Vol.2014-DBS-159 No.12
Vol.2014-IFAT-115 No.12
2014/8/2

マイクロブログを対象とした著者推定手法の提案 －10,000 人レベルでの著者推定－

奥野峻弥[†1]　浅井洋樹[†2]　山名早人[†3]

従来，著者推定研究は小説に対する著者推定を中心に研究が行われており，推定対象を限定した，少人数に対する著者候補者群が取り扱われてきた．これに対し，我々はマイクロブログを対象にした，不特定多数の候補者群に対する著者推定の提案を行った．その際，精度向上のためマイクロブログ特有の叫喚フレーズに対する正規化手法，および計算量削減のため推定に必要となるメッセージ数を削減する手法を提案してきた．本稿では，より多くのマイクロブログ利用者を対象にした著者推定を行う上での問題点，特に学習用データとテストデータの取得期間の差異が精度に与える影響について検証し，学習用データの取得期間が精度に与える影響を小さくする手法を提案する．実験では Twitter ユーザ 10,000 人に対して著者推定を行い，Precision@1 で 0.535，MRR で 0.602 を達成した．

1. はじめに

既存の著者推定手法[1][2][3]は小説などの文学的文章における著者推定を実現し，近年インターネットに投稿された日本語の文章に対して著者推定が応用[4][5][6]されている．このような文章の著者を推定する際には，大規模人数の著者候補者群に対して著者を推定する必要がある．なぜならば，インターネットに文章を投稿する著者は不特定多数であり，少人数に限定できないためである．そこで，我々はこれまでに掲示板ユーザに対する 10,000 人レベルでの著者推定手法[7]，およびマイクロブログユーザに対する 5,000 人レベルでの著者推定手法[13]を提案してきた．

しかし，これまでに提案してきた推定手法[13]においては，推定に用いる文章群が投稿された時間帯についての考慮を行ってこなかった．具体的には，著者推定対象となる文章群と，候補者ごとに予め用意した，各候補者の文体についての特徴量を取得するための文章群が投稿された時間の差が大きいほど，著者推定の精度は減少すると考えられるが，その低下についての検証を行っていない．実際にマイクロブログのデータを用いた著者推定を行う場合，推定対象となる各候補者のメッセージが，常に推定対象となるメッセージ群と近い時間帯に投稿されたものとは限らないため，時間による著者推定精度の低下を考慮することは必須である．

本稿では，上記の問題についての考察，および投稿時間が異なる複数の学習データセットを用いた著者推定手法を提案する．具体的には，学習データを作成する際に，テストデータに含まれるメッセージの投稿時刻から一定の期間離れているメッセージのみを用いて学習データを作成する．つまり，候補者ごとに，テストデータに含まれるメッセージの投稿時刻に比べ，１週間以上前に投稿されたメッセージのみを用いた文章群を準備する，同様に，1 ヶ月以上前に投稿されたメッセージのみを用いた文章群を準備する，というようになる．時間帯の異なる複数の文章群を用いた著者推定を行うことで．既存手法よりも高精度の著者推定を実現する．

本稿では以下の構成をとる．まず 2 節では，著者推定研究で取り扱われてきた著者推定タスクについて述べる．次の 3 節では，既存の著者推定手法について述べる．続く 4 節では，本稿で提案する著者推定手法について述べる．そして，5 節にて既存手法と提案手法とに対する評価実験の方法と結果について述べる．最後に 6 節で本稿をまとめる．

2. 著者推定タスク

著者推定とは，推定対象文章における文体の特徴から，その文章の著者を推定することである．推定対象文章とは，著者を推定する対象となる，著者不明の文章のことである．なお，本稿では日本語の推定対象文章を対象とした著者推定を取り扱う．また，本節で説明する著者推定タスクとは，数多く存在する著者推定手法を抽象化したものである．

従来，著者推定は文学研究[7][8][9]で行われてきたが，近年ではテキストマイニング技術を用いた著者推定の手法[1][3][6]が提案されている．これらの手法は計算機上で容易に実装可能であることから，インターネットに投稿された文章の著者推定に応用[4][5][6]されている．計算機によるテキストマイニングによる著者推定手法の研究では，著者推定手法をもって著者推定タスクを行い，この結果によって当手法の評価を行う．本節では，著者推定タスクについて，その内容と結果からの評価方法について述べていく．

