DEIM Forum 2020 C8-3(day2 p28)

災害情報抽出のためのReply関係を用いた話題抽出

藤田　俊之†　　小林　亜樹††

† 工学院大学情報学部情報通信工学科　〒 167–8677 東京都新宿区西新宿 1–24–2
†† 工学院大学情報学部情報通信工学科　〒 167–8677 東京都新宿区西新宿 1–24–2
E-mail: †j016241@ns.kogakuin.ac.jp, ††aki@cc.kogakuin.ac.jp

あらまし　文書中の話題検出手法は多く研究され，マイクロブログにおいても先行例がある．文書からの話題検出では Latent Dirichlet Allocation(LDA) 法の利用などが試みられているが，マイクロブログ上では短文であることによる困難さへの対処としての特有のモデルや 1 ユーザ=1 文書化の例が見られる．しかし，1 ユーザでも時間経過により話題は転換していくと考えられるため，本研究では 1 話題に留まることが期待される会話を用いた話題検出を試みる．ここでは，1 会話 1 文書として LDA トピックモデルを構築し，トピック空間上からクラスタリングをし，話題抽出を行うことを目的とする．比較手法として，1 tweet1 文書 モデルや，reply 関係を用いたクラスタ併合モデルを用いる．

キーワード　ツイート，自然言語処理，トピックモデル，災害情報

1 はじめに

災害が発生した際，被災者や，影響が少ない地域の人々も，その災害による被害や環境の変化を取得することで，より柔軟に災害状況に対応することができると考えられる．例として，地震や，その二次災害として，地盤沈下や障害物などでとある経路間の交通が困難になったり，停電などが発生したとする．このとき，これらの状況を示すような情報が得られれば，被災者が要求していると思われる支援物資の予測，被災地に向かう場合や被災者の行動などの災害に対応する際の参考に繋がると思われる．

話題，特に被災情報を得られる情報源の一つとして，マイクロブログサービスの一つである Twitter が挙げられる．Twitter は，140 文字以内の tweet と呼ばれる短文を投稿でき，投稿された tweet に対して，投稿したユーザ自身や，他ユーザが返信を行うことが出来る Reply 機能が提供されており，一つの tweet から，一種の会話を行うように reply のやりとりを行うことができる．また，多くのユーザが利用しており，2017 年末の時点で，3 億 3,000 万人の月間アクティブユーザ数が記録されている [1]．このような短文による性質や，利用者数の多さから，リアルタイムに近い形で，話題や，被災者などによる被災状況の情報が投稿されていると考えられる．災害時の情報インフラとして使われた事例として，2011 年 3 月 11 日に発生した東日本大震災では，避難場所や被災状況の共有，知人の安否確認などをリアルタイムで行う手段として活用された報告 [1] が挙げられる．

しかし，多くの tweet が投稿されるため，特定の話題に関する tweet 群を人手で取得していくのは困難である．tweet の分類方法として，Twitter 側が用意しているハッシュタグがあるが，tweet を投稿するユーザが自発的に付与する必要があり，設定されているのは一部の tweet であると考えられる．また，話題のキーワードを指定して検索を行うこともできるが，そのキーワードが tweet 中に含まれているものしか取得できない．例えば，台風に関する tweet を取得したいとき，台風をキーワードとして検索することで，台風という語が含まれる tweet を得ることが出来るが，台風という語を使わずに台風に言及している tweet は取得することができない．そこで，特定の話題について話している一連の tweet 群を特定のキーワードに頼らずに取得する必要があると考えられる．

そこで，本研究では，一般的なテキスト処理手法である LDA を用いてトピック分析を行い，K-means によるクラスタリングを行う．このとき，tweet 群を reply 関係で結ぶことで，特定の話題について言及していると考えられる tweet を，会話単位でまとめることができると考え，一つの話題に留まることが期待される会話に注目した話題検出を試みる．

1 : http://www.viewproxy.com/Twitter/2018/AnnualReport2017.pdf

2 関連研究

本研究では，Reply 関係に着目して Bag of words(BoW) 化したテキストに対して LDA によるトピック分析とクラスタリングを行い，話題抽出を試みる手法を提案する．

Twitter を利用して，情報の抽出を試みる研究は多く研究されている [2], [3]．James Benhardus ら [2] は，StreamingAPI で取得した tweet から，TF-IDF 法などを用いたトレンドワードの抽出を試みた．湯沢ら [3] は，大規模災害時に，StreamingAPI で取得した tweet から，感動詞との共起関係を利用し，災害に関連する検索語群の抽出を試みた．これらの研究は，tweet を対象として情報の抽出を行っているが，Reply による関係は利用していない．本研究では，Reply の関係に着目した手法を提案する．

マイクロブログ上での LDA を用いたトピック分析は数多く研究されている．LDA [4] は，文書 d に N 個のトピック n が存在すると仮定し，各文書のトピック分布 $\theta_{d,n}$，トピック n における語 w の出現確率 $\phi_{n,w}$ をそれぞれ，N 次元ベクトルを持

