RD-001

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
A Proposal of Word Weighting Method in Specific Field and its Adoption for the Estimation of Users' Expertise appearing in their Tweets

滝川 真弘† 山名早人†‡
Masahiro Takigawa Hayato Yamana

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．

単語の重要度を計算する有名な手法としては，TF-IDF[4]や BM25[5]などの手法がある．しかし，これらの手法は，ドキュメント集合に属する 1 ドキュメントを分別するための重要度であり，特定分野に対する重要度を測る手法ではない．また，専門用語は複合名詞で構成されるという仮定から C-Value[6]や中川らの提案した FLR[7][8]が存在する．しかし，特定分野の種類によっては重要な単語が複合名詞であるとは限らない．これに対して，相互情報量や Lan ら[9]の提案した tf*rf は，特定分野のコーパスと非特定分野のコーパスを用いることで，特定分野に特徴的な単語に高い重要度を付与する．しかし，これらの手法における重要度付与は，特定分野と特定分野以外を区別するための重要度であり，特定分野への専門性をこの重要度を用いて計算することはできない．

本稿では，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．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．重要度付与にあたっては，特定分野と特定分野以外のコーパスを用い，特定分野コーパスによく出現し，特定分野以外のコーパスにはあまり出現しない単語に高い重要度を付与する手法を提案する．

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，単語の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[4]と Okapi BM25[5]が有名である．

TF-IDF [4]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．

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

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

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

† 早稲田大学
‡ 国立情報学研究所
[1] https://twitter.com/

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，$\sum_k^K n(t_k,d)$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[5] 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 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．

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

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．k1 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 k1 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[10].

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 複合名詞に高い重要度を付与する手法

専門用語に対する重要度付与方法として，「専門用語は複合名詞である」という仮説に基づいて重要度を付与する手法が提案されている．複合名詞とは複数の語基から構成される単語であり，例えば「専門用語」は「専門」「用語」の二つの語基から成る複合名詞である．

同手法の一つである C-Value[6]は，TF に加え，単語を構成する語基数，当該単語が他の単語内に部分文字列として含有される頻度，当該部分文字列を含む単語の種類を用いて重要度を計算する．例えば，対象とする単語を「専門用語」とした場合，語基数は 2 である．また使用するコーパス内に「専門用語抽出」といった単語がある場合，「専門用語」は部分文字列として使用されていると判断する．

C-Value は，入れ子構造を持つ複合語の内, コーパス中で使用される頻度が高いものに高スコアを付与する．ここで，複合名詞 W，W を部分文字列として持つ単語の頻度を t(W)，W を部分文字列として持つ単語の種類数を c(W)，コーパス内の W の出現回数を tf(W)，W を構成する語基数を len(W)としたとき，C-Value の値は式(2.5)となる[6].

$$\text{C-Value(W)} = (len(W)-1)\left(tf(W) - \frac{t(W)}{c(W)}\right) \quad (2.5)$$

中川ら[7][8]は， FLR と呼ばれる手法を提案している．FLR は対象とする単語を単名詞と単名詞のみで構成される複合名詞とし，連接する単名詞の種類や頻度から当該単語の重要度を計算する．複合名詞 CN のスコア FLR(CN)は，CN を構成する単名詞 N_1，N_2，…，N_L が他の多くの複合名詞に使われているほど高くなる．

まず，単名詞の重要度計算手法について説明する．単名詞の重要度計算手法は連接種類から計算する方法と連接頻度から計算する方法の二つを提案している．連接種類から重要度を計算する方法では，単名詞 NW の両隣に来る単名詞の種類の異なり数を数える手法である．単名詞 N の左方に来る単名詞の種類の数を#LDN(N)，右方に来る単名詞の種類を#RDN(N)と定義する．一方，連接頻度から計算する方法は，単名詞 N の左隣に単名詞が連接している数を#LN(N)，単名詞 N の右隣に単名詞が連接している数を#RN(N)と定義する．

