Web を用いた人物の別名抽出

本間 大輝[†]　Danushka Bollegala[†]　松尾 豊[††]　石塚 満[†]

Web への主要なアクセス手段である検索エンジンの性能を向上させる試みが盛んに行われている．しかし，未解決の問題のひとつに別名の問題がある．たとえば「松井秀喜」で検索しても，「ゴジラ」と記述されたページはヒットしない．そこで，本研究では与えられた人物の本名に対する別名を Web から自動的に抽出する手法を提案する．提案手法により，別名を用いたクエリ拡張が可能となる．提案手法では，まず，検索エンジンのワイルドカード検索機能，および，本名と別名をつなぐパターンを利用し，別名の候補を抽出する．その後，正解データで学習を行った分類器により，候補の中から別名を選び出す．評価実験の結果，提案手法が十分な精度を持つことが確認できた．

1. はじめに

Web の発展に伴い，検索エンジンの利用が増加している．そのため，検索エンジンの性能向上に注目が集まっている．とりわけ，全検索の 30%を占める人名の検索の性能向上は重要と考えられている[1]．人名の検索の性能向上にあたって解決すべき問題は，同姓同名の問題と，別名の問題の 2 つである．これらの問題はどちらも，表記と実世界の人物との対応のあいまい性に起因する問題である．同姓同名は，1 つの表記に対して，複数の人物が対応する場合であり，検索において，精度の低下の原因となる．別名は，1 人の人物に対して，正式な表記と，正式な表記以外の複数の表記が対応する場合であり，検索において，再現率の低下の原因となる．なお，本論文では，その人物への参照として用いられる表記のうち正式な表記以外を別名と定義する．また，ある人物の正式な表記を単に名前と記述する．これらの問題のうち，同姓同名の問題に関しては，数多くの研究がなされ，十分な解決策が提示されているが[2]，別名の問題に関しては，ほとんど研究が行われておらず，十分な解決策は提示されていない．評判解析[3]のように別名の問題が性能に直結する分野もあるため，別名の問題の解決が強く望まれている．

別名の問題の単純な解決策としては，人物の名前を含むクエリに，自動的に別名を付加するクエリ拡張が考えられる．しかし，この方法の実装には名前と別名の対応表が必要となる．実在する人物の数と，次々と新しく生み出されるという別名の性質を考慮すると，人手による対応表作成･維持は現実的ではない．また，自動的に対応表を作成する手法も，十分な精度を持つものは存在しない．

そこで，本研究では，ある人物の名前が与えられたときに，その人物の別名を自動的に Web から抽出する高精度な手法を提案することを目的とする．従来手法では，文字列から別名に相当する部分を抽出する処理の精度，および，抽出した別名候補の評価の精度に問題があった．本研究では，前者の問題を形態素解析に頼らない統計情報を用いた別名の抽出手法によって解決する．後者の問題は，正解データ 50 件を利用した機械学習によって，さまざまな評価指標を統合した評価を実現することで解決する．

[†] 東京大学大学院 情報理工学系研究科 電子情報学専攻
[††] 独立行政法人 産業技術総合研究所

2. 関連研究

本研究と同じ目的を持つ研究として，外間・北川による研究[4]がある．しかし，外間・北川の研究では，形態素解析に基づく別名候補抽出処理を行っているために，抽出すべき別名を抽出しきれていない．たとえば，“荒川静香”に対する別名“しーちゃん”を“ちゃん”，“ーちゃん”として抽出してしまっている．本研究ではこのような形態素解析がうまくいかない別名にも対応する．

本研究とは目的がやや異なるが関連する研究としては，指定した語のペアの間の関係と似た関係にある語のペアすべてを，関係を表すパターンを用いて自動的に Web から抽出する研究[5]がある．この研究の手法を用いて，名前と別名の関係にある語のペアすべてを Web から抽出する方法も考えられる．しかし，名前と別名という関係はパターンだけでは評価しきれないと考えられる．それゆえ，この手法だけで精度のよい別名候補の評価ができるかは疑問である．本研究ではパターンによる評価指標とそれ以外の評価指標を組み合わせることにより，名前と別名という関係を精密に評価する．

