Vol.2012-NL-209 No.4
2012/11/22

隣接するツイート間の関係を考慮したマイクロブログのトピック推定

中村 直哉[1,a)]　笹野 遼平[2]　高村 大也[2]　奥村 学[2]

概要：近年，マイクロブログというサービスが登場し世界中で流行している．マイクロブログに投稿される情報は，従来のメディアとは異なる性質をもつと考えられ，マイクロブログを解析することにより従来のメディアからは得ることができなかった多くの有益な情報を得ることができると考えられる．しかし，マイクロブログを解析するにあたり，離散データの解析手法として用いられる LDA などの一般的なトピックモデルを適用しようとしても，隣接する投稿の間のトピックの連続性や 1 つ 1 つの投稿の短さといったマイクロブログの性質を考慮していないため，その意味内容を十分に捉えることができない．そこで本研究では，隣接する投稿間の関係を考慮したマイクロブログのための新しいトピックモデルを提案する．

キーワード：情報抽出，マイクロブログ，トピックモデル

Topic Estimation for Microblogs Taking into Account the Relationships between Adjacent Tweets

Naoya Nakamura[1,a)]　Ryohei Sasano[2]　Hiroya Takamura[2]　Manabu Okumura[2]

Abstract: In recent years, micloblogs have appeared and become prevalent all over the world. Since the characteristics of the information contained in micloblogs are considered to be different from those contained in conventional media, we can get a lot of useful information from micloblogs. However, commonly used topic models, such as LDA, do not work well on micloblogs, since these topic models do not take into account the relationships between adjacent posts and the shortness of each post. In this paper, we propose a new topic model for micloblogs that takes into account the relationships between adjacent posts.

Keywords: Information extraction, Microblog, Topic model

1. はじめに

近年，ブログと比較して 1 つ 1 つの投稿が短いマイクロブログと呼ばれるタイプのブログサービスが登場し，注目を集めている．本稿では特に人気が高く，広く利用されている Twitter について考える．Twitter はユーザが身の回りで起きた出来事であったり，自分の近況などを手軽に投稿でき，また，情報を共有する様々な仕組みが備わっている．この新しい形式のソーシャルメディアを有効活用するため，近年多くの研究がなされている．たとえば，リアルタイム情報を基にしたニュース推薦システムの構築 [7] や，Twitter のトポロジカルな特徴に注目した情報伝播の解析 [4]，リアルタイムなイベント検知の研究 [8] などの研究が行われている．

Twitter はユーザが手軽に投稿できること，情報の共有が容易であることなどから，Twitter で得られる有用な情報は従来のマスメディアとは異なる性質をもつと考えられる．しかし，離散データの解析手法として用いられる LDA などの一般的なトピックモデルを Twitter のデータに適用し

[1] 東京工業大学　総合理工学研究科
Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology
[2] 東京工業大学　精密工学研究所
Precision and Intelligence Laboratory, Tokyo Institute of Technology
[a)] nakamura0617@lr.pi.titech.ac.jp

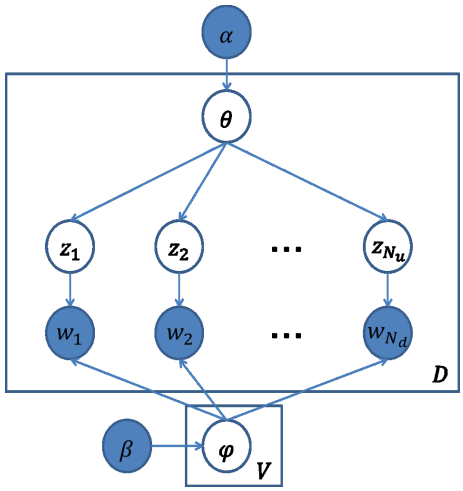

図 1 Latent Dirichlet Allocation を構成する変数の確率的依存関係

ようとしても，隣接する投稿の間のトピックの連続性や 1 つ 1 つの投稿の短さといったマイクロブログの性質を考慮していないため，その意味内容を十分に捉えることができない．

LDA は 1 つ 1 つが短い文書群に適用しても，上手く動作しないことから，Zhao らは各ユーザのツイートをまとめて 1 文書とモデル化し，さらに Twitter ユーザが 1 つのツイートで 1 つのトピックについてのみ言及する仮定に基づくトピックモデルである Twitter-LDA を提案した [10]．ツイートをまとめて 1 文書とすることで，ツイートが短いことでトピックが適切に推定できないという問題がある程度，解消された．しかし，Zhao らのモデルでは各ツイートごとのトピック生成確率は互いに独立と仮定しているが，経験的にユーザは同じ話題を連続してツイートすることが多く，この仮定では Twitter の性質を十分に捉えきれていないと考えられる．

