i}\right]\right)$$

で求められる．これをトピックのバースト度 $bz(t, z_i)$ の算出に応用すると，

$$bz(t, z_i) = \sum_{w} bw(i, w) \cdot p(w|z_i)$$

とすることができる．

4. 提案手法：ニュースの話題性の可視化

4.1 概　　要

本研究で提案するシステムは，ニュースの解析を動的トピックモデルとバースト解析の組み合わせでおこない，解析結果の可視化には ThemeRiver と動的検索を用いる．

動的トピックモデルを用いることで，全てのニュース記事を任意の数のトピックとしてまとめ上げることができる．これらのトピックごとのバースト度を算出し，値が高いものを話題性の高いトピックとする．

解析結果の可視化を ThemeRiver のような積み上げグラフでおこなうことで，トピックの話題性の増減が一目で分かる（図 1）．しかし，グラフ上で離れたトピック同士を比較することには向いていない．したがって本システムでは，スライダを用いた動的検索を導入してこの問題を解決する．

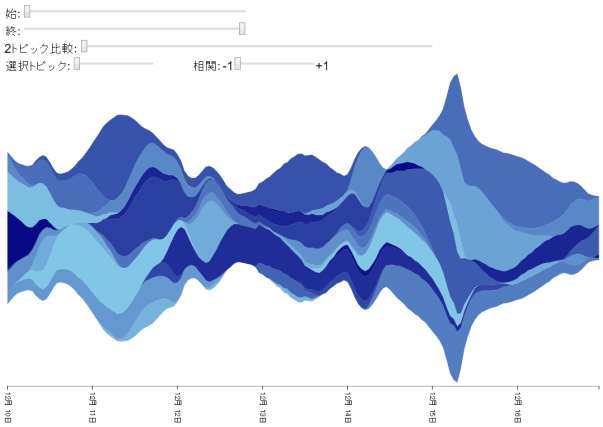

図 1　スライダ適用前のグラフ．

4.2 単純な 2 トピック比較

図 2 は，2 トピック比較のための動的検索の例を示している．動的検索のスライダを動かしていくと，総当り的に 2 つのトピックの色が濃くなり，2 つのトピックの単純な比較を支援する．可視化結果のグラフは，流れの太さが話題性の高さを表しており，左から右へ見ていくことでトピックの話題性の時間的推移を確認することができる．図 2 では，12 月 11 日から 14 日にかけて高い話題性を示したトピックがあることを確認することができる．

図 1 で示したように 20 トピック程度であればトピックの話題性を容易に比較することができるが，トピック数が増えると各要素の幅が狭くなってしまい，離れた要素同士の比較が困難になる．本システムでは 2 トピックを選択する方法として，本来は連続値を選択するスライダを採用している．トピックは離散値であるが，これは 2 トピック間の関係を掴むためにトピッ

クに割り振られる id を入力したり，ボタンを押して 1 つ 1 つ押して検査したりする仕様よりは，ユーザが流し見できるような仕様の方が負担が少ないと考えたからである．

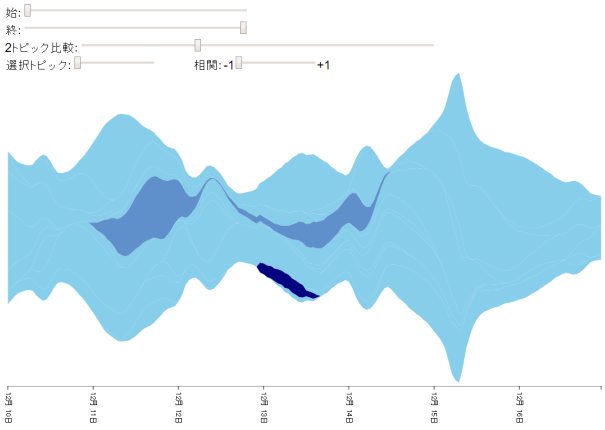

図 2　単純な 2 トピック比較の例．

4.3　相関を用いたトピック比較

図 3 と図 4 は，トピック同士の相関係数を計算した結果をもとに，特定のトピックに対して任意の相関係数を持つトピックの動的検索をおこなう例を示している．相関係数が +1 に近ければ 2 つのトピックは高い正の相関をもち，同じような増減をする（図 3）．逆に −1 に近ければ，反比例したような増減をする（図 4）．