つ $\alpha = (\alpha_1, \ldots, \alpha_N)$，総単語数を V として V 次元ベクトルを持つ $\beta = (\beta_1, \ldots, \beta_V)$ パラメータから生成されると仮定する．Dir をディリクレ分布として，式 (1)，(2) で表す．

$$\theta_d \sim Dir(\alpha) \tag{1}$$

$$\phi_k \sim Dir(\beta) \tag{2}$$

tweet に LDA を適用する最も単純な方式は，1 tweet を 1 文書とする手法である．しかし，短文であるが故の問題を指摘されることもあり，1 ユーザの全 tweets を 1 文書として扱う Author-topic model [5] や，1 tweet は 1 トピックであるという仮説に基づく Twitter-LDA モデル [6] といった改良モデルが提案されている．後者の Zhao らは提案した Twitter-LDA モデルに基づき話題を示すキーワードを抽出する手法も提案している [7]．マイクロブログが示す準実時間性を反映したトレンド分析も様々な手法が試みられている．基本的には，古くは Kleinberg が burst と呼んだ [8] 語出現頻度の急上昇を捉えて検出している [9]–[12]．

西村ら [13] は，有名人に関する tweet やその tweet を投稿したユーザのプロフィールから，LDA トピックモデルを用いてトピックの抽出を行い，得られたトピックの比率をもとに人物の類似関係を獲得し，有名人同士の人物関係を可視化した．

これらの関連研究に対して，本研究の目的は一定の時間帯から話題抽出にあり，そのために SNS 特有の利用者同士の会話を活用することが特徴である．

3 提 案 手 法

3.1 処 理 単 位

ここでは，提案手法で文書を処理する基本単位についての定義を行う．

3.1.1 slot

tweet 群を t 分ごとに時間で区切った tweet 集合を slot と呼び，1 処理単位とする tweet 集合を指す．本研究では，災害発生後の tweet より被害状況などの情報を抽出することを目標としている．そこで，処理開始時点を災害発生時刻に置く．ただし，処理の一部で，この開始時時刻以前の 1 slot 分の時間長のデータも用いる．なお，災害発生時刻の厳密な定義は難しいところだが，tweet の分析を目的としているため，分単位でわずかに前倒した時刻を発生時刻と見做し，処理開始時点と設定して問題ない．

災害発生時刻から t 分前の時刻を始点とした tweet 集合を slot_0 とし，時系列順になるように $\text{slot}_0, \text{slot}_1, \ldots$ とする．基本的に，注目している $\text{slot}_i (i > 0)$ における tweet 集合のみを対象として処理を行うが，会話の抽出は，slot_i と slot_{i-1} における tweet 集合を対象として行う．図 1 のように，災害発生時刻から t 分毎に区切られた tweet 集合を時系列順になるように slot_0, slot_1, ... となる．図 1 で，slot_1 に注目した場合，Reply 関係にある一連の tweet である tweet_{11}，tweet_{12}，tweet_{15} と，tweet_{31}，tweet_{32}，tweet_{33}，tweet_{35}，tweet_{36}，tweet_{37}，tweet_{38} がそれぞれ 1 つの会話として抽出される．

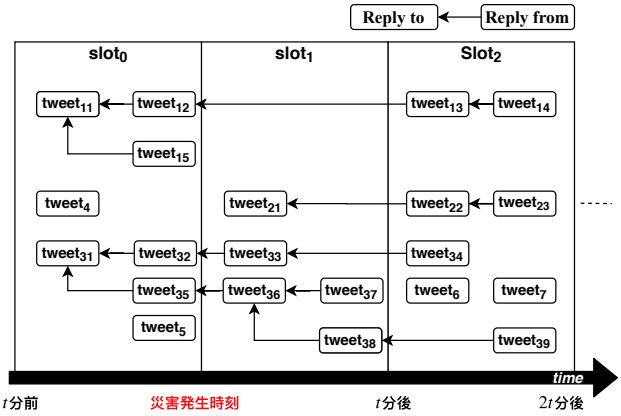

図 1　会話の抽出

3.1.2 会　　話

Twitter では，tweet に対して reply 機能を用いて返信を行うことができる．一つの tweet から連なる reply のやりとりは，一種の会話のように見ることができる．このとき，reply を行った tweet を reply 元 tweet と呼び，reply での参照先を reply 先 tweet と呼ぶ．tweet をノード（頂点）とし，reply 元 → reply 先の有向辺を持つグラフとしてモデル化すると，1 連の会話は根付き有向木になる．このとき，根となるのは reply ではなく単独 tweet として投稿された tweet であり，これを根 tweet と呼ぶ．会話の構造例を図 2 に示す．あるノードから根 tweet まで経由するノード数をそのノードの高さとする．一つの会話に属する各 tweet は，特定の話題について話していると考え，3.2 節では，この会話関係を用いた手法を提案する．

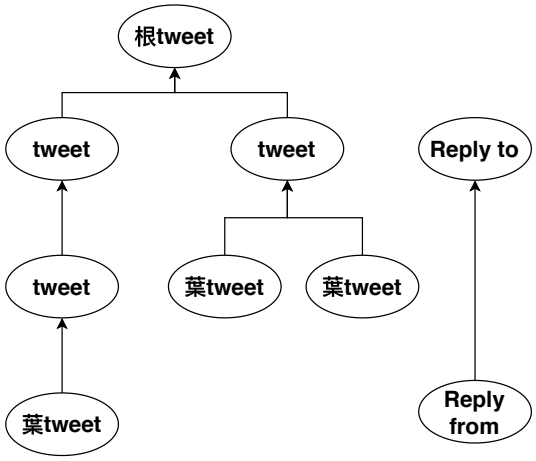

図 2　会話の構造例

3.2 1 会話 1 文書 モデル

会話関係を利用した，1 つの会話を 1 つの文書として扱う 1 会話 1 文書モデルの提案を行う．tweet 群から会話のみを抽出した会話 tweet 集合から，1 会話 1 文書として，bag of words 化した会話 tweet 集合をコーパスとして，トピック数を N とした LDA トピックモデルを構築する．トピックを $n = 1, \ldots, N$ とする．1 会話 1 文書 として，1 文書ごとに得られる N 次元のトピック分布を元に，各文書が持つトピック分布が点在する

