社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－DBS－140 (II) (56)
2006／7／13

固有名詞句を用いたWeb上の人物同定

向田直生† 田島敬史†,‡

人名を用いてWeb検索を行う場合，検索結果中には異なる人物に関するページが混在してしまう．そこで，ページ内容に基づいて検索結果をクラスタリングすることによって，各人物に対応するページ群へ自動分類することがよく行われる．本研究では，このようなクラスタリングにおいて，ページ中に現れる，その人物名と関連の深い固有名詞句を重視することによってクラスタリングの精度を向上させる手法と，また，そのような固有名詞句を検索結果のページから発見する手法について述べる．

Person Identification by Proper Noun Clauses on the Web

NAOKI MUKAIDA† and KEISHI TAJIMA†,‡

When we retrieve Web pages by using a person name as query keywords, the result often includes pages about other persons of the same name. One well-known approach to this problem is to classify the result pages into page groups corresponding to those persons with the same name by using clustering techniques based on the page contents to the search result. In this paper, we propose a method to improve the accuracy of such clustering by giving bigger weight values to the proper noun clauses related to the target person, and also propose a method to discover such related proper noun clauses appearing in the search results.

1. はじめに

現在，パーソナルコンピュータ及びブロードバンドをはじめとしたインターネット環境が普及し，個人が様々な情報を検索エンジンを用いてWeb上で検索するようになった．しかし，人名に代表されるように，固有名詞には，同じ字面で意味する実体が異なる「同音異義語」的なものがあるため，Google等の検索エンジンを用い，固有名詞を検索語として検索を行った場合には，同じ名前の異なる実体に関するページが入り混じって表示され，それらのうちのある特定の実体に関する情報を収集するには手間がかかる．

また，固有名詞には誤字や異形字等により字面は異なるが意味する実体は同じである「同義語」的なものがWebページ上に出現している場合も多く，記述の誤りはその固有名詞に関する公式なページであっても例外ではない．また，誤字等以外にも，別名や改名，表記の揺れなどによる「同義語」的な人物名も数多くある．このような「同義語」的な語が用いられているページがあった場合，そこに有益な情報があったとしても，既存の検索方法ではそのWebページは発見できない．誤字はそれほど多いものではなく，異形字は新・旧字体を考慮すれば十分と思うかもしれないが，例えば"斎藤千和"という人物の場合，正しく"斎藤千和"でGoogle検索すると約22100件，斎の旧字体である"齋藤千和"で290件，純粋な間違いの"斉藤千和"で980件，同じく"斎藤千知"で239件と，誤字は相当数に上り，関連する情報をもれなく収集したい再現率重視の検索を行う場合には大きな問題となる．

このような問題を解決する手法に，Webクラスタリングがある．Webクラスタリングは Web 文章に対するグループ化処理であり，類似する Web 文章は同じクラスタに，類似していない Web 文章は異なるクラスタに分類されるようにする手法である．固有名詞の同定では，ある固有名詞 α が含まれているページ集合をクラスタリングすることで，α が同じ実体を指しているページ同士は同じクラスタに分類され，α が同名の異なる実体を指しているページ同士は違うクラスタに分類されるようにして，各実体に対応するページ集合へと近似的に分類するということがよく行われる．

これまで，様々なWebクラスタリングの手法が提案されてきたが，Web文章自体の類似度によりクラスタ分けする手法では，内容の異なる複数の文章中に現れる，ある同一の固有名詞自体の意味による分類は困難であった．例えば，あるプロ野球選手を紹介するページが球団によって作られていたとする．また，その選手は自分の趣味であるゴルフに関するサイトも持って

† 北陸先端科学技術大学院大学
Japan Advanced Institute of Science And Technology
†† 京都大学
Kyoto University

いたとする．この選手の名前で検索した場合，球団によるページと選手個人のページの両方が出てくるが，これを文書自体の類似度によりクラスタ分けしてしまうと，球団によるページと選手個人のページは別のグループになってしまうことが予想される．