続いて複合名詞の重要度計算手法について説明する．単名詞 N_1，N_2，…，N_L がこの順で連接した複合名詞を CN とする．先に述べた 2 つの単名詞重要度計算関数を抽象化し，単名詞の左方の重要度計算関数を FL(N)，右方の重要度計算関数を RN(N)とする．複合名詞 CN の重要度は以下の式で表せる．

$$\mathrm{LR(CN)} = \left(\prod_{i=1}^{L}(FL(N_i)+1)(FR(N_i)+1)\right)^{\frac{1}{2L}} \quad (2.6)$$

中川らは，さらに複合名詞 CN が単独で出現した頻度 f(CN)を組み合わせ，式(2.6)を改善した式(2.7)を提案した．

$$\mathrm{FLR(CN)} = f(CN) * LR(CN) \quad (2.7)$$

以上，C-Value と FLR の 2 手法は，「専門用語は複数の名詞から成る複合名詞」という前提で重要度の計算を行っている．しかし，対象とする特定分野によっては，専門用語が複合名詞ではないこともある．例えば，特定分野をプログラミングとした場合，プログラム言語名などは複合名詞とはならない．このため，複合名詞ではない専門性の高い単語に対応した重要度付与方法を考える必要がある．

2.3 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[9]，tf*dc[11]，tf*bdc[11]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．

$$\mathrm{MI(t,c)} = \log\frac{P(t,c)}{P(t)P(c)} \quad (2.8)$$

これに対して，2009 年に Lan ら[9]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．

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

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[11]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 D_{C1}，D_{C2}…D_{Ci} 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 c_i 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$f(t, c_i)$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．

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

Wang らは同論文[11]で，tf*df 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 c_i 内における出現頻度を組み合わせた．カテゴリ c_i に属するドキュメント数を$f(c_i)$，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$f(t, c_i)$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．

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

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

3. 提案手法

本節では，ドキュメントが特定分野に属するか否かを判別するためでだけでは無く，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，「特定分野に属する単語の重要度計算手法」を提案する．ただし，前提条件として，Lan らの方法と同様，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することが提案手法の目的である．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．提案手法では，一般人があまり用い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 を使用する．また，同時に専門性がより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．逆にいえば「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することを本手法の目的とする．

2.1 項，2.3 項で紹介した手法は，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り，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．上で述べた例を用いると，「Swing」より「Java」という単語のほうが，重要度が高くなってしまう可能性がある．専門性を重要度に取り入れるためには，2.1 項，2.3 項の手法に加え，特定分野内での出現頻度をも考慮する必要がある．具体的には，出現頻度が低い単語を重要と判断する必要がある． しかし，出現頻度が低い単語にはノイズのようなものも含まれるため，適切に処理しなければならない．

提案手法では，まず，2.3 項で紹介した手法と同様に，特定分野に属するドキュメント集合 Dp = $\{dp_1, dp_2 \ldots dp_i \ldots dp_{|Dp|}\}$と特定分野に属さないドキュメント集合 Dn= $\{dn_1, dn_2 \ldots dn_i \ldots dn_{|Dp|}\}$を用意する．その上で，特定分野に関連する単語にその専門度合いに応じて重要度を付与する．なお，1 つの記事を 1 つのドキュメントと定義する．また，全体のコーパスを D = {Dp∪Dn}とする．

以下では，TF-IDF を拡張した手法と相互情報量を拡張した手法の二手法を提案する．

3.1 TF-IDF を拡張した単語重要度計算法 (提案①)

本項では，TF-IDF を拡張した単語重要度の計算手法について提案する． TF-IDF 自体は， 1) 1 つのドキュメント d 内の単語に対して重みを付与する手法である，2)カテゴリ（分野）を考慮しない，3)出現回数が高いと重要度が上がる，といった特徴を持つ．これに対して，提案手法では，予め与えられた専門辞書内に含まれる単語を対象に，Dp によく出現し，Dn にあまり出現しない単語に高い重要度を付与することを目指す．