2.1 著者推定タスクの分類

Stamatatos[10] は著者推定タスクを Profiled-Based Approach（PBA）と Instance-Based Approach（IBA）の 2 種類に分類した．本稿では，大規模候補者群に対する著者推定を行うため，PBA による著者推定タスクを取り扱う．これは，大規模候補者群に対する著者推定では，IBA による

†1 早稲田大学大学院 基幹理工学研究科
†2 早稲田大学大学院 基幹理工学研究科
早稲田大学グローバルエデュケーションセンター
†3 早稲田大学理工学術院，国立情報学研究所

著者推定タスクに 2.1.2 で示す問題があるためである．

2.1.1 PBA 及び IBA による著者推定タスク

PBA による著者推定タスクでは，事前に用意されている候補者の文章群と，推定対象文章を順に比較する．比較された候補者群の中から，推定対象文章の著者と文体が最も類似する候補者を得ることで，各著者推定手法は著者推定を行う．PBA に分類される著者推定タスクは，松浦ら[1], 安形ら[2], 中島ら[6], 及び井上ら[7][8]が取り扱っている．

一方で，IBA による著者推定タスクでは，機械学習により各候補者の文章群を学習し，推定対象文章を各候補者のいずれかに分類する．推定対象文章の分類先となる候補者を得ることで，各著者推定手法は著者推定を行う．IBA に分類される著者推定タスクは，金ら[3]，坪井ら[11]が取り扱っている．

2.1.2 IBA による著者推定タスクの問題点

大規模候補者群に対する著者推定における IBA の著者推定タスクでは，当該著者推定タスクにおける機械学習が上手く機能しない．これは，IBA の著者推定タスクで用いる学習データが不均衡データであるために起こる[12]．不均衡データとは，正例と負例の数に極端な差がある学習データを指す．IBA における著者推定タスクでは，学習データ中の文章群を，特定の 1 人の候補者の文章である正例，それ以外の複数候補者の文章である負例の 2 つに分類する．しかし，一般に負例を集めることは容易であるが，正例を多く集めることは困難である．このため， IBA における著者推定タスクでは，正例と負例の数に差が生まれ，学習データは不均衡データとなる．

不均衡データに対処するため，正例の数に合わせて負例の数を減らす，負例の数に合わせて正例の数を多くするといった対策が考えられる．しかし，前者の方法では学習が十分にできない問題が生じる．一方，後者の方法を講じることも難しい．これは，候補者ごとに集められる文章は数万文字の大量文章でなくてはならないが，マイクロブログを対象とした大規模候補者群においてこのような文章を 1 人の候補者に対し多く集めることは困難であるためである．

2.2 PBA による著者推定タスクの流れ

手順 1）学習データとテストデータの収集

学習データとは，著者が既知である文章群のことを指す．テストデータとは複数の推定対象文章を指す．ただし，著者推定タスクでは，推定したテストデータ中の文章の著者と実際の著者が同じであることを確かめるため，テストデータ中の文章の著者が既知であるものを用いる．また，テストデータ中の文章の著者は，学習データにおけるいずれかの文章の著者と同一であるとする．このような条件の下，著者推定の候補者群となる著者を決定した後，候補者ごとに学習データとテストデータの 2 種類の文章を取集する．

手順 2) 各文章の文体定量化

手順 1 で収集された学習データ及びテストデータ中のすべての文章に対して文体定量化を行う．文章の文体定量化とは，その文章の著者が持つ文体を，当該文章を用いて数値ベクトルに定量化することである．文体の定量化方法は，各著者推定手法によって異なる．

手順 3）各文章間の文体相違度計算

テストデータ中の文章ごとに，学習データ中の各文章との間の文体相違度をすべて計算する．2 つの文章間の文体相違度とは，各文章の著者の文体がどれほど異なるかを定量化したものである．2 つの文章間の文体相違度は，手順 2 で得られる定量化された文体を用いて算出される．文体相違度をどのように算出するかは，各著者推定手法によって異なる．

