DEIM2022 E23-1(day2 p25)

災害情報採集のための試作システムの提案と評価

藤田　俊之†　　小林　亜樹††

† 工学院大学大学院電気・電子工学専攻　〒 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: †cm20046@g.kogakuin.jp, ††aki@cc.kogakuin.ac.jp

あらまし　リアルタイムに災害情報を得るために，Social networking service (SNS) の 1 種である Twitter が注目されている．Twitter を情報源として自然言語処理を行う際，1 投稿単位である tweet を 1 処理単位である文書とする方法が考えられる．しかし，tweet は 140 文字の制限があり，短文性などの問題が指摘されている．この問題に対し，筆者らは reply-chain によりもたらされる reply-tree を会話木と定義し，1 会話木を 1 文書とすることで話題の連続性と自然言語処理の問題解決を提案してきた [1]．しかし，1 会話木中にも話題の遷移が見られることがあり，採集誤りを引き起こすことがわかった．そこで，本稿では細かい粒度での災害情報採集のために，会話木の部分を指す会話 window の導入を提案する．この会話 window を分類時に活用することで，より細かい粒度で会話を分析し，災害言及 tweet の採集を目指す．また，準リアルタイムでの採集を行うため，リアルタイム性を損なわない範囲の tweet 集合から教師なし分類器を用いたトレーニングデータの抽出も行う手法を提案する．これらの内容を元にした試作システムを用いて，実 tweet を対象とした評価実験を行い，会話 window と提案手法の有効性を示す．

キーワード　会話 window, 会話木，災害情報，自然言語処理

1 はじめに

Twitter は災害時に災害状況を知るための情報源として活用されており [2]，tweet から災害状況の分析を試みるような研究も多くされている [3]–[6]．しかし，tweet の短文性や人による基準の違いなどから従来の自然言語処理を適用する際の難しさが指摘されている [7]–[10]．

そこで，目的文書採集のための分類器における処理対象文書となる単位を，投稿アカウント全体とするなど文章量を増加させる手法が提案されてきた [8] [9]．このとき，処理対象文書は一つの話題にのみ言及していることが望ましいと言え，そのため，tweet に対する返信機能である reply でつながれた一連の tweet を会話と見做し，この会話を処理対象文書とする converstaion model [10] へと発展している．しかし，これらの手法では，分類器の学習時にしか会話単位での処理を導入しておらず，分類や話題抽出などは tweet 単位での処理となっており，複数 tweet によって意味をなすような場合を見落とす可能性を指摘できる．

これに対し筆者らは，災害時に投稿される tweet は単独では断片的で内容の解釈，分類処理が難しい場面があることを見出し [11]，処理対象文書を会話単位とすることを分類器の学習時のみばかりでなく，分類，採集時にも適用する，1 会話 1 文書モデルによる災害情報採集方式を提案してきた [1]．本方式では，tweet 単位での分類では見落とされるような tweet を会話としてまとめて採集できる良さがある反面，1 会話中でも話題遷移が生じるための，災害とは無関係の tweet 群を採集してしまう問題が生じる．

そこで本論文では，Twitter 上の投稿から災害情報の確認を行う人の負担軽減を目指して，災害に言及した投稿を会話の特性を考慮して準リアルタイムに，かつ，特段の学習用データの用意なく逐次自動的に採集できる方式を提案する．会話は，会話中での話題の遷移がある一方，一見，災害への言及度合いが少なそうに見える部分も，全体での意味疎通には必要な場合もあったりするなど人による判別も難しい場合も存在する [12]．そこで，会話中で災害言及部分が確実に存在することを識別するために会話 window 単位での処理を導入し（3. 1. 2 節），しかし最終的な出力は，会話単位での採集とする．会話 window は，文書内の単語数を増加させ分類をしやすくする一方で，話題の遷移の影響を軽減できる仕組みである．

また，提案手法では会話の言及度を求めるために，単純な代表語による災害情報の見逃しの可能性を考え語を指定した判別ではなく，教師あり分類器による出力結果を利用する．このとき，教師あり分類器に用いる教師データのために災害の主題を伝えるための 1 語だけは入力するが，以後は自己教師ありとして自動的に経時変化を捉えて学習し続けるようになっている．

2 関連研究

本研究では，会話の部分を 1 文書とすることで最終的な分類精度の向上を考える．この会話を活用した研究はいくつか見られる [10] [13]．Othman ら [13] は，製品に対する意見の抽出のために，時系列的に前の表現に依存する表現である Anaphora [14] の解決のために reply 関係を辿る手法を提案した．また，会話を LDA などの学習時に用いることで分類精度の向上の報告 [10] も見られる．これらから，Reply 関係を考慮することでより正確な処理を行うことができる可能性が考えられる．

また，本研究では準リアルタイムで災害時に投稿された tweet

群から災害に言及している tweet を会話単位での採集を試みる．本研究が目的としている災害に言及している tweet の採集には，検索語を含むかどうかで 1 tweet 単位で採集する方法が一般的である [3], [4]．検索語自体を拡張する研究も行われており，湯沢ら [6] は，感動詞との共起関係を利用し，リアルタイムに近い形での災害に関連する検索語群の抽出を試みた．

また，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 [15] 語出現頻度の急上昇を捉えて検出している [16]–[18]．また，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 [19]．James ら [20] は，Streaming API で収集した tweet 集合から単位時間毎に TF-IDF 法などを用いたトレンドワードの抽出を行った．

