Vol.2013-DBS-158 No.21
2013/11/26

文体及びツイート付随情報を用いた乗っ取りツイート検出

上里和也[†1]　奥谷貴志[†2]　浅井洋樹[†2†3]　奥野峻弥[†2]　田中正浩[†2]　山名早人[†4†5]

Twitter のユーザ数が増加を続ける一方で，不正に ID 及びパスワードを入手され，他人によってツイートを投稿される被害が増加している．これに対し，我々はアカウント乗っ取りによって投稿されるメッセージの一部であるスパムツイートの検出手法を提案し，8 割程度の正答率を得ている．同手法では特定の単語が含まれているスパムツイートを検出対象とし，検出の有効性を示している．本研究では同検出対象を広げ，アカウントの所持者以外が投稿したツイート全体を「乗っ取りツイート」として定義し，これを検出する手法を提案する．また本研究では，以前提案した手法に対してパラメータの再調整を行うと同時に，頻繁に用いるハッシュタグの種類及びリプライを送る相手が各アカウントにおいて特徴的であることを利用し，F 値の向上を図った．100 アカウントに対して評価実験を行った結果，我々が提案している従来手法と比較し，F 値を 0.1984 向上させ F 値 0.8570 を達成した．

1. はじめに

Twitter とは，ツイートと呼ばれる 140 字以内のメッセージの投稿を中心とした世界的に普及しているマイクロブログサービスである．Twitter のユーザ数は 2012 年 6 月の時点で 5 億人を超えている[1]．しかし一方では，Twitter の大規模なネットワークが悪用され，Twitter の快適な利用を妨げられている．その中でも，アカウントを他人に乗っ取られ，あたかもアカウント所持者自身が投稿しているかのようにツイートを投稿される被害が深刻な問題となっており，詐欺広告の拡散などを目的とした，スパムツイートと呼ばれる不正なツイートの 16%が常に不正なツイートを自動で投稿しているアカウントによって投稿されており，残りの 84%がアカウント所持者以外のユーザに乗っ取られているアカウントから投稿されているという報告[2]がある．Twitter におけるアカウントの認証で必要なものは ID とパスワードのみである．この性質を利用し，フィッシングサイトによって ID 及びパスワードを取得することで，他人のアカウントからスパムツイートを投稿している．このような問題が存在するにも関わらず，多くの既存の研究では常にスパムツイートを投稿するアカウントの検出を目的とした手法を提案しており，ツイート単位での検出は行われてこなかった．

これに対して，我々はアカウントの所持者が投稿したツイートと，スパムツイートに特有な表現を持つツイートをスパムツイートとして分類し，乗っ取られたアカウントから投稿されたツイートを検出する手法を提案し，8 割程度の正答率を得ている[3]．同手法では特定の単語が含まれているツイートを検出対象のスパムツイートとし，それらのツイートに対する検出の有効性を示した．

本研究では，我々の従来の手法[3]を拡張し,アカウントの所持者以外が投稿したツイートを「乗っ取りツイート」として定義し，これの検出を目標とした．これは，所謂スパムツイートだけでなく，アカウントの乗っ取りを自動的に判定することが重要であると考えたからである．さらに，本研究では，我々の従来手法[3]に対してパラメータの再調整を行うと同時に，頻繁にツイートに付加するハッシュタグの種類及びリプライを送る相手が各アカウントにおいて特徴的であることを利用し，扱う特徴量を追加することで検出性能の向上を図った．ハッシュタグとは，ツイートの本文中に書き込むことで付加することができる，ツイートのトピックを示すタグのことであり，リプライとは，特定のアカウントに宛てたツイートのことである．

本稿では以下の構成をとる．まず 2 節では，本研究に関連する既存研究である，著者推定に関する研究，スパムツイートを常に投稿するアカウントの検出に関する研究及びツイート単位でのスパムツイートの検出に関する研究について説明する．次の 3 節では，本研究の実験に用いるデータの収集方法や，データセットの作成方法について説明する．続く 4 節では，我々の従来手法[3]を乗っ取りツイートの検出に最適化するためのパラメータの調整の方法及び有効性について説明する．そして 5 節では，本稿で提案する乗っ取りツイートの検出手法について説明する．また 6 節では提案手法及びその比較対象となる我々の従来手法[3]に対する実験の結果を示し，評価を行う．最後に 7 節で本稿をまとめる．

2. 関連研究

2.1 著者推定に関する研究

著者推定の研究では，著者が既知である文章を用いて，著者が未知である文章の著者を著者候補の中から推定する．本研究においても同様に，対象のツイートがアカウントの所持者によって投稿されたものであるか，あるいはアカウントの所持者以外が投稿したものであるかを判定するため，著者推定の研究を本研究の関連研究として捉える．

中島らの研究[4]では，2 つの文章p, qにおいて，それぞれ形態素解析を行い，品詞 n-gram の出現頻度分布に対するピアソンの積率相関係数の逆数を式(1)に従って算出し，その

†1 早稲田大学　基幹理工学部
†2 早稲田大学大学院　基幹理工学研究科
†3 早稲田大学メディアネットワークセンター
†4 早稲田大学理工学術院
†5 国立情報学研究所

値を文体相違度$Dissim(p,q)$としている．品詞 n-gram とは，文章中に表れる長さ n の品詞の列である．なお，集合Cは文章p,qに現れる全ての品詞 n-gram の和集合，f_{px}は文章pにおける品詞 n-gram$x$の頻度，$\overline{f_{pq}}$は品詞 n-gram の頻度の平均を示している．

$$Dissim(p,q)=\frac{\sqrt{\sum_{i\in C}(f_{px}-\overline{f_{pq}})^2}\sqrt{\sum_{i\in C}(f_{qx}-\overline{f_{pq}})^2}}{\sum_{i\in C}(f_{px}-\overline{f_{pq}})(f_{qx}-\overline{f_{pq}})} \quad (1)$$

