ノイズに頑健な分野別単語排他度の提案と Twitter ユーザの専門性推定への適用

滝川 真弘† 山名 早人‡

†早稲田大学大学院基幹理工学研究科 〒169-8555 東京都新宿区大久保 3-4-1

‡早稲田大学理工学術院 〒169-8555 東京都新宿区大久保 3-4-1

E-mail: †‡{temy0501, yamana}@yama.info.waseda.ac.jp

あらまし 本研究の目標は，特定領域に対する著者の専門性を如何に短い文章から判定するかにある．例えば多くの Twitter ユーザは少数の tweet しかしておらず，再現率の高い「著者の専門性判定」を実現するには短い文章から判定することが求められる．この目標を達成するため，我々はこれまでに「特定の専門性と強く関連するが，他分野ではあまり使われない単語」に大きな重要度を付与する手法を提案している．本稿では，我々が提案した以前の手法の精度向上を目的に，ノイズとなる単語の重要度を下げる仕組みを導入する．ノイズは，本手法で用いている「専門性が高い文書のコーパス」と「一般文書のコーパス」が必ずしも完璧でないが故に発生する．ノイズを低減するため，各コーパス中の文書群にどのように単語が出現しているかという単語出現頻度の分散に着目した．Twitter を対象としたユーザの専門性推定に適用し評価したところ，従来手法に比較して，人手で作成した専門性に基づく正解ランキングとの相関係数を 0.08 向上させることができた．

キーワード 単語，重要度，専門用語，Twitter

1. はじめに

Twitter[1]は，代表的なソーシャル・ネットワーク・サービス(SNS)の一つであり，国内外問わず多くのユーザ数を抱えている．Twitter で用いられる投稿(ツイート)は，長さが 140 文字以内に限られ，多くのユーザは，有益なことから些細なことまで気軽に多くのツイートを行う．ツイートを対象とした研究は数多く存在し，ユーザの属性推定[1]や専門性の推定[2]など多くの研究が行われている．また，これらの研究ではツイートだけでなくフォロー関係やメンション関係を用いた研究まで幅広い．

こうした既存研究では，SVM などの機械学習や LDA などのトピックモデルが一般的に用いられているが，機械学習やトピックモデルを使用するには十分なデータ量が必要となる．しかし，Twitter を利用するユーザ全てが多くのツイートを行っているわけではない．我々の調査によれば，2013 年 1 月~12 月にツイートを日本語で発信した約 800 万人の Twitter ユーザの内，1 週間以内に 30 ツイート以上発信した Twitter ユーザは 12.5%である[3]．つまり，機械学習や LDA などが適用できる Twitter ユーザは限定される．

本稿では，こうしたツイート数が少ない場合にも有効に機能する手法として，一定数のツイートをまとめて 1 ドキュメントとして扱い解析するのではなく，1 ツイートを 1 ドキュメントとして扱い解析する手法に取り組む．そのためには 1 ツイート内の単語自体に適切な重み(重要度)を付与することが重要となる．具体的には，適切な重みは対象とする分野毎に異なることを前提に「特定分野を対象とした単語重要度の計算法」について提案する．

以前我々は，特定分野における単語重要度を「一般人が使わない単語であり，かつ特定分野の中でも出現頻度の低い単語がより重要度が高い」という仮説をもって単語重要度を付与する手法を提案した[4]．具体的には，予め専門辞書が与えられている時，当該専門辞書内の単語に重要度を付与することによって実現する．重要度付与にあたっては，特定分野コーパスと特定分野以外のコーパス（一般分野コーパス）を用い，一般分野コーパスにはほとんど出現せず，かつ特定分野コーパスにおいても出現頻度が低い単語に高い重要度を付与する手法である．いわゆる idf（逆文書頻度）と類似するが，逆文書頻度ではなく逆コーパス頻度と言える．その上で，特定分野コーパスにおいても低頻度出現する単語に高い重みを付与している．しかし，各コーパス内での単語出現頻度によって重み付けを行っているため，各コーパスに特異な文書が存在した場合，単語の重要度計算に大きな影響を与えることがわかった．

これに対して本稿では，一般分野コーパス中で特定文書のみにしか出現しない単語はノイズの可能性が高いことに注目する．具体的には，tf（単語頻度）のコーパス内での分散を用いる．一般分野コーパスにおいて tf の分散値が高い単語はノイズと考え重要度を下げる．