このように，検索語の拡張や検索語自体の候補としてはこれらのトレンド分析によって得られるトレンドワードを用いるなど様々考えられる．一方，検索語による災害情報の採集では，検索語が含まれていない tweet を見逃してしまう可能性が考えられる．そのため，本研究で提案する手法では， tweet 群を直接分類することをベースとするために，教師あり分類器を用いる．一般的に教師あり分類器には教師データが必要だが，準リアルタイムでの採集を行えるように，採集対象とする tweet よりも過去に投稿された tweet 群から機械的に教師データを作成する．このとき，検索語に相当する災害を代表するような 1 語の代表語を用いるが，あくまでも教師データの作成時のみに利用し，直接分類時には用いない．

マイクロブログ上での Latent Dirichlet Allocation (LDA) [21] などのトピック分析は数多く研究されている．tweet に LDA を適用する最も単純な方式は，1 tweet を 1 文書とする文書モデルの活用である．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 ユーザの全投稿を 1 文書として扱う Author-topic モデル [22] 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 [7] といった改良モデルが提案されている．しかし，同一ユーザの投稿する tweet には，時間の経過による話題の転換が考えられる．また，学習時に一連の reply のやりとりを 1 会話と見做し 1 文書として扱うことで， 1 tweet を 1 文書として扱う文書モデルや同一ハッシュタグの tweet を 1 文書として見做す文書モデル [23] などより分類精度の向上が見られた報告 [10] がある．これに対し，会話は会話としてユーザに提示することを念頭に分類時にも会話を用いることを提案した [1]．

会話内においても，reply のやりとりが続くうちに，同じ会話内においても話題が転換していく場合があり，これに対して本研究では会話をより細かい粒度で分析するための会話 window を導入することで，会話内の話題の転換に対応することを提案する．

3 提案手法

本研究の目的は，投稿された多数の tweet から災害に言及している tweet を 3.1.1 節で定義する会話木単位で採集することである．ここで採集とは，投稿を会話木単位にまとめ，会話木単位に災害への言及度合いを推定，一定の閾値を超える会話木のみを抽出する処理のことを指す．このとき，準リアルタイムで逐次的に処理可能で，特段の学習用データを要しないことを制約条件とする．

この採集動作を実現する提案手法は，図 1 に示す構成である．この提案手法は，大きく分けて教師データ作成部と言及度算出部の 2 つの部から構成される．本方式では会話単位にまとめた投稿について，災害への言及度と呼ぶ投稿内容の言及度合いの指標を算出することで，採集を実現する．この部分を言及度算出部と呼び，基本的な構造は教師あり分類器によって構成される．そのため，この分類器を適切動作させるための教師データが必要であり，これを自動的に生成し，全体として自己教師あり分類器として動作させるための部分が教師データ作成部である．なお，この 2 ついずれにも共通する前処理については 3.2 節で説明する．

災害に言及している会話木の採集では，会話中の各所に災害関連語が散在するなどして，全体としては災害に言及していないにも関わらず誤分類される場合が考えられる．そこで，会話中の話題遷移を捉えて，会話の一部分で十分に災害に言及しているような会話を採集できるようにするため，会話木の部分を指す会話 window という概念を導入する（3.1.2 節）．会話 window 単位で災害への言及度合いを算出した上で，会話毎の採集に反映させる．

この言及度算出部のための教師データ作成部は，時間経過に伴う災害状況の推移などによって変化する投稿の話題に追従しつつ準リアルタイムで動作する条件を満たすように，一定時間間隔で教師データ作成を随時その直前期間の投稿データから作成する仕組みとする．これらの手順については 3.3 節で述べる．

生成された教師データを用いて，言及度算出部の分類器は一定時間間隔で新しい分類基準をもつに至る．この分類器出力を用いて，採集対象とする会話木の災害に言及している度合である言及度を推定する．言及度推定のために，教師データ作成部によって得られた教師データをもとにし学習を行った教師あり分類器を用いる．分類対象は，会話内の会話 window 単位であり，多数の会話 window の言及度を処理して会話毎の言及度を推定する．この推定結果が提案手法における実質的な出力であり，一定の閾値を超える言及度を持つ会話を利用者に提示することで，利用者の災害情報採集を支援する．これらの手順は 4.1 節で説明する．

3.1 会　　話

3.1.1 会 話 木

Twitter における投稿は tweet と呼ばれる．Tweet には，他の（一つの）tweet を「返信先」として指定する機能が存在し，In-Reply-To-Status-Id 属性と名付けられている．In-Reply-To-Status-Id 属性に値のある，すなわち，他の tweet への「返信」である tweet に対しても，さらに別の tweet における In-Reply-To-Status-Id 属性の対象先 tweet として指定するこ

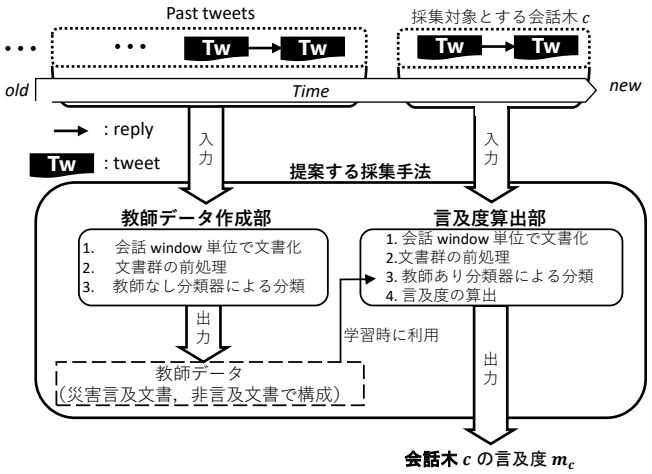


図 1: 提案手法の概要図

とは制限されない．したがって，多数の tweet が In-Reply-To-Status-Id 属性をポインタとして，数珠つなぎのように「返信」（reply）で接続された状況が存在する．Tweet 毎の投稿主に特に制約はなく，例えば，2 者が交互に reply する会話的用法などが観測される．