また，誤字・異形字等により字面が異なる「同義語」的な場合には，そもそも検索している単語とは異なる字面のため検索結果に現れず，クラスタリングの対象にすらならない．

そこで，本研究では，検索エンジンを用いて固有名詞を Web 上で検索し，その結果にクラスタリングを適用して各実体に対応するページ集合に分類しようとするアプローチにおける，上記の二つの問題点を改善する手法について提案する．

まず，一つ目の，同じ自体を指す固有名詞が登場しているが，主要な話題が大きく異なるページを同じクラスタに分類できるようにするための手法として，現在広く用いられている，検索語の含まれているページ全体の文章の類似度によるクラスタリングではなく，ページ内の特徴的な名詞を抜き出し，それらを用いてクラスタリングを行う手法を提案する．

ここでいう特徴的な名詞とは，そのページ内に多く出てきて，かつほかのページではあまり出てこない名詞のことである．先ほどの野球選手の例で特徴的な名詞を抽出すると，その選手自体に関係する単語，すなわち，居住地・愛車・同僚の野球選手の氏名等が抽出されるであろう．このような特徴的な名詞に大きい重みを与えてクラスタリングをすれば，文章全体を用いてクラスタリングすると別のグループになっていたものを，同じグループに分類できる可能性がある．

しかし，単純に上のような手法で特徴的な「名詞」を抽出するだけでは不十分な場合がある．例えば，俳優や歌手，芸術家，音楽家，作家などの有名人の情報を，その人名を使って検索し，結果の中に同姓同名の有名人の情報が混在している場合を考える．そのような場合，これらの同姓同名の人物の識別に最も有用な情報は，その作品名等である．しかし，映画や音楽，絵画，小説等の作品名は，それ自体が「ゴジラ」などのような単純かつ，その作品名以外の意味では使われないような特徴的な名詞になっていることは稀で，むしろ，「世界の中心で愛をさけぶ」等のように，形態素解析を行った場合には一つの作品名としては認識されずに「名詞＋助詞＋... ＋動詞」と分割されてしまい，その結果，個々の「世界」，「中心」等の単語自体は多くのページに出現するような一般的な名詞になってしまう場合の方が多い．

そこで，本研究では，このような作品名などの「固有名詞句」にあたる単語列を自動的に抽出する手法を提案し，この手法によって抽出された固有名詞句に重みをつけてクラスタリングを行うことで，人物同定の精度を改善することを提案する．

また，二つ目の問題点は，誤字や改名などによる「同義語」の問題である．本研究では，同義語の可能性まで考慮して，まず広く候補ページを収集し，それらのページ集合に対して上述のクラスタリング手法を適用することによって，異なる人物名が用いられているが同じ人物に関するページであるようなものを同じクラスタに分類できるようにする．

2. 関連研究

同姓同名の人物や，同一人物の名前の異なる表記などの同定に関する研究としては，文献データベースにおける各文献の著者名や参考文献の記述中に現れる著者名の同定に関する研究が古くから行われており，現在でも様々な研究が行われている[1]．これらの研究では，例えば，人物名の表記に特有の慣例や，共著関係，文献間の関連度などの様々な情報を用いる方法などがこれまでに提案されている．これらの研究では，文献のタイトルや，ある学術文献が発表された雑誌や会議の名称などの固有名詞句に当たる情報を用いることも多いが，それらの研究が対象とするデータは，データベース化された文献情報や，一般的な形式によって記述された参考文献リストであることが多く，これらのデータからのタイトルや会議名等の抽出は比較的簡単であるため，本研究で考えるような，対象となる Web ページ中で特徴的な語の抽出や，固有名詞句にあたるものの抽出などの手法は必要にならない．

また，最近では，自らのコンピュータの中にあるメールアドレスなどの情報と PDF ファイル中などに出てくる文章の著者名の対応を，文字列の類似度，同じファイル中に著者として出てくる共著者にはどのような人物がいるか，どのような人物とメールのやり取りをしているかなどの情報を用いて発見する技術も提案されている[2]．この場合も，文献の PDF ファイルからの著者名の抽出やメールからの送信者，受信者のメールアドレスの抽出などの処理の部分は比較的容易であり，本研究で考えるような特徴語，固有名詞句の抽出は必要にならない．