式(1)を用いて 2 つの文章の文体相違度を算出し，文体相違度の値が小さいほど文体が類似しているとすることで，対象の文章と最も文体が類似している文章を書いた著者候補を真の著者として推定する．

松浦らの研究[5]では，品詞 n-gram を用いて文体相違度を算出する中島ら[4]の手法に対し，文字 n-gram を用いて文体相違度を算出する．文字 n-gram は，文章中に現れる長さ n の文字列のことを示す．具体的には，文章pに現れる全ての文字 n-gram 中に含まれる文字 n-gram$x$の割合を$P_p(x)$，集合Cを文章p,q双方に表れる文字 n-gram の和集合とし，文章p,qの文体相違度$Dissim(p,q)$を式(2)に従って算出する．

$$Dissim(p,q)=\frac{1}{|C|}\sum_{x\in C}\left|\log_{10}\frac{P_p(x)}{P_q(x)}\right| \quad (2)$$

式(2)で算出した 2 つの文章の文体相違度を算出し，中島らの手法[4]と同様，文体相違度の値が小さいほど文体が類似しているとすることで，対象の文章と最も類似している文章を書いた著者候補を真の著者として推定する．

2.2 スパムツイートを常に投稿するアカウントの検出に関する研究

本項では，Twitter 上の不正な行為の検出に関する研究の中で多く行われている，スパムツイートを常に投稿する不正なアカウントの検出に関する先行研究について説明する．

Benevenuto らの研究[6]では，5,400 万アカウント以上を収集し，手作業によってスパムツイートを常に投稿する不正なアカウントと正規のアカウントに分類した上で，各々のアカウントの特徴をサポートベクタマシンによって学習し，分類を行った．なお，正規のアカウントは不正なアカウント以外の全てのアカウントを指す．サポートベクタマシンの学習には，対象のアカウントのツイートの内容に関する特徴及びアカウント自体に関する特徴をそれぞれ 3 種類用いている．ツイートの内容に関する特徴としては，URL を含んでいるツイートの割合，スパムツイートに頻繁に含まれる単語を含んでいるツイートの割合及びハッシュタグを含むツイートの割合を用いている．またアカウント自体に関する特徴としては，フォローされているアカウントの数に対するフォローしているアカウントの数，アカウントの使用期間及びリプライを受け取る数を用いている．ここで Twitter におけるフォローとは，対象のアカウントのツイートを自分のタイムラインに表示するように設定する行為のことである．これらの合計 6 つの特徴を学習させたサポートベクタマシンによってアカウントの分類を行った結果，不正なアカウントを 69.7%の精度で検出することに成功している．

Wang らの研究[7]では，あらかじめ手作業によって不正なアカウントと正規のアカウントに分類した 500 アカウントを対象とし，各アカウントの最も新しい 20 ツイート及びアカウント周辺のフォロー関係から多数の特徴を抽出し，ベイジアンフィルタによって学習，分類を行うことで，分類に有効である特徴量を実験的に調査している．実験を行った結果，正規のアカウントと比べ，不正なアカウントはフォロー数に対し，被フォロー数が極端に小さい，あるいは極端に大きい傾向があることが確認され，また不正なアカウントは同様な内容のツイートを複数投稿する傾向があることが確認されている．ベイジアンフィルタによる分類の精度は 89%である．

2.3 ツイート単位でのスパム検出に関する研究

本項では，対象のツイートをツイート単位で検出するという点で本研究と類似している，スパムツイートのツイート単位での検出に関する先行研究について説明する．

我々は以前に，ツイート本文の文字 n-gram の分布，ツイートを投稿したクライアントの種類及びツイートを投稿した時間帯を用いて対象のアカウントの 1,000 件の過去のツイートどうしの相違度を取得し，同様に算出した新着ツイートと過去のツイートの相違度が過去のツイートどうしの相違度から算出した閾値より大きい場合に，その新着ツイートをスパムツイートと判定する手法[3]を提案した．

具体的には，まず対象のアカウントの過去の 1,000 ツイートを直近の 100 ツイートとその他の 900 ツイートに分割し，それぞれのツイートの集合を$\mathbf{A'}=\{a'_1,\dots,a'_j,\dots,a'_{100}\}$, $\mathbf{A}=\{a_1,\dots,a_i,\dots,a_{900}\}$とする．式(3)によってツイート$a_i$に対するツイート$a'_j$の文体相違度$dissim(a_i,a'_j)$を算出し，式(4)によって$1\leqq i\leqq 900$の文体相違度$dissim(a_i,a'_j)$の中央値を求め，各ツイート$a'_j$のツイート集合Aに対する文体相違度$Dissim(\boldsymbol{A},a'_j)$とする．

$$dissim(a_i,a'_j)=\frac{1}{|C|}\sum_{x\in C}\left|\log_{10}\frac{P_{a_i}(x)}{P_{a'_j}(x)}\right| \quad (3)$$

$$Dissim(\boldsymbol{A},a'_j)=\widetilde{dissim(a_i,a'_j)} \quad (4)$$
$$1\leqq i\leqq 900$$

$P_{a_i}(x)$はツイートa_iに出現する全ての文字 2-gram 中に含まれる文字 2-gram xの割合であり，同様に $P_{a'_j}(x)$はツイートa'_jに出現する全ての文字 2-gram 中に含まれる文字 2-gram xの割合である．ここで文字 n-gram の n の値として用いている 2 は，実験的に求めた最適な値である．また集合Cはツイートa_i及びツイートa'_jの双方に現れる文字 2-gram 全体の集合を表している．なお，文体相違度の算出の際はノイズを除去するために，リプライに含まれる「@ユーザ名」とハッシュタグ「#文字列」，URL をツイートか