このような一連の reply のやりとりのうち，時間的に古い側を reply 元 tweet, reply 元 tweet に対して reply した tweet を reply 先 tweet とする．すなわち，reply 元 tweet に対して，reply 先 tweet の In-Reply-To-Status-Id 属性が reply 元 tweet の識別子を保持する関係性である．Reply 元 tweet と reply 先 tweet が連鎖している一連の tweet 群を一つの会話と呼ぶことにする．この会話は，tweet をノードとし，reply 元 → reply 先の reply 関係を有向辺として持つグラフをモデル化すると，一般に根付き有向木となる．これを会話木と呼ぶ．ここで，会話木の辺の向きは時系列に沿った向きである．会話木の条件は，会話木 C の頂点（tweet）集合 $T = V(C)$ とするとき，その要素数を用いて，$|V(C)| \geqq 2$ とし，reply 関係を持たない tweet は会話木と呼ばないこととする．

Tweet t の持つ属性を表 1 に示す．text は tweet 本文，id は Twitter 社による tweet ID，timestamp は投稿時刻，account は投稿アカウント ID である．会話を構成する tweet は，reply 元 tweet の id を In-Reply-To-Status-Id 属性を示す inreplyto 属性に保持する．したがって，会話を構成しない単一 tweet や，会話木の root となる tweet についてはこの属性の値は null とする．

4 つの tweet で構成される会話木の構造例を図 2 に示す．左下の tweet t_4 の inreplyto 属性の値は，t_2 の id となっており，直接の親にあたる reply 元 tweet を指す構造である．Inreplyto 属性の値は会話木の root を指さず，また，この一種のポインタの向きが会話木定義の辺の向きと逆向きである点に注意を要する．

3.1.2 会話 window

会話 window は，会話木内の近傍関係にある一定範囲の tweet を指す．本論文では，会話木におけるエッジ（reply 関係）を距離 1 とし，各 tweet から k-近傍に含まれる tweet 群を会話 window と呼ぶ．以後，この k を会話範囲段数 r と呼ぶことと

表 1: Description of the attributes of t.

Notation	Meaning
id	Tweet id
timestamp	Tweet の投稿時刻
account	Tweet を投稿したユーザの ID
inreplyto	親である reply 元 tweet の id

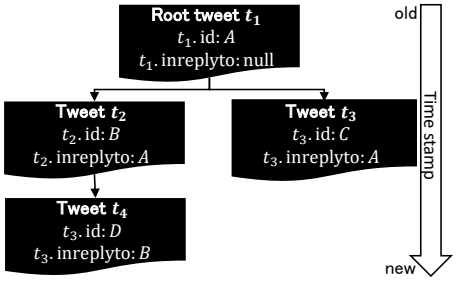


図 2: 会話木の構造例

する．本研究では，分類器等での扱い上，この会話 window に含まれる tweet を連結した文字列を 1 文書として扱う．

この会話 window は，会話範囲段数 r をパラメータとして r の値によって例えば次のような意味を持つ．

- $r = 0$: tweet 単位がバラバラに文書として処理される．
- $r = 1$: 各 tweet 自身とその reply 元 tweet, reply 先 tweet を 1 文書とする．
- $r = \infty$: 会話木全体が 1 文書としてまとめて処理される．

3.2 前 処 理

教師データ作成部，言及度算出部の双方で共通となる前処理について説明する．前処理は，分類器の動作で不具合が起きないようにする文字列処理とデータ化全般を指し，tweet の連結による文書生成，Bag of words (BoW) 表現への変換，Bot account の検出と除外，不要語の削除などである．

3.2.1 Bot account の検出と除外

Twitter 上に投稿される tweet には，自動的に生成された tweet から reply のやりとりを行う bot account が存在する．このような bot account が投稿する tweet は，機械的に生成されたもので災害情報を採集する目的ではノイズになるため，bot account による投稿は除外したい．そこで，bot account 同士による会話は，人同士の会話よりも長く reply のやりとりが続く会話が発生する傾向があることを利用し，bot account を検出し，当該アカウントによる投稿は処理対象から除外する．

具体的には，適当な閾値回以上の回数 reply のやりとりが発生していることをもって，当該投稿元 account を bot account と判定する．本論文では，この閾値に 500 回を用いる．ここで判定された bot account により投稿された tweet は以後の処理対象から除外する．

3.2.2 文書ベクトル

会話木を会話 window 単位に（重複させながら）分割したものを 1 処理単位として，これを文書 d と呼ぶ．文書はその後の処理に備えて，その本文を形態素解析器である MeCab[1]で名詞,

1：http://taku910.github.io/mecab/

形容詞，動詞と判定された語のみの Bag of Words (BoW) へと変換する．このとき，記号，絵文字は除外し，各語は原形で用いることとする．また，ひらがな（あ – ん）だけを含む語のうち，二文字以下の語は除外する．BoW 文書ベクトル v_d は，この BoW 表現から，成分となる各単語を指す値に TF-IDF 値を用いたベクトルとして構成する．TF-IDF 値には，3.3 節における教師データ作成部への入力とする複数の会話木から会話 window によって得られた文書群をコーパスとする．

3.3 教師データ作成部

言及度算出部の分類器を学習するために用いる教師データを作成する部分である．本方式では LDA 分類器（3.3.1 節），Twitter-LDA 分類器 (3.3.2 節) の 2 種の教師なし分類器を用いて教師データを生成する．手順はおおまかに次の通りである．