[1] http://twitter.com

一方，特定分野コーパスにおいて tf の分散値が高い単語は専門用語の中でも特殊な分野でのみ利用されると考え重要度を高くする．このようにして，ノイズとなりうる単語の重要度を下げる．

以下，2 節にて関連研究，3 節にて提案手法．4 節にて実験に使用するデータセット，5 節にて評価方法，6 節にて実験結果を示し，7 節にて本稿をまとめる．

2. 関連研究

出現頻度と分野(カテゴリ)の観点から，単語重要度を計算する手法について紹介する．なお，以下では，単体で意味を持つ最小単位を語基と定義し，語基一つから成るもの，また複数の語基から成り立つものを単語と定義する．

2.1. 単語重要性を測る手法

文章中に表れる単語の重要性を測る手法としては，TF-IDF[5]と Okapi BM25[6]が有名である．

TF-IDF[5]はドキュメントに索引を付ける際の重みづけを目的として考案された．あるドキュメント集合中に存在する一つのドキュメントにおける特徴的な単語を表現するために用いられる．単語 t のドキュメント d に対する重要度 w(t,d)は，式(2.1)により計算する．TF(Term Frequency)は単語出現頻度であり，式(2.2)の tf(t, d)は、単語 t のドキュメント d 内での出現頻度を示す．DF(Document Frequency)は，単語が出現するドキュメント頻度である．DF の逆数の値が IDF(Inverse Document Frequency)であり，この値が大きいと特定のドキュメントのみに出現する傾向が高いことを示す．idf(t)は，式(2.3)により計算する．

$$w(t,d) = tf(t,d) * idf(t) \tag{2.1}$$

$$tf(t,d) = \frac{n(t,\ d)}{\sum_k^K n(k,\ d)} \tag{2.2}$$

$$idf(t) = log\left(\frac{|D|}{df(t)}\right) \tag{2.3}$$

ここで，n(t, d)はドキュメント d 中の単語 t の出現回数，K は全単語集合，$n(t,\ d)$はドキュメント d の中に出現する全単語の出現回数の和，|D|はドキュメント数，df(t)は単語 t が現れるドキュメント d の数である．

Okapi BM25[6]は，情報検索の分野で TF-IDF よりも精度が高いとされる重み付け手法であり，TF-IDF を拡張したものである．ある単語の出現回数が同一である 2 つのドキュメントがある時，「ある単語の重要度は 2 つのドキュメントに対して同等ではなく，短い文章に対する重要度がより高くなる」という考えを TF-IDF に追加している．Okapi BM25[6]を用いたドキュメント d に対する単語 t の重要度 w(t,d)は以下の計算式で表せる．

$$w(t,d) = \frac{tf(t,d) * (k_1 + 1)}{tf(t,d) + k_1 * \left(1 - b + b * \frac{len(d)}{avgdl}\right)} * \log\frac{|D|}{df(t)} \tag{2.4}$$

ここで，len(d)はドキュメント d の長さ，avgdl は総ドキュメントの平均長を示す．k_1 と b は調整パラメータであり，エッセイ長に対して tf をどれだけ補正するかを表す．一般的に k_1 は 1.2 から 2.0 の値で，b は 0.75 で高い精度を示すことが知られている[7].

上記二手法は，ドキュメント群に対する 1 つのドキュメント内に存在する各単語の重要度を算出することにより，対象とするドキュメントの特徴語を抽出している．これらの手法は，文章の検索インデックスなどに使用することを目的としている．このため，ある分野おける単語重要度を算出することはできない．特定分野での重要度算出のためには，各ドキュメントが属する分野を考慮する必要がある．

2.2. カテゴリと単語の関係から重要度を計算する手法

本節では，カテゴリ(特定分野)が付与されたドキュメント集合について，カテゴリに対する単語の出現頻度の偏りから重要度を計算する従来手法について説明する．以下では，相互情報量，tf*rf[8]，tf*dc[9]，tf*bdc[9]の 4 つの手法について述べる．

