テレビ番組視聴時におけるTwitter投稿からのトピック検知

中原　孝信，前川　浩基，羽室　行信

本研究は，特定のテレビ番組を視聴しながら投稿されたツイートの内容を解析することで，急激に投稿数が増加したときの内容などを検出し，それらを要約する手法を提案する．提案手法では，まず単語の共起関係に基づいたクラスタリングから概念を生成する．そして，バースト時の投稿と番組の台詞に一致した投稿を興味対象のツイートとして考え，それらのツイートをできる限り多く被覆するような少数のクラスタをナップサック制約付き最大被覆問題を用いて抽出する．抽出されたクラスタは，興味対象のツイートから得られたトピックを表していると考え，膨大なツイートから特定の目的に関係する投稿内容を要約することが可能である．計算実験では，テレビアニメーション番組「宇宙兄弟」を対象にして提案手法の有効性を示す．

キーワード：バースト検知，編集距離，マイクロクラスタ，ナップサック制約付き最大被覆問題

1. はじめに

情報通信技術の急速な発展と普及により，2000年代前半から掲示板やブログなどのテキスト情報は急激に増加した．その後インターネットを利用したコミュニケーションツールとして，Facebookやmixiなどのソーシャルメディアが出現し，他者とのつながりをより意識したコミュニケーションが可能となった．さらに，Twitter，Jaiku，mixiボイスなどに代表されるように，マイクロブログの利用者が増加している．マイクロブログは，ブログとチャットの性質を併せ持ったサービスで，手軽に文章を投稿できることから，投稿までに要する時間は短く，リアルタイム性を持ったコミュニケーションツールとして利用されている．

マイクロブログの流行によって，既存のメディアからは得ることが困難であった膨大なユーザの率直な意見をリアルタイムに入手することが可能になった．そのなかでもソーシャルビューイング（以下SV）と呼ばれる，テレビ番組を視聴しながらマイクロブログへ番組の感想や意見を投稿する視聴スタイルが盛んになってきている．Twitterユーザの54%はSVを経験しており，他人のツイートをきっかけに番組を視聴したことのあるユーザは30.5%という調査報告[9]がある．テレビを見ながら家族やお茶の間で話題を共有するというスタイルから，不特定多数の人とSNSを通じて，話題の共有や一体感を得たいという視聴スタイルへの変化が生じていると考えられる．

本研究では，Twitter投稿のなかでもSVに着目し，特定の番組を視聴しながら投稿しているTwitterの内容を解析することで，解析者が興味を持つツイート（以下，興味対象ツイートと呼ぶ）を要約する方法を提案する．興味対象ツイートとしては，投稿数の急激な増加を表すバースト時のツイートと，台詞と投稿内容が一致したツイートをそれぞれ取り上げる．

提案する方法では，まず単語の共起関係に基づいて関連する単語から構成される概念を生成する．そして，興味対象ツイートをできる限り多く被覆するような少数のトピックをナップサック制約付き最大被覆問題を用いて抽出する．ここで「トピック」という言葉は，興味対象ツイートに対する要約として利用する．例えば，対象番組についてバースト時の投稿を興味対象ツイートとした場合は，バースト時のツイートを要約したものがトピックである．提案手法を用いた実験では，「宇宙兄弟」を分析対象の番組として利用し，番組視聴時の投稿内容からトピックを抽出して，それらの有効性を評価する．

2. 関連研究

ニュース記事やTwitterに投稿された内容からトピックを抽出する研究は，投稿数（文章数）の急激な増加をバーストとして検知し，トピックモデルを利用して，

なかはら　たかのぶ
関西大学データマイニング応用研究センター
〒564–8680 大阪府吹田市山手町3–3–35
まえがわ　ひろき
（株）Magne-Max Captial Management
はむろ　ゆきのぶ
関西学院大学経営戦略研究科
〒662–8501 兵庫県西宮市上ヶ原一番町1–155

バースト時に出現する単語や文章を概念化している．そして特定のトピックを抽出する方法を提案している [8, 10]．これらはいずれも Kleinberg のバースト検知 [6] を利用した方法で，ドキュメント出現数の急激な増加に着目することでバーストを検知している．一方で，ドキュメントの急増を見つけるのではなく，時間区間内で出現した単語の生成確率分布からバーストしている単語を検知し，分布が類似した単語をグループ化することで，バーストイベントを抽出する方法も提案されている [4]．これらの研究は，バーストを検知してからそのトピックを抽出することを目的にしているが，本研究では，最初に文章に含まれる単語の共起情報に基づいて，互いに関連の強い単語からなるクラスタを概念として生成している．そして，文章の自動要約で用いられる手法を応用し，興味対象ツイートからトピックを抽出する．