N 次元のトピック空間において K-means 法で K 個のクラスタ $k = 1, \ldots, K$ にクラスタリングを行う．

話題の抽出として，3.5 節で，クラスタごとに簡単な要約や代表すると思われる代表語集合の取得することで，話題の抽出をこころみる．代表語集合は，各クラスタに属する語ごとに，トピック分布などを利用した計算を行うことで，選出を行う．

3.3 tweet 集合の取得

Twitter 社の提供する API で，全 tweet の 1%を取得できる StreamingAPI(statuses/sample)[2] で取得した tweet 集合 T_S から，返信先 tweet の ID である in_reply_to_status_id をもとに，指定した tweet_id の tweet を取得できる LookupAPI(statuses/lookup) [3] に対するクエリとして in_reply_to_status_id を指定し，得られた tweet から再帰的に reply 先 tweet を取得して tweet 集合 T_L を得る．このように tweet を取得することで，図 3 のように，StreamingAPI で得た tweet 集合 T_S と LookupAPI で取得した tweet 集合 T_L を合わせることで，会話を含んだ tweet 集合の取得を行う．このとき，日本語 tweet のみを使用するため，tweet の属性である language プロパティが 'ja'，である tweet のみを使用し，他の tweet は破棄している．また，リツイートや，引用ツイートも除外する．

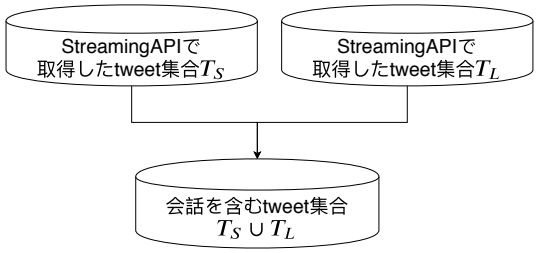

図 3　tweet 集合

3.4 前　処　理

3.4.1 BOT の削除

Twitter には，互いに自動的に生成された tweet で reply のやりとりを行う，BOT 同士とみられるやりとりが存在したため，人間同士のやりとりよりも高い高さの葉 tweet を持つ傾向を利用して除去する．$T_S \cup T_L$ 中の tweet 集合から抽出された会話のうち，高さが 500 以上の葉 tweet を持つ会話に参加した tweet の tweet 元 usr_id を BOT アカウントと見做す．判定された BOT アカウント集合を A_b，$T_S \cup T_L$ 中の BOT アカウントによる tweet を T_b とするとき，以後の分類における処理の対象となる tweet 集合を $T_S \cup T_L - T_b$ とし，BOT アカウントによる tweet を処理対象から除外する．

2：https://developer.twitter.com/en/docs/tweets/sample-realtime/api-reference/get-statuses-sample

3：https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/get-statuses-lookup

あそこ, あたり, あちら, あっち, あと, あな, あなた, あれ, いくつ, いつ, いま, いや, いろいろ, うち, おおまか, おまえ, おれ, がい, かく, かたち, かやの, から, がら, きた, くせ, ここ, こっち, こと, ごと, こちら, ごっちゃ, これ, これら, ごろ, さまざま, さらい, さん, しかた, しよう, すか, ずつ, すね, すべて, ぜんぶ, そう, そこ, そちら, そっち, そで, それ, それぞれ, それなり, たくさん, たち, たび, ため, だめ, ちゃ, ちゃん, てん, とおり, とき, どこ, どこか, ところ, どちら, どっか, どっち, どれ, なか, なかば, なに, など, なん, はじめ, はず, はるか, ひと, ひとつ, ふく, ぶり, べつ, へん, べん, ほう, ほか, まさ, まし, まとも, まま, みたい, みつ, みなさん, みんな, もと, もの, もん, やつ, よう, よそ, わけ, わたし, ハイ, 上, 中, 下, 字, 年, 月, 日, 時, 分, 秒, 週, 火, 水, 木, 金, 土, 国, 都, 道, 府, 県, 市, 区, 町, 村, 各, 第, 方, 何, 的, 度, 文, 者, 性, 体, 人, 他, 今, 部, 課, 係, 外, 類, 達, 気, 室, 口, 誰, 用, 界, 会, 首, 男, 女, 別, 話, 私, 屋, 店, 家, 場, 等, 見, 際, 観, 段, 略, 例, 系, 論, 形, 間, 地, 員, 線, 点, 書, 品, 力, 法, 感, 作, 元, 手, 数, 彼, 彼女, 子, 内, 楽, 喜, 怒, 哀, 輪, 頃, 化, 境, 俺, 奴, 高, 校, 婦, 伸, 紀, 誌, レ, 行, 列, 事, 士, 台, 集, 様, 所, 歴, 器, 名, 情, 連, 毎, 式, 簿, 回, 匹, 個, 席, 束, 歳, 目, 通, 面, 円, 玉, 枚, 前, 後, 左, 右, 次, 先, 春, 夏, 秋, 冬, 一, 二, 三, 四, 五, 六, 七, 八, 九, 十, 百, 千, 万, 億, 兆, 下記, 上記, 時間, 今回, 前回, 場合, 一つ, 年生, 自分, ケ所, ヵ所, カ所, 箇所, ケ月, ヵ月, カ月, 箇月, 名前, 本当, 確か, 時点, 全部, 関係, 近く, 方法, 我々, 違い, 多く, 扱い, 新た, その後, 半ば, 結局, 様々, 以前, 以後, 以降, 未満, 以上, 以下, 幾つ, 毎日, 自体, 向こう, 何人, 手段, 同じ, 感じ