相互情報量は，2 つの確率変数間の相互依存量を表す尺度である．ここで，単語 t のカテゴリ c に対する相互依存量を求めるとする．単語 t の文章集合全体での出現確率を P(t)，カテゴリ c の文章集合全体での出現確率を P(c)，P(t)と P(c)の同時確率を P(t,c)とする時，相互情報量 MI(t,c)は式(2.8)で表せる．しかし，相互情報量は，低頻度で出現する単語については必ずしも正しく計算することができない．例えば，出現回数が少ない場合，同単語のノイズ的な出現によって相互情報量が大きく変化するからである．

$$MI(t,c) = log\frac{P(t,c)}{P(t)P(c)} \tag{2.5}$$

これに対して，2009 年に Lan ら[8]は，あるドキュメントがカテゴリ C に属するか否かを推定することを目的として，tf*rf と呼ばれる単語重要度計算手法を提案した．同手法は，単語 t のドキュメント内での単語出現頻度 tf に加え，単語 t の出現が，あるカテゴリに属するドキュメント集合と当該カテゴリに属さないドキュメント集合でどれだけ異なるかを示す rf を用いる．具体的には，事前にカテゴリ C に属するドキュメント集合 Dp と属さないドキュメント集合 Dn を用意する．単語 t についての rf 値である rf(t)は，Dp 内で単語 t を含むドキュメント数を a, Dn 内で単語 t を含むドキュメント数を c とした時，式(2.9)で表される．

$$rf(t) = \log\left(2 + \frac{a}{\max(1, c)}\right) \tag{2.9}$$

なお，tf(t,d)は対象とするドキュメント d の中の単語 t の出現頻度であり，tf*idf の tf と同値であり，tf*rf は，

tf(t,d)と rf(t)の積により求める．

また，Wang ら[9]は 2015 年に tf*dc を提案した．Lan らと同様に，あるドキュメントがカテゴリ C に属するか否かを推定することを目的としている．Wang らの手法は，Lan らとは異なり，カテゴリが複数あることを想定している．tf*dc は，ドキュメント内での単語出現頻度 tf とカテゴリ毎のエントロピーから算出される dc を組み合わせている．

カテゴリ数|C|が i の時，用意するべきコーパスは DC_1, DC_2…DC_i となる．単語 t について，すべてのドキュメント群に出現する単語 t の出現数を f(t)，あるカテゴリ ci に属するドキュメント内に出現する単語 t の出現数をと$f(t,c\mathrm{R})$する時，dc(t)は式(2.10)で表せる．なお，H(t)はすべてのカテゴリにおける単語 t のエントロピーを示す．

$$dc(t) = 1 - \frac{H(t)}{\log(|C|)} \tag{2.10}$$
$$= 1 + \frac{\sum_{i=1}^{|C|} \frac{f(t,\ c_i)}{f(t)} log \frac{f(t,\ c_i)}{f(t)}}{\log(|C|)}$$

Wang らは同論文[9]で，tf*dc の他に tf*bdc を提案している．目的は同様であるが，bdc(t)は dc(t)とは異なり，エントロピーに加えて単語 t のカテゴリ ci 内における出現頻度を組み合わせた．カテゴリ ci に属するドキュメント数を$f(c\mathrm{R})$，カテゴリ ci に所属するドキュメント内に出現する単語 t の出現数を$f(t,\ c\mathrm{R})$とすると，bdc(t)は式(2.11)で表せる．なお，BH(t)はすべてのカテゴリにおける単語 t のエントロピーを平均したものを示す．

$$bdc(t) = 1 - \frac{BH(t)}{\log(|C|)} \tag{2.11}$$
$$= 1 + \frac{\sum_{i=1}^{|C|} \frac{p(t,c_i)}{\sum_{i=1}^{|C|} p(t,\ c_i)} log \frac{p(t,\ c_i)}{\sum_{i=1}^{|C|} p(t,\ c_i)}}{\log(|C|)}$$

$$p(t,c_i) = \frac{f(t,\ c_i)}{f(c_i)}$$

上記で述べた手法は全て，文書集合中のある文書を特徴づける(特定分野への片寄のある)単語の重みを大きくするものである．したがって，専門性を測るものではない．このため，特定分野にどれだけ精通しているかを判断することを目的として，単語に重要度を付与することができない．

そこで，以前我々は専門用語に対して，専門度を付与するための単語重要度計算手法[4]を提案した．以前提案した手法は，専門用語には一般人も使用する単語が含まれていることを想定し，一般人があまり用いない単語に高い重要度を付与する．また，特定分野にどれだけ精通しているかを判断することを目的としているため，該当分野に精通していないと知り得ない単語に高い重要度を付与する．このために，特定分野のコーパス Dp と一般分野のコーパス Dn を使用した．Dn にはほとんど出現せず，かつ Dp 内でも出現頻度が低い単語ほど重要であるという仮説の元提案した．実際の計算式を(2.12)に示す．

$$W(t) = MI(t,c) * TF_{MAX}(t) * \log\left(\frac{|D|}{DF'(t)}\right) \tag{2.12}$$