手順 4）文体類似度順位の算出

テストデータ中の文章ごとに文体類似度順位を算出する．文体類似度順位とは，文体相違度の低い順に候補者群を並び替えたとき，推定対象文章の著者が何位に順位付けされたかを表す．

手順 5）著者推定手法の評価

手順 4 で得られたテストデータ中の各文章に対する文体類似度順位に基づいて，手順 2 及び手順 3 で用いた著者推定手法の評価を行う．得られた文体類似度順位からどのように著者推定手法を評価するかは，著者推定手法評価方法によって異なる．

2.3 評価方法

既存の著者推定研究[1][2][3][5][6][7]で行われる著者推定手法の評価は，2.2 で述べた著者推定タスクの手順 5 において，テストデータ中の文章群の中で文体類似度順位が 1 位となる文章の割合である，PRECISION@1 を指標として評価を行ってきた．これは，テストデータ中の各文章に対して著者推定を行うとき，著者推定タスクの手順 4 で並び替えられる候補者群において 1 位となる候補者を推定対象文章の著者であると推定するためである．

井上ら[7]は大規模候補者群に対する著者推定手法評価方法として，文体類似度順位の累積相対度数分布を定量的に評価する MRR 及び，正解が上位 k 件以内に入っていれば 1 と，そうでなければ 0 としてその平均をとる mean top-k call を評価方法として用いた．具体的には，MRR については式(1)によって算出される．ここで，Q はテストデータ中の文章の著者の集合，N_qは出力される候補者群順列中における候補者の順位である．

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{q \in Q}\frac{1}{N_q} \quad (1)$$

井上らが MRR 及び mean top-k call による評価方法を用いたのは，著者推定タスクにおける候補者群の並び替えにおいて，実際の著者が 1 位に順位付けされているかだけでなく，上位に順位付けされているかを評価するためである．これは，誤った推定をしない著者推定手法が存在しない以上，推定結果を実用するためには複数の候補から人手によ

って選択することが要求されるためである．特に，推定精度低下が顕著となる大規模候補者群に対する著者推定では，人手による確認が要求される．人手による推定を行う際は，複数の推定結果から著者を精査することで，正しい著者推定を行うことができる．しかし，そのためには 2 位以降の上位に正解が含まれていなければならない．よって，大規模候補者群に対する著者推定の評価には，MRR による評価方法が適しているといえる．

3. 従来の著者推定手法

3.1 大規模候補者群に対する著者推定手法

我々は以前，マイクロブログの文章を用いた 5,000 人レベルの候補者群に対する著者推定手法[13]を提案した．その際の文体定量化手法として，井上ら[7]が提案した，品詞タグ・文字混合 n-gram 頻度分布を用いている．ここで，品詞タグ・文字混合 n-gram とは，文章を文字または品詞タグの羅列に変換したときに，当該羅列中に存在する n 個の連続した要素順列を指す．

我々の手法で用いる文章中の文体定量化は，文章 p 中における品詞タグ・文字混合 n-gram x の生起回数d_{px}の集合D_pを得ることで行う．文章を文字または品詞タグの羅列に変換するために以下の手順をとる．まず，形態素解析器を用いて文章を形態素に分割する．なお，形態素解析器は lucene-gosen [a]を用いている．次に，「動詞」「接続詞」「記号」「副詞」「形容詞」「感動詞」「未知語」の形態素については，文字列をそのまま採用し，これら 6 種類の品詞以外について品詞タグを用いる．