文章の自動要約は，限られた文字数制限のもとで，文章を重複なく含める問題であり，その研究は 2000 年頃から行われている．初期の研究は，逐次的に文を選択する方法 [5] で文章要約が実現されていたが，2000 年代の中頃からは，最適化の問題として扱われており，Filatova ら [2] は，初めて文章要約を最大被覆問題として定式化し，貪欲アルゴリズムを提案した．高村ら [11] は，文章要約に対して最大被覆問題で提案されてきたアルゴリズムの詳細な比較実験を行っている．その実験によると，Filatova らの方法は，性能は若干劣るが，計算時間は最も早いことが示されている．本研究では膨大な Twitter データを扱うため，このアルゴリズムを基にして興味対象ツイートからトピックを抽出する．

3. 手法

本節では，興味対象ツイートからトピックを抽出する手法について論じる．図 1 に本稿で提案するトピック抽出に関する手法の概略を示す．まず，TV 番組に関するツイートデータを形態素解析により単語に分割し，共起頻度に基づいて単語をクラスタリングする（図 1 (1)）．ここで得られたクラスタは，対象とする番組における類似概念を単語集合として構成したものと解釈できる．そしてクラスタ間での要素の重複は許容し，また共起頻度のパラメータを調整することでサイズの異なるクラスタが多数列挙される．クラスタリングに利用した手法は，比較的小さなクラスタ（数個の単語を含むクラスタ）が多数列挙されることに特徴があり，それゆえマイクロクラスタリングと呼ばれる．

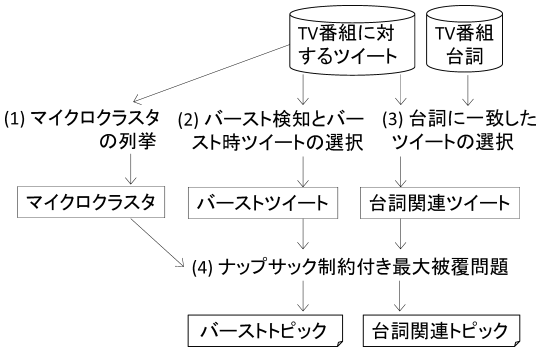

図 1 解析の流れ

次に，興味対象ツイートを選択する（図 1 (2)，(3)）．本稿では，バースト時のツイートと台詞に関係するツイートを興味の対象として取り上げている．バーストは，投稿間隔時間が統計的に十分短くなったと判断されたツイートを選択し（以下「バーストツイート」と呼ぶ），台詞との関連は，台詞との編集距離が小さいツイートを選択する（以下「台詞関連ツイート」と呼ぶ）．

以上の流れで得られた台詞関連ツイート，およびバーストツイートを，マイクロクラスタを使って要約する．そこでは，興味対象ツイートをできるだけ多く被覆するような少数のマイクロクラスタを選択するために，ナップサック制約付き最大被覆問題を適用する．

以下では，図 1 の (1)〜(4) の各手法について論述していく．

3.1 マイクロクラスタの取得

取得したツイートから関連の強い単語をクラスタリングするために，単語を節点に，関係の強い単語に枝を張ったネットワークを構成し，そこから密な部分グラフを抽出することで，意味の近い単語のクラスタを抽出する．ただし，単語の品詞としては，動詞，形容詞，名詞，副詞，感動詞を利用した．

関係性の強さは PMI (pointwise mutual information) によって定義した．単語 u の生起確率を $p(u)$，単語 v との共起確率を $p(u,v)$ で表すと，u と v の PMI は式 (1) で定義される．

$$\mathrm{pmi}(u,v) = \log_2 \frac{p(u,v)}{p(u)p(v)} \tag{1}$$

PMI の値が 0 より大きければ，二つの評価表現は共起しやすく，0 より小さければ共起しにくいと解釈できる．そしてユーザが指定した最小 PMI の γ について，$\mathrm{pmi}(u,v) \geq \gamma$ を満たすような二つの単語 u,v に枝を張る．