ら取り除く．

ツイートを投稿したクライアントの種類による重み$Weight(q)$は，クライアントqから投稿されたツイートa'_jに対し，ツイート集合 **A** 内でクライアントqから投稿されたツイート数が占める割合$P(q)$を用いて，式(5)をもとに算出する．同様に，ツイートを投稿したクライアントの種類及び投稿時間帯を用いた重み$Weight(q,t)$は，時間tにクライアントqから投稿されたツイートa'_jに対し，ツイート集合**A**内で，時間tから前後 1 時間に投稿されたツイートの中で，クライアントqから投稿されたツイートが占める割合$P(q,t)$を用いて，式(6)によって算出する．

$$Weight(q) = \begin{cases} 0.9\big(1 - P(q)\big) & (P(q) > 0) \\ 1.0 & (P(q) = 0) \end{cases} \quad (5)$$

$$Weight(q,t) = \begin{cases} 0.9\big(1 - P(q,t)\big) & (P(q,t) > 0) \\ 1.0 & (P(q,t) = 0) \end{cases} \quad (6)$$

式(5)，(6)における係数 0.9 は実験的に求めた最適な値を用いている．式(7)，(8)に従い，式(5)，(6)で求めた重み$Weight(q)$，$Weight(q,t)$及び式(4)によって求めた文体相違度$Dissim(\boldsymbol{A}, a'_j)$を掛け合わせることで，重み付き文体相違度$Score(\boldsymbol{A}, a'_j)$を求める．

$$Score(\boldsymbol{A}, a'_j) = Dissim(\boldsymbol{A}, a'_j) * Weight(q) \quad (7)$$

$$Score(\boldsymbol{A}, a'_j) = Dissim(\boldsymbol{A}, a'_j) * Weight(q,t) \quad (8)$$

式(4)で算出した文体相違度$Dissim(\boldsymbol{A}, a'_j)$，あるいは式(7)，(8)で算出した重み付き文体相違度$Score(\boldsymbol{A}, a'_j)$の標準偏差σを用いて式(9)，(10)によって，アカウントの所持者のツイートとスパムツイートを分類する際に用いる閾値$\alpha(\boldsymbol{A}, \boldsymbol{A'})$を算出する．

$$\alpha(\boldsymbol{A}, \boldsymbol{A'}) = \sigma + 0.08\overline{Dissim(\boldsymbol{A}, a'_j)} \quad (9)$$

$$\alpha(\boldsymbol{A}, \boldsymbol{A'}) = \sigma + 0.08\overline{Score(\boldsymbol{A}, a'_j)} \quad (10)$$

式(9)における$\overline{Dissim(\boldsymbol{A}, a'_j)}$は 100 件のツイート$a'_j$の文体相違度の平均であり，式(10)における$\overline{Score(\boldsymbol{A}, a'_j)}$は 100 件のツイート$a'_j$の重み付き文体相違度$Score(\boldsymbol{A}, a'_j)$の平均である．また式(9)，(10)の右辺第 2 項の係数 0.08 は実験的に求めた最適な値である．

評価実験では，100 アカウントに対してそれぞれアカウントの所持者の 30 ツイート及び特定の単語を含む 30 件のスパムツイートから成る 60 ツイートを新着ツイート集合$\boldsymbol{B} = \{b_1, \ldots, b_k, \ldots, b_{60}\}$とし，それぞれ文体相違度$Dissim(\boldsymbol{A}, b_k)$，あるいは重み付き文体相違度$Score(\boldsymbol{A}, b_k)$を同様の手順で算出し，閾値より大きい値となったツイートb_kをスパムツイートとして分類する．文字 2-gram の分布による文体相違度のみを用いる手法，式(7)を用いてツイートを投稿したクライアントの種類によって重み付けを行う手法，式(8)を用いてツイートを投稿したクライアントの種類及び投稿時間帯によって重み付けを行う手法の，合計 3 つの手法によって分類を行った．その結果，ツイートを投稿したクライアント及び投稿時間帯によって重み付けを用いたときに最高の正答率となり，その値は 82.8%である．

2.4 先行研究の問題点と解決方法

2.1 項，2.2 項及び 2.3 項では，本研究の関連研究について説明した．そこで本項では，本研究に関連する研究の手法，あるいは評価手法の問題及び本研究におけるその解決方法について説明する．

2.1 項で説明した著者推定手法では，小説やウェブ上のブログの文章などを対象としており，本研究で対象としている 140 字以内であるツイートに対し，文章量が多い．したがって，中島ら[4]や松浦ら[5]の研究では品詞 n-gram や文字 n-gram の分布によって著者の推定を行うために充分な特徴を取得することができる．それに対し，本研究では 1 ツイートのみを対象として著者がアカウントの所持者であるか否かを推定しなければならず，n-gram のみでは充分に特徴が取得できない．そこで本研究では，著者推定手法の文体相違度の算出方法によって文体相違度を取得し，その文体相違度に対してツイートデータに付随する情報を特徴量として用いて重み付けを行うことで，検出性能の向上を図った．また形態素解析を利用した品詞 n-gram を用いる場合，ツイートのように短文であり，かつ未知語が多い文章が対象となると，正確に文体相違度を算出することが困難である．したがって本研究では，文字を対象とするため，短文においても取得できるデータ数が比較的多く，未知語にも頑強である文字 n-gram を用いた文体相違度の算出方法を用いた．