$$MI(t,c) = \frac{P(t,c_{D_p})}{p(t) * p(c_{D_p})} \tag{2.13}$$

$$TF_{MAX}(t) = \max_{dp \subseteq Dp}(tf(t,dp)) - \max_{dn \subseteq Dn}(tf(t,dn)) * \beta \tag{2.14}$$

$$DF'(t) = DF(t)_p - DF(t)_n * \alpha \tag{2.15}$$

ただし，すべてのドキュメントの集合 D に対して特定分野のドキュメント Dp である確率を$P(c_{D_p})$，D に対する t の出現確率を P(t)，$P(c_{D_p})$と P(t)の同時出現確率を$P(t,\ c_{D_p})$，特定分野 Dp 内のドキュメントを dp(dp∈Dp)，非特定分野 Dn 内のドキュメントを dn(dn∈Dn)，t が dp 内に出現する回数を tf(t，dp)，t が dn 内に出現する回数を tf(t，dn)とする．また，α，β は Dn に出現した単語重要度を下げる割合を表すハイパーパラメータである．なお，α は以下の条件を両方共満たすとする．

$$\begin{Bmatrix} \alpha \geq 0 \\ DF'(t) < |D| \end{Bmatrix} \tag{2.16}$$

また，β は以下の条件を満たす．

$$\beta \geq 0 \tag{2.17}$$

なお，実験の結果最適なα，βの値はそれぞれ 1.1，2.7 という結論を得ている．

3. 提案手法

3.1. 従来手法[4]と問題

2.2 で述べた通り，我々は以前特定分野にどれだけ精通しているかを判断することを目的とした単語重要度計算手法[4]を提案した．前提条件として，特定分野に属する単語群(専門辞書)が事前に与えられているものとし，重要度(専門度)に応じて単語に重みを付与した．

しかし，専門辞書には一般人も使用する単語が含まれるのが一般的である．例えば，特定分野をプログラミングとした場合，「ファイル」や「インストール」は専門用語であるが，一般人も使用する単語である．また，同時に専門性が高い単語に高い重要度を付与する．例えば，「Java」という著名なプログラミング言語は，一般人はほとんど使わないが，プログラミングを少し触った人間なら誰でも知りえる単語である．一方で，「Swing」という単語は，Java をある程度使いこなさないと知りえない単語である．すなわち，「Swing」という単語を知っている人間はプログラミングに関してある程度精通していることが予測される．このように特定分野に精通していないと知りえない単語により高い重要度を付与しなければならない．

このため，[4]では，特定分野のコーパス Dp と一般分野のコーパス Dn を使用し，Dn にはほとんど出現せず，Dp にもあまり出現しない単語の重要度を高くする計算手法を提案した．しかし，本手法はノイズのような単語の重要度も上がってしまうという問題が生じた．

3.2. ノイズに対する頑健性の実現

ノイズに対する頑健性を実現するための手法を提案する．ノイズに頑健な分野別単語排他度にはコーパス Dp，Dn それぞれに対して，tf（単語頻度）のコーパス内での分散の値を用いた．この分散の値が低い単語は，コーパス内の多くの文章で出現する単語である．つまり，この分散の値が高いということは少ない文章に偏って出現するということになる．Dn における分散が高い単語はノイズであることが予想される．一方，Dp における分散が高い単語は専門性が高い単語の可能性がある．そこで，Dn における分散が高い単語の重要度を低くし，Dp における分散が高い単語の重要度を高くする．

以上から，提案するノイズに頑健な分野別単語排他度を式(3,1)に示す．Dp における出現頻度の分散値$tfv(t,Dp)$が高いほど重要度が上がり，逆に Dn における出現頻度の分散値$tfv(t,Dn)$が高いほど重要度は下がる．また，γ は Dn に出現した単語重要度を下げる割合を表すハイパーパラメータである．

さらに，式(3,1)を用いた改善を行った手法を式(3.3)に示す．なお，式中の$MI(t,c)$，$TF_{MAX}(t)$，$DF'(t)$およびハイパーパラメータ α，β はそれぞれ式(2.13)，式(2.14)，式(2.15)，式(2.16)，式(2.17)に準ずるものである．

$$V'(t) = tfv(t,Dp) - tfv(t,Dn) * \gamma \quad (3.1)$$

$$tfv(t,Dx) = \frac{\left(\sum_d^D tf(t,d)\right)^2}{|Dx|} - \left(\frac{\sum_d^D tf(t,d)}{|Dx|}\right)^2 \quad (3.2)$$

$$W(t) = MI(t,c) * TF_{MAX}(t) * \log\left(\frac{|D|}{DF'(t)}\right) * V'(t) \quad (3.3)$$