γ を小さな値にすると密なネットワークとなり，逆

に大きな値にすると疎なネットワークが構成されることになる．さらに，直接の隣接関係だけでなく，間接的な隣接関係も考慮に入れることでネットワークからノイズ的な枝を除去することができ，結果として，より小さく密な部分グラフを多く含むネットワークに変換することができる．

以上のようにして構成された単語ネットワークからクリークを列挙することで，クラスタを構成する．$G=(V,E)$ を節点集合 V と枝集合 E をもつ無向グラフとすると，節点集合 V の G の誘導部分グラフで任意の節点に枝があるようなものをクリークと呼ぶ．また，あるクリークがほかのクリークの真部分集合でなければ，それは極大クリークと呼ぶ．単語ネットワークから極大クリークを列挙することで，お互いに関係の強い単語集合を抽出することが可能となる．得られた極大クリークをわれわれはマイクロクラスタと呼ぶ．

3.2 バースト検知手法

これまでの多くのトピック抽出の研究において用いられてきたKleinbergのバースト検知手法 [6] は，メッセージの平均到着間隔についての確率分布の変化を検出することでバースト状態を検知する．この手法は，時系列データのモデル化手法の一つであるHMM (Hidden Malkov Model) をベースにしており，本稿でもこれと同等の手法を用いる．以下でその内容について説明する．

HMMは確率的状態遷移モデルとデータ生成モデルから構成され，観測される系列データは，隠れ状態におけるデータ生成モデルに従うと考える．時刻 t において観測されたデータ x_t は，隠れ状態 $z_t \in \{1,2,\ldots,K\}$ に定義された確率分布 $p(x_t|z_t;\boldsymbol{\phi})$ に従って生成されるようにモデル化される．ここで，$\boldsymbol{\phi}$ は生成モデルのパラメータベクトルで，t に依存せず一定であると仮定する．

また，隠れ状態 z_t は直前の状態 z_{t-1} にのみ依存して遷移し，その確率分布は $p(z_t|z_{t-1};\mathbf{A})$ で表される．ここで $\mathbf{A}=\{a_{i,j}|i,j=1,2,\ldots,K\}$ は，状態 i から状態 j への遷移確率表で，t に依存せず一定であると仮定する．ただし，$\sum_j a_{i,j}=1.0$ で，また初期状態 z_1 は確率ベクトル $\boldsymbol{\pi}$ に従うものとする．

以上より，観測データ系列 $X=x_1,x_2,\ldots,x_T$，および状態系列 $Z=z_1,z_2,\ldots,z_T$ の同時確率は式 (2) で与えられる [1]．

$$p(\mathbf{X},\mathbf{Z};\boldsymbol{\pi},\mathbf{A},\boldsymbol{\phi}) = p(z_1;\boldsymbol{\pi})\left[\prod_{i=2}^{T} p(z_i|z_{i-1};\mathbf{A})\right]\prod_{j=1}^{T} p(x_j|z_j;\boldsymbol{\phi}) \quad (2)$$

Kleinbergのバースト検知手法は，パラメータ $\boldsymbol{\pi},\mathbf{A},\boldsymbol{\phi}$ が与えられたなかで，データ系列 $\mathbf{X}$ を観測したときに，式 (2) で示された同時確率を最大化するような $\mathbf{Z}$ を見つける問題としてとらえることができる (式 (3))．

$$\mathbf{Z}^* = \underset{\mathbf{Z}}{\operatorname{argmax}}\ p(\mathbf{X},\mathbf{Z};\boldsymbol{\pi},\mathbf{A},\boldsymbol{\phi}) \quad (3)$$

本研究においては，観測データ系列 $\mathbf{X}$ がツイートの投稿間隔時間（秒単位）に対応し，隠れ状態は，定常状態とバースト状態の二状態 ($K=2$) である．そしてデータ生成モデルには指数分布 $f(x;\phi)=\phi e^{-\phi x}$ を用いている．指数分布は，単位時間（秒）あたりのツイート平均投稿数 ϕ をパラメータとしたときに，ツイート投稿間隔が従う分布である．

定常状態における平均投稿数 ϕ_1 は，全ツイートの単位時間あたりの平均投稿数とし，バースト時の平均投稿数は $\phi_2=s\phi_1$ で与えられる．ここで，$s>1.0$ はスケーリングパラメータで，この値を大きく設定すれば，より際立ったバーストのみを検知することになる．