図 4　SlothLib [14] が公開している不要語リスト

3.4.2 不要語の削除

形態素解析エンジンである MeCab[4] を用いて，tweet を形態素解析し，次の 3 条件に当てはまる語を各 tweet から除外する．ここで，不要語リストとして用いた，SlothLib ライブラリ [14] が公開している不要語リスト [5] を図 4 に示す．

- 名詞以外
- SlothLib [14] による不要語リストと，独自に用意した不要語リスト (明日，今日，くん，ん，ー，@，_，の，などの語と絵文字) に該当する語

3.5 代表語集合の選出

各クラスタの代表語集合を選出するために，クラスタに出現する w に対して，score_w を式 (4) のように計算する．クラスタに属する m 個の文書を $d_i (1 \leqq i \leqq m)$ として，$M = \{d_1, d_2, \ldots, d_m\}$ とする．語 w ごとに，トピック n における d_i のトピック分布 $\theta_{d_i,n}$ と，トピック n における語 w の出現確率 $\phi_{n,w}$ を掛け合わせた値に対して，$\mathrm{IDF}(w)$ を重みとして掛けた値を score_w とする．$\mathrm{IDF}(w)$ は，クラスタごとの語集合を W，語 w が含まれる W の数を $\mathrm{df}(w)$ としたとき，式 (3) で計算する．score_w が上位 5 件の語を，クラスタの代表語集合とする．

$$\mathrm{IDF}(w) = \log \frac{N}{\mathrm{df}(w)} \tag{3}$$

$$\mathrm{score}_w = \mathrm{IDF}(w) \sum_{d \in M} \sum_{n=1}^{N} \phi_{n,w} \theta_{d,n} \tag{4}$$

3.6 文書分類の評価

本研究では，クラスタ毎に災害 A に言及した tweet 群，あるいは言及していない文書群と見做して利用者に提示することを想定している．そのため，各クラスタ内の文書が災害 A に言及しているものだけ，あるいはそうでないものだけ，となっていることが望ましい．そこで，クラスタ毎にクラスタ内の文書がいずれのクラス（分類）に属しているかの比率を，災害 A に言及している精度 $\hat{P}$ を用いて表示する．

本稿では全文書を用いての評価が難しいため，両クラス 200

4：https://taku910.github.io/mecab/

5：http://svn.sourceforge.jp/svnroot/slothlib/CSharp/Version1/SlothLib/NLP/Filter/StopWord/word/Japanese.txt

文書ずつを人手によりラベル付けを行い，正解と見做した上で，各クラスタ毎にそれぞれにラベル付けされた tweet 数の比率として評価する．このとき，任意のクラスタにおける推定精度 $\hat{P}$ は，

$$\hat{P} = \frac{T}{T+F} \tag{5}$$

となり，これがクラスタ毎に 0 または 1 にどれだけ近いのかの分布で手法毎の違いを考察する．

3.7 比較手法

ここでは，1 会話 1 文書モデルに対する比較手法について説明していく．1 tweet を 1 つの文書として扱うものを 1 tweet1 文書モデル，1 tweet1 文書モデルのクラスタリング段階で，会話関係でのクラスタ併合を加えたものをクラスタ併合モデルと呼ぶ．

3.7.1 1 tweet1 文書モデル

1 tweet1 文書モデルでは，1 tweet を 1 つの文書として扱う．提案手法と異なる点として，会話を抽出せずに，1slot ごとの tweet 集合から 1 tweet1 文書として LDA トピックモデルを構築する．

3.7.2 クラスタ併合モデル

ここでは，クラスタ併合モデルについて説明していく．クラスタ併合モデルは，1 tweet = 1 文書 モデルにおけるクラスタリングを行うときに，初期クラスタリングである K-means によるクラスタリングと，reply 関係を用いたクラスタ併合から構成される階層的クラスタリングを行う．

初期クラスタリング時におけるクラスタ数 K を，後述のクラスタ併合処理を念頭に一般に用いられるよりも過大な数値を割り当てて実行する．このとき，各クラスタの index を，クラスタ $\mathrm{K}_h(h = 1,\ldots,K)$ とする．初期クラスタリングによるクラスタリングを行ったとき，図 5 のように，クラスタをまたいで reply 関係にある tweet が存在する場合，j 個の tweet からなる reply 元 tweet が属するクラスタを $Q = \{\mathrm{tweet}_{Q,1}, \mathrm{tweet}_{Q,2}, ..., \mathrm{tweet}_{Q,j}\}$，$h$ 個の tweet からなる reply 先 tweet が属するクラスタを $P = \{\mathrm{tweet}_{P,1}, \mathrm{tweet}_{P,2}, ..., \mathrm{tweet}_{P,h}\}$ として，その和集合 $P \cup Q$ を新たに $j+h$ 個の tweet からなるクラスタ Q' とすることで，クラスタの併合を行う．クラスタ併合後のクラスタ数を E として，各クラスタの index を，クラスタ $\mathrm{L}_f(f = 1,\ldots,E)$ とする．