（1） まず，直近一定時間の過去に投稿された tweet 群から，会話 window 単位の文書集合を得て，LDA 分類器（3.3.1 節），Twitter-LDA 分類器 (3.3.2 節) の入力とする．ここで，LDA 分類器，Twitter-LDA 分類器は入力文書を二値（災害言及, 非言及）に分類する分類器である．

（2） 両分類器による判定結果が同一となった文書とその分類結果だけを用いて，教師データとして用いる文書集合を得る．

3.3.1 LDA 分類器

LDA 分類器は，入力文書を二値分類すること目的とし，以下に示す 5 step で構成される．

（1） 全ての文書のベクトル v_d を LDA の入力データとし，LDA トピックモデルを得る．

（2） 得られた LDA トピックモデルにより，文書 v_d に対するトピック分布ベクトル t_d が対応づけられるため，全ての v_d を対応する t_d に変換したトピック分布ベクトル集合を得る．

（3） 全 t_d をトピック空間において，k-means 法でクラスタリングする．

（4） 得られたクラスタ群から 3.3.3 節による代表語割合アルゴリズムを用いて，文書ごとの判定結果を得る．

（5） こうして得られた文書 d 毎の災害言及か災害非言及かの判定結果をこの分類器の出力結果とする．

3.3.2 Twitter-LDA 分類器

Twitter-LDA 分類器は，3.3.1 節の step 1, 2 において，LDA ではなく Twitter-LDA モデル [7] を用いる．Twitter-LDA ではユーザ u 毎にトピック分布 θ_u を持つと仮定しているが，本論文では，これを会話 window 単位になるよう変形して用いる．すなわち，Twitter-LDA モデルにおける 1 文書単位がユーザ u による全 tweet 集合であるのに対して，会話 window によって文書化される会話木の部分木 s を 1 文書として処理する．したがって，この部分木 s 毎にトピック分布 θ_s が得られる．このトピック分布 θ_s を 3.3.1 節の step 3 における t_d と置き換え，以後の処理は 3.3.1 節の Step 4, 5 と同じように行う．

3.3.3 代表語割合アルゴリズム

代表語割合アルゴリズムは，人手で指定された災害を代表するような一語を用いて，一定の文書集合が災害に言及しているか否かを 2 値判定するアルゴリズムである．この人手によって設定する一語は代表語 w と呼ぶ．また，ここまでで得られたクラスタに含まれる文書を判定対象の文書集合とする．したがって，代表語 w を用いて災害言及か非言及文書でまとまっているクラスタの選択を自動的に行うことになる．

手順はまず，k-means で得られた i 番目のクラスタに含まれる文書集合を $D(i)$ とするとき，クラスタ i の代表語割合 p_i を

$$p_i = \frac{|D_w(i)|}{|D(i)|} \tag{1}$$

として定義する．ここで，

$$\begin{cases} D(i) & = \{v_d \in i\text{−th cluster}\} \\ D_w(i) & = \{v_d \in D(i) \mid w \in W(v_d)\}, \end{cases} \tag{2}$$

ただし，$W(v_d)$ は，文書ベクトル v_d に含まれる単語集合を示す．k 個のクラスタそれぞれに対応する代表語割合のうち，最小値 $p_{\max}$, 最大値 $p_{\min}$ を

$$\begin{cases} p_{\max} = \max\{p_i\} \\ p_{\min} = \min\{p_i\} \end{cases} \tag{3}$$

としたとき，各 p_i を (4) 式のように max–min normalization 化した値を ρ_i とする．

$$\rho_i = \frac{p_i - p_{\min}}{p_{\max} - p_{\min}} \tag{4}$$

別に定める閾値 β_0, β_1 を用いて，この ρ の値でクラスタの判定を行う．すなわち，

$$\text{クラスタ } i\text{: } \begin{cases} \text{災害言及} & \rho_i > \beta_1 \\ \text{災害非言及} & \rho_k < \beta_0 \\ \text{決定しない} & \beta_0 \leqq \rho_k \leqq \beta_1 \end{cases} \tag{5}$$

と判定する．

3.3.4 教師データの構成

災害言及，災害非言及と判定されたクラスタ内に属する文書は，それぞれそのクラスタの判定と同一の判定とする．一方，決定しないとしたクラスタに属する文書は全て破棄する．

このようにして，文書（会話 window）単位で，両方の分類器による災害言及，非言及の判定が行われる．教師データは，両方の分類器で一致した判定となった会話 window のみをその判定（災害言及，または，非言及）であるとして構成する．

4 言及度算出部

ここでは，採集対象とする会話木を入力として，会話木単位で災害に言及している度合である言及度を推定する．言及度の推定時には，3.3 節によって得られた教師データを元に学習した 4.1 節に示す教師あり分類器を用いる．

言及度算出部の概要について，言及度算出例である図 3 を用いて説明する．まず，採集対象とする会話木 c を会話 window 単位で文書として切り出しを行い文書群 $D(\ni d_i)$ を得る．次に，文書群 D を教師あり分類器の入力とし，文書 d_i 毎に文書