また著者推定の研究では，全ての著者候補に対して著者が既知である文章から特徴を取得することが可能である．しかし本研究では，アカウントの所持者が投稿したツイートに限り特徴を取得することができるため，著者推定手法をそのまま転用することはできない．そこで本研究では，アカウントの所持者の過去のツイートどうしの文体相違度の標準偏差を用いて閾値を定め，新着ツイートとアカウントの所持者が過去に投稿したツイートの文体相違度の値が閾値を越えた場合にそのツイートを乗っ取りツイートとして検出する方法をとった．

2.2 項で説明した，常にスパムツイートを投稿するアカウントの検出手法では，スパムツイートを常に投稿するアカウントのフォロー関係などの特徴を学習及び分類に用いるため，正規のアカウントから投稿される乗っ取りツイートの検出を行うことができない．そこで本研究では，アカウントの所持者の過去のツイートの特徴のみを用いて，新着ツイートがアカウントの所持者によって投稿されたツイートであるか，あるいはアカウントの所持者以外によって投稿された乗っ取りツイートであるかを判定する手法を提案する．

2.3 項で説明した，スパムツイートの検出手法では，特定の内容を含むツイートに対する検出の有効性を示している．そこで本研究では，検出対象を広げ，アカウントの所持者以外が投稿したツイートを「乗っ取りツイート」とし

て定義し，これを検出する手法を提案する．そのため，評価実験において，対象のアカウント以外のアカウントから投稿されたツイート全体から無作為に選んだツイートを検出対象とする．また本研究では，我々の従来手法[3]を参考にする．しかし我々の従来手法[3]におけるパラメータは特定の表現を含むスパムツイートの検出に最適化されている．そのため，多様な文体をもつ乗っ取りツイートを検出対象とする上で，我々の従来手法[3]におけるパラメータを乗っ取りツイートの検出に最適化する必要がある．そこで本研究では，我々の従来手法[3]におけるパラメータを乗っ取りツイートの検出に最適化した上で，検出性能をさらに向上する手法を提案する．

また我々の従来手法[3]では，検出性能の評価基準として正答率を用いていた．それに対して本研究では，乗っ取りツイートのフィルタリングを行う目的に対してより適切であると考えられる，情報フィルタリングの評価基準であるF値を用いることとする．

3. 実験データ

3.1 データの収集方法

本項では，本研究で用いたデータの収集方法を説明する．

データの収集は Twitter API1.1 を用いて行い，2013 年 7 月 24 日から同年同月 25 日までの間にツイートを投稿した，ユーザインターフェースの言語設定が英語であるアカウントを 5,117 アカウント取得した．本研究では英語のツイートを対象としているが，ユーザインターフェースの言語設定が英語であっても英語でツイートを投稿していないアカウントが存在する．そのため，英字及び半角記号以外の文字を含むツイートが直近の 1,000 ツイート内に半分以上の割合で存在するアカウントを排除した．また本研究においてアカウントの所持者によるツイートは最低 1,030 ツイート必要であるため，1,030 ツイート以上取得できなかったアカウントを排除した．最終的に得られたデータは 2,427 アカウントのデータである．

3.2 パラメータ最適化に用いるデータセットの作成方法

本項では，3.1 項で説明したデータの中からパラメータの最適化のためのデータセットを作成する方法について説明する．

3.1 項で説明したデータの中からアカウントを選択する方法の概要図を図 1 に，選択したアカウントのデータからのツイートの取得方法の概略図を図 2 に示す．無作為に選んだ 100 アカウントを 1 組として，アカウントの重複を許さず 9 組のデータを取得し，これらの 100 アカウント×9 組のデータをアカウントの所持者によるツイートを取得するために用いる．なおアカウントの所持者のツイートを取得するためのアカウントのデータに関しては，各アカウントの直近の 1,030 ツイートを取得する．ここで得られた 1,030 ツイートに対して，最も新しい 30 ツイートを分類対象の新着ツイート集合**B**，その次の 100 ツイートをツイート集合**A′**，その他の 900 ツイートをツイート集合**A**として分割する．

また別途無作為に取得した 30 アカウントからそれぞれアカウントの重複を許さず無作為に 1 ツイートずつ選択する．ここで得られた 30 ツイートを乗っ取りツイートとし，アカウントの所持者によるツイート内のツイート集合**B**に要素として追加し，合計 60 ツイートを分類対象の新着ツイート集合**B**とする．なお乗っ取りツイートに関しては，全てのアカウントに対する実験において常に同一のデータセットを用いた．