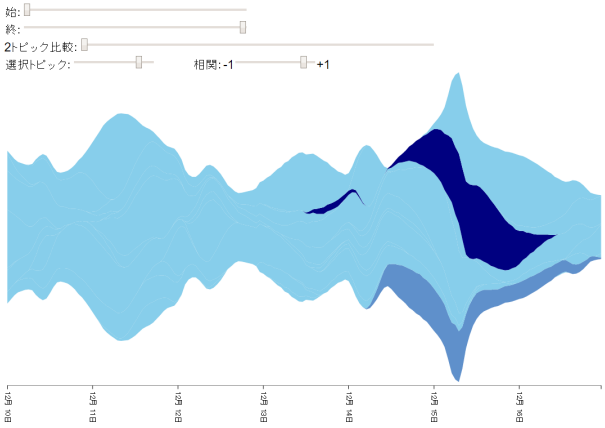

図 3　選択トピックと正の相関を持つトピックの表示．

ユーザは，選択トピックのスライダで比較対象としたいトピックを選択する．このトピックはグラフにおいて最も濃い色で表示される．次に相関スライダで数値を選ぶ．このスライダは 20 ステップで，一番左の状態だと −1.0 から −0.9 を，1 つ右にスライドさせると −0.9 から −0.8 を選択したことになる．比較対象となったトピックと相関スライダで選択した範囲の相関係数を持つトピックが次に濃い色で表示され，それ以外のトピックは最も薄い色で表示される．以上のように，スライダを使った可視化結果の動的な変化を示すことによって，ユーザは話題性の高いトピックやトピック同士の関連性を知ることができる．

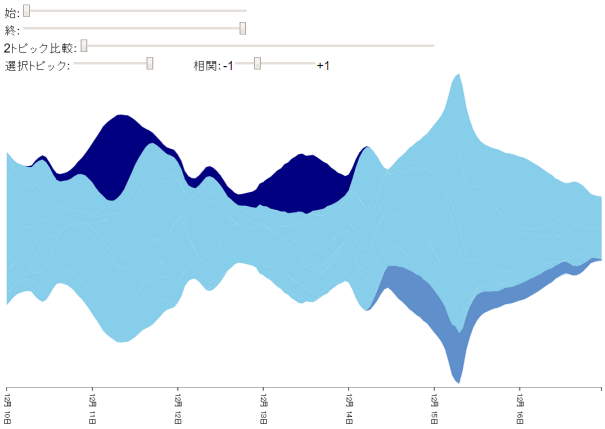

図 4　選択トピックと負の相関を持つトピックの表示．

相関係数の計算は以下のようにおこなった．期間 n のある時点 i における 2 つのトピック A と B の話題性をそれぞれ A_i, B_i とし，A と B の話題性の平均を $\overline{A}$, $\overline{B}$ とすると， A と B の相関係数 ρ_{AB} は次式で得られる．

$$\rho_{AB} = \frac{\sum_{i=1}^{n}(A_i - \overline{A})(B_i - \overline{B})}{\sqrt{\{\sum_{i=1}^{n}(A_i - \overline{A})^2\}\{\sum_{i=1}^{n}(B_i - \overline{B})^2\}}}$$

4.4　ズ　ー　ム

本システムでは，解析に使う元のデータや解析対象とする期間はあらかじめ入力する．しかし，解析対象とする期間を長くしすぎると，可視化結果のサイズはブラウザのサイズに依存しているため，潰れた形のグラフになってしまう．