井上らが提案する著者推定タスクにおける文体相違度計算では，文章 p および q についてのD_p, D_qだけではなく，C_{pq}およびa_pを用いる．C_{pq}は，文章 p と文章 q の各々に存在するすべての品詞タグ・文字混合 n-gram の和集合である．a_pは，文章 p を構成する記事の数である．記事とは，マイクロブログにおける 1 件のメッセージのように，一度に投稿する文のまとまりを指す．井上らはC_{pq}, D_pおよびD_pを用いることで，2 つの文章 p, q における文体相違度$Dissim_{pos}$を以下のように定義している．

$$Dissim_{pos}(p,q) = \frac{\sqrt{\sum_{i\in C_{pq}}\left(f_{pi}-\overline{f_{pq}}\right)^2}\sqrt{\sum_{i\in C_{pq}}\left(f_{qi}-\overline{f_{qp}}\right)^2}}{\sum_{i\in C_{pq}}\left(f_{pi}-\overline{f_{pq}}\right)\left(f_{qi}-\overline{f_{qp}}\right)} \quad (2)$$

$$\overline{f_{pq}} = \frac{\sum_{i\in C_{pq}} f_{pi}}{|C_{pq}|} \quad (3)$$

$$f_{pi} = \begin{cases} 0.4 & (f'_{pi} > 0.4) \\ f'_{pi} & (f'_{pi} \le 0.4) \end{cases} \quad (4)$$

$$f'_{pi} = \frac{d_{pi}}{a_p} \quad (5)$$

文体相違度$Dissim_{pos}$は，その値が小さいほど 2 つの文章 p, q の文体が似ていることを表す．

さらに，我々は浅井ら[9]が提案した，マイクロブログ上で投稿される突発的な感情を表わす「叫喚フレーズ」と呼ばれる表現に着目し，それらの表現を除去することによる推定精度向上を試みた[13]．

叫喚フレーズを以下のようになっている．

- 語尾の母音が 3 回以上繰り返して付加されている
- 母音は大文字，小文字を区別しない
- 母音はひらがな，カタカナの大小文字すべて

この定義から，我々は以下の正規表現に基づいて，叫喚フレーズの含まれるメッセージの正規化を行った．

`[あ|ぁ|ア|ァ]{3,}|[い|ぃ|イ|ィ]{3,}|[う|ぅ|ウ|ゥ]{3,}|[え|ぇ|エ|ェ]{3,}|[お|ぉ|オ|ォ]{3,}`

具体的には，以下の手順のようになる．

1. 叫喚フレーズの含まれる文章を，本項で説明した正規表現を用いて抽出する．
   例) うわあぁあどうしようぅぅぅう
2. 繰り返される母音を大文字化する．
   例) うわあああどうしようううううう
3. すべての繰り返される母音部分に対して，母音一文字とそれ以前の文字列を削除する．
   例) うわあどうしよう

3.2 既存手法の問題点

3.1 で説明した，我々がこれまでに発表したマイクロブログのデータを用いた大規模候補者群に対する著者推定[13]では，推定に用いるメッセージの投稿時刻についての考慮をしてこなかった．つまり，学習データに用いる各候補者のメッセージが，テストデータに用いるメッセージからどの程度時間的に離れたものであるのか，またそのことによる推定精度の低下についての議論を行っていない．

一般に，マイクロブログに投稿されるメッセージでは，投稿されるメッセージの話題は多岐に渡る．投稿時刻が近いメッセージは近い話題について，投稿時刻が離れているメッセージについては異なる話題について述べていると考えられる．しかし，これまでの我々の手法では，話題に敏感な特徴を除去しきれていないことがわかっている．そのため，テストデータおよび学習データ内に用いるメッセージを選択する際に，投稿時刻が近いものを選択することで，改めてメッセージ中に含まれる，話題に敏感な特徴の除去についての議論を行わなくてはならない．

また，これまでの我々の手法では，各ユーザの文体は，時間経過によって変化することはないという前提での著者推定を行っている．そこで，テストデータに含まれるメッセージ，および学習データに使用するメッセージの選択を行う際，投稿時刻が離れているメッセージを選択することで，時間経過による著者の文体の変化についての議論を行わなくてはならない．

a lucene-gosen, https://code.google.com/p/lucene-gosen/

Vol.2014-DBS-159 No.12
Vol.2014-IFAT-115 No.12
2014/8/2

4. 提案手法

4.1 概要

マイクロブログユーザの文体が時間経過によって変化した場合も精度よく著者推定を行うための手法を提案する．具体的には，時間経過によって文体が変化することを考慮し，各ユーザの学習データセットについて，投稿期間を変えたものを複数用意し，テストデータセットとの間で各々文体相違度の算出を行う．そして，テストデータセットと最も文体が似ている学習データセットとの文体相違度（最も小さい文体相違度）を用いて著者推定を行う．これによって，ユーザがたまたま学習データ取得時に通常とは異なるメッセージの投稿を行っていた場合も，当該学習データを用いることなく，代わりに他期間に取得した学習データを利用できるため，精度向上が実現できると考えられる．なお，比較のため複数の学習データセットとテストデータとの間の文体相違度の平均を用いる手法との比較も行う．