そこで本研究では，Twitter-LDA と同様にユーザが 1 つのツイートで 1 つのトピックについてのみ言及すると仮定した上で，さらに，隣接するツイート間の関係を考慮したトピックモデルを提案する．実験の結果，隣接したツイート間の関係を考慮する方が，独立と考えるよりトピックを適切に推定できることが確認できた．

2. 関連研究

2.1 Latent Dirichlet Allocation

Blei らは，潜在トピックを考慮した文書生成モデルの一種である Latent Dirichlet Allocation (LDA) を提案した [1]．LDA では，各文書中の単語は，有限個の要素からなるトピック集合のうちの 1 つから生成されると仮定する．各単語は，多項分布に従って生成されるものとし，この分布をトピックごとに仮定する．また，トピックは多項分布に従って生成されるものとし，この分布を文書ごと仮定する．ここで，文書ごとに仮定された多項分布と，トピックごとに仮定された多項分布が，それぞれディリクレ分布から生成されるとする．変数間の確率的依存関係を視覚化したものを，図 1 に示す．パラメータ推定の方法は大きく分けて 2 つが知られており，変分ベイズによる推定とギブスサンプリングによる推定がある．推定の結果，トピック-単語分布と文書-トピック分布に対するディリクレ事後分布が得られる．このような分布を得ることで，文書ごとのトピック分布や，各トピックにどのような単語が含まれているのか知ることができる．

2.2 Twitter-LDA

LDA は短い文書に対しては上手くいかないことが知られており，例えば Twitter の各ツイートを 1 つの文書として LDA を適用することは難しい．この問題を解決するモデルとして，Twitter-LDA がある [10]．このモデルの特徴として，以下の 2 点が挙げられる．

(1) 各ユーザのツイートをまとめて 1 つの文書とする
(2) 1 つのツイートにつき 1 つのトピックが割り振られる

(1) は，同じユーザのツイート集合は，そのユーザのトピック分布に基づいて生成されるという仮定に基づいている．(2) については，通常，1 つのツイートでは 1 つのトピックについてしか言及しないという仮定に基づいている．

次に，Twitter-LDA の生成プロセスについて説明する．T をトピック数，U をユーザ数，$\phi=\{\phi^1,\ldots,\phi^T\}$ をトピック-単語分布，$\theta=\{\theta^1,\ldots,\theta^U\}$ をユーザ-トピック分布，$z=\{z_1,\ldots,z_{N_u}\}$ を各ツイートに割り振られるトピックとおく．また，ディリクレ分布のパラメータを α, β とする．まず，全てのトピックについて ϕ^t はディリクレ分布から生成される．次に，各ユーザについて θ^u が生成され，各ツイートに対して θ^u をパラメータとする多項分布からトピックが生成される．この時，各ツイートに対するトピックの生成確率は条件付き独立を仮定しており，ここが本研究の提案手法と異なる点に注意する．次に，各ツイート中の各単語に対して，ϕ^{z_s} をパラメータとする多項分布から単語が生成される．ここで，z_s はツイート s のトピックを意味する．

2.3 トピックの局所性に注目したトピックモデル

たとえば，オーディオプレイヤーのレビューテキストでは，文の初めの方に音質についての言及が多く，真ん中の方では価格についての言及が多く，最後の方では操作性についての言及が多いなど，文書より小さい単位でトピックの分布を考えた方が適切なことがある．このようなトピック分布の局所性に注目した研究は多く行われている．Titov らは，文書あたりのトピック分布に加え，連続した複数の文ごとのトピック分布まで考慮した Multi-grain LDA (MG-LDA) を提案している [9]．Gruber らは，文単位のトピック遷移をモデル化した Hidden Topic Markov Model (HTMM) を提

案している [3]．また，Diao らは，Twitter のユーザごとのトピック分布に加え，タイムスタンプごとのトピック分布を考慮した TimeUserLDA を提案している [2]．

3. 隣接するツイート間の関係を考慮したトピックモデル

3.1 隣接するツイートの性質

Twitter を観察すると，同じ話題について連続してツイートするユーザが多いことに気付く．この傾向は，短い間隔で投稿されたツイートの場合，特に顕著である．

一方，Twitter における 1 つ 1 つのツイートは短いため，単独のツイートからそのトピックを適切に判断するのが困難な場合がある．たとえば以下のような 2 つのツイート列があった場合，2 番目のツイートはいずれも「ACL の日程が発表されていた.」であるが，ツイート列 1 におけるトピックは “計算言語学” であるのに対し，ツイート列 2 におけるトピックは “サッカー” であると考えられる．

ツイート列 1
(1) ようやくネットに繋がった.
(2) ACL の日程が発表されていた.
(3) 論文投稿の締切は 1 月 15 日らしい.