これを解決するためにズームができるスライダを用意した．「始」と「終」とラベル付けされたスライダによって，可視化する期間の開始時刻と終了時刻を指定することで，注目したい期間へのズームができる．図 5 は，図 2 などの中央部分（12 日から 14 日）をズームしたものである．

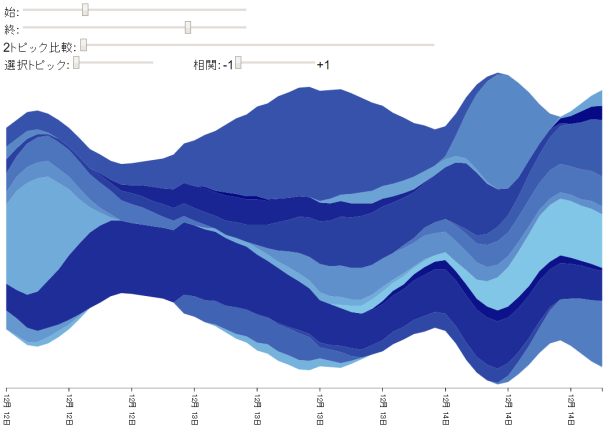

図 5　特定期間へのズーム．

5.　実　　装

本研究では，ニュースを収集するシステム，収集したニュースを解析して話題性を算出するシステム，結果を可視化するシステムを実装した．

ニュース収集システムは，日本の主要 5 紙（朝日・産経・日本経済・毎日・読売新聞）と海外の通信社 3 社（AFP・CNN・ロイター）の RSS 配信を 1 時間に 1 度取得する．使用したプログラミング言語は Java である．

次に，取得したニュースのタイトルを Java の日本語形態素解析器 Kuromoji を用いて形態素解析し，キーワードとなり得る名詞を抽出する．このとき，記号などが紛れ込まないようにするなどのクリーニングも同時におこなう．そしてキーワードごとに出現した回数，含まれるニュースの URL，共起するキーワードを記録する．この記録をもとに，動的トピックモデルを用いて複数のトピックに分類し任意の期間におけるバースト度を算出する．この 2 つ部分に関しては公開されているパッケージを用いた．

算出した話題性は，JavaScript のライブラリ D3.js を用いることで ThemeRiver のように上下対象の積み上げグラフとして可視化する．ユーザはこの可視化結果を Web ブラウザ上で閲覧することができる．

6. 実　　験

実験に使うデータは 2015 年 8 月 1 日から 12 月 31 日に取得したニュースである．また，解析の対象は 12 月 10 日 0 時台から 12 月 16 日 23 時台に取得したニュースである．本実験では，一般的にニュースを読む目的である「話題性の高い最新トピックについて知ること」を目指した．本研究において開発した話題性可視化システムの使い方を提案するとともに，ユーザは話題性の高いトピックを視覚的に把握できるという本研究の有用性を示す．

まず，上記のデータをニュース解析のシステムにかけた結果を入力とし，可視化システムを起動する．起動した画面では図 2 から図 5 で示したように，ThemeRiver を用いた可視化結果が表示される．この時点で，ThemeRiver の効果として流れが太い要素，つまり話題性の高いトピックがある程度分かる．ここで本研究が提案するのは，2 トピック比較のスライダを用いて，流れが太い 2 つの要素を探す作業をすることである．なお，話題性が最大の 2 要素を厳密に探す必要はない．スライダを動かしているうちに話題性の高い要素がおおよそ見つかるのが，動的検索を用いた利点である．

この作業をおこなって見つけ出した 2 つの要素 A と要素 B を図 6 に示す．12 月 10 日頃話題性が高く，12 月 15 日頃から再び話題性が高くなったトピックを表す要素を A とした．また，12 月 11 日から 12 月 14 日にかけて話題性が高いトピックを表す要素を B とした．それぞれデータを参照すると，A のトピックが含むキーワードはオリンピック関係のものであり，B はバドミントンとフィギュアスケートに関するトピックであった．