4.2 学習データとテストデータの作成

学習データ及びテストデータを作成する手順は以下に示す通りである．収集した全てのメッセージをD_{all}とする．文字列として採用する品詞群集合を P とし，以降 P を文字列採用品詞群と呼ぶ．また，各ユーザについてのテストデータ及び学習データセットを作成する際に用いるメッセージ数を k とする．さらに，ここでは n 名に対する著者推定タスクを行うものとする．

step 1. ユーザ ID 集合を UID とし，UID=∅とする．

step 2. D_{all}から，ランダムに 1 つのユーザ ID u_iを抽出し，step3 から step8 を適用した後，ユーザ ID 集合である UID に追加する．これを｜UID｜=n になるまで繰り返す．

step 3. u_iが投稿したメッセージを，投稿時刻を用いて降順に並び替える．

step 4. u_iが投稿したメッセージのうち，図 1 に示すように投稿時刻が最新のものから k 件を選択し，$T^{u_i}_{test}$とする．なお，k 件選択できない場合は，step2 に戻る．

step 5. step 4 で選択したメッセージのうち，最も投稿時刻が古いメッセージが投稿された時刻をt_{last}とおく．

step 6. u_iが投稿したメッセージのうち，t_{last}から(t_{last}—1 週間)の間に投稿されたメッセージのうち，投稿時刻が新しいものから順に k 件を選択し，$T^{u_i}_{train,1}$とする．ここで，k 件選択できない場合は，step2 に戻る．

step 7. (t_{last}—1 週間) から(t_{last}—1 ヵ月)の間に投稿されたメッセージのうち，投稿時刻が新しいものから順に k 件を選択し，$T^{u_i}_{train,2}$とする．先と同様に，k 件選択できない場合は，step2 に戻る．

step 8. (t_{last}—1 ヵ月)から(t_{last}—3 ヶ月)の間に投稿されたメッセージのうち，投稿時刻が新しいものから順に k 件を選択し，$T^{u_i}_{train,3}$とする．同様に k 件選択できない場合は，step2 に戻る．

step 9. $T^{u_i}_{test}$，$T^{u_i}_{train,1}$，$T^{u_i}_{train,2}$，及び$T^{u_i}_{train,3}$に対して叫喚フレーズの正規化を行うと共に形態素解析を行い，メッセージに含まれる各形態素を品詞もしくは文字列の混合列に変換する．ここで，文字列採用品詞群 P に含まれる品詞に変換される形態素については文字列を採用する．変換についての具体例を図 2 に示す．

以上により生成されてた$T^{u_i}_{test}$は，ユーザ ID u_iを持つユーザについてのテストデータとなり，$T^{u_i}_{train,1}$，$T^{u_i}_{train,2}$及び$T^{u_i}_{train,3}$はそれぞれユーザ ID u_iを持つユーザについての学習データとなる．つまり，各ユーザは k 件のメッセージをそれぞれ品詞もしくは文字列の混合列に変換したものからなるデータセットである$T^{u_i}_{test}$を 1 つと，k 件のメッセージをそれぞれ品詞もしくは文字列の混合列に変換したものからなるデータセットである$T^{u_i}_{train,1}$, $T^{u_i}_{train,2}$および$T^{u_i}_{train,3}$の 3 つのデータセットの組を 1 つ持つこととなる．