一方，人名による Web の検索結果の中の人名の出現の同定に関する研究も数多く行われている[3)~5)]．しかし，これらの研究では，人物間の複雑な関係のグラフ解析や，人物に関する属性値の自動抽出などの手法が用いられており，本研究で提案するような固有名詞句などを発見して Web 上の人物同定に用いる手法は，われわれが知る限りはこれまでに提案されていない．

また，本研究で固有名詞句と呼ぶような特徴的な単語の並びの発見に関連する研究としては，6)，7) 等がある．7) では，他の文献から文章を引用しているために，非常に特徴的な同じ単語の並びが異なるページに共通して現れているようなパターンを発見することによって，文章の引用を自動発見しようとしている．こ

れらの研究とわれわれの研究は，各単語がその単語の一般的な出現頻度に応じてランダムに出現するのに比べて有意に高い頻度で現れる単語の並び方を発見しようとしているという点では共通しているが，そのような単語の並び方には，6)，7) 等の研究が主に対象としているような「引用」によるもの，われわれが対象としているような作品名などの「固有名詞句」，日本語でごく一般的に使われる「決まり文句のフレーズ」などいくつかの種類があり，それぞれについて，それを発見するための手法が異なる．

なお，本論文の内容は，文献 8) での内容に新たな実験結果等を加えたものであり，基本的な部分の詳細については，文献 8) を参照されたい．

3. システムの概要

ある単語の列が，固有名詞句であるかどうかには次の二つの要素を考える必要がある．

- その単語の並びの出現頻度が，構成要素となっている各単語がその単語の一般的な出現頻度に基づいてランダムに出現した時に生じうる出現頻度に比べて有意に高いか．すなわち，その並び方が特に頻繁に使われるような並び方であるかどうか．
- その単語の並びが，日本語でごく一般的に，頻繁に使われる決まり文句のような「一般フレーズ」であるのか，それとも作品名などのような「固有名詞句」であるのかどうか．

そこで，本研究で提案するシステムは，以下の各処理を行うモジュール群からなる．

(1) Web ページ集合からの特徴的な単語列の発見
対象としている人物名による Web 検索の結果のページ集合から，それらのページ集合中で同じ単語の並びが繰り返し使われているようなものを固有名詞句の候補として抽出する．

(2) 単語列が固有名詞句かの判定
前段階で抽出された固有名詞句が，検索語の人物に特に関係の深い固有名詞句なのか，それとも日本語としてごく一般的に用いられる決まり文句のようなフレーズなのかを判定する．

(3) 誤字の可能性を想定した再検索
検索語の人物に関する情報を書いているページ内の人物名の表記に誤りがあった場合等でもクラスタリング対象となるよう，再度検索を実行する．

(4) クラスタリング
適切な手法を用いてクラスタリングを行う．

よって，本システムの構成は図 1 のようになる．

4. 各モジュールの処理の詳細

この章では，各モジュールの処理の詳細について順に説明する．

図 1 システム

4.1 固有名詞句発見モジュールの処理

検索したい人物が俳優などの場合，作品名などは一般名詞・動詞・未知語などの集まりから構成された固有名詞句であることが多いため，形態素解析を行った結果が一般名詞・動詞・未知語等複数の単語にわかれてしまう．そこで，このモジュールでは，このような複数の単語の集合からなる固有名詞句を発見する．処理の概略は以下のようになる．

(1) 検索語により検索された Web ページに現れる頻度が一定閾値以上の単語を発見する．

(2) これらの単語を始点として，その単語の前後に隣接してどんな単語が現れるかを調査し，複数単語の同じ並び方が繰り返し現れているようなものを固有名詞句である可能性の高い単語列として発見する．