3. 方法

3.1 概要

提案するシステムの全体像は図 3 のようになる．システムへの入力は人物の名前であり，出力は別名らしいものがより上位にくるような別名候補のランキング

である．システムの内部では，Web へのアクセスに検索エンジン（注1）を用いる．検索結果のページそのものはダウンロードせず，ヒット件数とスニペット上位 100 件のみを解析することで，高効率な処理を実現する．また，システム内部では，正解データ 50 件を利用する．正解データの 1 件は，有名人の名前と，その名前に対応する 1 つ以上の別名から成る．

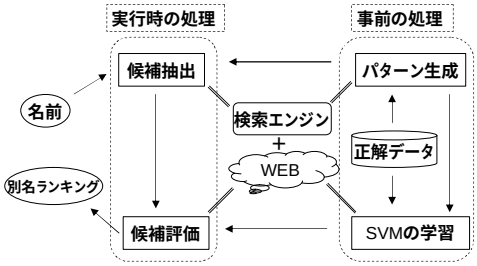

図 1　システムの全体像

提案手法の詳細を述べる前に，提案手法の基本的アイディアを簡単な例に沿って示す．たとえば，松井秀喜の別名（ゴジラ，Godzilla などがある）を抽出したいとする．おそらく，別名は Web に存在する全ページのどこかでは，名前と同一ページ内で現れているはずである．特に，“ゴジラこと松井秀喜”のように特定のパターンでつながれて出現することが多いと考えられる．したがって，検索エンジンで，「こと松井秀喜」のように検索し，スニペットを見て，“こと”の直前に現れる文字列を解析すれば，松井秀喜の別名候補が得られると考えられる．その後，得られた別名候補を“こと”の前に現れた回数で単純にランキングすることもできるが，それだけでは，適切なランキングになるとは限らない．たとえば，“息子に教えられた”，などが上位にくる可能性がある．（“父に教えられたこと，息子に教えられたこと”というタイトルの本を松井秀喜が書いているため．）したがって，出現頻度以外の評価もしなければならない．頻度以外の評価としては，別名候補ごとに「松井秀喜 and ゴジラ」，「松井秀喜 and 息子に教えられた」のようにあらためて検索しなおし，そのヒット件数を利用する方法が考えられる．そのような評価と，頻度による評価を組み合わせて総合的に評価すれば，”ゴジラ“や”Godzilla”が上位にくるようなランキングを生成できると考えられる．

以降の節では，提案手法の詳細を述べる．3.2 節では名前と別名をつなぐパターンを生成する手法を述べる．3.3 節では生成したパターンを用いて別名候補を抽出する方法を述べる．3.4 節では抽出した別名候補を評価しランキングを生成する方法を述べる．

3.2　名前と別名をつなぐパターンの生成

名前と別名をつなぐパターンとしては“こと”が広く知られているが，他にも有効なパターンは存在すると考えられる．そこで，本研究では名前と別名をつなぐパターンを自動的に生成する．

本研究では，パターンとして，“名前パターン別名”のように現れるもの，および，その逆の“別名パターン名前”のように現れるものの 2 種類を扱う．なお，前者を順方向のパターンと呼び，後者を逆方向のパターンと呼ぶことにする．また，順方向のパターンで「名前パターン」と検索する場合も，逆方向のパターンで「パターン名前」と検索する場合も，両方とも，「パターン（名前）」で検索する，と表現する．さらに，スニペットにおける，順方向のパターンの直後，および，逆方向のパターンの直前のどちらも，パターン近傍と表現する．

パターンは次のような処理を，正解データ 50 件について行うことで生成する．

1　「名前*別名」，「別名*名前」というワイルドカードを含むクエリを検索エンジンに投げる．
2　検索結果の上位 100 件のスニペットを解析し，名前と別名に挟まれている部分から，記号を除いたものをパターンとして抽出する．

(2) で記号を除いているのは，現在の検索エンジンはクエリ内の記号を無視するようになっているため，記号をパターンとして抽出しても，別名候補抽出時に役に立たないからである．これらの処理の結果，出現頻度でパターンをランキングすることができる．