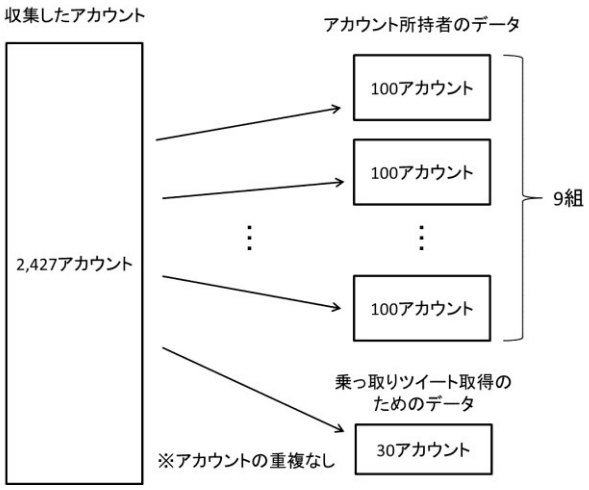

図 1　パラメータの最適化に用いるアカウントの選択方法

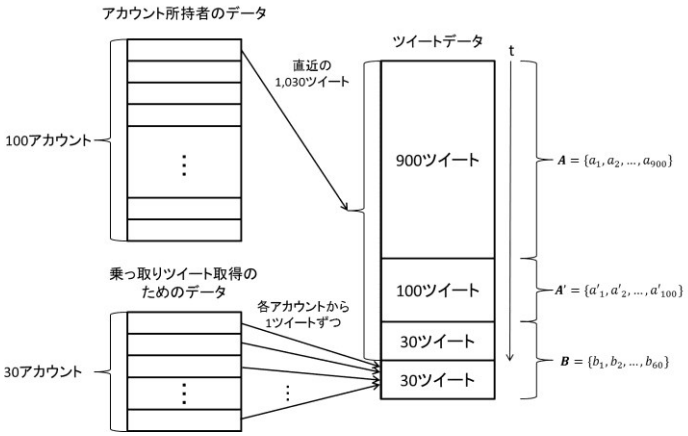

図 2　ツイートの取得方法

3.3 評価実験に用いるデータセットの作成方法

本項では，3.1 項で説明したデータから評価実験のためのデータセットを作成する方法について説明する．

3.1 項で説明したデータの中からアカウントを選択する方法の概要図を図 3 に示す．選択したアカウントのデータからのツイートの取得方法の概略図は 3.2 項と同様に図 2

を参照されたい.

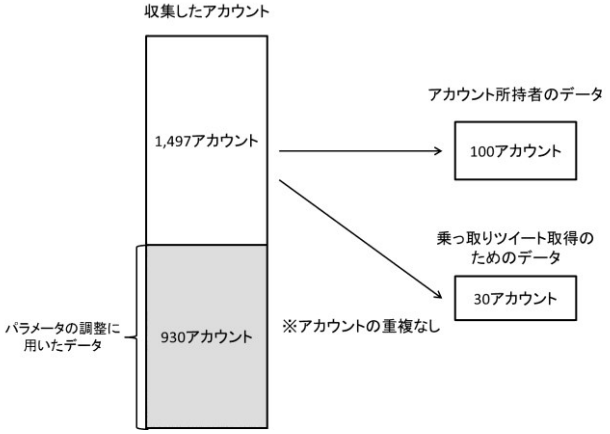

図 3　評価実験に用いるアカウントの選択方法

3.2 項で作成した，パラメータの最適化に用いたデータとは一切重複せず，かつ無作為に 100 アカウントを選出する．ここで得られた 100 アカウントのデータに対して，各アカウントの直近の 1,030 ツイートを取得し，アカウントの所持者によるツイートとする．1,030 ツイートのアカウントの所持者によるツイートに対して,最も新しい 30 ツイートを分類対象の新着ツイート集合**B**，その次の 100 ツイートをツイート集合**A′**，その他の 900 ツイートをツイート集合**A**として分割する．

また同様に，パラメータの最適化に用いたデータ及び評価実験で用いるアカウントの所持者によるツイートを取得するためのデータとの重複を許さず,かつ無作為に 30 アカウントを選択し，それぞれのアカウントのツイートを無作為に 1 ツイートずつ選出する.このとき得られた 30 ツイートを乗っ取りツイートとする．ここで得られた乗っ取りツイートをアカウント所持者のツイート内の 30 件の新着ツイート集合**B**に要素として追加し，合計 60 ツイートを分類対象の新着ツイート集合**B**とする．乗っ取りツイートに関しては，パラメータの調整と同様，常に同一のデータセットを用いた.

4. 乗っ取りツイート検出に対するパラメータの最適化

本節では，我々の従来手法[3]をもとに，パラメータを乗っ取りツイートの検出に最適化する方法について説明する.

4.1 パラメータを最適化する方法

本項では，パラメータとして最適な値を実験的に取得する方法について説明する．3.2 項において説明した方法によって作成した 9 組のアカウントの所持者による過去のツイートに対して，それぞれ全通りのパラメータの値を用いて F 値を算出し，各々のデータにおいて F 値が最大となったパラメータの値を取得する．ここで取得した 9 つのパラメータの値の中に最も多く出現した値を最適なパラメータとして採用する.

4.2 最適化されたパラメータを用いた手法

本項では，4.1 項で説明した方法によって実際に得られた値をパラメータとして採用した手法について説明する.

我々の従来手法[3]に対して最適化を行うパラメータは，以下の通りである.

- 文体相違度を取得する際に用いる文字 n-gram の n の値
- 重みを算出する式の係数
- ツイートを投稿したクライアントの種類による重み付けに用いる過去のツイートの数
- ツイートを投稿したクライアントの種類及び投稿時間帯による重み付けを行う際に，重み付けに用いる過去のツイートを決定するための時間帯の幅
- 閾値を算出する式の係数

全てのパラメータの初期値は，我々の従来手法[3]のパラメータと同一の値とし，実験的に最適な値が求められたパラメータから値を固定していくことで全てのパラメータの最適な値を求めた.

文体相違度の算出に用いる文字 n-gram の n の値は 1,2,3,4,5 と変動させた結果，1 が最適であったため，本研究では文字 1-gram を用いて式(3)，(4)によって文体相違度を算出する.