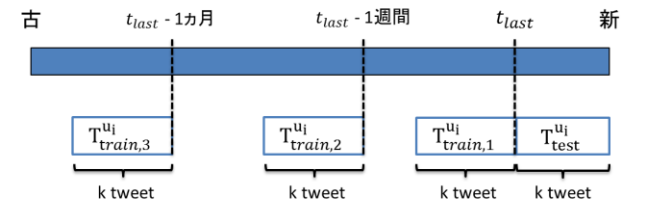

図 1 学習データセットにおけるメッセージの選択

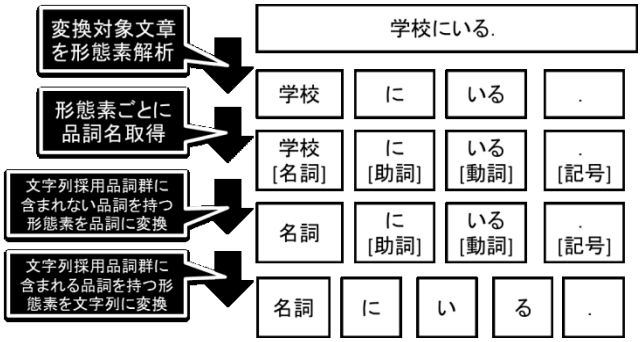

図 2 P={動詞，助詞，記号}のときの変換例

4.3 文体相違度の決定手法

文体相違度の計算は，これまでの我々の手法[13]の方法と同様に以下の方法により行う．すなわち，$T^{u_i}_{train,1}$, $T^{u_i}_{train,2}$および$T^{u_i}_{train,3}$に対し，テストデータセット$T^{u_i}_{test}$に対する文体相違度を計算する．

- 各学習データセットから得られた文体相違度のうち，最も数値が小さなものを$Dissim_{top}$とし，これを文体類似度順位算出時に用いる文体相違度として採用する．
- 各学習データセットから得られた文体相違度の相加平均を算出し，$Dissim_{average}$とし，これを文体類似度順位算出時に用いる文体相違度として採用する．

5. 評価実験

評価実験では，既存手法としては我々がこれまでに用いてきた著者推定手法[13]を用い，提案手法との比較を行う．

5.1 実験環境

本稿では Twitter から収集した tweet をデータセットとして用いた．データセットの概要は以下の通りである．

- データ収集期間: 2013 年 1 月-12 月
- 総収集 tweet 数: 7,955,714 名×最大 2,000 件

本実験で使用するデータセットに含まれるすべてのメッセージには，そのメッセージを投稿したユーザに固有の情報である，ユーザ ID が付随する．ここで，Twitter を代表とするマイクロブログにおいては，引用やアプリによる投稿など，アカウントを所持するユーザ以外によるメッセージの投稿が頻繁に行われる．我々の手法では，メッセージを記述した人物の文体を特徴量として用いるため，当ユーザ以外によって投稿されたメッセージは全て除く必要がある．そのため，前処理としてデータセット内のメッセージに含まれるメンション（@username），ハッシュタグ(#hashtag)，他人の文章であるリツイート(RT)をデータセットから除去した． また，各メッセージについて，メッセージに付随するクライアントアプリについての情報から，bot による投稿など，ユーザ以外の文章であると判断したものについては除外を行った．

また，評価実験では形態素解析器として lucene-gosen を利用する．辞書については，IPAdic[b]のライセンス問題を解決した NAIST-Japanese Dictionary[c]を形態素解析に用いる基本の辞書とする． NAIST-Japanese Dictionary は IPA 品詞体系に基づく辞書であるため，本実験での品詞体系は IPA 品詞体系に依存したものとなる．

5.2 評価実験全体の流れ

4.2 で作成した学習データとテストデータの組について，著者推定タスクにおける手順 2 と手順 3 の方法で文体相違度を算出する．文体相違度算出には，3.1 で説明した手法を用いる．また，文体定量化手法に関しては，表 21 に示す我々が提案してきた手法[13]を用いる．