ここで，Dx は抽象化したコーパス，$|Dx|$は抽象化したコーパスの数，dは抽象化されたコーパスに属するドキュメントである．なお，γ は以下の条件を満たすとする．

$$\gamma \geq 0 \quad (3.4)$$

また，式(3.3)が負の値をとる単語は重要単語とみなさないとする．

4. 実験データ

本節では，実験に用いるデータについて述べる．なお，今回の実験では対象とする特定分野を「プログラミングに関する専門性」とした．

4.1. 特定分野関連単語を抽出するために使用する辞書

特定分野関連単語として，IT 用語辞書のサイトである e-words[2]と多種多様な辞書を持つサイトである Weblio[3]から情報セキュリティ用語集，OSS 用語集，NET Framework 用語集，IT 用語辞書バイナリ，コンピュータ用語辞典の計 5 種類の辞書を利用し，のべ 36,895 の専門用語（単語）を収集した．本辞書に出現する単語を対象に 4.2 のコーパスにより単語重要度を付与する．

4.2. 単語重要度を算出するためのコーパス

本実験では，特定分野のコーパス Dp として，プログラマー・IT エンジニア用記事投稿サイトである Qiita[4]から 69,395 の記事を利用した．Qiita は Web 系の言語の記事から専門的なアルゴリズムについて，プログラマーや IT エンジニアが使用する知識を網羅しており，特定分野のコーパスとして適切であると判断した．

一方，特定分野外のコーパス Dn(一般分野コーパス)として，プログラマー・IT エンジニアではない人物約 30 人が書いたブログ群，ニュースメディアサイト，英語のニュースサイトを利用し，計 76,058 の記事を使用した．多種類の記事から構成したのは，偏りを少なくするためである．またプログラムは英語で書かれることも多いため，ノイズとなりうる一般的な英単語を除去するために英語ニュースサイトもコーパスに含めた．

なお，特定分野のコーパス・一般分野のコーパスは共に Mecab[10]を用いて形態素解析を行い，名詞のみを抽出した．使用した辞書は ipadic[5]に 4.1 で収集した単語を追加したものを使用した．

4.3. Twitter ユーザ

本実験では新卒採用に使われるケースを想定し，Twitter ユーザ（ただし学生）の専門性を推定することとした．Twitter のプロフィールから学生であること及び所属学部が明らかである Twitter ユーザ 150 名と，学生であることは明らかだが，所属学部が不明な Twitter ユーザ 50 名を人手により抽出し，各ユーザの直近 100 ツイートを 2016 年 1 月 11 日に TwitterAPI で取得した．所属学部が明らかな 150 人のうち，100 人は情報系学部に所属している学生，残り 50 人を情報系の学部ではない学部に所属しているユーザとした．

前処理として，まず，ツイートに含まれる URL や他ユーザの ID など、投稿内容ではないものを正規表現により削除した．その後 Mecab[10]を用いて形態素解

[2] http://e-words.jp/
[3] http://www.weblio.jp
[4] http://qiita.com/
[5] https://osdn.jp/projects/ipadic/

析を行い，名詞のみを抽出し，抽出した名詞群を単語として用いた．

5. 評価方法

本稿で提案した「ノイズに頑健な分野別単語排他度」の有効性を確認するため，Twitterユーザを対象とした「プログラミングに関する専門性推定」を行う．専門性推定タスクでは，専門辞書（単語の重要度付）を作成し，専門辞書に含まれる単語とユーザのツイート内容をパターンマッチングさせることで，ユーザの専門性を定量的に推定する．その後，単語重要度の値を用いてユーザをランキングし，そのランキングの妥当性を評価する．

ユーザ毎のツイート数は，10ツイートとし，少ないツイート数でも適切にランキングできるかどうかについて検証する．

5.1. ベースライン手法

提案手法の比較対象（ベースライン）として，既存の7手法（2.1で述べたTF-IDFとokapi BM25，2.2で述べた相互情報量，tf*rf，tf*dc，tf*bdcに加え，我々の以前提案した手法）を用いる．

