ナップサック制約付き最大被覆問題を用いたTwitterからのトピック検知

中原 孝信　　関西大学
前川 浩基　　(株)Magne-Max Data
羽室 行信　　関西学院大学

2013年6月28日　ERATO湊離散構造処理系プロジェクト 2013初夏のワークショップ

背景と目的

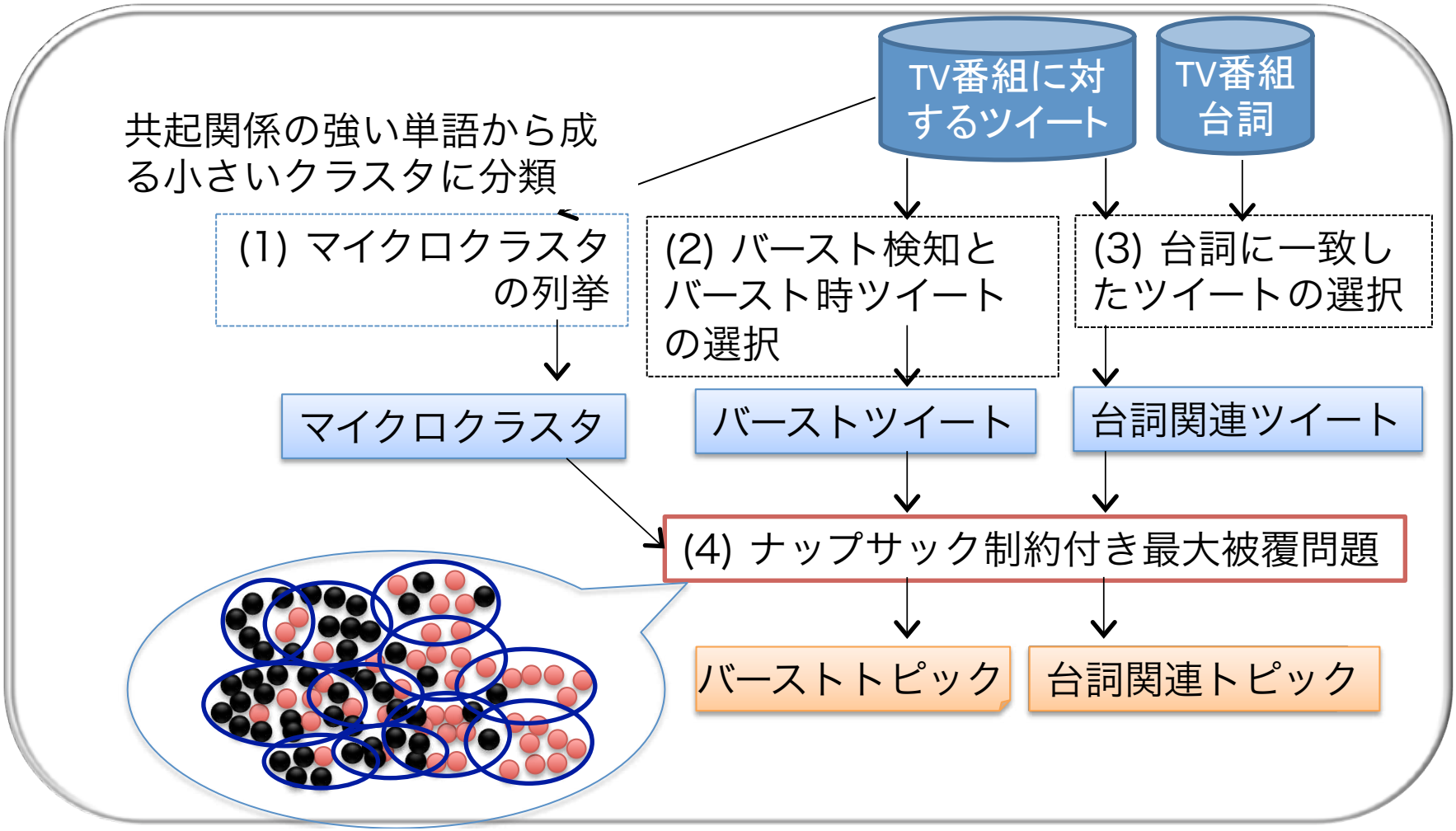
- テレビ番組を視聴しながらSNSへ意見を投稿する視聴スタイル（以下SV)が盛んであり、Twitterユーザの約半数がSVを経験
- Twitterを解析して投稿内容を集約することで、意味のある情報を抽出しマーケティングに活用したい
- 既存研究は、バーストを検知してからトピック抽出しているので、のっぺりしたトピックは抽出できない。
- 抽出するトピック数を決める必要がある