ツイート列 2
(1) スポーツニュースが始まった.
(2) ACL の日程が発表されていた.
(3) 決勝は 11 月に行われるらしい.

このようなトピックの違いは前後のツイートを考慮に入れることで認識することができると考えられる．たとえばツイート列 1 では直後のツイートに「論文」や「投稿」などの表現が含まれることから学術的なトピックであること，ツイート列 2 では直後のツイートに「決勝」という表現が含まれることからスポーツなどに関するイベントであることが推測できる．そこで，本研究では，隣接するツイートのトピックが連続して同じになる傾向がある点に注目したトピックモデルを提案する．

3.2 提案手法

Twitter-LDA では，新たなツイートのトピックは常に θ^u からトピックから生成される．一方，提案手法においては，隣接するツイートの性質を考慮し，一定の確率で直前のトピックを引き継ぐというモデルを考える．すわなち，提案手法においては，新たなツイートのトピックは以下の 2 通りのうちの 1 つが選択されるとする．

- 直前のツイートのトピックを引き継ぐ
- θ^u からトピックが生成される

Twitter-LDA，提案手法について，モデルにおける変数間の確率的依存関係を視覚化したものを図 2 と図 3 に示す．提案手法においては，隣接するトピック間に依存関係があり，直前のツイートのトピックを引き継ぐかどうかを制御する変数 y が追加されている点が，Twitter-LDA と異なっている．

提案モデルの生成プロセスは次のようになる．

---

(1) **for each** topic $t = 1, ..., T$
　draw $\phi^t \sim Dir(\beta)$
(2) **for each** user $u = 1, ..., U$
　(a) draw $\theta^u \sim Dir(\alpha)$
　(b) draw $z_1 \sim Multinomial(\theta^u)$
　(c) **for each** tweet $s = 2, ..., N_u$
　　(i) draw $y_s \sim Bernoulli(\gamma)$
　　(ii) **if** $y_s = 0$ **then** draw $z_s \sim Multinomial(\theta^u)$
　　else $z_s \leftarrow z_{s-1}$
　　(iii) **for each** word $i = 1, ..., N_{u,s}$
　　draw $w_i \sim Multinomial(\phi^{z_s})$

---

(2)(b) までは通常の Twitter-LDA と同じである．提案手法では，(2)(c) においてツイートに割り当てられるトピックが，前と同じになるか，または，分布から生成するか選択する変数が生成され，続いて，生成された変数に従って新たなツイートのトピックが生成される．

3.3 学習

パラメータ推定には，ギブスサンプリングを用いた. ギブスサンプリングは，モデルのパラメータを推定する際に，事例の一部を観測された状態にしてサンプリングを繰り返す手法である．モデルの状態数を小さくすることができるので，実用上有効で，広く用いられている．例えば，LDA だと，文書 d のトピック列 $\mathbf{z_d}$ の確率分布を推定するにあたり，普通に考えると T^{N_d} 通り（T はトピック数，N_d は文書 d の単語数）の状態を考慮する必要があるが，これは現実的ではない．そこで，ある単語以外の全てのトピックを観測された状態にしたうえで，その単語の分布からトピックをサンプリングし，次は，別のある単語以外が観測された状態にし，同様のサンプリングを繰り返すことにより，全体の確率分布を推定できる．この方法では，一度に考える状態数は T 通りであり，十分計算可能である．先行研究の Twitter-LDA の場合，1 回のサンプリングで 1 つのツイートのトピックを推定している．

提案手法では，ある確率分布に従って，隣接するツイートのトピックが同じになるかどうかが決まる．ここで，単純に考えるのなら，各ツイートごと，前から順番にサンプリングしていき，前と同じトピックなのであれば前と同じトピックを割り当て，違うのであればサンプリングすれば