以下，これらの処理の詳細について説明していく．

4.1.1 検索結果ページに現れる頻度が高い単語の発見

ある固有名詞を検索語として Google により検索されたページのうち，ある程度数（実験では 100 件）の Web ページを解析し，出現頻度の高い単語を記録して，その前後を調査すべき単語（以降，始点単語と呼ぶ）とする．具体的には，

(1) 検索された個々のページの文章を茶筅[10] を用いて単語ごとに区切り，全ての単語を記録する．

(2) 上の処理で記録された全ページの単語を KH Coder[11] を用いて解析し，単語の出現頻度を計測する．

(3) 出現頻度が 30%以上の単語を記録する．このとき，「1 文字のみ」，「数字のみ」，「記号のみ」の単語は省く．

単に，有意に高い確立で連続して現れるような単語の列を発見することだけを考えるのであれば，任意の単語を始点単語とすべきである．しかし，ここでは以下の二つの理由から，対象としている人物名による検索結果中に出現する頻度が高い単語のみを始点単語とすることにする．

- すべての単語の前後を調査すると必要な処理の量が多くなり過ぎ，システムの反応時間が長くなっ

て使い勝手が悪くなる．

- ここで発見したいのは，対象となる人物名に特に関連の高い固有名詞句であり，必ずしも全ての頻出単語列を発見したいわけではない．

4.1.2 出現頻度の高い単語を始点とした前後の調査

出現頻度の高い単語から順に，その単語の出現の前後にどのような単語が現れるを調査する．

(1) 始点単語が現れるページで，始点単語の前にどのような単語が現れるかを調査する．

(2) 始点単語が出てくるページの中で，始点単語とその前のある単語（以降，始点前単語と呼ぶ）が続けて現れている割合が60%を超えており，かつ始点前単語が記号ではない場合に，始点単語と始点前単語は固有名詞句の一部であると仮定し記録する．

(3) 始点単語の後ろの単語についても同様に調査する．(以降，始点単語の後の単語を始点後単語と呼ぶ)

(4) 始点前単語と始点単語，もしくは始点単語と始点後単語のどちらか1つ以上が固有名詞句の一部であると仮定された場合，始点前単語・始点後単語のうち固有名詞句の一部であると仮定された単語の並びを始点単語として再度1. から実行する．

(5) 1〜4までを，始点単語と前・後の単語が続けて現れる割合が60%を下回るまで繰り返し，固有名詞句の候補となる単語列の長さを延ばしていく．

4.2 固有名詞句判別モジュールの処理

このモジュールでは，モジュール1において発見された固有名詞句の候補となる単語列が検索語の人物に関係する「固有名詞句」なのか，それとも例えば「更新履歴」,「ページを更新しました」のような一般の日本語で，あるいは特にWebページ上で，頻繁に見られる一般的なフレーズなのかを判定する．

具体的には，以下のような方法を用いる．対象としている人物名を用いてGoogle検索をした際の検索結果件数と，発見された単語列を検索語としてGoogle検索を行った際の検索結果件数を比較し，後者の件数が10倍以上多い場合にはその単語列は一般的なフレーズであると判定し，その単語列は次以降のモジュールには渡さない．

4.3 誤字発見モジュールの処理

このモジュールでは，誤字等の理由で表記が異なる人物名がWebページ上にあった場合でも検索語の人物と同一人物と思われる場合にはクラスタリング対象に含めるための処理を行う．

具体的には，検索語から一文字を取り除いてできる全ての文字列についてor検索を実行する．例えば,「石毛佐和」が検索語であるならば，”毛佐和” OR ” 石* 佐和” OR ” 石毛*和” OR ”石毛佐”で再度検索する．

4.4 クラスタリングモジュールの処理

このモジュールでは，モジュール1〜3までの結果を用いてクラスタリングを行う．クラスタリングには群平均法を用いる．具体的には，まず，モジュール1において抽出された固有名詞句のうち，全ページに対する出現頻度が一番高い単語を始点単語とした固有名詞句を，その人物名に対する最も特徴的な固有名詞句とし，この固有名詞句が含まれるページをグループ化し，これを初期クラスタとする．