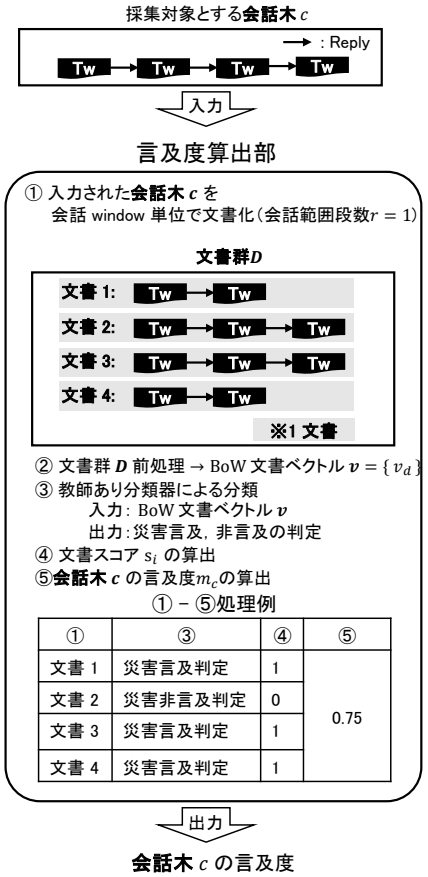


図 3: 言及度算出部の動作例

スコア s_i を算出する．最後に，文書スコア s_i をもとに会話木 c 単位での言及度 m_c を推定し，言及度 m_c を言及度算出部の出力とし，提案手法における採集結果として用いる．

4.1 教師あり分類器

本論文で用いる教師あり分類器は，入力文書を災害に言及しているか非言及かの二値に分類する分類器である．この分類器は，教師あり分類アルゴリズムとしてよく知られているランダムフォレスト，ロジスティック回帰，ナイーブベイズを 3 種を弱分類器とした，スタッキングによるアンサンブル学習によって構成する．

4.2 言及度の推定

まず，3.3 節によって得られた教師データを元に 4.1 節における教師あり分類器を学習しておく．

次に，ある会話 c を会話 window 単位に分割 [2] し，文書集合 $D = \{d_1, \ldots, d_N\}$ を得る．この文書集合 D 中の各文書が分類器への入力となり，文書 $d_i (d_i \in D)$ が災害言及判定の場合は文書スコア $s_i = 1$，災害非言及判定の場合は $s_i = 0$ とする．このとき，会話 c の言及度 m_c は D に含まれる各文書のスコアの平均値とし，次の式で求める．

$$m_c = \frac{1}{N} \sum_{i=1}^{N} s_i \tag{6}$$

2：複数の会話 window に同一の tweet が含まれうる点に注意.

5 試作システム

ここでは，提案手法に則って試作したシステムについて紹介する．

まず，会話を含む tweet 集合の取得のため，Twitter 社の提供する API のうち，Streaming API (statuses/sample) と，Lookup API (statuses/lookup) を用いる．Streaming API を利用し tweet 集合 T_S を取得する．このとき，無料ライセンスの範囲内で取得をするため，tweet 集合 T_s は投稿された全 tweet の 1%で構成され，また，このとき日本語 tweet のみを用いるために language 属性が 'ja' である tweet のみを用いる．次に，LookupAPI に対するクエリとして tweet 集合 T_s から取得した t.inreplyto を指定する．こうして得られた tweet から再帰的に reply tweet を取得し，tweet 集合 T_L を得る．こうして，会話を含む tweet 集合 $T_S \cup T_L$ の取得を行う．

また，3.2.1 節の bot account の検出と除外に基づき，bot によると判定された tweet を処理対象から除外する．検出された bot account 集合を A_b，$T_S \cup T_L$ に含まれる bot account の 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 T_E は，

$$T_E = (T_S \cup T_L) \setminus T_b \tag{7}$$

となる．

こうして得られた tweet 集合 T_E を，始点時刻 t_s から時間幅 w ごとに区切った tweet 集合を順番に $\mathrm{slot}_0, \mathrm{slot}_1, \ldots$ とする．分類対象の tweet 群を $\mathrm{slot}_j (j > 0)$ としたとき，教師データ作成部で入力とする tweet は過去の情報だけを使うため，slot_{j-1} を参照する．

この試作システム上では，tweet が投稿された時刻によっては処理されるまでに最悪の場合で slot の時間幅 w 分程度の待ち時間が発生する．本稿で扱う試作システムは提案手法と会話 window の有効性を示すために用いることを想定しているためであり，この制約はあくまで試作システムにおける実装上の問題であり，提案手法の理論上では投稿された tweet から準リアルタイムに次々と災害情報の採集を行うことが可能である．

6 評　　価

提案手法による効果を評価するために，実 tweet データを対象として実験し，代表語を含むか否かで言及度を求める 6.2 節の比較手法と比較を行う．同時に会話 window における会話範囲段数による言及度に対する効果を確認する．会話 window における会話範囲段数 $r = 1$ の場合に対し，$r = 0$，つまり従来の 1 tweet 1 文書分類と等価である場合を比較することで，1 文書とする tweet の範囲を reply 関係を辿ることで拡張することによる言及度への影響を確認する．

6.1 実　　験

6.1.1 条　　件

ランダムフォレスト，ロジスティック回帰，ナイーブベイズ，

スタッキングは python のライブラリである scikit-learn による実装とし各種パラメータはデフォルト値とする．LDA は python ライブラリである gensim による実装，Twitter-LDA は Minghui らによって GitHub に公開されているプログラム [3] を用いた．また，各種パラメータを表 3 に示す．代表語割合におけるパラメータは $\beta_0 = 0.2, \beta_1 = 0.7$ とした．教師データ作成部では会話 window の会話範囲段数 $r = 1$ とした．また，提案手法と比較手法における会話範囲段数 r を変化させたときの手法名称を表 2 の通りに呼ぶこととする．Tweet 単位処理は会話範囲段数 $r = 0$, 会話全体処理は $r = \infty$ の場合を示す．