重みの式の係数を 0.1～1.0 の間で 0.1 ずつ変動させて最適化した結果，ツイートを投稿したクライアントの種類を用いた重み付け手法では 1.0，ツイートを投稿したクライアントの種類及び投稿時間帯を用いた重み付け手法では 0.8 が最適な値であった．またツイートを投稿したクライアントの種類及び投稿時間帯を用いた重み付け手法では，対象のツイートが投稿された時間tから前後 1 時間に投稿された過去のツイートの投稿に用いたクライアントの種類を用いていたが，この時間の幅を前後 1 時間，2 時間，3 時間と変動させたところ,時間tの前後 1 時間に投稿された過去のツイートを重み付けに用いたときに最も F 値が大きくなった．したがって，ツイートを投稿したクライアントの種類を用いた重み$Weight(q)$は式(11)，また対象のツイートの投稿時間tの前後 1 時間に投稿された過去のツイートの投稿に利用したクライアントの種類を用いた重み$Weight(q,t)$は，式(12)によって算出する.

$$Weight(q) = \begin{cases} 1 - P(q) & (P(q) > 0) \\ 1.0 & (P(q) = 0) \end{cases} \tag{11}$$

$$Weight(q,t) = \begin{cases} 0.8\big(1 - P(q,t)\big) & (P(q,t) > 0) \\ 1.0 & (P(q,t) = 0) \end{cases} \tag{12}$$

次に，閾値の式の係数を 0.1～1.0 の間で 0.1 ずつ変動させて最適化した結果，文体相違度$Dissim\big(\boldsymbol{A}, a'_j\big)$を用いた場合及び重み付き文体相違度$Score\big(\boldsymbol{A}, a'_j\big)$を用いた場合のいずれの場合も 0.7 が最適な値であった．したがって，式(4)で算出した文体相違度$Dissim\big(\boldsymbol{A}, a'_j\big)$，あるいは式(11)，(12)

で算出した重み付き文体相違度$Score(\boldsymbol{A}, a_j')$の標準偏差σを用いて式(13), (14)によって，アカウント所持者によるツイートとスパムツイートを分類する際に用いる閾値$\alpha(\boldsymbol{A}, \boldsymbol{A}')$を算出する．

$$\alpha(\boldsymbol{A}, \boldsymbol{A}') = \sigma + 0.7\overline{Dissim(\boldsymbol{A}, a_j')} \quad (13)$$

$$\alpha(\boldsymbol{A}, \boldsymbol{A}') = \sigma + 0.7\overline{Score(\boldsymbol{A}, a_j')} \quad (14)$$

4.3 パラメータ最適化の有効性の検証

4.2 項ではパラメータを調整し，我々の従来手法[3]を乗っ取りツイートの検出に最適化した．また我々は以前，文字 n-gram を用いた文体相違度のみを用いる手法, ツイートを投稿したクライアントの種類を重み付けに用いる手法, ツイートを投稿したクライアントの種類及び投稿時間を重み付けに用いる手法の合計 3 つの手法[3]を提案している．そこで，各々の手法に対して我々の従来手法[3]のままのパラメータを採用し, 乗っ取りツイートの検出を行う場合と，パラメータの最適化を行った上で乗っ取りツイートの検出を行う場合の比較実験を行い，4.1 項におけるパラメータの調整が有効であるか否かを検証する．本項における実験では，3.3 項で説明した方法で作成した評価用のデータセットを用いる．我々の従来手法[3]のままのパラメータを採用した場合の検出結果を表 1 に，またパラメータの最適化を行った場合の検出結果を
表 2 に示す.

表 1　我々の従来手法を用いたときの検出結果

手法	適合率	再現率	F 値
文体相違度	0.5132	0.6827	0.5859
文体相違度×クライアントの種類	0.7630	0.5687	0.6516
文体相違度×クライアントの種類及び投稿時間帯	0.7552	0.5840	0.6586

表 2　パラメータを調整した手法を用いたときの検出結果

手法	適合率	再現率	F 値
文体相違度	0.7806	0.9000	0.8360
文体相違度×クライアントの種類	0.9052	0.7763	0.8358
文体相違度×クライアントの種類及び投稿時間帯	0.9048	0.7857	0.8410

表 1,
表 2 を見ると，全ての手法において，パラメータの調整を行った後の方が高い F 値を得られていることが確認できる．

5. 提案手法

5.1 概要

本項では，乗っ取りツイートの検出に最適化された手法に対し，新たにハッシュタグの種類及びリプライを送る相手を特徴量として用いた重み付けを行う手法について説明する．具体的には，ハッシュタグの種類を用いた重み付けを行う手法，リプライを送る相手の ID を用いた重み付けを行う手法，ハッシュタグの種類及びリプライを送る相手の ID を同時に用いた重み付けを行う手法の合計 3 種類の手法を提案する．なお，本節で説明する 3 種類の重み付け手法はいずれも，4.3 項における実験において F 値が最高値となった，ツイートを投稿したクライアントの種類及び投稿時間帯を重み付けに用いた手法に対し，追加で重み付けを行い，更なる F 値の向上を図るものである．

提案手法を用いて，対象のアカウントの新着ツイートをアカウント所持者が投稿した通常のツイートであるか，あるいは乗っ取りツイートであるかを判定するシステムの概要図を図 4 に示す.

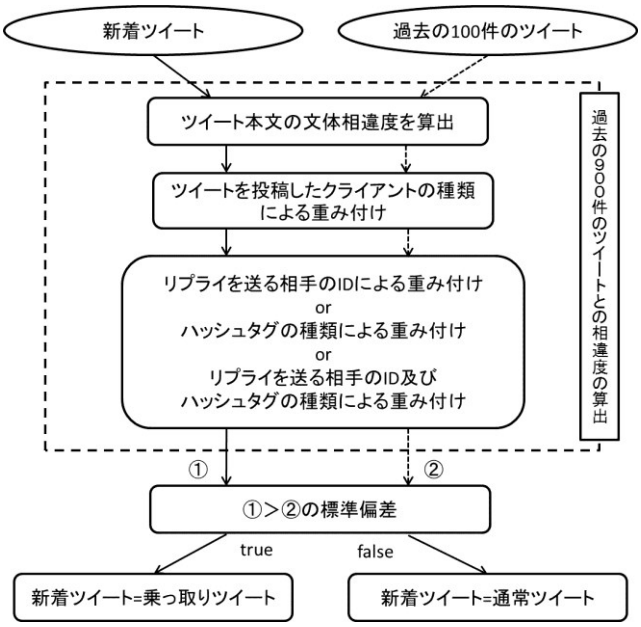

図 4　システム概要図

ハッシュタグは「#文字列」のフォーマットでツイート内に書き込むだけで新たなものを作成できるため，その種類は無数に存在する．しかし，同一アカウントが複数回用いるハッシュタグの種類は限られている．そこで我々は，新着ツイート集合**B**に含まれるハッシュタグと同一のハッシュタグがアカウントの所持者の過去のツイートに含まれていた場合に，文体相違度の値を小さくするように重み付けを行う手法を提案する．