次に，前項でまとめられたページ以外のページについて，そのページ内に含まれるモジュール1で抽出した固有名詞句に加えて，tf-idf値が高い語を，固有名詞句との合計が10個になるまで順に抽出し，これらの特徴語を用いて結果を10次元の特徴空間上に配置し，距離の近いものをまとめてグループ化する．

5. 予備実験によるパラメータの決定

前章で説明した処理では，固有名詞句判定の閾値等，いくつかのパラメータを用いている．これらのパラメータに関して最適な値を求めるために，予備実験を行った．実験に用いた人名は，田中理恵，豊口めぐみ，野中藍，水樹奈々，斎藤千和，池澤春菜，松岡由貴，小林沙苗，門脇舞，松来未祐，茅原実里，杉田智和，小野大輔，井上和彦，大塚明夫，桑谷夏子，森川智之，緒方賢一，関智一，金田朋子の20人である．

5.1 固有名詞句発見モジュールの予備実験

ここでは，

- 始点単語として全ページ中にどの程度の頻度以上で現れる単語を用いるか
- 始点単語と始点前・後単語が繋がって現れる割合がどの程度なら固有名詞句とするか

の二つの閾値を決定するために実験を行った．まず，前者の閾値を決定するために，上記の人物名を用いてGoogle検索を実行し，検索されたページのうち上位100ページに現れる単語を分析し，頻度を計算した．結果のグラフを図2に示す．このグラフは横軸に，単語の出現頻度，縦軸にそのような出現頻度を持つ単語の数をとっている．

このようなグラフの全体の形がこのような曲線となることは広く知られているが，具体的な数値としては，このグラフからわかるように，30%以下の出現頻度になるあたりから，該当する単語数が数十個を越えて急激に増加していくため，今回は出現頻度が30%以上の単語を始点単語として用いることとした．また，同姓同名の人物が複数存在する場合には「30%」という数字に達する単語が減少すると考えられるため，出現頻度が上位の単語50個か，もしくは出現頻度が30%以上の全単語のうち，個数が多いほうを選択するものとした．ここで，上位50件としたのは，出現頻度が30%の単語数の平均値がおよそ50個であったためで