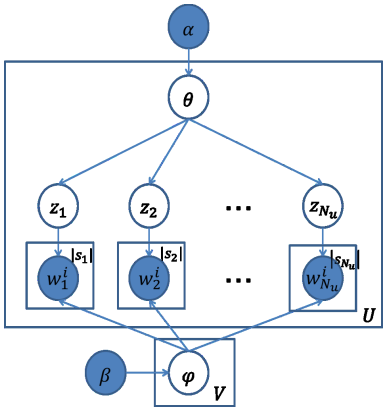

図 2 Twitter-LDA を構成する変数の確率的依存関係

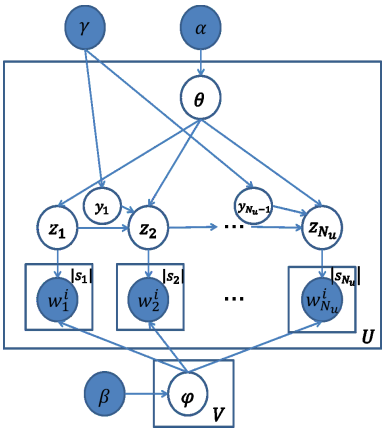

図 3 提案手法を構成する変数の確率的依存関係

良い．しかし，この方法だと，連続して同じトピックが割り当てられるツイート列のトピックを決定するための単語情報として，先頭のツイートしか使えないという問題がある．そこでまず，連続で同じトピックが与えられる箇所を決定した上で，連続で同じトピックが割り当てられるツイートを 1 つにまとめ，まとめたツイート列単位でサンプリングを行う必要がある．

しかし，単純に前から順番にサンプリングしていく方法を用いた場合，図 4 に示すような場合に問題が生じる．図 4 は，n 回目と n+1 回目のイテレーションで，まとめたツイート列が異なる状況を示している．通常ギブスサンプリングでは，現在サンプリングの対象となっている事例が観測されていないものとし，サンプリングを行うが，この場合，n+1 回目で初めの 2 つのツイートがまとまった列のトピックを決めるに際し，どこを観測された状態にしたサンプリングを行えばいいのか自明でない．そこで，実際の計算では次のような工夫をした．

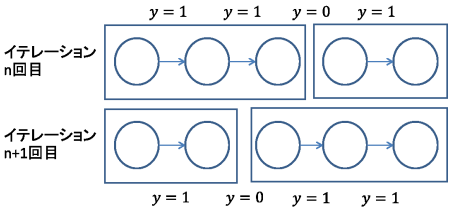

図 4 同じトピックのツイート列がずれた状態

(1) ツイート列を M 個のツイートからなる小区間に分割
(2) 各小区間ごとにまとめてサンプリングを行う

このように，M 個のツイートからなる区間をまとめてサンプリングすることで，どこを観測された状態としてサンプリングを行えば良いのかが明らかとなる．提案手法では，長さの決まった小区間で分割することで，観測された事例の範囲が各イテレーションで統一される．

同一のトピックを引き継ぐツイート列のトピック推定は，以下の式に基づき行う．

$$
\begin{aligned}
& p(z_{ij} \mid \mathbf{z}_{-\mathbf{i}}, \mathbf{S}_{-\mathbf{i}}, \mathbf{y}_{-\mathbf{i}}) \\
& \quad \propto p(\mathbf{s_{ij}}|z_{ij}) p(z_{ij}) \\
& \quad \propto \left\{ \frac{1}{\prod_{k=0}^{|\mathbf{s_{ij}}|-1} (n_t^{(\cdot)} + V\beta - k)} \prod_{v=1}^{V} \frac{\Gamma(n_t^{(v)} + \beta)}{\Gamma(n_{t,-\mathbf{S_i}}^{(v)} + \beta)} \right\} (n_u^t + \alpha)
\end{aligned}
$$

$n_t^{(v)}$ は，トピック t，単語 v の頻度，$n_t^{(\cdot)}$ は，トピック t の単語の頻度，n_u^t はユーザ u，トピック t の頻度を表わす．また，$\mathbf{S}$ は全ツイートの集合，$\mathbf{s_{ij}}$ は i 番目の区間にある j 番目のツイートを表し，$\mathbf{S_i}$ は，i 番目の区間にある全ツイートの集合を，$\mathbf{z_i}$ は，i 番目の区間にあるツイートに割り当てられたトピック列，$\mathbf{y_i}$ は i 番目の区間にあるツイートに割り当てられた，隣り合ったツイートが同じかどうか判定するための変数とする．図 5 に，ツイート列を小区間に分割した様子を図示する．図 5 には，$M = 8$ の状態を示している．