表 1 評価実験の対象となる著者推定手法

手法名	文字列採用品詞群	頻度分布
提案手法	「動詞」「接続詞」「記号」「副詞」「形容詞」「感動詞」「未知語」	2gram 頻度分布

次に，テストデータ中のすべての文章に対して，著者推定タスクにおける手順 4 より，4.3 で示した手順を用いて文体類似度順位を算出し，MRR を算出する．

5.3 実験結果の評価

評価では，n=10,000，k=50 として，4.2 の手順を用いてテストデータ及び学習データをそれぞれ作成した. ここで，k=50 としたのは, 文献[13]での結果との比較のためである．評価では，MRR を用いた．MRR は，文体類似度順位の累積相対度数分布を定量的に評価したものであり，すべてのテストデータにおいて文体類似度順位が高くなるときに，MRR の値も高くなる．

MRR を用いた評価結果を表 2 に示す．ここで， 提案手法における Top，Ave は，それぞれ 4.3 で説明した文体相違度決定手法を用いた際の MRR を示す．つまり，Top は 4.3 における$\mathrm{Dissim}_{\mathrm{top}}$を採用した場合，Ave は 4.3 における $\mathrm{Dissim}_{\mathrm{average}}$を採用した場合の推定結果である．同様に，Train1，Train2，Train3 については，$\mathrm{T}^{\mathrm{u_i}}_{\mathrm{train},1}$, $\mathrm{T}^{\mathrm{u_i}}_{\mathrm{train},2}$および $\mathrm{T}^{\mathrm{u_i}}_{\mathrm{train},3}$により算出された文体相違度のみを用いて算出した MRR を示す．

表 2 MRR による評価結果

	提案手法		既存手法		
	Top	Ave	Train1	Train2	Train3
MRR	0.602	0.560	0.539	0.445	0.353

表 2 の結果から, 既存手法においては, Train1 から Train3 と，テストデータから投稿時刻が離れるにつれ，MRR が低下していくことがわかる．しかし，実験の結果から，提案手法における Top が既存手法を上回る，最も高い MRR を持つことが分かった．仮に，テストデータと投稿時刻が離れるにつれ，常に推定精度が低下していくと仮定すると，テストデータから投稿時刻が最も離れた学習データセットを用いた文体相違度を併用している Top の推定精度は低下することになる．しかし，実験結果はそのようになっていない．つまり，投稿時刻がテストデータに対して最も近い学習データセットを用いた場合の推定精度に比べ，投稿時刻がテストデータから離れたデータセットを用いた場合の推定精度が高いメッセージを投稿しているユーザが存在することを示しており，提案手法の有効性が確認できた．

5.4 さらなる推定精度改善に向けて

上記の結果が得られたのは，著者（Twitter ユーザ）のメッセージの文体が時間と共に変化したことが原因であると考えられるが，より詳細にその原因を調べるため，投稿時刻がテストデータに対して最も近い学習データセットを用いた場合の文体類似度順位に比べ，投稿時刻がテストデータから離れたデータセットを用いた場合の文体類似度順位が高い結果を得た学習データセットを持つユーザのメッセージを目視により確認した．その結果，次の 2 点がわかった．

① 学習用に用いたデータセットに含まれる総文字数が少ないため，特徴量を十分に取得できず，特定の学習

b IPADic legacy, http://sourceforge.jp/projects/ipadic/
c NAIST-Japanese Dictionary, http://sourceforge.jp/projects/naist-jdic/

データセットに対して精度が向上しない.

② メッセージ中に含まれる顔文字について, 形態素解析器による分割を十分な精度で行うことができず, 推定において副作用が発生している. このため, 顔文字を多く含む学習用データセットを用いた場合の精度が向上しない.

これら①及び②は, 著者（Twitter ユーザ）のメッセージの文体が, 時間と共に変化したことに起因している. しかし, これらの影響をできるだけ排除するための文体相違度計算手法を考えることも重要である. そこで, 以下では①と②についての対策を考えた.

まず①の理由により, テストデータセットと学習用データセットとの間で文字数が大きく異なる場合が発生する. これはすなわち、我々が定義している文体相違度計算に影響を及ぼすことを意味する. これを避ける一手法として, 文体特徴量を増加させ, 文字数に起因して特徴量が大きく影響を受けないようにすることを考える. 具体的には, 品詞タグ・文字混合 n-gram 頻度分布によるメッセージの文体定量化を行う際に, 複数の n-gram を併用することを考える.