本研究の目的

特定の番組を視聴しながら投稿したツイート内容を解析し、解析者が興味を持つ投稿内容を要約する手法を提案する。

提案手法の特徴と概要

- ボトムアップ的なアプローチでマイノリティも抽出可能
- ナップサック制約付き最大被覆問題で重複なくトピックを検出可能



マイクロクラスタの取得

①ツイートデータを形態素解析によって単語に分割

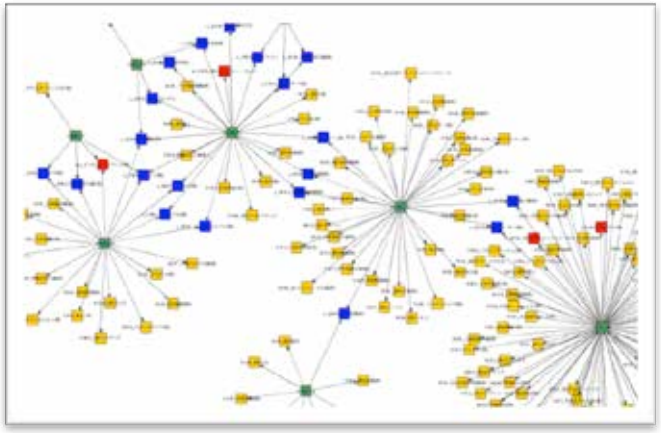
②単語を節点、共起度をpmiで表し最小pmiのγについて、$\mathrm{pmi}(u,v) \geq \gamma$
を満たすような二つの単語u,vに枝を張る。