ニュースを読むことにあてられる時間は場合によって様々である．A と B のトピックに関するニュースを読んで，まだ時間があてられるならば次の作業に進むことを提案する．次は 4.3 節で説明したように，相関スライダを用いて別のトピックについても見ていく．例えば要素 A に関するトピックに興味があれば，A と高い正の相関をもつ要素をスライダ操作によって見つける．これが A と関連をもつことは保証できないが，動的トピックモデルによって統合できなかった，似ているトピックを表す要素の可能性がある．

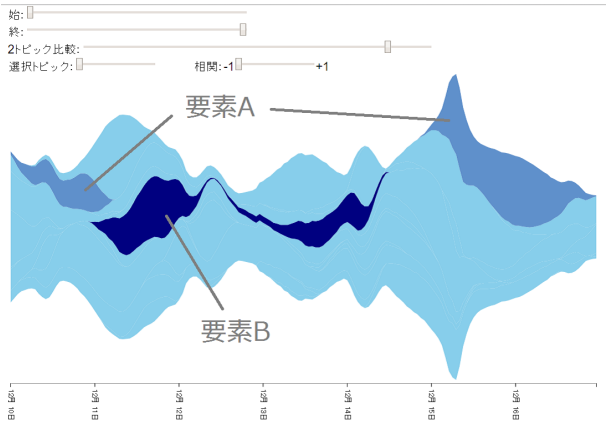

図 6　見つけ出した 2 要素．

逆に，A に関するトピックのニュースは十分だと感じたら，A と負の相関をもつ要素をスライダ操作によって見つける．この例を図 7 に示す．要素 C よりも長期間にわたって高い話題性を示した要素 D について詳しく見ると，要素 D を表すトピックが含むキーワードはパリ協定に関するものであった．このように，相関スライダを使った作業をある程度繰り返すことで，解析期間におけるトレンドやトピック同士の関連性を掴むことができる．

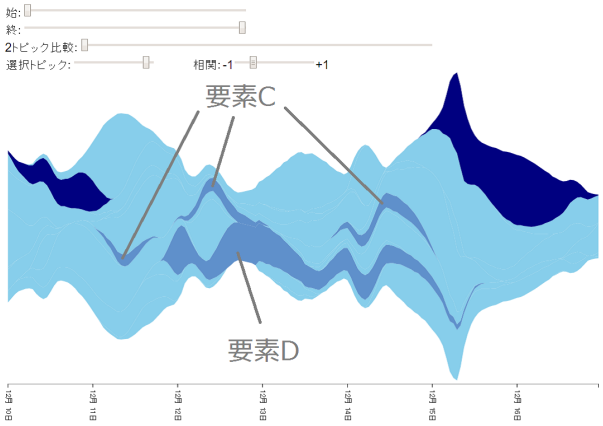

図 7　要素 A と負の相関をもつ 2 つの要素．

以上の結果を，Google の急上昇ワードのランキングと比較していく．要素 A について，オリンピック関係の急上昇ワードは 12 月 10 日には見られなかったが，15 日には「新国立競技場」が 4 位である．要素 B については，12 月 11 日は「羽生結弦」「グランプリファイナル（この期間中におこなわれていた大会）」がそれぞれ 1 位と 6 位，12 日は「浅田真央」が 1 位，14 日は「安藤美姫」が 2 位と，フィギュアスケートに関するキーワードが急上昇している．このことから，本システムで話題性の高いトピックとされたものは，これを見る限りでは社会的にも話題性の高いものであることが分かった．

ズームのスライダを使って表示する期間を短くすることで，

特定の期間を拡大して見ることができると 4.4 節で説明した．最初に長めの期間を表示しておくと，長期間におけるトピックの話題性の概観を見ることができる．しかし，各トピックの話題性の相関係数に差が出にくくなる．