TF-IDF及びokapi BM25を用いた専門辞書作成では，提案手法で使用した特定分野のコーパスDpのみを使用した．今回の重みづけは当該特定分野にどれだけ精通しているかを判断できることを目的としているため，一般分野のコーパスDnは用いなかった．Dpは4.2節で述べた通り，Qiitaに投稿された69，395の記事である．単語tのドキュメントd∈Dpに対する重要度w(t，d)の計算には，式(5.1)を用いる．

$$W(t) = \max_{d \in Dp} w(t,d) \tag{5.1}$$

相互情報量，tf*rf，tf*dc，tf*bdc，及び我々が以前提案手法を用いた専門辞書の作成では，提案手法と同様に種類のコーパスDp，Dnを使用する．また，tf*rf，tf*dc，tf*bdcで用いるtf値は特定分野のコーパスDpに属する単語に対して，ドキュメント（ツイート）毎に求め，ユーザ単位でその最大値をtf値として用いる．

また，我々が以前提案した手法で用いるハイパーパラメータの値は，α，βをそれぞれ1.1，2.7の値を採用した．これは，以前提案した時に最適と判断した値である．

5.2. Twitterユーザの専門性の定量的推定手法

ツイートごとにトピックが変わる可能性があるため，ツイート毎に専門性スコア（単語重要度）を計算する．この方法により，ツイート単位でのノイズ，つまり特定分野とは関係のないツイートを除去することが可能になる．その後，専門性の高いツイートを多くしているユーザは専門性が高いと考え，専門性の高いと判断されたツイートの数をユーザの専門性スコアとする．この考え方は，ベースライン手法にも適用する．

5.2.1. ツイートの専門性の計算方法

ユーザuが発信したツイート$tweet_{u,i}$の専門性スコアをTweetScore(u, i)とする．また，使用する専門辞書に含まれる単語集合をTとし，単語$t_j(t_j \in T,\ 1 \le j \le |T|)$がユーザuがツイートした$tweet_{u,i}$の中で出現した回数を$C_{u,i}(t_j)$とする．単語$t_j$の重みは$W(t_j)$とする．単語の出現回数から生成した|T|次元のベクトルをTweetVec(u，i)=[$C_{u,i}(t_1)$，$C_{u,i}(t_2)$，…$C_{u,i}(t_j)$，…$C_{u,i}(t_{|T|})$]，|T|次元の単語重要度ベクトルをWeightVec=[$W(t_1)$，$W(t_2)$，…$W(t_j)$，…$W(t_{|T|})$]とした時，TweetScore(u，i)を式(5.2)に示す．

$$TweetScore(u,i) = TweetVec(u,i) \times \text{Weight Vec} \tag{5.2}$$

5.2.2. ユーザの専門性度合いの計算方法

ツイート毎に「専門性が高いかどうか」を2値で判断し，専門性が高いと判断されたツイート数をユーザの専門性スコアとする．つまり，10ツイートを用いる際には，最大の専門性スコアは10となる．$tweet_i$が「専門性が高いかどうか」の判断は，$tweet_i$に含まれる単語が持つ単語重要度の総和であるTweetScore(u,i)が閾値thresholdより以上であるか否かにより判断する．ユーザuの専門性スコアを式(5.3)に示す．ただし，Nを実験時に使用する各ユーザあたりのツイート数とし，Count(Cond(i))を条件式Cond(i)を満たすiの数とする．Cond(i)はiを引数とする条件式を抽象化したものである．なお，本稿では閾値thresholdを使用する専門辞書中の単語を重要度降順で並べた際，上位k番目の単語の重要度の値とする．なお，k番目より下位の単語のみが$tweet_i$に出現している場合でも，同ツイート内で複数の重要単語が出現した場合，TweetScore(u，i)がthresholdより高くなる可能性がある．

$$Score(u) = Count(TweetScore(u,i) > threshold) \quad (1 \le i \le N) \tag{5.3}$$

5.3. ランキングを用いた評価手法

5.2.2で算出した専門性スコアをもとに学生ユーザ群をランキングし，正解ランキングと比較することで評価する．具体的には，Rafaelら[11]が用いた評価方法を採用する．Rafaelらは，Twitter上における影響度の高いユーザをランキングする研究において，提案手法によるランキング上位20名と，ベースライン手法によるランキング上位20名からなるのべ40名から異なる32名を抽出し，プーリングにより正解ランキングを作成した．具体的には，被験

提案手法	以前の我々の手法	相互情報量	tf*idf	bm25	tf*rf	tf*dc	tf*bdc
0.72	0.65	0.58	0.50	0.50	0.50	0.49	0.48

表 1 各手法により生成したランキングと正解ランキングとの順位相関係数

者に協力を得て,プーリングされた 32 名にスコアを付与し，そのスコア平均を元に正解ランキングを構築した．そして，正解ランキングと提案手法のランキングを比較することで評価を行っている．本稿でも同様の流れで評価を行う．