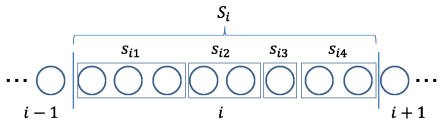

図 5 ツイート列を小区間に分割した様子

4. 実験

4.1 実験設定

実験で使用したデータについて説明する．まず，Twitter のパブリックタイムラインからツイートをランダムサンプリングし，ツイート収集のためのユーザリストを作成した．次に，作成したユーザリストの各ユーザごとに，最大 600 件のツイートを収集した．ここで，対象とするユーザは主に英語を使用するユーザに限定した．次に，全ユーザの全ツイートに含まれる単語の頻度を調べ，頻度の高過ぎる単語と低すぎる単語をストップワードとして除去した．また，ストップワードを除去した後，含まれる単語が 2 語以下の短いツイートを除去した．最後に，全ユーザの全ツイートに対して，URL とハッシュタグを除去し，入力デー

タとした．コーパスの統計情報を，表 1 に示す．

表 1 コーパスの統計情報

全単語数	2,388,883
全ツイート数	348,517
ユーザ数（訓練用）	1,073
ユーザ数（topic coherence 計算用）	120

モデルを構築する際に使ったパラメータとして，前のトピックと同じになる確率 γ は 0.0 から 0.9 まで 0.1 刻みで 9 つの値でモデルを構築した．ここで $\gamma = 0.0$ の場合は提案手法は Twitter-LDA と同じモデルとなる．その他のパラメータはいずれも固定値を使用した．具体的には，$\alpha = 0.01$, $\beta = 0.01$ と，トピック数 T は 250 とした．また，小区間に区切ったツイート列の 1 つの長さ M は，$M = 10$ とした．

4.2 評価

モデルの性能を評価するため，まず，perplexity の計算を行った．perplexity は情報理論的な意味での平均分岐数 (幾何平均) を表しており，言語モデルの良さを定量的に評価する尺度として使用される．perplexity は，値が低いほど良いモデルであることを表しており，以下の式により算出される．

$$perplexity = \exp\left\{-\frac{1}{N_s}\sum_{u}\log p(\mathbf{s_u})\right\}$$

ここで，$\mathbf{s}$ はツイート，u はユーザ，N_t は全ツイート数である．ツイートの生成確率にあたる $p(\mathbf{s})$ は，以下の式により計算される．本研究の提案手法では，ツイート間のトピックに確率的依存関係があるため独立な場合と比べ式が複雑になるが，動的計画法を用いることで効率的に計算できる．

$$\begin{aligned} p(\mathbf{s}) &= \sum_{\mathbf{z}} p(\mathbf{s},\mathbf{z}) \\ &= \sum_{\mathbf{z}} p(z_1)p(s_1|z_1)\prod_{j=2} p(z_j|z_{j-1})p(s_j|z_j) \\ &= \sum_{\mathbf{z}} p(z_1)p(s_1|z_1)p(s_2,...,s_{|s|}|z_1) \end{aligned}$$

前のトピックと同じになる確率 γ ごとの，perplexity の計算結果を図 6 に示す．横軸が，前のトピックと同じになる確率 γ で，縦軸が perplexity を表す．既に述べたとおり，$\gamma = 0.0$ の場合は，ツイート間の依存関係を考えないモデルとなり，Twitter-LDA と同等であることから，以下では，$\gamma = 0.0$ の場合をベースラインと呼ぶ．$\gamma = 0.0$ から γ を大きくしていくと，perplexity が単調減少し，$\gamma = 0.8$ の場合に最も小さい値となった．ベースラインと比べ提案モデルの perplexity は小さい値となっていることから，隣接するトピックを考慮することでより良い言語モデルとなっていることが確認できる．