次に，遷移確率表 $\mathbf{A}$ であるが，本研究では，定常状態からバースト状態への遷移確率 $a_{1,2}=0.8$ とし，逆の遷移確率 $a_{2,1}=0.5$ とした．そして，初期状態を決定する確率ベクトル $\boldsymbol{\pi}$ は $\pi_1=1.0,\pi_2=0.0$ とすることで必ず定常状態から始まるように設定した．

なお，TV番組に対する投稿は，一般的に番組の最初と最後，そして途中のTV広告時に増加する傾向がある．このようなデータに対してバースト検知を実施すると，それら増加時のみをバーストとして検知することになる．この問題を回避するために，番組の平均的な投稿分布によって投稿時刻を基準化する方法 [3] を用いた．

以上の手法を「宇宙兄弟」が放映される30分間につぶやかれたツイートに適用し，全ツイートに対してバーストかどうかの判定を行った．

3.3 シソーラス編集距離

ツイートが番組の内容に関係があるかどうかを台詞との距離によって測定する．距離の定義には，文章を単語列として見た編集距離を用いる．編集距離とは，二つの文字列について，挿入，削除，置換の三つの文字操作によって，一方の文字列 x を他方の文字列 y に変えるために必要な操作列に基づいて定義された距離で，

式 (4) のとおり再帰的に定義される．

$$d(x_i, y_j) = \min \begin{cases} \text{置換コスト}(x_i, y_i) + d(x_{i-1}, y_{j-1}), \\ \text{削除コスト}(x_i) + d(x_{i-1}, y_j), \\ \text{挿入コスト}(y_i) + d(x_i, y_{j-1}) \end{cases} \quad (4)$$

ここで，x_i は文字列 x の i 番目の文字を表し，$i = 1, 2, \ldots, |x|$，$j = 1, 2, \ldots, |y|$ で，また $d(x_0, y_0) = 0$ である ($|x|, |y|$ は，それぞれ文字列 x, y の長さ)．一般的に，挿入コストと削除コストは 1.0 とし，置換コストは 2 つの単語 x_i と y_j が同じであれば 0，異なれば 1.0 を設定する．また，この定義によると，文字列長が距離に影響を及ぼすため，変換に要した文字操作長で除する方法である正規化編集距離を用いる [7]．正規化編集距離の最大値は，二つの文字列長に関係なく，削除と挿入のみの操作によって変換されるケースで，その値は 1.0 となる．逆に最小値は，二つの文字列が完全一致するケースで 0.0 となる．本実験では 0.5 以下の編集距離を持つツイートを台詞関連ツイートとして選択した．

さらに，予備実験の結果，概念的に台詞に関係のあることをつぶやいていても，単語が完全に一致するケースは稀であることがわかった．そこで，本研究ではシソーラス辞書 [12] を用い，親の概念が同じであれば同一の単語であると判定することで置換コストを定義した．図 2 にシソーラス辞書の例を示す．

3.4 ナップサック制約付き最大被覆問題

本稿での目的は興味対象ツイート（例えばバースト時のツイート）を要約することである．そこで，できる限り多くの対象ツイートを被覆するような，少数のマイクロクラスタを選択する問題を考える．

いま，マイクロクラスタ集合 $M = \{m_1, m_2, \ldots, m_{|M|}\}$，および全ツイート集合 $W = \{t_1, t_2, \ldots, t_{|T|}\}$，および W の中から選ばれた興味対象ツイート W' が与えられているとする．またツイート集合 W においてクラスタ m が出現するツイート集合を $Occ(W, m)$ で表す．