これを実現するため，まず TF 値を 1 つのドキュメントに対しての値ではなく，Dp，Dn それぞれのドキュメント集合に対して求める．つまり，ドキュメント集合 Dp，Dn における単語 t の出現回数（$TF(t)_P$，$TF(t)_N$）を使用する．また DF 値についても， D 全体に対して求めず，Dp，Dn それぞれのドキュメント集合に対して算出する．つまり，Dp，Dn それぞれにおいて単語 t を含有するドキュメント数（$DF(t)_P$，$DF(t)_N$）を用いる．単語 t∈T の重要度 W(t)は，式(3,1)～(3.3)により計算する．

$$W(t) = \log\left(TF'(t)\right) * \log\left(\frac{|D|}{DF'(t)}\right) \quad (3.1)$$

$$TF'(t) = TF(t)_P - TF(t)_N * \alpha \quad (3.2)$$

$$DF'(t) = DF(t)_P + DF(t)_N * \alpha \quad (3.3)$$

ここで，α 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．

$$\left\{ \begin{array}{c} \alpha \geq 0 \\ \max_{t \in T} DF'(t) < |D| \end{array} \right\} \quad (3.4)$$

なお，以下の条件のいずれかを満たす単語 t は重要用語でないとみなす．

$$\left\{ \begin{array}{c} TF'(t) < 0 \\ DF(t)_P < DF(t)_N * \alpha \end{array} \right\} \quad (3.5)$$

3.2 相互情報量を拡張した単語重要度計算法(提案②)

本項では，相互情報量を拡張した単語重要度計算手法を提案する．相互情報量の利点として，カテゴリ（分野）ごとの出現の偏りを考慮できる点と低頻度の単語ほど高い重要度を付与できる点がある．低頻度の単語に対して重要度を高くできることは，高い専門的知識を持つ人間しか使わない単語ほど重要度を高く設定できることを意味する．一方で，たまたま使われた単語の重要度も上がるという副作用を持つ．また，単語の出現頻度のみを用いているため，文書頻度が考慮できていないという欠点を持つ．

そこで，提案手法では Dp，Dn のそれぞれのドキュメント集合において，各ドキュメントにおける TF 値の最大値と IDF を組み合わせ，重要度を計算する．単語集合 T に属する単語 t(t∈T)の重要度 W(t)を以下のとおり定義する．

$$W(t) = MI(t,c) * TF'_{MAX}(t) * \log\left(\frac{|D|}{DF'(t)}\right) \quad (3.6)$$

$$MI(t,c) = \frac{P(t, c_{D_p})}{P(t) * P(c_{D_p})} \quad (3.7)$$

$$TF'_{MAX}(t) = \max_{dp \in Dp}(\mathrm{tf}(t, dp)) - \max_{dn \in Dn}(\mathrm{tf}(t, dn)) * \beta \quad (3.8)$$

$$DF'(t) = DF(t)_P + DF(t)_N * \alpha \quad (3.9)$$

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$P(c_{Dp})$，D に対する t の出現確率を P(t)，$P(c_{Dn})$と P(t)の同時出現確率を$P(t, c_{Dn})$，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．

ハイパーパラメータ α の条件は 3.1 項と同様である． β もハイパーパラメータであり，以下の条件を満たすとする．

$$\beta \geq 0 \quad (3.10)$$

なお，以下の条件のいずれかを満たす単語 t は，重要単語ではないとみなす．

$$\left\{ \begin{array}{c} TF'_{MAX}(t) < 0 \\ DF'(t)_P < DF'(t)_N * \alpha \end{array} \right\} \quad (3.11)$$

4. 実験に用いるデータ

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[1] と多種多様な辞書を持つサイトである Weblio[2] 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 項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[3] 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[11]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 [4]に 4.1 節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．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 名を人力で発見し，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

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

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[12]を用いて形態素解析を行い，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「ある特定分野の単語重要度を算出する手法」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, 50, 100 と 3 種類を用意し，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 6 手法（2.1 節で述べた TF-IDF と okapi BM25，2.3 節で述べた相互情報量，tf*rf，tf*dc，tf*bdc）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．