クラスタ併合処理は，Relational Database(RDB) を用いて実装を行う．tweet_id，in_reply_to_id，cluster_id の三つの属性を持つ table を用意する．この table には，取得した tweet 群を格納する．各属性は，表 1 の意味を持つとする．データベース上で，tweet_id が，以下の 2 条件に当てはまり，in_reply_to_id と合致する tweet_id があるとき，in_reply_to_id を reply 元 tweet，reply 元 tweet を tweet_id として持つ tweet_id を reply 先 tweet とする．

- 注目する tweet_id 自身ではない
- 異なる cluster_id を持つ

このとき，reply 先 tweet が属する cluster_id を reply 元 tweet が属する clsuter_id に UPDATE することで，クラスタの併合アルゴリズムを構築する．

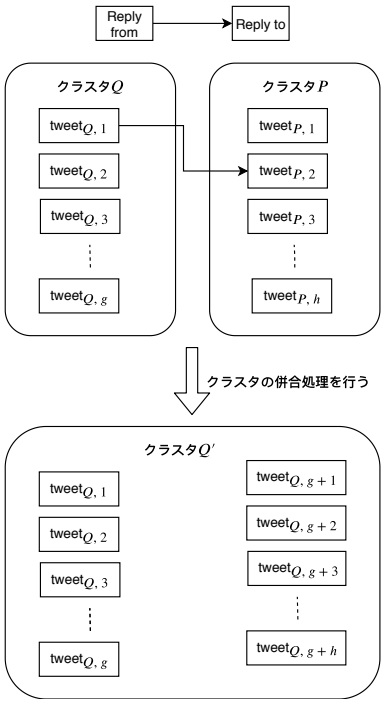

図 5 クラスタの併合処理

表 1 各属性の説明

tweet_id	tweet ごとに一意に振り分けられる ID
in_reply_to_id	reply 先の tweet_id
cluster_id	tweet が属するクラスタ番号

4 実験

4.1 環境

本実験では，形態素解析器，システム辞書，プログラミング言語，ライブラリとして表 2 に示すものを使用した．LDA トピックモデルや K-means 法の実装のために，プログラミング言語である python の，gensim ライブラリ [6] を用いる．トピック数や，クラスタ数以外の各種設定は，デフォルトのまま使用している．

表 2 実験環境

形態素解析器	MeCab 0.996
システム辞書	mecab-ipadic-Neologd(2018 年 05 月 14 日時点)
gensim	2.2.0
python	3.6.9

4.2 条件

本実験では，典型的な災害例として，北海道胆振東部地震と，台風 19 号を対象とするため，各種パラメータを表 3 に示すものとする．台風 19 号の災害発生時刻は，台風 19 号が伊豆半島に上陸したと考えられる時刻とした．各手法において，災害 A に言及している文書，災害 A に言及していない文書がそれぞれ

6 : https://radimrehurek.com/gensim/

表 3　対象とする災害

災害 A	災害名	災害発生時刻
地震	北海道胆振東部地震	2018/9/6 3:08
台風	台風 19 号	2019/10/12 19:00

表 4　各種条件

各手法	1slot の間隔 t	トピック数 N	クラスタ数 K
1 会話 1 文書モデル	60[min]	10	6
1 tweet1 文書モデル			
クラスタ併合モデル			100

同一のクラスタに属するかを確認する．

各手法における 1slot の間隔 t，K-means 法によるクラスタ数 K，LDA トピックモデルにおけるトピック数 N は表 4 に示すものとした．このとき，クラスタ併合モデルにおけるクラスタ数 K を 100 としているが，これは初期クラスタリングにおけるクラスタ数のため，過大な値を設定した．他のパラメータは，事前に試したいくつかの組み合わせのうち，筆者が適切であると思われるものを設定した．

StreamingAPI では，災害発生時刻から 24 時間分の tweet を収集し，tweet 集合 $T_S \cup T_L$ から BOT の削除を行った tweet 集合 $T_S \cup T_L - T_b$ を対象に各手法に基づきクラスタリングを行い，話題の抽出を行った．

4.3　結　　果

4.3.1　北海道胆振東部地震

北海道胆振東部地震を対象とした，災害発生時刻から 24slot 分の，各 slot における会話数は図 6 の通りとなった．1 会話 1 文書，1 tweet1 文書，クラスタ併合モデルの slot_1 における各クラスタの代表語集合，文書数を，それぞれ表 5，6，7 に示す．人手で，各モデルにおけるクラスタリング結果から，各クラスタに属する文書を人手で確認を行い，1 会話 1 文書モデルは 1 tweet1 文書，クラスタ併合モデルよりも地震に言及する文書同士で同一のクラスタとなっている傾向が見られた．この要因の一つとして，1 文書における語量の差が考えられる．図 7 に，slot_1 における 1 tweet1 文書，1 会話 1 文書としたときの語量を比較したグラフを示す．縦軸は，文書数を相対度数で示したものである．図 7 より，1 tweet1 文書では，1 会話 1 文書よりも語量が少ない文書が多い傾向にあることが分かる．この語量の差が，会話を用いてクラスタリングを行うことでより良いクラスタリングができた要因の一つであると考えられる．

4.3.2　台風 19 号

台風 19 号を対象として，精度の計算を行い，表 8 の結果を得た．また，1 会話 1 文書，1 tweet1 文書，クラスタ併合モデルにおける精度を 95%信頼区間でグラフに表したものを図 8，図 9，図 10 に示す．図 8,9 より，1 会話 1 文書モデルは 1 tweet1 文書モデルよりも精度が 0.00, 1.00 に寄っている傾向にあり，他の手法に比べ分類がされていると考えられる．クラスタ併合モデルは，95%信頼区間が 0.00 から 1.00 の内，半分以上占めているクラスタがいくつか存在しており，判断が難しいが，精度が