しかし，出現頻度のランキングの上位のパターンを用いて「パターン（名前）」と検索したときに，検索結果のスニペットにおいて，パターン近傍に別名が本当に現れるかは確かではない．そこで，パターン p の別名候補取得力を以下の指標 *F-Measure* で評価する．

$$F-Measure(p)=\frac{2}{\frac{1}{Precision(p)}+\frac{1}{Recall(p)}}$$

$$Precision(p)=\underset{name\in namesExceptZeroHit(p)}{Ave}\left(\frac{positiveSnippets(name,p)}{totalSnippets(name,p)}\right)$$

$$Recall(p)=\underset{name\in allNames}{Ave}\left(\frac{ExtractedAliases(name,p)}{totalAliases(name)}\right)$$

$$\underset{x\in X}{Ave}(f(x))=\frac{1}{|X|}\sum_{x\in X}f(x)$$

注1 本研究では Google を利用した．

ここで，*totalSnippets(name,p)*は *p* で *name* を検索したときに得られるスニペットの数（最大 100）であり，*positiveSnippets(name,p)*はそのうち，正しく別名を抽出できたスニペットの数である．*totalAliases(name)*は *name* のもつ別名の個数であり，*ExtractedAliases(name,p)* はそのうち，*p* で *name* を検索したときに得られた別名の個数である．*allNames* は指標算出に使用する正解データに含まれるすべての名前の集合であり，*namesExceptZeroHit(p)*はそのうち，*p* で別名候補の検索をしたときに，少なくとも 1 件はスニペットが得られる名前の集合である．

出現頻度ランキング上位 50 パターンについて，パターンの *F-Measure* を算出し，*F-Measure* のランキングの上位 6 パターンを候補抽出用のパターンとして採用する．なお，この場合は，パターン近傍に別名が現れていれば必ずそれを正しく抽出できるとして *F-Measure* を算出する．

3.3 パターンを用いた別名候補の抽出

前節で生成したパターンを用いて，「パターン（名前）」と検索することにより，別名候補を含むようなスニペットを得ることができる．そこで問題となるのは，スニペット内のパターン近傍の文字列のうち，どこまでを別名候補として抽出するかである．単純なやり方としては，形態素解析して，パターン近傍の名詞をとるというやり方が考えられる．しかし，別名の多くは形態素解析器の辞書に登録されていない未知語であるために，形態素解析器はうまく動作しない．そこで，本研究では形態素解析に頼らない，統計情報に基づく別名候補抽出を行う．

本研究における別名候補抽出の手順は次のようになる．まず，スニペット上位 100 件について，パターン近傍の文字 10 文字を重ね合わせる処理を各パターンについて行い，図 2 のような木構造を作成する．木構造は順方向のパターン，逆方向のパターンでそれぞれ 1 本ずつ作る．木のノードは 1 文字であり，その文字が各パターンで何回出現したかが記録されている．エッジは親ノードに対する子ノードの相対頻度で重み付けされている．

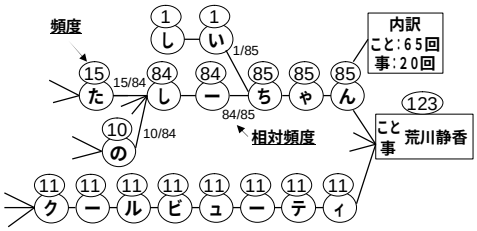

図 2　別名候補抽出のための木構造の例

この相対頻度がある閾値より低くなった場合，もしくは，次の文字が存在しない場合は，そこまでを抽出すべき別名候補とする．ただし，ひらがなカタカナ 1 文字のときは別名候補としない．閾値は，0 から 1 まで 0.05 刻みで 21 通りを試し，もっとも候補抽出の *F-Measure* が高くなる値を選択する．

3.4 別名候補の評価

パターンによって抽出した別名候補の評価は事前に学習させたランキング SVM[6]によって行う．本研究では，ランキング SVM への入力はある名前に対する複数の別名候補であり，出力は別名候補のランキングである．別名候補は，9 個の素性を持つベクトルとして表す．9 個の素性のうち 6 個は別名候補抽出時に用いたパターンでその候補が出現した回数である．たとえば，図 2 の“しーちゃん”が候補とすると，“こと”というパターンに関する素性値が 65，“事”というパターンに関する素性値が 20 となる．残りの 3 個は以下の Web 全体での共起指標 3 つを用いる．

$$Dice(name, candidate) = \frac{Hits(name, canididate)}{Hits(name) + Hits(candidate)}$$

$$OverlapC(name, candidate) = \frac{Hits(name, canididate)}{Hits(candidate)}$$

$$OverlapN(name, candidate) = \frac{Hits(name, canididate)}{Hits(name)}$$

ここで，*Hits(x,y)*は「x and y」で検索した際のヒット件数であり，*Hits(x)*は「x」で検索した際のヒット件数である．

ランキング SVM の学習には，別名候補の素性ベクトルの正しいランキングが複数必要である．本研究では次のような処理を正解データの名前 50 件について行うことによって学習用のランキング 50 個を生成する．