また 1 節で述べたように，Twitter のユーザは 2012 年 6 月の時点で 5 億人を超えており，リプライの相手の種類はアカウントの数だけ存在する．しかし，ツイートに含まれるハッシュタグの種類と同様，同一アカウントがリプライを複数回送る相手は限られている．そこで我々は，新着ツイートb_kのリプライの相手と同一の相手に対するリプライ

が過去のアカウント所持者のツイートに含まれていた場合に，文体相違度の値を小さくするように重み付けを行う手法を提案する．

5.2 ハッシュタグの種類を用いた重み付け手法

本項では，ツイートに含まれるハッシュタグの種類を用いた重み付けの手法について説明する．ツイートb_kに含まれるハッシュタグをhとし，ツイート集合 **A** 内に含まれる，ハッシュタグhを含むツイートの割合を$P(h)$としたとき，ハッシュタグの種類を用いた重み$HashtagWeight(h)$を式(15)によって算出する．

$$HashtagWeight(h) = \begin{cases} 0.3(1 - P(h)) & (P(h) > 0) \\ 1.0 & (P(h) = 0) \end{cases} \quad (15)$$

式(15)における係数 0.3 は，4.1 項で説明した方法と同様の方法で実験的に求めた最適な値を用いている．ツイートを投稿したクライアントの種類及び投稿時間帯を用いた手法に対してこの重みを用いるため，式(12)によって求めたツイートを投稿したクライアントの種類及び投稿時間帯を用いた重み$Weight(q,t)$及び式(15)によって求めたツイートに含まれるハッシュタグの種類を用いた重み$HashtagWeight(h)$を用いて，重み付き文体相違度$Score(\boldsymbol{A}, a_j')$を式(16)によって算出する．

$$Score(\boldsymbol{A}, a_j') = Dissim(\boldsymbol{A}, a_j') * Weight(q,t) * HashtagWeight(h) \quad (16)$$

式(16)で算出した重み付き文体相違度$Score(\boldsymbol{A}, a_j')$の標準偏差σを用いて式(14)によって，アカウント所持者のツイートと乗っ取りツイートを分類する際に用いる閾値$\alpha(\boldsymbol{A}, \boldsymbol{A}')$を算出する．

5.3 リプライを送る相手の ID を用いた重み付け手法

本項では，リプライを送る相手の ID を用いた重み付け手法について説明する．ツイートb_kのリプライの相手の ID をrとし，ツイート集合 **A** 内に含まれる，rに対するリプライの割合を$p(r)$としたとき，リプライの相手の ID を用いた重み$ReplyWeight(r)$を式(17)によって算出する．

$$ReplyWeight(r) = \begin{cases} 0.2(1 - P(r)) & (P(r) > 0) \\ 1.0 & (P(r) = 0) \end{cases} \quad (17)$$

式(17)における係数 0.2 は，4.1 項で説明した方法と同様の方法で実験的に求めた最適な値を用いている．5.2 項と同様，ツイートを投稿したクライアントの種類及び投稿時間帯を用いた手法に対してこの重みを用いるため，式(12)によって求めたツイートを投稿したクライアントの種類及び投稿時間帯を用いた重み$Weight(q,t)$及び式(17)によって求めたリプライを送る相手を用いた重み$ReplyWeight(r)$を用いて，重み付き文体相違度$Score(\boldsymbol{A}, a_j')$を式(18)によって算出する．

$$Score(\boldsymbol{A}, a_j') = Dissim(\boldsymbol{A}, a_j') * Weight(q,t) * ReplyWeight(r) \quad (18)$$

式(18)で算出した重み付き文体相違度$Score(\boldsymbol{A}, a_j')$の標準偏差σを用いて式(14)によって，アカウントの所持者のツイートと乗っ取りツイートを分類する際に用いる閾値$\alpha(\boldsymbol{A}, \boldsymbol{A}')$を算出する．

5.4 ハッシュタグの種類及びリプライを送る相手の ID を用いた重み付け手法

本項では，5.2 項及び 5.3 項で説明した重み付け手法を同時に用いる手法について説明する．まずツイートに含まれるリプライを送る相手の ID を用いた重みを，5.3 項で説明した方法と同様に式(17)から求める．

次にハッシュタグの種類を用いた重み$HashtagWeight(h)$を求める．ここでは式(15)の係数を，4.1 項で説明したパラメータの最適化を行う方法と同様に最適化する．係数を 0.1~1.0 の間で 0.1 ずつ変化させ，最適化を行った結果，0.5 が最適な値であった．したがって，ハッシュタグの種類を用いた重み$HashtagWeight(h)$を式(19)によって算出する．

$$HashtagWeight(h) = \begin{cases} 0.5(1 - P(h)) & (P(h) > 0) \\ 1.0 & (P(h) = 0) \end{cases} \quad (19)$$