5.3.1. 正解ランキングの生成

正解ランキングの作成では，5.1 で述べたベースライン手法によって出力された 6 つのランキングを用いた．具体的には，使用するツイート数 N を 10 にした時の，各々のランキングから，上位 25 名を抽出したのべ 150 名から，異なる 44 名のユーザを抽出した．同様に，ランキングの中間に位置する 70 位から 130 位までの 60 名を抽出したのべ 360 名から，異なる計 114 名を抽出した．これら，上位 44 名，中間部 114 名の中から，ランダムに上位ユーザから 15 名，中間部から 10 名を抽出し，合計 25 名のユーザを抽出した．抽出された 25 名の Twitter ユーザのツイートを 5 名の被験者に確認してもらい,「プログラミングに関する専門的知識量」を 5 段階で判断してもらった．5 名の評価結果の平均を元に正解ランキングを生成した．なお，5 人の 5 段階判断において，分散が 1 を超えたものは除くこととしたが,実際には分散が 1 を超えるユーザは存在しなかった．

5.3.2. 正解ランキングとの比較方法

Rafael らは，ランキング評価に Precision@k とスピアマンの順位相関係数の 2 つの指標を用いているが，本稿では，スピアマンの順位相関係数を評価指標として用いる．これは，今回の実験ではランキング同位のものが多数出現し，Precission@k では正確な評価ができないためである．

スピアマンの順位相関係数とは二つのランキングの一致率を測る指標である．評価するランキングを X, 正解ランキングを Y とすると式(5.4)で表せる．

なおスピアマンの順位相関係数の結果は 1.0 から -1.0 であり，1 の場合はランキングが完全に一致していることを示し，-1.0 のときはランキングが完全に逆順になっていることを示す．

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})\,(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \qquad (5.4)$$

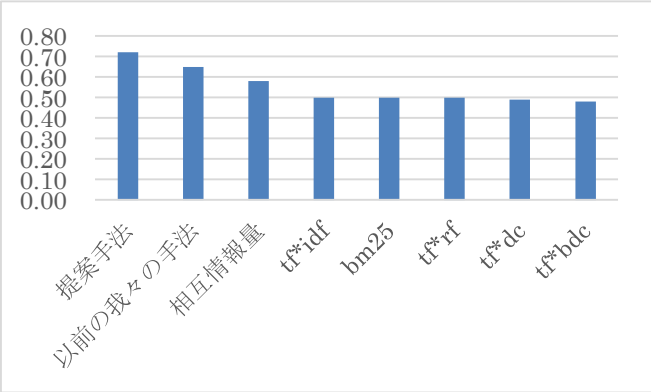

図 1 各手法により生成したランキングと正解ランキングとの順位相関係数

5.4. パラメータ決定のための事前実験

本研究では，3.1，3.2 の提案手法で用いるハイパーパラメータと，5.2.2 で述べた threshold の決定のために，4.3 のデータとは別のデータを用いた事前実験を行う．

事前実験用に,プロフィールから情報系学部に所属している学生であることが明らかな Twitter ユーザ 4 名分と情報系学部に所属していない学生であることが明らかな Twitter ユーザ 6 名分を人力により収集した．その後，3 人の被験者により，各 Twitter ユーザに対して「専門性の高さ」を 5 段階の評価をしてもらい,その平均によりランキングを作成し，事前実験用の正解ランキングとした．

次に，提案手法を用いてグリッドサーチを行い，正解ランキングとの相関が高くなる時パラメータを求めた．この時に使用するツイート数は 10 とする．その結果，k は 4500，提案手法で用いるハイパーパラメータ α，β，γ はそれぞれ 1.1，3.0，6.0 の値を採用した．

6. 実験結果・考察

使用するツイート数が 10 の時の，スピアマンの順位相関係数を用いて評価を行った結果を表 1，図 1 に示す．同結果から，提案手法が最も相関が高いことが分かる．以前の手法に比べ，今回は分散の値を計算式に取り入れた．特定コーパス内の一つの文章内の出現頻度の分散が高い単語ほど重要度が上がり，一般コーパス内の一つの文章内の出現頻度の分散が高い単語ほど重要度は下がる．この分散の値を取り入れたことで精度が向上したことから 2 つのことが言える．1 つ