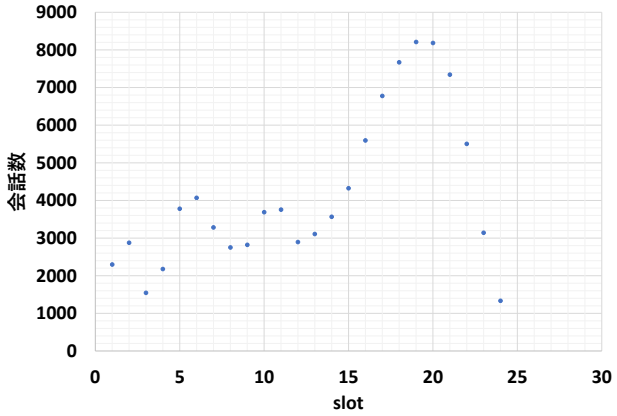

図 6　会話数の推移

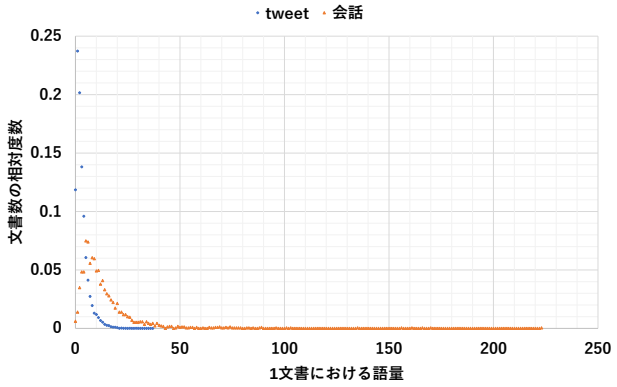

図 7　語量の比較

0.5 に近いクラスタが複数存在していると考えられ，1 会話 1 文書モデルよりも文書の分類がされていないと考えられる．また，クラスタ併合モデルや，1 tweet1 文書モデルでは，文書が全体の半分以上を占めているクラスタが存在しており，そのクラスタも精度が 0.5 に近く，台風に言及している文書と言及していない文書の多くが混ざってしまっていることが分かる．

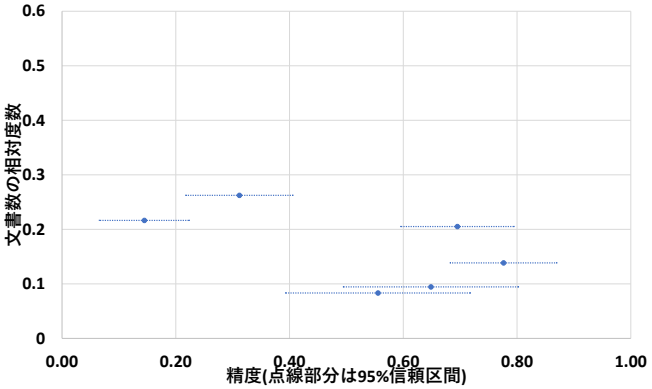

図 8　会話モデルにおける精度分布

4.3.3　クラスタ併合モデル

ここでは，クラスタ併合モデルにおける，併合処理に注目した考察を行う．slot_1 の各クラスタには，似たような印象を受ける語群や，完全に一致する語群が代表語集合として選出された．これは，どのクラスタにも，同じ話題に関する文書が多く含まれたからだと考えられる．一度も他クラスタの併合が行わ

表 5　1 会話 1 文書モデルによる結果

クラスタ	代表語集合					文書数
	1	2	3	4	5	
1	オリオン座	フィギュア	避難経路	残	札幌	258
2	模様	では	発生	とりあえず	札幌	299
3	札幌	全滅	食器棚	ペンライト	全滅	820
4	ちくわぶ	引用	学校	ノシ	お題	617
5	日本	ストーブ	上司	呑気	営業	187
6	腹	犬	ゲーム	家で	エネルギー	117

表 6　1 tweet1 文書モデルによる結果

クラスタ	代表語集合					文書数
	1	2	3	4	5	
1	女子	垢	裏	じしん	最高	3155
2	報	深さ	中東	参戦	参加者募集	13060
3	ニュース	ペンラ	ヤバい	発生	室蘭	6699
4	1 回	1 日	死	事前登録	プラチナオーディションガチャ	1940
5	おはよう	草	ラジお	懐中電灯	前日比	2117
6	ぇ	配信中	キャス	ゆれ	ぶじ	2145

表 7　クラスタ併合モデルによる結果

クラスタ	代表語集合					文書数	併合数
	1	2	3	4	5		
L_1	北海道	震度 6 強	星	綺麗	揺れ	63	1
L_2	台風	最後	笑	びっくり	仕事	75	1
L_3	北海道	無事	心配	笑	揺れ	10858	29
L_4	北海道	無事	心配	揺れ	さ	373	1
L_5	北海道	無事	心配	笑	充電	4516	23
L_6	北海道	無事	笑	心配	揺れ	852	2
L_7	北海道	無事	心配	笑	充電	2600	11
L_8	北海道	無事	心配	笑	充電	5446	20
L_9	北海道	無事	品ぴ	笑	びっくり	2874	5
L_{10}	北海道	無事	心配	笑	揺れ	1459	7