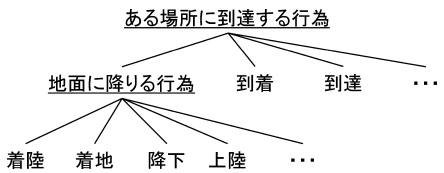

図 2 シソーラス辞書の例シソーラス辞書における，単語とその親概念（アンダーラインを付した文）の関係を示している．「着陸」と「着地」は同じ親を共有しているので同一の単語と判断するが，「着陸」と「到着」は親が同じでないので同一の単語とは判断されない．

e_{ij} を，マイクロクラスタ m_i がツイート t_j に出現していたとき 1 をとり，出現しなかったときに 0 となる定数とする（出現の定義は後述）．そして，マイクロクラスタ m_i を選択すれば 1，選択しなければ 0 となる 2 値変数を x_i としたとき，この問題は，式 (5) のとおり定式化できる．

$$\text{maximize} \quad \left| \left\{ j \;\middle|\; \sum_i e_{ij} x_i \geq 1 \right\} \right| \quad (5)$$

$$s.t. \sum_i c_i x_i \leq \kappa; \quad \forall i, \; x_i \in 0, 1$$

ここで c_i は，クラスタ m_i のコストであり，κ はユーザによって任意に与えられる総コストの上限値パラメータである（c_i の設定方法は後述）．

以上の問題は，ナップサック制約付き最大被覆問題と呼ばれる問題で，NP 困難問題であることが知られている [2, 11]．そこで図 3 に示される貪欲アルゴリズムを利用する．アルゴリズムのポイントは 6 行目で，すでに選択されたクラスタ S が被覆するツイート以外で，コスト c_i あたりの興味対象ツイート数が多いクラスタを優先的に選択していく．

1: κ : 総コスト上限値
2: W':興味対象ツイート集合
3: $M = \{m_1, m_2, \ldots, m_{|M|}\}$:クラスタ集合
4: $S = \phi$; $C = 0$
5: **while** $M \neq \phi$
6: $\quad m_i = \underset{m_i \in M}{\text{argmax}} \dfrac{|Occ(W', m_i) \backslash \bigcup_{d \in S} Occ(W', d)|}{c_i}$
7: $\quad$ **break if** $C + c_i > \kappa$
8: $\quad C = C + c_i$
9: $\quad$ insert m_i into S
10: $\quad$ delete m_i from M
11: **end**
12: output S.

図 3 ナップサック制約付き最大被覆問題の貪欲アルゴリズム

さて，「出現する」の定義であるが，あるツイート t_j がマイクロクラスタ m_i を構成する単語を μ 個以上含んでいれば，t_j に m_i が出現したと考える．μ の値を大きくすると，マイクロクラスタにより関連の強いツイートを得ることができるが，一方でサイズが μ 以下のマイクロクラスタはいずれのツイートにも出現しないこととなり，サイズの小さなクラスタは選択されなくなる．また全ツイートに対するマイクロクラスタの出現確率が σ より小さいマイクロクラスタは対象外とした．本研究では $\mu = 1$，$\sigma = 0.004$（バーストツイート），$\sigma = 0.001$（内容関連ツイート）として実験して

いる．

そして次にコスト c_i の設定方法であるが，ここでは式 (6) により与える．

$$c_i = |m_i| \cdot \left(1 - \frac{|Occ(W', m_i)|}{|Occ(W, m_i)|}\right) \tag{6}$$

これは，クラスタサイズ $|m_i|$（クラスタを構成する単語数）にクラスタ m_i の出現を条件としたときの興味対象でないツイートの確率を掛け合わせたものである．このコスト定義により，要約に利用する総単語数をできる限り減らしながら，対象ツイートへの関連が強いクラスタが選ばれるようになる．

4. 実験

4.1 実験データ

本研究では，TV アニメーション番組「宇宙兄弟」に関するツイートを分析対象とした．具体的には，「宇宙兄弟」「#uchukyodai」などを検索キーワードとして，2012 年 10 月 26 日から 2013 年 2 月 20 日までの約 28 万ツイートを Twitter から取得し，そのなかから番組放送時間中（日曜日午前 7 時〜7 時 30 分）のツイート（31〜41 話）を抽出して実験に用いた．また，台詞についても同様に 31〜41 話放送分の台詞をすべて人手により入力した．

4.2 マイクロクラスタの生成

マイクロクラスタは，話ごとに最小 PMI である γ を 0.1〜0.9 まで 0.1 ずつずらしながら生成した．そして，得られたクラスタのなかには，内容が同一のクラスタが複数存在する可能性があるためそれらを一意にする．この方法によって，多様なクラスタを生成することが可能である．表 2 の最右列が話ごとのクラスタ数で，平均すると約 3,000 のクラスタが生成されている．1 クラスタあたりの語数の平均は約 4 語，最大は 387 語であった．なおソフトクラスタリングであるため， 複数のクラスタに属する語も存在する．得られたクラスタの一部を表 1 に示す．ツイート内で共起する確率の高い語がクラスタを構成しているため，各クラスタは意味的に同質の概念であると考える．