5.2 項及び 5.3 項と同様，ツイートを投稿したクライアントの種類及び投稿時間帯を用いた手法に対してこの重みを用いるため，式(12)によって求めたツイートを投稿したクライアントの種類及び投稿時間帯を用いた重み$Weight(q)$，式(17)によって求めたリプライを送る相手の ID を用いた重み$ReplyWeight(r)$及び式(19)によって求めたツイートに含まれるハッシュタグの種類を用いた重み$HashtagWeight(h)$を用いて，重み付き文体相違度$Score(\boldsymbol{A}, a_j')$を式(20)によって算出する．

$$Score(\boldsymbol{A}, a_j') = Dissim(\boldsymbol{A}, a_j') * Weight(q,t) * HashtagWeight(h) * ReplyWeight(r) \quad (20)$$

式(20)で算出した重み付き文体相違度$Score(\boldsymbol{A}, a_j')$の標準偏差σを用いて式(14)によって，アカウントの所持者のツイートと乗っ取りツイートを分類する際に用いる閾値$\alpha(\boldsymbol{A}, \boldsymbol{A}')$を算出する．

6. 評価実験

本節では，5 節で説明した 3 種類の提案手法を用いて乗っ取りツイートの検出を行い，各手法の評価を行う．

6.1 評価実験に用いるデータ

本項では評価実験に用いるデータについて説明する．評価実験に用いるデータは 3.3 項で説明した方法で選択した 100 アカウントの直近の 1,030 ツイート及び 30 件の乗っ取りツイートである．アカウントの所持者による 30 件の直近のツイート及び 30 件の乗っ取りツイートに対し，アカウントの所持者の過去の 1,000 ツイートの特徴を基に，分類を行う．

6.2 実験結果の評価

本項では，実験の結果を示し，評価を行う．5 節で説明した 3 種類の提案手法に対して評価実験を行ったときの検

出結果及び提案手法である重み付けを行う元の手法である，クライアントの種類及び投稿時間帯による重み付けを用いた手法に対して実験を行ったときの検出結果を表 3 に示す．

表 3　提案手法及び提案手法の元となる手法の実験結果

手法	適合率	再現率	F 値
文体相違度×クライアントの種類及び投稿時間（元の手法）	0.9048	0.7857	0.8410
文体相違度×クライアントの種類及び投稿時間×ハッシュタグの種類	0.9062	0.7860	0.8418
文体相違度×クライアントの種類及び投稿時間帯×リプライの相手	0.8553	0.7933	0.8553
文体相違度×クライアントの種類及び投稿時間帯×リプライの相手×ハッシュタグの種類	0.8156	0.9300	0.8570

表 3 を見ると，提案手法の重み付けを行う元の手法である，ツイートを投稿したクライアントの種類及び投稿時間による重み付けを用いた手法に対し，3 種類の提案手法による重み付けを行った場合に，いずれの提案手法においても F 値が向上していることが確認できる．また提案手法のうち最高の F 値を得られた手法は，リプライを送る相手，ハッシュタグの種類の両方を特徴量として追加し，重み付けを行った手法である．

7. まとめ

本稿では, 既存の Twitter におけるスパムツイート検出に関する研究では扱われてこなかった，アカウントの所持者以外のユーザによるツイートを検出対象とした，乗っ取りツイートの検出手法を提案した．また本稿で説明した提案手法を用いることで, 最高で 0.8570 の F 値で乗っ取りツイートの検出を行うことが可能であることがわかった．これは，我々の従来手法[3]に対してパラメータの再調整を行い，乗っ取りツイートの検出に最適化すると共に，複数回用いるハッシュタグの種類及び複数回リプライを送る相手がアカウントによって特徴的であることを効果的に利用しているためである．

本研究の課題としては，手法の評価方法の検討が挙げられる．本稿では，より F 値の高い手法の提案を目的としていた. しかし, 実際に乗っ取りツイートを検出する際には，アカウントの所持者によるツイートを乗っ取りツイートとして誤分類する場合よりも，乗っ取りツイートをアカウントの所持者によるツイートとして誤分類する場合の方がユーザに対してより大きな不利益をもたらすことが考えられる．したがって，F 値のみでなく，適合率及び再現率の数値にも注目し，乗っ取りツイートの検出という目的により適合した方法で評価を行う必要がある．したがって今後の研究では，より適切な評価方法を考案した上で，その評価方法に基づいて，さらに高い評価を得られる手法を検討していくことが求められる．

参考文献

1) Twitter reaches half a billion accounts More than 140 million in the U.S., http://semiocast.com/en/publications/2012_07_30_Twitter_reaches_half_a_billion_accounts_140m_in_the_US. (2013 年 10 月 21 日アクセス)
2) C. Grier, K. Thomas, V. Paxspm, and M. Zhang :“@spam: The Underground on 140 Characters or Less”, Proc of the 17th ACM conference on Computer and communications security, pp.27-37, 2010.
3) 和田なぎさ，奥谷貴志，山名早人: ”Twitter におけるアカウント乗っ取りによるスパムツイートの検出”, DEIM 5th, 2010.
4) 中島泰，山名早人:“品詞と助詞の出現パターンを用いた類似著者の推定とコミュニティ抽出”, DEIM 6th, 2011.
5) 松浦司，金田康正:“近代日本文学者 8 人による文章における文字 n-gram の分布を利用した近代日本語分の著者推定”，計量国語学, Vol.22, No.6, pp.1-9, 2000.
6) F.Benevenuto, G. Magno, T. Rodrigues, and V.Almeida :“Detecting Spammers on Twitter”, Proc of the Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS), 2010.
7) A. Wang:“Don’t follow me: Spam detection in twitter”， Proc of the 2010 International Conference on Security and Cryptography (SECRYPT), pp.1-10, 2010.