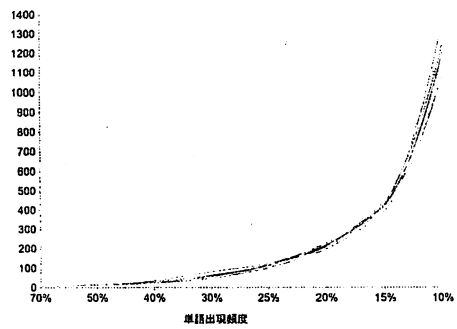

図 2　単語出現頻度と単語数

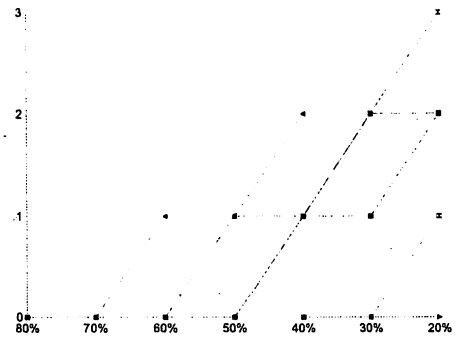

図 4　閾値を変化させた際の絞り過ぎの固有名詞句数

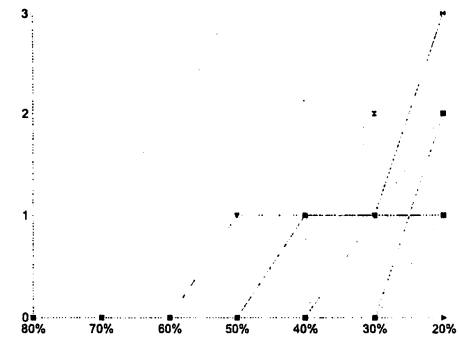

図 3　閾値を変化させた際の一般名詞句出現個数

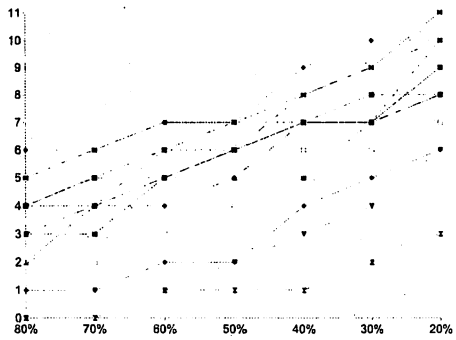

図 5　閾値を変化させた際の正解固有名詞句抽出個数

ある．

次に，始点単語と始点前・後単語が繋がって現れる割合がどの程度なら固有名詞句とするかの閾値の決定するために，上記の実験で抽出された始点単語を用いて，始点単語と始点前・後単語が繋がって現れる割合をどの程度まで下げると一般的なフレーズであるような単語列が現れるかを実験した．固有名詞句を作成する際の閾値を変化させた際の結果を，図 3 に示す．横軸が閾値，縦軸がその閾値を用いた場合に固有名詞句と判定されてしまうが人間の判定では一般的なフレーズと考えられるような単語列の数である．

また，ある程度以上に閾値を下げると，ある作品シリーズの中の一つの回の題名であるものなど，関連する固有名詞句ではあるものの特殊化しすぎていると思われるものが抽出される．例えば，「Star Wars」で十分固有名詞句としての役目を果たすのに，閾値を下げていくと「Star Wars Episode III」等のような単語列が固有名詞句として抽出されてしまい，ある役者に関するページでは「Star Wars」のことにしか言及していないが，実際にはその役者は「Star Wars Episode III」にも出ている，というような場合が生じる．閾値を変化させた時に，そのような固有名詞句がどの程度出現するかを表すグラフを結果を図 4 に示す．

また，閾値を変化させた時に，固有名詞句と判定するのが正しいような単語列がどの程度抽出されるかを図 5 に示す．

このように，閾値を 80%にしてもある程度の個数の固有名詞句を取得できたものもあるが，検索語しか抽出できないなどの場合もある．また，逆に閾値を 40%以下に設定した場合にはエラーの発生・特殊すぎる固有名詞句の抽出などの現象が発生した．以上の結果より，閾値は 60〜70%が妥当な範囲であると考え，その範囲でより多くの固有名詞句を取得できると期待できる 60%を閾値とすることとした．

5.2　固有名詞句判別モジュールの予備実験

次に，上記の閾値を用いて抽出した固有名詞句候補の単語列の中から，その人物名に特に関連する固有名詞句ではなくて一般的なフレーズであるような単語列を識別するための最適な閾値を求める予備実験を行った．実験には，モジュール 1 の予備実験において抽出された一般的な言い回しである単語列に関して，以下の仮説が成り立つかを検証した．

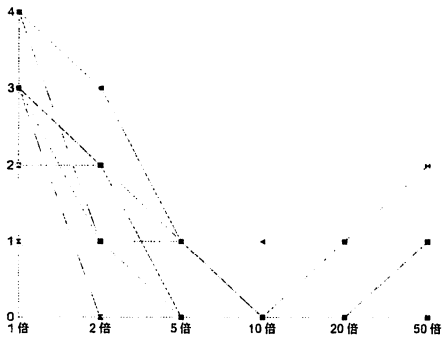

図 6 検索件数の開きとエラー件数

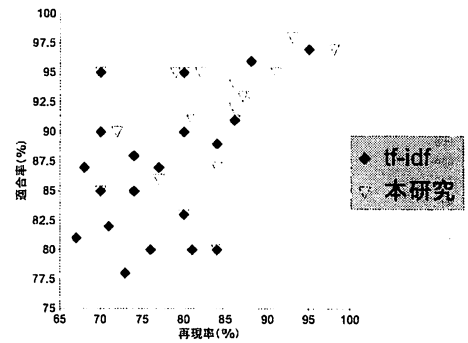

図 7 従来手法と提案手法の比較

仮説: ある閾値が存在し，抽出された単語列が出現している総ページ数の，対象としている人物名が出現している総ページ数に対する比が，その閾値以上の場合は，その単語列は一般的なフレーズである確率が高く，その閾値以下の場合は，その単語列がその人物名に特に関連した固有名詞句である確率が高い．