4.3 結果の考察

興味対象ツイートの要約を目的に，ナップサック制約付き最大被覆問題から得られた結果を表 2（バーストツイート），表 3（台詞関連ツイート）に示す．それぞれ κ を 5, 10, 15, 20 の 4 段階で動かして実行した結果を示している．

評価指標としては，式 (7) に示される *Precision* と

表 1 クラスタの抜粋

話	No.	クラスタ
31	1162	{ コントロール，人生，空，誰 }
31	1574	{ 感動，やばい，ヒビト }
32	781	{ 公転，聞ける，ば，頭，出勤，自転 }
32	155	{ オープニング，好きだ，誰，曲，エンディング，やっぱり，シド }
37	720	{ なれる，人生，君，士，月面，飛行 }
37	447	{ ある，ねる，運，おめでとう，ムッタ }

表 2 バーストを対象に抽出されたトピックの精度

話	κ	*Precision*	*Recall*	*Supp*	#Bs	#Tw	#Cls
31	20	0.913	0.358	0.256	558	854	2183
32	20	0.830	0.194	0.105	454	1010	2337
33	15	0.708	0.071	0.037	478	1305	2841
35	15	0.730	0.066	0.028	407	1332	3015
36	10	0.733	0.069	0.024	317	1261	2386
37	10	0.743	0.058	0.022	446	1617	3020
38	20	0.739	0.077	0.035	666	1980	3621
39	20	0.800	0.092	0.048	743	1754	3022

#Bs はバーストツイート数，#Tw はツイート数，#Cls はクラスタ数を表す．

表 3 台詞関連ツイートを対象に抽出されたトピックの精度

話	κ	*Precision*	*Recall*	*Supp*	#Sc	#Tw	#Cls
31	5	0.765	0.542	0.020	24	854	2183
33	5	0.778	0.412	0.007	17	1305	2841
37	10	0.735	0.581	0.021	43	1617	3020
38	5	0.813	0.173	0.008	75	1980	3621
40	5	0.733	0.440	0.009	25	1590	2912

#Sc は，台詞関連ツイート数を表す．

式 (8) に示される *Recall*，そして式 (9) に示される *Supp* を利用した．*Precision* は選択されたクラスタによって被覆されたツイートのなかで興味対象ツイートの出現割合を表している．*Recall* は興味対象ツイートの中で被覆した興味対象ツイートの割合を表している．また *Supp* は，全ツイートのなかで被覆した興味対象ツイートの割合を表している．

$$Precision = \frac{|\bigcup_{m \in S} Occ(W', m)|}{|\bigcup_{m \in S} Occ(W, m)|} \tag{7}$$

$$Recall = \frac{|\bigcup_{m \in S} Occ(W', m)|}{|W'|} \tag{8}$$

$$Supp = \frac{|\bigcup_{m \in S} Occ(W', m)|}{|W|} \tag{9}$$

そして，各表は *Precision* が 0.7 以上で *Recall* の最も高い κ を話ごとに一つ選択した結果をそれぞれ示している．κ は大きすぎると要約にならないため最大で 20 とした．バーストに関しては，34，40，41 話は *Precision* が 0.7 以上になる結果を抽出することができなかったため，表 2 から省いている．また，台詞関連ツイートに関しても同様に，32，34，35，36，39，41 話は *Precision* が 0.7 以上になる結果を抽出することができなかったため，表 3 から省いている．

表 2 に示した結果は，抽出されたクラスタに被覆されているツイートの 7 割以上がバーストツイートであり，バースト時のツイートを少数のマイクロクラスタでとらえることができている．しかし，全体的に *Recall* は低く，バーストツイートのすべてを被覆できているわけではない．より多くのバーストツイートをカバーするようにクラスタを選択したいのであれば，*Precision* よりも *Recall* を優先した選択を行うことになる．

表 3 の台詞関連ツイートはいずれの話も *Supp* が小さい．これは全ツイートに比べて，台詞関連ツイートの数が少ないことを意味している．しかし，*Precision* と *Recall* は比較的高い値になっているので，ツイート数が少なくても有効なトピックが抽出できていると考えられる．