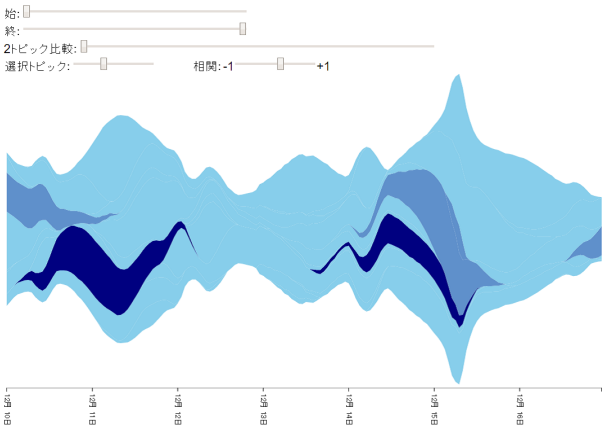

図 8　グラフと相関係数．

図 8 のように，14 日と 15 日の部分を見て正の相関が高そうなトピックがあるにも関わらず，その他の期間（10 日から 11 日など）で変化が似ていないと，もちろん相関係数は 0 に近づいていく．実際，図 8 で示しているのは選択トピックと 0 から +0.1 の相関係数もつトピックである．

そこで本システムでは表示期間を変えると，トピック同士の相関係数を再計算するようにした．先述のように長い期間で見ると似ていないトピックが，実は短い期間で見ると高い正の相関をもつことをユーザが発見しやすくなっている．逆に，長期間で見ると高い正（または負）の相関があっても，特定の期間では 0 に近い相関係数であることも操作している内に発見できる．これらがなぜ起こっているのかはそれぞれの場合によって異なり，一概にどのような意味があるのかは言えないが，読むニュースを選ぶための情報の 1 つとなり得る情報を可視化できる．

具体的に，要素 A と負の相関をもつものを 10 日から 12 日にかけての期間から探してみる．するとズーム前では見つけづらかったが，要素 A と −0.9 から −0.8 の高い負の相関をもつ要素を示すことができる．

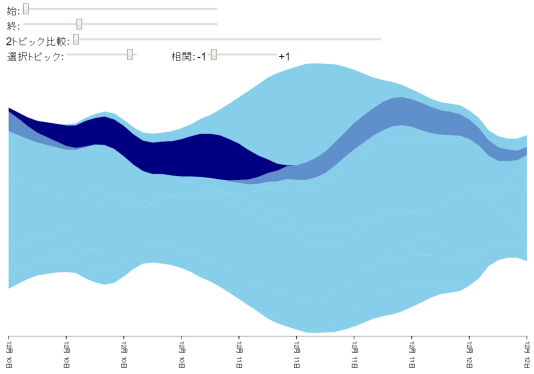

図 9　10 日から 12 日へのズーム．

今回の実験では 20 個のトピックについて 1 週間分の解析結果を出力としたが，トピックの数を増やしたり，表示する期間を長くしたりすることで，ズームできることがユーザにとってより有用な機能となることが期待できる．

7. 議　　論

前節において，要素 A も B もスポーツに関するトピックであったが，意図的な選択はおこなっていない．そもそもバースト解析の特性として急激な情報の増加を検知するため，日々ある程度報道されている政治や経済に関するトピックよりも，スポーツなどのトピックの方がバースト状態になりやすい．

要素 B のトピックで異なる競技のキーワードが混ざった原因として，先述のフィギュアスケートの大会と同時期に「バドミントン SS ファイナル」という大会も開催されていたからだと分かった．これを含む問題を解決するためには，より長期間のニュースを取得し，より多くのトピックに分割して解析する必要がある．

要素 C について実験の節では言及しなかったが，要素 C を表すトピックには政治に関するキーワードが含まれていた．要素 C のような細切れのトピックを可視化して得られる情報は少なく，これに対する改善が必要である．