表 2: 手法名称

手法名称	会話範囲段数 r
提案手法 ($r = 1$)	1
提案手法 (tweet 単位処理)	0
提案手法 (会話全体処理)	∞
比較手法 ($r = 1$)	1
比較手法 (tweet 単位処理)	0
比較手法 (会話全体処理)	∞

6.1.2 対象 tweet

言及度算出部の入力には表 4 に示す slot のうち，採集対象とする現 slot は $\mathrm{slot}_1, \mathrm{slot}_2, \mathrm{slot}_3$ である．Slot_0 は教師データ作成時にしか用いない．教師データ作成部には採集対象とする現 slot の前 slot から行う．ここで，現 slot に対して前 slot は一つ前の slot を指す．例えば，slot_2 が採集対象の slot である場合，slot_2 を現 slot, slot_1 を前 slot となる．また，採集対象とする現 slot に含まれる会話木における reply 先 tweet の追跡は前 slot までとする．slot_0 は始点時刻から slot 幅分過去の時刻まで追跡を行う．

実災害例として北海道胆振東部大震災，令和元年東日本台風，フンガ・ハアパイ火山の大噴火に伴う津波を対象とし，以後それぞれの災害を北海道震災，台風水害，火山噴火と略称する．各 slot における文書数を表 5 に示す．会話 window の定義より，表 5 における会話範囲段数 $r = 1, 0$ の文書数は両方で同一となっている．北海道震災は発生直後付近の時刻，火山噴火は日本列島の沿岸沿いに津波警報が発令した前後の時刻，台風水害は伊豆半島に上陸したと考えられる付近の時刻を slot_0 の始点時刻とした．

表 3: 教師なし分類器の各種パラメータ

	トピック数	クラスタ数	β_0	β_1
Twitter-LDA 分類器	7	4	0.2	0.7
LDA 分類器	10	6		

6.1.3 正解言及度データセット

実験参加者を募り，人手による tweet の言及度判定を行い，これを利用して正解言及度データセットを作成した．

対象会話木は，評価対象 slot 内の会話木から無作為抽出した．人手による判定は，3–5 人の評価者に各会話木それぞれに属する tweet を，その tweet が属する会話の流れも併せて災害に非言及，言及しているかどうかを 2 値で判定してもらった．

この人手による判定結果から，会話木ごとの正解言及度 y を求める．まず，tweet 単位で，各評価者ごとの文書 d_i のスコア s_i を

$$s_i = \begin{cases} 0 & \text{非言及判定} \\ 1 & \text{言及判定} \end{cases} \tag{8}$$

と点数化し，会話木の正解言及度 y は，木に含まれる tweet の点数，全評価者の算術平均とした．

3：https://github.com/minghui/Twitter-LDA

表 4: 対象 tweet と条件

災害例	採集対象 slot (現 slot)	始点時刻	前 slot	Slot 幅 [分]
台風水害	slot_0	2019/10/12 18:00	-	60
	slot_1	2019/10/12 19:00	slot_0	
	slot_2	2019/10/12 20:00	slot_1	
	slot_3	2019/10/12 21:00	slot_2	
北海道震災	slot_0	2018/9/6 3:00	-	120
	slot_1	2018/9/6 5:00	slot_0	
	slot_2	2018/9/6 7:00	slot_1	
	slot_3	2018/9/6 9:00	slot_2	
火山噴火	slot_0	2022/1/16 0:00	-	60
	slot_1	2022/1/16 1:00	slot_0	
	slot_2	2022/1/16 2:00	slot_1	
	slot_3	2022/1/16 3:00	slot_2	

表 5: 文書数内訳

災害名称	条件		Slot			
	名称	会話範囲段数 r	0	1	2	3
北海道震災	前 slot	∞	-	5370	8430	10215
		1	-	20780	29431	32822
		0	-	20780	29431	32822
	現 slot	∞	5370	8430	10215	11030
		1	20780	29431	32822	36163
		0	20780	29431	32822	36163
台風水害	前 slot	∞	-	4636	10897	16286
		1	-	20639	53258	77724
		0	-	20639	53258	77724
	現 slot	∞	4636	10897	16286	17233
		1	20639	53258	77724	84751
		0	20639	53258	77724	84751
火山噴火	前 slot	∞	-	14570	12095	6052
		1	-	12095	47173	24347
		0	-	12095	47173	24347
	現 slot	∞	14570	12095	6052	2804
		1	12095	47173	24347	11869
		0	12095	47173	24347	11869

6.1.4 評価指標

6.1.3 節で得られた各会話木の正解言及度 y と試作システムによって得られた推定言及度 $\hat{y}$ との平均二乗偏差 RMSE を次のように求め，評価指標として用いる．このとき，台風 Hagibis の slot_1 は 100 個の会話木を対象とするため $n = 100$，その他は 50 個の会話木を対象とするため $n = 50$ である．

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y_i} - y_i)^2} \tag{9}$$

6.2 比較手法

提案手法では，代表語を教師データ作成時に用いる．これに対し，教師あり分類器を用いらずに文書スコア s_i を，単純に代表語が含まれているかどうかで次の式の通りに求める手法を比較手法とする．

$$s_i = \begin{cases} 1 & (w \in d_i) \\ 0 & (w \notin d_i) \end{cases} \quad (10)$$

6.3 結　　果

まず，教師データ作成部による教師データとして抽出された会話数を表 6 に災害例毎に示す．

採集のための言及度推定結果については，台風 Hagibis の slot_1 における提案手法と比較手法の推定言及度 $\hat{y}$ と正解言及度 y の分布を図 4 に示す．各図における (a), (b) はそれぞれ tweet 単位処理, $r = 1$ の場合を示している．