4.3.1 バーストツイートに関する考察

図 4 は，32 話を対象に放送時間中のツイートから上述のバースト検知手法を適用して得られた結果を示している．スケーリングパラメータ $s = 1.1$ でバースト検知を行った．横軸は秒，細い線は投稿間隔を表しており，波の振れ幅が小さい箇所は，投稿間隔が短いことを示している．太い線で囲まれている時間帯がバーストしている時間帯を表している．32 話は全体でバーストが 5 回起こっており，前半の約 200〜400 秒の間

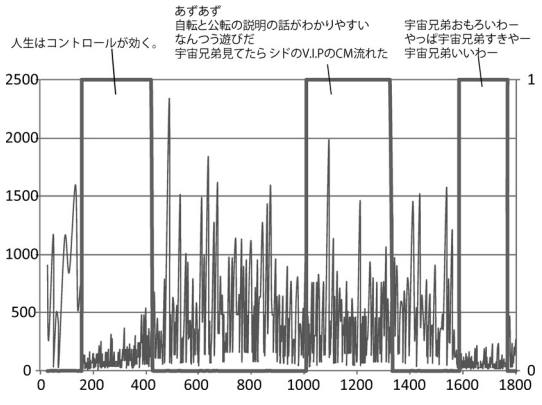

図 4 検知されたバースト

が最も長いバースト状態であった．

バーストツイートのトピック抽出を試みた結果以下の 16 個のマイクロクラスタが選ばれた．これは表 2 の 32 話の結果に対応している．{ あずあず }，{ 自転　公転　出勤　頭 ば }，{ 公転　自転 }，{ 人生　コントロール 効く }，{ シド　流れる }，{ 月面　着陸 }，{ ず　誕生日 }，{ はじ }，{ 色　ムッタ }，{ 孤独だ }，{ ムッ}，{ 出勤　頭　ちょっと }，{ 泣く }，{ 聞ける }，{ 言う さん }，{ 遊ぶ }．

{ あずあず } は主人公のムッタが，先輩宇宙飛行士の吾妻さんに「あずあず」というあだ名をつけようとしたときで，そのあだ名と人物のギャップが面白く 7 時 19〜21 分にバーストしいてる．また，{ 自転　公転　出勤　頭　ば }，{ 公転　自転 } は，自転と公転遊びという，ムッタの周りを吾妻さんの息子がボールを持って走り回るというシーンがあり，「自転と公転の説明の話がわかりやすい」や，「なんつう遊びだ」など，自転と公転に関する投稿から，7 時 19〜21 分にバーストしている．それ以外にも，{ 人生　コントロール　効く } は，「人生はコントロールが効く」という名言に反応したバーストが，7 時 3〜4 分に起こっている．{ シド　流れる } は，「宇宙兄弟見てたらシドの V.I.P の CM 流れた.」など，放送中の CM に反応して起こったバーストが検知できている．これらは，バースト中の投稿内容を要約したトピックであり，図 4 からわかるように同じバースト時間中に複数のトピックが出現しており，多様なトピックを抽出することができている．また，抽出したトピックは複数の単語から構成されたものが多く，マイクロクラスタリングが有効に機能していたことが示される．

4.3.2 台詞関連ツイートに関する考察

台詞関連ツイートのトピック抽出を試みた結果，37 話では以下 10 個のクラスタが選択された．{ しそ }，{ 人生 }，{ 仲間 }，{ 君 }，{ キミ }，{ モジャ }，{ 下 }，{ 足りる }，{ 迎える }，{ 間違う }．

{ しそ } は，主人公である六太の母親が，「お味噌汁」を「おしそみる」といったことから，「**おしそ**みる」というツイートが複数回行われており，それを検知している．{ 人生 } は，「今日から君の宇宙飛行士**人生**が始まるぞ」という台詞がそのまま投稿されており，その内容を検知した．そのほかにも「**仲間**に頼ると言うことだ」，「**君**を宇宙飛行士として迎えます.」，「おめでとう，**キミ**には運がある」「どうした**モジャ**君，そのおでこ」，「おれはこういうことは空の**下**で伝えたい」など台詞を真似たツイートが検知されている．しかし，こ

れらのクラスタはすべて一つの単語からなるクラスタであり，台詞関連ツイートのなかの一語がクラスタ内の一語と一致しているだけであり，バーストに比べるとマイクロクラスタの効果は少ない．