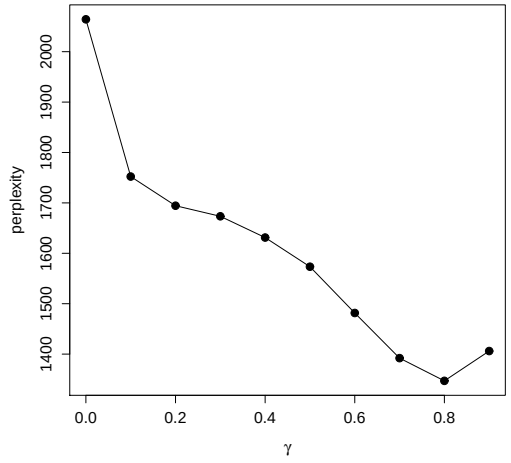

図 6 実験結果 perplexity

しかしながら，提案モデルは，ベースラインモデルで考慮してない隣接するツイートを考慮した言語モデルとなっていることから perplexity が向上するのはある意味，自明であると言え，perplexity が向上したからといって，推定したトピックが良いものとなっていることは言えない．そこで推定したトピック自体の評価を行うため，topic coherence[5], [6] による評価も行う．これは，モデルが適切であるほど，出力されるトピック語は意味的なまとまりを持ち，同一トピック内の単語は共起しやすいことに注目した指標であり，topic coherence が高いほど良いトピックモデルであることといえる．topic coherence は以下の式により計算される．

$$C(t;V^{(t)}) = \sum_{k=2}^{K}\sum_{l=1}^{k-1}\log\frac{D(v_k^{(t)}v_l^{(t)})+1}{D(v_l^{(t)})}$$

ここで，$D(v^{(t)})$ は，トピック t が割り振られた単語 v が出現するツイートの頻度，$D(u^{(t)},v^{(t)})$ はトピック t が割り振られた単語 u と v が共起するツイートの頻度，K は評価に使うトピック語の数である．評価に使うトピック語は，推定確率の高い順に K 個使うこととし，トピック t におけるトピック語の集合を $V^{(t)}$ とする．

確率 γ ごとの，topic coherence の計算結果を図 7 に示す．横軸が，前のトピックと同じになる補正確率 γ で，縦軸が topic coherence を表す．$\gamma = 0.0$ から γ を大きくしていくと，$\gamma = 0.6$ までは topic coherence が単調増加し，$\gamma = 0.6$ が最も大きな値となった．さらに γ を大きくすると，全体的に topic coherence は減少した．この結果から提案手法はベースライン手法である Twitter-LDA よりも，トピックを正しく推定できていると言える．

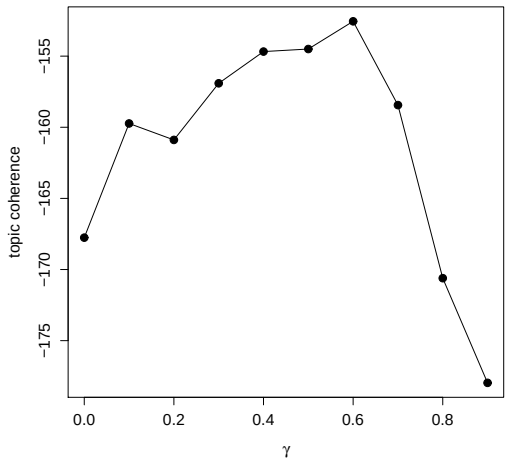

図 7 実験結果 topic coherence

5. まとめと今後の課題

本研究では，隣接するツイート間の関係を考慮したトピックモデルを提案し，実験の結果，提案手法が有効であることを示した．今後，さらに発展させるための方針として，ツイート間でトピックが同じになる確率をツイート間の時間差に基づき算出することが考えられる．たとえば，あるツイートをした 5 分後に次のツイートする場合，同じトピックのツイートである場合が多いと考えられる．一方，あるツイートをして，次にツイートするまでに数日あいてしまうと，仮に隣接するツイートであったとしても，これらのツイートの内容にはほとんど関係がないと考えられる．このため，ツイート間の時間が短いほど同じトピックになりやすいという仮定を導入することで，さらなるモデルの改善が期待される．

参考文献

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[2] Qiming Diao, Jing Jiang, Feida Zhu, and Ee-Peng Lim. Finding bursty topics from microblogs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 536–544, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[3] Amit Gruber, Michal Rosen-zvi, and Yair Weiss. Hidden topic markov models. In *In Proceedings of Artificial Intelligence and Statistics*, 2007.

[4] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM.

[5] David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 262–272, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[6] David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. Evaluating topic models for digital libraries. In *Proceedings of the 10th annual joint conference on Digital libraries*, JCDL '10, pages 215–224, New York, NY, USA, 2010. ACM.

[7] Owen Phelan, Kevin McCarthy, and Barry Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 385–388, New York, NY, USA, 2009. ACM.

[8] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 851–860, New York, NY, USA, 2010. ACM.

[9] Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 111–120, New York, NY, USA, 2008. ACM.

[10] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 338–349, Berlin, Heidelberg, 2011. Springer-Verlag.