本可視化システムは PC で使用することを目的として開発した．しかし，はじめにの章でも言及したようにオンラインニュースの閲覧は PC からだけでなく，スマートフォンなどサイズが小さいディスプレイの端末からもおこなわれる．JavaScript で開発した本システムは，JavaScript に対応したウェブブラウザであれば使用可能であり，Android 端末のエミュレータから起動した状態を図 10 に示す．

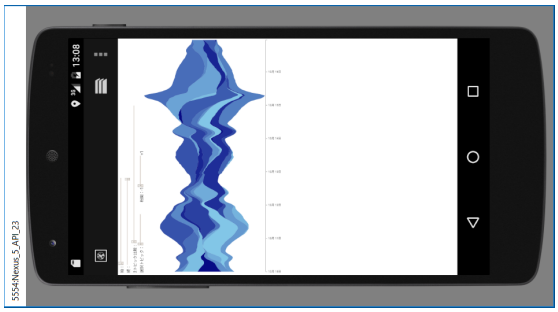

図 10　スマートフォンからの起動．

当然 PC のディスプレイよりも小さなサイズでの可視化となるため，グラフが細かくなってしまう．このような場合にも本システムが提供するズーム機能は有効である．ブラウザが提供するズームではグラフやスライダが画面に収まらなくなってしまい，グラフを見たりスライダを操作したりするためにスクロールが必要になる．スライダによるズームであれば，画面に収まるグラフのまま拡大され，表示されている期間におけるトピック同士の再計算された相関係数も動的に見ることができる．しかし，タップによるスライダの操作性や時間を示す横軸の見やすさなどは PC とは違い，改善する必要がある．

動的検索について，100 ミリ秒以内の更新速度を基準とし

ていたが，実験に使用した PC (CPU は Intel Core i7-6500U 2.50GHz，メモリは 8GB) とエミュレータで動く Nexus 5 (API Level 23) ではこれを満たすことができた．しかしデータやスライダがさらに増加または複雑化すると，100 ミリ秒の更新ができなくなる可能性もある．今回は実装途中の段階で基準の更新速度を満たすことが予想できたため，高速化は特におこなっていないが，基準を満たせなくなった場合は必要になる．

8. おわりに

本研究では，時系列ニュースの話題性を可視化するシステムとその使い方を提案した．ニュースサイトで読むニュースを選んだり，多くの記事を読んだりする負担をユーザにかけずに，本システムによって話題性の高いトピックやトレンドを可視化できることを示した．

追加の実験として，どれほどニュース読者の負担が減るかなどを，実際にニュースサイトでニュースを読む場合と比較などが考えられる．今後の課題として，ニュースの収集・解析・可視化のプログラムが現状では 1 つのシステムとしてまとまっていないため，収集から可視化まで自動でできるようにすることが必要である．また，可視化結果からニュース記事へのリンクを表示して，すぐにトピックに対応するニュースを読めるようにする必要がある．

JavaScript でプログラムした本可視化システムの Web での公開に対するハードルは低い．現状の課題を解決した後に，一般ユーザに利用してもらうことで新たな改善点の発見が見込め，本研究の発展が期待できる．

謝　　辞

本研究は JSPS 科研費 JP25540029 の助成を受けたものである．

文　　献

[1] 長谷川幹根, 石川佳治. T-Scroll：時系列文書のクラスタリングに基づくトレンド可視化システム. 情報処理学会論文誌：データベース, vol. 48, no. SIG 20 (TOD 36), pp. 61–78, 2007.

[2] D. Blei and J. Lafferty. Dynamic Topic Models. *Proc. ICML*, pp. 113–120, 2006.

[3] J. Kleinberg. Bursty and Hierarchical Structure in Streams. *Proc. KDD*, pp. 91–101, 2002.

[4] S. Havre, B. Hetzler and L. Nowell. ThemeRiver: Vizualizing Theme Changes over Time. *Proc. InfoVis*, pp. 115–123. 2000.

[5] B. Shneiderman Dynamic Queries for Visual Information Seeking. *IEEE Software*, vol. 11, no. 6, pp. 70–77, 1994.