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

相互情報量，tf*rf，tf*dc，tf*bdc 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．

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 $\mathrm{tweet_{u,i}}$ 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$t_j (t_j \in T,\ 1 \le \mathrm{j} \le |\mathrm{T}|)$がユーザ u がツイートした $\mathrm{tweet_{u,i}}$ の中で出現した回数を$\mathrm{C_{u,i}}(t_j)$とする．単語t_jの重みは $\mathrm{W}(t_j)$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)=[$\mathrm{C_{u,i}}(t_1)$，$\mathrm{C_{u,i}}(t_2)$，…$\mathrm{C_{u,i}}(t_j)$，…$\mathrm{C_{u,i}}(t_{|T|})$]，|T|次元の単語重要度ベクトルを WeightVec=[$\mathrm{W}(t_1)$，$\mathrm{W}(t_2)$，…$\mathrm{W}(t_j)$，…$\mathrm{W}(t_{|T|})$]とした時，TweetScore(u,i)を式(5.4)に示す．

$$\mathrm{TweetScore(u,i)} = TweetVec(\mathrm{u,i}) \times WeightVec \qquad (5.3)$$

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．$\mathrm{tweet_i}$ が「専門性が高いかどうか」の判断は，$\mathrm{tweet_i}$ 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 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(5.4)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)を引数とする条件式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 $\mathrm{tweet_i}$ 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．

$$\mathrm{Score(u)} = Count(TweetScore(u,i) > threshold) \qquad (5.4)$$

$$(1 \le \mathrm{i} \le \mathrm{N})$$

5.3 ランキングを用いた評価手法

5.2.2 項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[13]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，ベースライン手法（5.1 項）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．

表 1 スピアマンの順位相関係数

ツイート数	提案①	提案②	相互情報量	tf*idf	bm25	tf*rf	tf*dc	tf*bdc
10	0.45	**0.62**	0.57	0.50	0.50	0.43	0.38	0.46
50	0.65	0.74	**0.75**	0.70	0.70	0.67	0.67	0.73
100	0.67	**0.78**	0.77	0.67	0.67	0.65	0.68	0.69

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(5.5)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．

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \qquad (5.5)$$

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 項の提案手法で用いるハイパーパラメータと， 5.2.2 項で述べた threshold の決定のために，4.3 節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①及び提案②の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 7500，提案①の α を 0.2，提案②の α，β は 1.1，2.7 の値を採用した．

6. 実験結果・考察

使用するツイート数を 10，50，100 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～図 3 に示す．同結果から，ツイート数が 10 の時，提案②が最も相関が高いことが分かる．一方でツイート数が 50，100 の時は，相互情報量を用いた手法と提案②がほぼ同等の結果となっている．以上から，提案②は，ツイート数が少ない時（ツイート数 10）に相互情報量などの既存手法に比べて有効であることがわかる．

相互情報量が，提案②と同様によい結果となっているのは，相互情報量の計算時に，特定分野の単語群を重要単語（4.2 項）とし，ノイズが極めて少ない単語群とすることができたことに起因していると考えられる．このことは，相互情報量だけでも，特定分野に属する単語の中で適切に

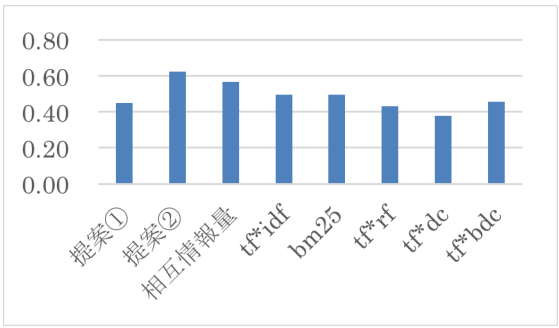

図 1 スピアマンの順位相関係数(ツイート数 10)