この 2 次元ヒストグラムにおいて，縦軸と横軸はそれぞれ正解言及度 y と推定言及度 $\hat{y}$ の階級を示し，会話木の分布は左下から右上への対角線上に乗ることが理想的な結果といえる．各マス目は対応する階級に収まる会話木の度数を示しており，同図に記載されているカラーバーと対応した色付けがされている．

また，提案手法と比較手法における推定言及度 $\hat{y}$ と正解言及度 y に対する RMSE 値を表 7 に示す．これは，前述の結果を集約した性能指標であり，値が小さいほど精度良く言及度を推定できたといえる．

6.4 教師あり分類器の利用による効果

ここでは，教師あり分類器を用いた提案手法と代表語によって言及度を推定する比較手法を比較し，提案手法の有効性を確認する．まず，表 7 による RMSE 値から，会話範囲段数が同一の場合において全ての slot で比較手法よりも提案手法の方が推定誤差が小さくなったことから，提案手法の方がより良い採集結果を得られたと考えられる．

次に，具体的に台風水害における正解言及度 y と推定言及度 $\hat{y}$ の散布図を示した図 4 より，会話木の分布を確認する．まず，提案手法 ($r = 1$) における推定言及度よりも比較手法 ($r = 1$) における推定言及度は全体的に推定言及度 $\hat{y}$ が小さい傾向にあることが見られる．また，比較手法 ($r = 1$) と比較手法 (tweet 単位処理) は $0.9 \leqq y$ における正解言及度 y が高い値の範囲において，多くの会話木の推定言及度 $\hat{y}$ が 0.1 未満の範囲に分布していることが見られる．これは提案手法 ($r = 1$) において同じ正解言及度の範囲の推定言及度 $\hat{y}$ が $0.9 \leqq \hat{y}$ の範囲にほとんどの会話木が分布していることと対照的である．

以上のことから，正解言及度が高い会話木において，比較手法では推定言及度が低い会話木が提案手法では推定誤差が小さくなっていることが見られた．これらのことから，代表語による採集よりも提案手法で教師生成を自動的に行った上での教師あり分類器を利用することで，推定誤差を減らした良い採集結果を得られることが分かった．

表 6: 各 採集対象 slot における教師データとして用いる文書数.

条件		採集対象 slot		
災害略称	会話範囲段数 r	1	2	3
北海道震災	1	1032	840	2088
台風水害	1	1038	1754	2356
火山噴火	1	6150	5272	884

表 7: 各条件における提案手法と比較手法の RMSE 値

災害名称	手法名称	slot		
		1	2	3
北海道震災	提案手法 ($r = 1$)	**0.297**	**0.288**	**0.203**
	提案手法 (tweet 単位処理)	0.344	0.363	0.264
	提案手法 (会話全体処理)	0.314	0.321	0.218
	比較手法 ($r = 1$)	0.617	0.583	0.471
	比較手法 (tweet 単位処理)	0.642	0.614	0.484
	比較手法 (会話全体処理)	0.615	0.620	0.468
台風水害	提案手法 ($r = 1$)	**0.198**	**0.210**	0.241
	提案手法 (tweet 単位処理)	0.225	0.227	**0.208**
	提案手法 (会話全体処理)	0.298	0.239	0.296
	比較手法 ($r = 1$)	0.461	0.497	0.454
	比較手法 (tweet 単位処理)	0.480	0.527	0.480
	比較手法 (会話全体処理)	0.475	0.490	0.476
火山噴火	提案手法 ($r = 1$)	**0.167**	**0.190**	0.402
	提案手法 (tweet 単位処理)	0.187	0.242	**0.299**
	提案手法 (会話全体処理)	0.189	0.216	0.455
	比較手法 ($r = 1$)	0.420	0.453	0.410
	比較手法 (tweet 単位処理)	0.456	0.496	0.440
	比較手法 (会話全体処理)	0.416	0.440	0.408

6.5 会話範囲段数による効果

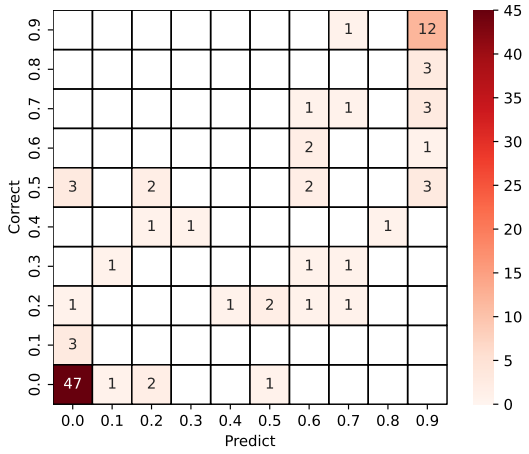
ここでは，会話 window の独自性をもたらす場合である会話範囲段数 $r = 1$ による効果を確認する．提案手法 ($r = 1$) と 1 tweet を 1 文書と見做す提案手法 (tweet 単位処理), 1 会話 1 文書とする提案手法 (会話全体処理) について見ていく．いずれも教師の自動生成などを用いた提案手法の枠組みの中ではあるが，tweet 単位処理は会話概念を用いない既存手法と類似の場合に対応し，会話全体処理は会話内の詳細情報を利用しない手法に対応する．

まず，全般的な性能比較のため表 7 で推定誤差を RSME で比較すると，9 slot 中 7 slot において提案手法 ($r = 1$) が他 2 種の提案手法 (tweet 単位処理), 提案手法 (会話全体処理) よりも RMSE 値による推定誤差が小さく，最良の結果である．提案手法 (会話全体処理) と比較した場合は，提案手法 ($r = 1$) が全ての slot において優位である．