1　3.3 節の方法で名前に対する別名候補を抽出する．
2　別名候補の出現頻度ランキングを作成し上位 10 件を選ぶ．
3　選んだ別名候補に関して，本節で述べた 9 素性を計算する．
4　正解データの別名と一致する別名候補がすべて 1 位，一致しない別名候補がすべて 2 位となるランキングを作る．

4. 実験

4.1 パターンの生成

3.2 節で述べた方法で，パターンの *F-Measure* のラン

キングを生成した結果は，表 1 のようになった．“こと”以外にも，Precision が高い“通称”，“愛称”というパターン，*Recall* が高い“以下”というパターンなどを生成できた．

表 1　パターンの *F-Measure* ランキング上位 10 件

パターン		*F-Measure*	*Precision*	*Recall*
こと	名前	0.843	0.762	0.943
名前	通称	0.551	0.819	0.415
名前	こと	0.499	0.569	0.444
事	名前	0.341	0.461	0.270
名前	愛称	0.284	0.824	0.171
名前	以下	0.239	0.212	0.276
ちゃん	名前	0.094	0.071	0.140
名前	主演	0.075	0.060	0.100
名前	は	0.071	0.041	0.278
名前	社長	0.068	0.105	0.050

4.2　別名候補抽出のための最適な閾値の設定

3.3 節で述べたように別名候補の切れ目を決めるための閾値によって，”こと名前“ というパターンの別名候補抽出性能がどう変化するかを調べた結果，図 3 のようになった．

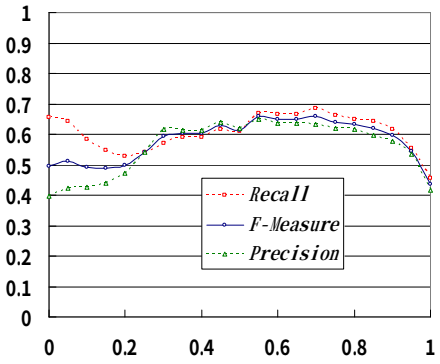

図 3　閾値による別名候補抽出性能の変化

F-Measure 最大となるのは閾値 0.7 のときであることが分かる．また，*Recall*，*Precision* も閾値 0.7 付近で最大値をとることがわかる．これは，閾値が 0.7 より上がると，切れ目の判定が厳しすぎて，少しでも書き間違いがあるだけで，正しい別名の途中で切ってしまうためである．また，閾値が 0.7 より下がると，切れ目の判定が甘くなりすぎて，別名以外の部分まで含めて別名候補として抽出してしまう．Recall だけ閾値 0 でかなり高い値をとっているが，これは，別名から文が始まる場合があるためである．

4.3　テストデータにおける提案手法の性能評価

提案手法の出力するランキングを評価するために評価指標 *MRR* を以下のように定義した．

$$MRR = \frac{\sum_{Alias \in ExtractedAliases} \frac{1}{Rank(Alias)}}{\sum_{i=1}^{totalAliases} \frac{1}{i}}$$

ここで，*ExtractedAliases* はランキングの別名候補のうち，正解データの別名と一致するものの集合を，*Rank(Alias)*はその別名のランキングにおける順位を，*totalAliases* はその名前がもつ別名の個数を表す．*MRR* はすべての別名がランキングに現れ，かつ，ランキングの上位に集まっているとき最大値の 1 をとる．

この *MRR* の平均値を学習データとは別の正解データ 50 件について求めた結果，表 2 のようになった．表 2 において，SVM(Linear)，SVM(RBF)はそれぞれ，線形カーネル，RBF カーネルを用いるランキング SVM でランキングした場合である．こと名前，名前通称は，そのパターンで別名候補を抽出し，出現頻度でランキングした場合である．*Dice*,*OverlapN*,*OverlapC* は *F-Measure* 上位 6 パターンで別名候補を抽出し，出現頻度ランキング上位 10 件の別名候補について，それぞれの係数の値でランキングした場合である．出現頻度や共起係数単独で評価するより，SVM でそれらを統合するほうがより良い性能が出ることが分かる．