は，専門性の推定においては特定コーパス内の出現する文章の偏りが大きい単語ほど重要であるということ．もう一つは，一般コーパス内の出現する文章の偏りが大きい単語はノイズの可能性が大きいため重要ではないということである．

7. おわりに

本稿では，特定分野における重要な単語に対する重要度計算手法を行う際，ノイズに頑健な分野別単語排他度として出現頻度の分散の値が有効であることを提案した．以前の我々の手法では，ノイズのような単語の重要度も高く付与してしまうという問題があった．これに対し本稿では，ノイズに頑健な分野別単語排他度を提案し，以前の我々の手法の拡張を行った．我々は，ノイズとなりうる単語はコーパスに対して文章集合に対してそれぞれの文章内の出現頻度の分散の値が高くなることに着目した．しかし，同時に特定分野コーパスに対して分散の値が高くなる単語は重要用語の可能性もある．そこで，特定分野コーパスに対して分散の値が高い単語の重要度を高くし，一般分野コーパスに対して分散の値が高い単語の重要度を低くするような拡張を行った．提案手法を Twitter ユーザの「プログラミングに関する専門性」判定に適用した評価実験の結果，我々の以前提案した手法などの既存手法と比べて，提案手法の方が専門性に応じたランキングをスピアマンの順位相関係数で 0.08 ほど高く行うことができた．同時に，ツイート数が 10 の時のように，ツイート数が少なくても効果があることを確認した．

提案手法はあらかじめ特定分野に属する重要単語が存在することを前提としており，重要単語の選定が難しい場合の対処方法も今後の課題である．さらに，本手法では計算する特定分野に関する大量のコーパスが必要である．今回は特定分野を「プログラミング」としたため，エンジニアブログ記事を用いることで大量のコーパスを入手することができた．しかし，一般的な特定分野に関するコーパスは論文や専門書である．数万種類の一つの分野に関する論文や専門書を集めることは極めて難しい．少ないコーパスで計算可能となるような拡張も今後の課題である．

謝辞

本研究実施にあたり助言をいただいた同研究室の石山雄大氏，また実証実験に協力いただいた方々に感謝する．

参 考 文 献

[1] 池田和史，服部元，松本一則."マーケット分析のための twitter 投稿者プロフィール推定手法"，情報処理学会論文誌コンシューマ・デバイス＆システム（CDS），Vol.2，No.1，pp.82-93 (2012)I. Tanaka and J. Suzuki, “Web and Database Technologies”, Proc. of ACM SIGMOD, pp. 10-22, 2010.

[2] X.Shao，Z.Chunhong and J.Yang. "Finding Domain Experts in Microblogs" Proc. of the 10th Int’l Conf. on WEBIST (2014).

[3] 奥野峻弥."マイクロブログを対象とした 100,000 人レベルでの著者推定手法の提案，" 早稲田大学修士論文 (2015).

[4] 滝川真弘，山名早人.“特定分野を対象とした単語重要度計算手法の提案と Twitter における専門性推定への適応”, FIT2016(第 15 回情報科学技術フォーラム)，第 2 分冊，pp.1-7 (2016)

[5] G.Saltion， E.A.Fox and H.Wu. ”Extended Boolean Information Retrieval”，CACM，Vol.26，No.11，pp.1022-1036 (1983).

[6] S.E.Robertson，S.Walker，S.Jones，M.M.Hancock-Beaulieu，and M.Gatford.“Okapi at TREC-3”，Proc. of TREC-3，pp.109-126 (1995).

[7] Stanford IR-book HTML Edition ，http://nlp.stanford.edu/IR-book/html/htmledition/okapi-bm25-a-non- binary-model-1.html

[8] M. Lan，C.L.Tan，J.Su and Y.Lu. "Supervised and traditional term weighting methods for automatic text categorization" IEEE Trans. on Pattern Analysis and Machine Intelligence，Vol.31，No.4， pp.721-735 (2009).

[9] T.Wang， Y.Cai， H.f.Leung， Z.Cai and H.Min. "Entropy-based Term Weighting Schemes for Text Categorization in VSM,” Proc. of the 27th Int’l Conf. on ICTAI，pp.325-322 (2015).

[10] T.Kudo，K.Yamamoto and Y.Matsumoto. "Applying Conditional Random Fields to Japanese Morphological Analysis，" Proc. of the 2004 Conf. on EMNLP，pp.230-237 (2004).

[11] C.Rafael and N.Sastry. "IARank: Ranking users on Twitter in near real-time， based on their information amplification potential，" Proc. of the 2012 Int’l Conf. on Social Informatics， pp.70-77 (2012).