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

単語u の生起確率を$p(u)$
単語v との共起確率を$p(u, v)$

*直接の隣接関係と間接的に接続された節の隣接関係を考慮することで、
最終的にノイズ的な枝を除去することが可能

③密な部分グラフを多く含むネットワークに変換し、そこから極大クリークを列挙



NII宇野毅明先生が開発されたツールを利用

バースト検知

Kleinbergの提案したバースト検知手法を利用し、番組放送中の投稿の中で急激に投稿数が増加したバーストイベントを検知

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

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

$\mathbf{X}\{x_1, x_2, ..., x_T\}$ は観測データ系列

$\mathbf{Z}\{z_1, z_2, ..., z_T\}$ は状態系列

$\boldsymbol{\pi}$ は初期状態 z_1 を定める確率ベクトル

$\mathbf{A}\{a_{i,j} \mid i, j = 1, 2, ..., K\}$ は状態 i から状態 j への遷移確率表

ϕ は 生成モデルのパラメータベクトル

x_t は時刻 t において観測されたデータ

z_t は時刻 t における隠れ状態（定常／バースト）

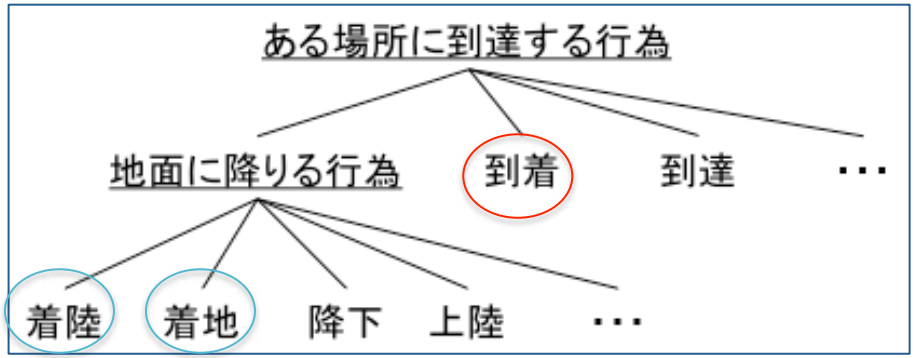
シソーラス編集距離

投稿内容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

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

x_iは文字列xのi番目の文字、y_jは文字列yのj番目の文字
$d(x_0,y_0)$=0,　挿入・削除コスト=1.0,　置換コスト(一致なら0, 異なれば1.0)

- 文字列長に依存しないように、変換に要した文字列操作長で除する正規化編集距離を利用
- シソーラス辞書を利用して、単語の親概念が一致すれば同一の単語であると判定し置換コストを計算



シソーラス辞書の例

ナップサック制約付き最大被覆問題

興味対象ツイートを要約するために、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

$$maximize|\{j|\sum_i e_{ij}x_i \geq 1\}|$$

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

e_{ij}はマイクロクラスタm_iがツイートt_jに出現していたとき1、出現しなかったときに0となる定数とする

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

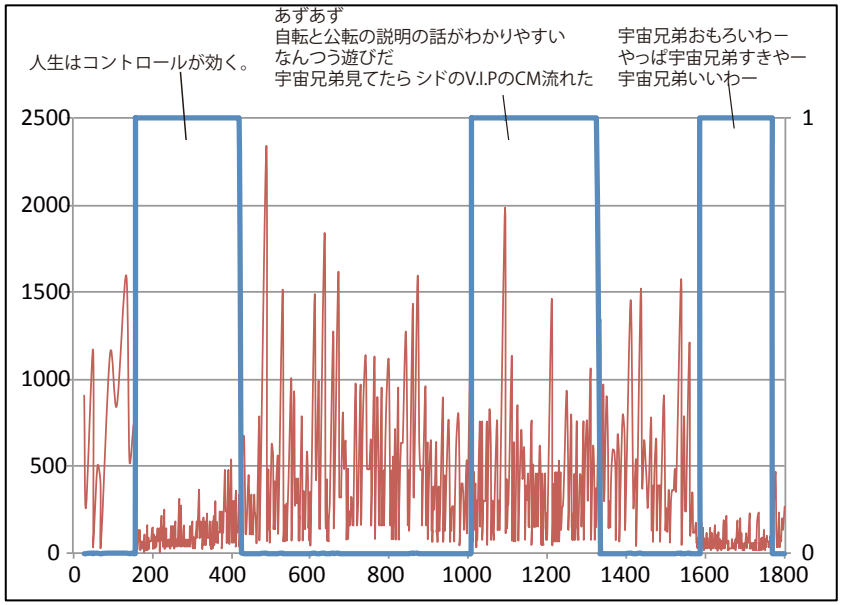
$$c_i = |m_i| \cdot \left(1 - \frac{|Occ(W', m_i)|}{|Occ(W, m_i)|}\right)$$

コストは、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時の興味対象でないツイートの確率を掛けあわせたもの

適用結果

宇宙兄弟を視聴しながら投稿したツイートを提案手法によって解析

バーストツイートに関するトピック検知



話	κ	Precision	Recall	Supp	#Bs	#Tw	#Cls
32	20	0.83	0.194	0.105	454	1010	2337

16個のクラスタでバーストツイートを要約

{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}.

台詞関連ツイートに関するトピック検知

10個のクラスタで台詞関連ツイートを要約

話	κ	Precision	Recall	Supp	#Bs	#Tw	#Cls
37	10	0.735	0.581	0.021	43	1617	3020

「お味噌汁」を「おしそみる」といった

今日から君の宇宙飛行士人生が始まるぞ

{しそ}, {人生}, {仲間}, {君}, {キミ}, {モジャ}, {下}, {足りる}, {迎える}, {間違う}.