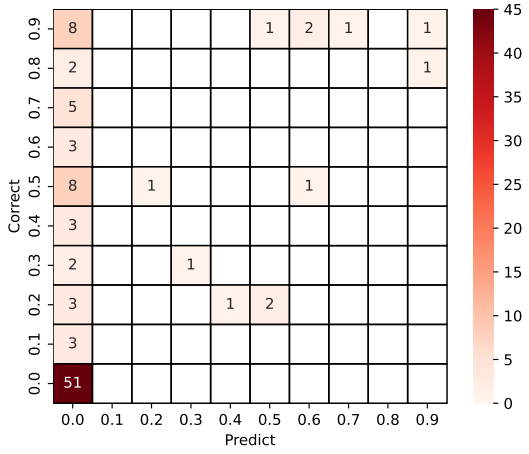
残る台風水害における slot_3 と火山噴火における slot_3 については，提案手法 (tweet 単位処理) が最良である．このため，提案手法（$r = 1$）が常に最良とはいえないが，多くの場合において会話内の会話 window 単位での処理が有効であるといえる．会話範囲段数の設定をデータ状況に応じて適応的に設定することができればさらに改善することも予想されるが，これは今後の課題である．

7 おわりに

災害時における災害言及 tweet の採集を準リアルタイムに実行できる手法について提案した．提案手法は tweet から自己教師生成を行い言及度を算出する手法であり，単純に代表語を含むか否かによる判定を行う比較手法より精度良く災害言及度を推定できることを示した．同時に，会話 window の効果について提案手法 ($r = 1$), 提案手法 (tweet 単位処理), 提案手法 (会



(a) 提案手法 ($r = 1$).



(b) 比較手法 ($r = 1$).

図 4: 台風 Hagibis, 採集対象を slot_1 としたときの比較手法における正解言及度 y と推定言及度 $\hat{y}$ の散布図.

話全体処理) における推定言及度を確認し，提案手法 ($r = 1$) による会話 window における適当なパラメータを見出した.

本論文での会話 window は，会話木におけるエッジ距離を常に 1 とした場合の k-近傍を 1 文書としたが，より話題の変化を捉えるために，枝の分岐を考慮した方法を取り入れるなどが考えられる．1 会話木内での言及度の高低を視覚化して表示することで，読者の理解を支援するなどの周辺と併せ今後の課題である.

文　　献

[1] T. Fujita, K. Shibutani and A. Kobayashi, “Tweet classification using con-versational relationships, ” 2020 Eighth International Symposium on Computing and Net-working Workshops (CANDARW), pp.406–410, Nov.2020.

[2] 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.

[3] H. Shekhar and S. Setty, “Disaster analysis through tweets,” 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, pp.1719–1723, 2015.

[4] Sakaki, T., Okazaki, M., and Matsuo, Y, “Earthquake ShakesTwitter Users: Real-time Event Detection by Social Sen-sors,” Proc. 19th International Conference on World WideWeb (WWW 2010), pp.851–860, 2010.

[5] A. Manaka et al., ”Collection of disaster-related information by focusing on Twitter posts immediately after retweeting announcement posts,” 2016 IEEE Region 10 Conference (TENCON), Singapore, pp.2251–2255, 2016.

[6] 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.

[7] 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.

[8] Hong, L., and Davison B. D. “Empirical study of topic modeling in twitter, ” Proceedings of the first workshop on social media analytics, pp.80–88, 2010.

[9] Mehrotra, Rishabh, et al. “Improving lda topic models for microblogs via tweet pooling and automatic labeling, ” Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, pp.889–892, 2013.

[10] Alvarez-Melis, D., Saveski, M. “Topic modeling in twitter: Aggregating tweets by conversations”, Proceedings of the Tenth International AAAI Conference on Web and Social Media (ICWSM 2016) , Vol.10,No. 1, pp.519 – 522 . 2016.

[11] 藤田 俊之, 小林 亜樹, “災害情報抽出のための Reply 関係を用いた話題抽出,” 第 12 回データ工学と情報マネジメントに関するフォーラム (DEIM2020), 2020.

[12] T. Fujita and A. Kobayashi, “Proposal of Window on Conversation Tree to Disaster Information Collecting,” 2021 International Conference on Emerging Technologies for Communications (ICETC2021), 2021.

[13] R. Othman, R. Belkaroui, and R. Faiz, “Extracting product features for opinion mining using public conversations in Twitter,” Procedia Comput. Sci., vol.112, pp.927–935, Jan. 2017.

[14] S. A. Bahrainian and A. Dengel, “Sentiment analysis and summarization of twitter data,” In Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on, pp.227–234. 2013.

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

[16] 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.

[17] 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.

[18] 鳥海不二夫, 榊剛史, “バースト現象におけるトピック分析,” 情報処理学会論文誌, Vol.58, No.6, pp.1287–1299, 2017.

[19] 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.

[20] J. Benhardus, Jugal Kalita, “Streaming trend detection in Twitter,” Int. J. Web Based Communities, Vol. 9, No.1, pp.122–139, 2013.

[21] David M. Blei, Andrew Y. Ng, Michael I. Jordan, “Latent Dirichlet Allocation, ” Journal of Machine Learning Research 3, pp.993–1022, 2003.

[22] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths, “Probabilistic author-topic models for information discovery,” Proc. of SIGKDD 2004, 2004.

[23] Mehrotra, R.; Sanner, S.; Buntine, W.; and Xie, L. 2013. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In SIGIR.

[24] Fleiss, J. L, “Measuring nominal scale agreement among many raters, ” Psychological Bulletin, Vol.76, No.5 pp.378–382, 1971.