この原因の一つは，表 3 の *Supp* からもわかるように，編集距離によって選択された台詞関連ツイートが少なく，複数の単語からなるクラスタは，台詞関連ツイート以外のツイートを被覆する可能性が高くなってしまい，複数の単語をもつクラスタの価値が相対的に低くなったと考えられる．

5. おわりに

本研究は，宇宙兄弟を視聴しながらツイートした内容を対象にして，マイクロクラスタリングにより概念を生成し，バースト検知と編集距離を利用して興味対象ツイートを選択した．そして，ナップサック制約付き最大被覆問題を応用して，興味対象ツイートが多く被覆されるような少数のクラスタを選択しトピックを抽出する手法を提案した．抽出されたトピックは興味対象ツイートを要約した内容になっており，効率的に Twitter の内容を把握することが可能である．

バーストを対象としたトピック抽出では，「名言」によるバーストや，「笑い」によるバースト，そして，CM 中に起こったバーストまで，多様なバーストをトピックとして要約することができた．同じバースト時間中に複数の異なるトピックが存在しており，あらかじめ特定のトピックだけを対象にした抽出方法では，これらすべてをとらえることは困難である．提案手法は，最初にマイクロクラスタを利用してさまざまな概念を生成し，興味対象ツイートを要約する概念をトピックとして効率的に選択できることから，提案手法の有効性を示した．今後は，情報番組などに提案手法を適用することで，要約されたトピックからマーケティング施策への応用が期待できるため，よりビジネスへの応用を意識した研究を進めていきたい．

謝辞　本研究で利用したマイクロクラスタリングは，国立情報学研究所の宇野毅明准教授が実装されたツールを利用させていただいた．本研究の一部は，ERATO 湊離散構造処理系プロジェクト，および文部科学省の科研費若手研究 (B) 4730375 の研究助成を受けている．

参考文献

[1] C.M. ビショップ著，元田浩，栗田多喜夫，樋口知之，松本裕治，村田昇（編），パターン認識と機械学習（下）：ベイズ理論による統計的予測，13 章，323–370，2008.

[2] E. Filatova and V. Hatzivassiloglou, “A Formal Model for Information Selection in Multi-sentence Text Extraction,” *Proceedings of the International Conference on Computational Linguistics (COLING)*, 397–403, 2004.

[3] 藤木稔明，南野朋之，鈴木泰裕，奥村学，「document stream における burst の発見」，情報処理学会研究報告，自然言語処理研究会報告，一般社団法人情報処理学会，**23**，85–92, 2004.

[4] G. Fung, J. Yu, P. Yu and H. Lu, “Parameter Free Bursty Events Detection in Text Streams,” *Proceedings of the 31st International Conference on Very Large Data Bases*, **12**, 181–192, 2005.

[5] J. Goldstein, V. Mittal, J. Carbonell and M. Kantrowitz, “Multi-Document Summarization By Sentence Extraction,” *Proceedings of the ANLP/NAACL Workshop on Automatic Summarization*, 40–48, 2000.

[6] J. Kleinberg, “Bursty and Hierarchical Structure in Streams,” *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, **11**, 91–101, 2002.

[7] A. Marzal and E. Vidal, “Computation of Normalized Edit Distance and Applications,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **15**, (9), 926–932, 1993.

[8] 中澤昌美，帆足啓一郎，小野智弘，「Twitter を用いたテレビ番組からのイベント検出及びラベル付与手法」，一般社団法人情報処理学会，第 3 回データ工学と情報マネジメントに関するフォーラム，517–519，2011.

[9] SNS × TV 連携の現状と展望 Twitter/Facebook，mixi/LINE の取り組み，http://av.watch.impress.co.jp/docs/news/20121018_566709.html

[10] 高橋佑介，横本大輔，宇津呂武仁，吉岡真治，「ニュースにおけるトピックのバースト特性の分析」，情報処理学会研究報告．自然言語処理研究会報告，一般社団法人情報処理学会，**6**，1–6，2011.

[11] 高村大也，奥村学，「最大被覆問題とその変種による文書要約モデル」，人工知能学会論文誌，社団法人人工知能学会，**23**，(6)，505–513，2008.

[12] 独立行政法人，情報通信研究機構 日本語 WordNet (1.1) 最新版，http://nlpwww.nict.go.jp/wn-ja/