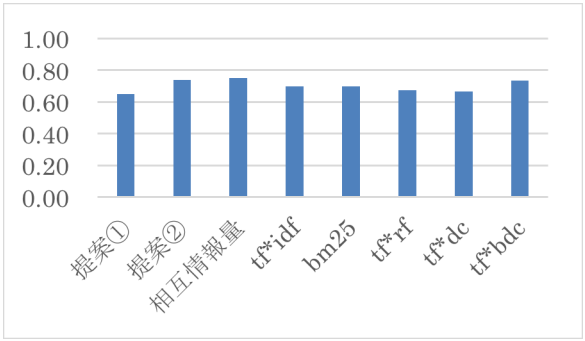

図 2 スピアマンの順位相関係数(ツイート数 50)

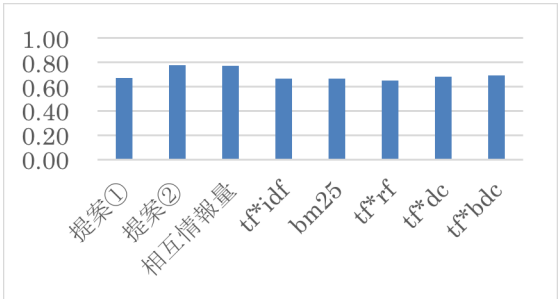

図 3 スピアマンの順位相関係数(ツイート数 100)

重要単語を選択できれば一定の精度を出すことができることを示している．

次に，ツイート数が 10 の時に，提案②で成功例と失敗例について述べる．成功した例は，「HTML って，プログラミング言語じゃないからプログラミングとは言わないけど，コーディングとは言う？」や「Go は Node と同じく unix philosophy 系だから，そこまでフルスタックフレームワークは向かないんだよね．」といったツイートがある．また，「冬休み Lisp のやつ読む予定だったのに機械学習のやつ読み終わりそう．」というツイートは，相互情報量を用いた

手法では専門性が高いとされないが，提案②の手法を用いたときは専門性が高いと判断される．これは，単語 「機械学習」が提案②の手法では重要度が高くなるが（Dp 内での IDF 値が高いことに起因），相互情報量では，高い重要度が付与されなかったことによる．

一方，失敗例として「さすがに設計しなさすぎて自業自得感がでてきた．」「コーディングの定義なー」というツイートがある．これらのツイートは，提案②の手法では専門性が低くなるが，tf*idf や tf*rf，tf*dc といった tf 値に重みを置く手法を用いたときは専門性が高いと判断される．この時の重要単語は「設計」と「コーディング」であり，これらの単語は二つとも Dp，Dn によらず多くのドキュメントに出現するため，提案②では重要度が低くなった．

「設計」「コーディング」という単語は，「プログラミング」に関する専門用語として使われるわけではないため，提案手法では重要度が低くなったが，ツイート内ではプログラミングに関連して用いられていた．このような，複数の意味で用いられる単語に対する弱点を確認することができた．

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提案した．従来の重要度計算手法は，カテゴリ（分野）を分割するための重要度付与はできるが，特定分野内での重要度（専門性）に応じた重要度付与ができなかった．これに対し本稿では，特定分野内での重要度をも単語重要度として付与する手法を提案した．具体的には，当該特定分野における単語の出現頻度を重要度計算に追加した．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，相互情報量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 0.05 ほど高く行うことができた．また，ツイート数が 10 の時のように，ツイート数が少ない時に特に効果があることを確認した．

提案手法は，多義語に対してはうまく適用できず，これらの単語に対して適切な重要度を持たせることが今後の課題である．また，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．なお，本研究の一部は JSPS 科研費 25280113 の助成を受けたものである．

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)

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

[5] 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).

[6] K.T.Franzi and S.Ananiadou. "Extracting Nested Collocations," Proc. of COLING, pp.41-46 (1996) .

[7] H.Nakagawa. "Automatic Term Recognition based on Statistics of Compound Nouns," Terminology, Vol.6, No.2, pp.195-210 (2000).

[8] 中川裕志，湯本紘彰，森辰則．"出現頻度と連接頻度に基づく専門用語抽出"，自然言語処理，Vol.10，No.1，pp.27-45 (2003).

[9] 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).

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

[11] 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).

[12] 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).

[13] 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).