表 8　各手法における精度

		クラスタ									
		1	2	3	4	5	6	7	8	9	10
会話モデル	文書数	907	1509	1029	2859	2235	2358	-	-	-	-
	T	20	59	24	29	57	11	-	-	-	-
	F	16	17	13	64	25	65	-	-	-	-
	精度	0.56	0.78	0.65	0.31	0.70	0.14	-	-	-	-
単 tweet モデル	文書数	35508	7791	5820	6480	9404	7722	-	-	-	-
	T	64	33	16	36	28	23	-	-	-	-
	F	123	12	11	8	29	17	-	-	-	-
	精度	0.34	0.73	0.59	0.82	0.49	0.58	-	-	-	-
クラスタ併合モデル	文書数	511	39431	7350	678	2242	11852	4765	4850	373	673
	T	1	86	18	3	8	43	16	20	2	3
	F	2	114	22	5	34	13	8	0	2	0
	精度	0.33	0.43	0.45	0.38	0.19	0.77	0.67	1	0.5	1

れていない，併合数が一つのクラスタが，各 slot に見られる．この，各クラスタの併合数の偏りによって，各 slot において，文書数が最も多いクラスタと，文書数が最も少ないクラスタで大きな差が表れたと考えられる．

また，災害発生時刻から 24 時間分の，24slot の各 slot の併合後のクラスタ数を，表 9 に示す．最大クラスタ数は，12，最小クラスタ数は 10 であった．1slot の間隔を 60 分としたとき，概ねクラスタ数は 11 前後となる傾向があることが分かる．

次に，$slot_1$ に注目して，併合後のクラスタと，クラスタに併合された，併合前クラスタについて，併合後クラスタに属する，

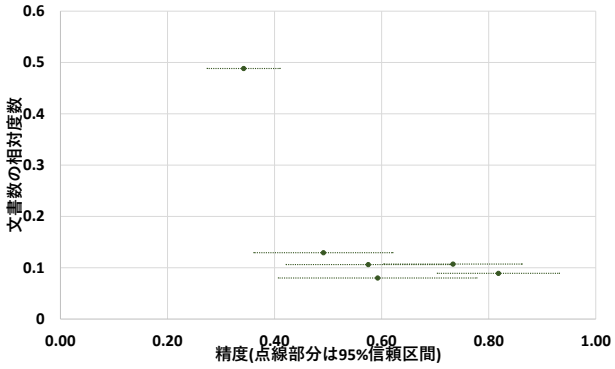

図 9　単 tweet モデルにおける精度分布

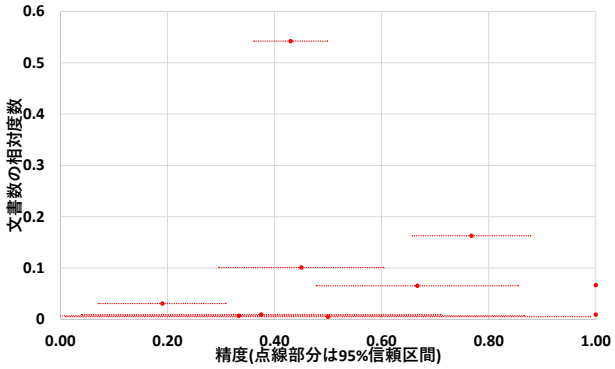

図 10　クラスタ併合モデルにおける精度分布

表 9　各 slot のクラスタ数

slot	クラスタ数	slot	クラスタ数
1	10	13	10
2	11	14	10
3	12	15	12
4	10	16	10
5	10	17	10
6	10	18	10
7	11	19	10
8	10	20	11
9	10	21	10
10	12	22	10
11	10	23	12
12	10	24	10

初期クラスタリングにおけるクラスタに属する文書が，地震に言及しているか，言及していないかどうかを確認した．このとき，表 7 において，クラスタの併合が行われていないクラスタ L_1，クラスタ L_2，クラスタ L_4 は確認を行う対象から除外した．併合後クラスタ L_6, L_9, L_7 に併合されていたクラスタは，それぞれ地震に言及している文書が多いクラスタ同士，少ないクラスタ同士で併合されていた．クラスタ L_6 におけるクラスタ K_3, K_{87} に属する文書例を表 10 に示す．文書例では，メンション先である@に続くユーザ名は■としている．また，URL 部分は URL としている．併合後クラスタ L_3, L_5, L_8，L_{10} に併合されていたクラスタは，地震に言及している文書が多いクラスタと少ないクラスタ同士が併合されていた．一例として，L_{10} として併合されたクラスタ K_9, K_{90}, K_{94}, K_{42}, K_{63}, K_{53}, K_{47} が挙げられる．L_{10} に併合された各クラスタの文書例を表 11 に示す．クラスタ K_9, K_{94}, K_{42}, K_{63}, K_{53}, K_{47} では，地震に関連する tweet がほとんどであったが，クラスタ K_{90} では，地震に言及する tweet も見られたが地震に関連しない tweet が多く見かけられた

4.3.4　ま　と　め

マイクロブログ上でトレンドと言われる様な流行が発生している場合，当該話題に関する投稿量は，他の話題に関する投稿量を凌駕していると考えられる．したがって，K-means 法のような手法でのクラスタリングは不適である．提案手法である 1 会話 1 文書モデルでは，単純に 1 tweet = 1 文書 とするよりも，災害 A に言及している文書と言及していない文書がクラスタ間で分かれる傾向にあった．