表 2　各ランキング手法の比較

ランキング手法	*MRR*
SVM(Linear)	0.762
SVM(RBF)	0.729
こと名前	0.702
Dice	0.650
OverlapN	0.602
OverlapC	0.554
名前通称	0.194

また，最も性能の良かった SVM(Linear)によって生成されたランキングのうち上位 5 件を外間・北川の手法の結果と比較すると表 3 のようになった．ただし，外間・北川の研究では別名のリストを返すという形式になっているため順位は示されていない．本研究手法は全体として上位に別名がまとまるようなランキングを生成できており，SVM による評価がうまくいってい

ることが分かる．ただし，小泉純一郎に対して“相”，“という”などの意味をなさない語が得られてしまっている点，および，荒川静香に対して“上杉香緒里”などの無関係な語が得られてしまっている点では外間・北川の手法に本研究手法は劣っている．このような意味を成さない語が別名候補として取れてしまう理由は統計情報が少ない別名候補に関しては，切れ目の判定がうまくいかないためである．また，無関係な語が得られる理由は，別名候補評価において，出現回数と Web 全体での共起という表層的な評価して行っていないためと考えられる．

しかし，荒川静香に対して“しーちゃん”，松井秀喜に対して“にしこり”（顔を表現したアスキーアート）などの外間・北川の手法で得られていない別名が得られている点では本研究手法が優れる．

5. おわりに

統計情報を用いた別名候補抽出処理と，ランキング SVM を用いた別名候補の総合的評価によって，高精度の別名抽出を行う手法を提案した．評価実験の結果，ランキング SVM による評価の統合が有効に働いていること，従来手法では取れなかった別名が抽出できることが確認できた．

別名候補抽出の精度向上のために，相対頻度以外の別名候補の切れ目の判定基準を考えること，別名候補のより詳細な評価を得るために，名前と別名候補の分布類似度をランキング SVM の素性として与えることなどが今後の課題である．

参考文献

1) J. Artiles, J. Gonzalo, and F. Verdejo. A testbed for people searching strategies in the WWW. in Proc. SIGIR, 2005.

2) R. Guha, and A. Garg. Disambiguating People in Search. in Proc. WWW, 2004.

3) P. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. in Proc. ACL, 2002.

4) **外間智子，北川博之**. Web **データを用いた人物の呼称抽出**. DBSJ Letters, 2006.

5) M. Pasca, D. Lin, J. Bigham, A. Lifchits, A. Jain. Names and similarities on the web: Fact extraction in the fast lane. in Proc. ACL, 2006.

6) T. Joachims. Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning. MIT-Press, 1999.

表 3 本研究手法と外間・北川の手法の比較

（ただし，出力の形式が異なる．本研究手法ではランキング，外間・北川の手法ではリストとなっている．）

本研究手法						
順位	荒川静香	小泉純一郎	坂本龍一	中田英寿	堀江貴文	松井秀喜
1	しーちゃん	純ちゃん	教授	ヒデ	ホリエモン	ゴジラ
2	上杉香緒里	相	多東 Key	HIDE	ほりえもん	ゴジラ松井
3	演歌歌手	変人	アホアホ	NAKATA	僕が伝えたかった	にしこり
4	ダイエーリンク復活	という	世界のサカモト	ファンタジスタ	要は法律を守る	という
5	クールビューティ	ポチ	シスターM	中田ドット英寿	ふーみんの	私にとってはヒデキ
外間・北川の手法						
	荒川静香	小泉純一郎	坂本龍一	中田英寿	堀江貴文	松井秀喜
	ちゃん	ポチ	教授	ヒデ	ホリエモン	松井
	イナバウアー	純ちゃん	世界のサカモト	Hide	ほりえもん	ゴジラ
	ビューティー	ジュン様	龍一教授		社長ホリエモン	ゴジラ松井
	ーちゃん	ライオンハート	キョージュ		ホリえもん	マツイ
	クールビューティー		天才音楽家教授		事件ホリエモン	