実際に, n=100 の時, 品詞タグ・文字混合 n-gram について, 1-gram から 4-gram までの n-gram の組み合わせを用いて品詞タグ・文字混合 n-gram 頻度分布を取得し, MRR による評価を行った. その結果, 1-gram, 2-gram および 3-gram までを使用した際の MRR が 0.937 となり, 既存手法である 2-gram のみを使用した際の MRR である 0.894 を上回ることができた.

②の問題を避けるためには, 正確に顔文字を認識し, 1 つの顔文字を複数の特徴量に分割しないことが重要となる. そこで, 顔文字辞書を形態素解析器に追加し実験を行った. 実験で用いる顔文字辞書として, 顔文字ステーション[d]から提供されている顔文字辞書を用いた. また, 顔文字は著者の感情を表す記号であるため, 著者の文体ではなく話題に近い単語であると考え, 顔文字については品詞をフィラーとして扱い, 文体定量化の際には品詞タグとして扱うこととした. n=1,000 としたときの実験結果においては, 顔文字辞書を使用した際の MRR が 0.744, 使用しない場合の MRR が 0.736 となり, 若干の性能向上がみられた.

6. まとめ

本稿では, マイクロブログデータに対する著者推定手法について, 推定に使用するメッセージの投稿時間を考慮した手法の提案を行った. 本稿で提案した著者推定手法を用いることで, マイクロブログのデータを用いた大規模候補者群に対する著者推定において, より高精度の推定が行えることがわかった.

今後は, 5.4 で考察した内容について, より詳細に検討を行っていきたい.

謝辞 本研究の一部は科研・基盤（B）（No.25280113）によるものである.

d 顔文字ステーション, http://kaosute.net/jisyo/

参考文献

1. 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文の著者推定”, 計量国語学, Vol.22, No.6, pp.1-9, 2000.
2. 安形輝 : “圧縮プログラムを応用した著者推定”, J. of Library and Information Science, 三田図書館・情報学会, No.54, pp.1-18, 2005.
3. 金明哲, 村上征勝: “ランダムフォレスト法による文章の書き手の同定”, 統計数理, Vol.55, No.2, pp.255-268, 2007.
4. 石川尚季, 西村涼, 渡辺靖彦, 村田真樹,岡田至弘: “コミュニケーションサイトに投稿されたメッセージに対する著者の推定”, 信学技報(NLC) ,Vol.109, No.142, pp.79-84, 2009.
5. 佐藤進也, 原田昌紀, 風間一洋: “文字列出現頻度比較による情報源間の類似性判定”, 情処研報(DD), Vol.2002, No.28, pp.119-126, 2002.
6. 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”, DEIM2011, B6-5, 2011.
7. 井上雅翔, 山名早人: “大規模候補者群に対する著者推定手法の提案と評価”, DEIM2013, C6-6, 2013.
8. 井上雅翔, 山名早人: “品詞 n-gram を用いた著者推定手法 : 話題に対する頑健性の評価”, 日本データベース学会論文誌, Vol.10, No.3, pp.7-12, 2012.
9. 浅井洋樹, 秋岡明香, 山名早人: “きたあああああああああああああああああ！！！！！１１：マイクロブログを用いた教師なし叫喚フレーズ抽出”, DEIM2013, A4-1, 2013.
10. Stamatatos, E. : “A Survey of Modern Authorship Attribution Methods, J. of the American Society for Information Science and Technology”, Vol.60, No.3, pp.538-556, 2009.
11. 坪井祐太, 松本裕治: “異なるタイプのドキュメントに対する著者推定”, 情処研報(NL), Vol.2002, No.20, pp.17-24, 2002.
12. N.V. Chawla, N. Japkowicz and A. Kotcz : “Editorial: special issue on learning from imbalanced data sets”, ACM SIGKDD Explorations Newsletter, Vol.6, No.1, pp.1-6, 2004.
13. 奥野峻弥, 浅井洋樹, 山名早人: “マイクロブログを対象とした 5,000 人レベルでの著者推定手法の提案－5,000 人レベルでの著者推定”, WebDB Forum 2013, pp. 1-8, 2013.