クラスタ併合モデルでは，テキストによる細分類となる初期クラスタリングを行った上で，会話関係を利用して関連するクラスタを併合している．これは，reply の存在を利用者一種の階層化クラスタリングであるとも捉えられ，クラスタ間，すなわち，話題間での投稿量の差異を吸収できる可能性を示した．北海道胆振東部地震では，大地震というドミナントな話題でもなんらかの分割が得られると言え，得られたいくつかのクラスタでは同一話題に閉じていることが観測された．一方で，多数のクラスタが併合された巨大クラスタでは，異なる話題が混在している状況も観察された．

5　お わ り に

本論文では，マイクロブログ上でのトレンドに関するような話題を抽出することを目的に，マイクロブログの特徴でもある会話関係を利用した 1 会話 1 文書モデルによるクラスタリングや，比較手法として階層的なクラスタリング手法を取り上げた．Twitter での災害発生時を対象とした実験では，一定の効果が見られることがわかったが，改善すべき点も多々存在することを確認した．先行研究のトピック分析の改良手法の採用を考慮することや，その他の属性を用いたクラスタリングの改善，精度 $\hat{P}$ による評価を，北海道胆振東部地震に対しても行うこと，比較手法として，ユーザごとに発言された tweet を 1 文書として扱う 1 ユーザ=1 文書モデルの追加，測定項目の統一化に取り組んでいく予定である．

文　　献

[1] 河井 孝仁，藤代 裕之，“東日本大震災の災害情報における Twitter の利用分析”，広報研究 = Corporate communication studies，Vol.17，pp.118–128, 2013.
[2] James Benhardus，Jugal Kalita，“Streaming trend detection in Twitter”，Int. J. Web Based Communities, Vol. 9, No. 1, pp. 122–139, 2013.
[3] 湯沢昭夫，小林亜樹 “マイクロブログにおける感動詞との共起を利用した検索語の抽出”，情報処理学会論文誌データベース (TOD) ，Vol. 12, No. 3, pp. 1–17, 2019.
[4] David M. Blei, Andrew Y. Ng, Michael I. Jordan, “Latent Dirichlet Allocation”Journal of Machine Learning Research 3, pp.993–1022, 2003.
[5] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths.

表 10　tweet 例 1

slot	併合前クラスタ	tweet 本文
1	K_5	停電してる地域結構あるみたいですね…
		@■ 停電しますた笑
		@■ よかったです！家族がどうしてるか心配 w
	K_{87}	北海道にいる友達が停電して TV 見れないから状況も何掴めなく困ってる。携帯充電できんそんなに使え外避難した方がいいか聞かれたけど、どうしたらいんやろ？#震度 6 強地震
		@■ 最初気付かんかったけど信号無くて気付いた
		あ、生きてます…めっちゃ怖かった

表 11　tweet 例 2

slot	併合前クラスタ	tweet 本文
1	K_9	ご当地は震度 5 強。マンションが斜めになったの分かった免震構造重要
		震度いくつ !?
	K_{90}	日清のカップヌードル、またはシーフードヌードルをもらおう！その場で当たりがわかる即時抽選合計 2,4000 名様にプレゼント。URL
		[3 位：ホーム＆キッチン] アイリスオーヤマ扇風機タワーファン \4,158 URL
	K_{94}	ウチも停電中。窓の外真っ暗こんな初めてだ
		と思ったけどまた停電したでこれ…アカン
	K_{42}	@■ 今日いる主任誰も知らなかった…
		最近、長時間眠れない昼寝すると深夜に目がさえる
	K_{63}	北海道電力全部落としたな!
		まだ、微妙に揺れるの怖すぎ
	K_{53}	@■ 大丈夫だけど停電してる。
		自信やばい！おおいきい
	K_{47}	停電だしめっちゃ揺れて家具落ちまくる
		北海道民大丈夫ですか!!??

Probabilistic author-topic models for information discovery. Proc. of SIGKDD 2004, 2004.

[6] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Compareing twitter and traditional media using topic models. Proc. of ECIR 2011, 2011.

[7] Wayne Xin Zhao, Jing Jiang, Jing He, Yang Song, Palakorn Achananuparp, Ee-Peng Lim, and Xaoming Li. Topical keyphrase extraction from twitter. Proc. of The Anuual Meeting of the Association for Computational Linguistics 2011, pp. 379–388, 2011.

[8] J. Kleinberg. Bursty and hierarchical structure in streams. Proc. of SIGKDD2002, pp. 1–25, 2002.

[9] M. Cataldi, L. Di Caro, and C. Schifanella. Emerging topic detection on twitter based on temporal and social terms evaluation. Proc. of MDMKDD '10, pp. 4:1–4:10. ACM, 2010.

[10] Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on twitter. Proc. of ICWSM 2011, 2011.

[11] Edward Benson, Aria Haghighi, and Regina Barzilay. Event discovery in social media feeds. Proc. of ACL 2011, pp. 389–398, 2011.

[12] 中島伸介，張建偉，稲垣陽一，中本レン：大規模なブログ記事時系列分析に基づく流行語候補の早期発見手法，情報処理学会論文誌データベース（TOD），Vol.6, No.1, pp.1–15 , 2013.

[13] 西村章宏，土方嘉徳，三輪祥太郎，西田正吾，“一般ユーザの観点に基づく Twitter からの人物関係の可視化と事例の考察”，情報処理学会論文誌 ，No. 56，Vol. 3，pp. 972–982，2015.

[14] 大島裕明, 中村聡史, 田中克己，“SlothLib:Web サーチ研究のためのプログラミングライブラリ”, 日本データベース学会 Letters, 6, 1, pp. 113–116, 2007.