検証のために，前節の実験で抽出された単語列に関して，その単語列が出現する総ページ数を求め，これと人物名の出現ページ数との比を求めて上の仮定のもとで判定を行い，閾値としてどのような値を用いるとどの程度誤って判定されるものが出るかをグラフにしたものを図 6 に示す．

このように，人名と一般的なフレーズには出現ページ数において，ほぼ 10 倍以上の開きがある．また，たとえ検索語に関係している固有名詞句だったとしても，それほど多く出現するような固有名詞句はクラスタリングに利用しても有効でない可能性が高い．よって，出現ページ数において対象人物名と 10 倍以上の差が有る場合には，そのような固有名詞句はクラスタリングのための特徴語としては利用しないこととした．

6. 評価実験

前節で説明した予備実験をもとに各閾値を決定し，これらの閾値のもとでシステムの評価実験を行った．本節では，この評価実験の結果について報告する．各モジュール毎の出力結果に関する詳細な評価については，文献 8) を参照されたい．

この実験では，同姓同名の人物が居る人物名をウィキペディア[9] の同姓同名の項 (2005 年 11 月 3 日 12:16 の版) より選択して用いた．実験に用いた人物名は以下のとおりである．

(1) 石田敦子（漫画家/毎日放送社員）
(2) 伊藤美紀（声優/女優・歌手）
(3) 江川卓（元プロ野球選手/ロシア文学者）
(4) 木村洋二（札幌テレビ放送/新潟大学教授）
(5) 黒田清子（元皇族（さやこ）/行政書士（きよこ））
(6) 斉藤圭（男性ラジオディレクター・斎藤 K/女性声優・真堂圭の旧芸名）
(7) 鈴木俊一（元東京都知事/衆議院議員）
(8) 高橋英樹（俳優，タレント/元広島東洋カープの投手）
(9) 西村博之（アニメーター/「2 ちゃんねる」管理人）
(10) 宮村優子（声優 /脚本家）

また，比較のために，従来手法の例として群平均法によるクラスタリングを行った．結果として得られた再現率と適合率を横軸を再現率，縦軸を適合率としてプロットしたものを図 7 に示す．ここでの再現率と適合率は，手動によりページを分類し，そのページ数を分母として計算した．

このように，本研究の提案手法は多くの場合に，ごく単純なクラスタリング手法よりも再現率・適合率共に改善することが出来た．江川卓（ロシア文学者）の結果が従来手法と本研究提案手法とほぼ同じなのは，元プロ野球選手に比べてページ数が少ないため，固有名詞を抽出できなかったためである．同様に，ラジオディレクターの斉藤圭，元広島東洋カープ投手の高橋英樹，脚本家の宮村優子についても固有名詞を抽出できなかった．また，全体的に，再現率が余り高くならない理由としては，文章が全くない，画像のみのページがあるなどしたためである．

7. まとめと今後の課題

本研究では，人物名を用いた Web 検索結果のページ集合にクラスタリングを適用して，同姓同名の各人物毎のページ集合へと分類する際に，各人物名に特に関連が深いと思われる固有名詞句を重視することによってクラスタリングの精度を改善する手法，および，そのような固有名詞句を発見する手法について提案した．簡単な評価実験の結果，本研究の提案手法によっ

て，ごく単純なクラスタリングよりもクラスタリングの精度を改善することができた．人物名による Web 検索結果に対して人物同定を行う手法に関するこれまでの研究のうち，クラスタリングと，クラスタリング以外のなんらかの手法（リンク解析など）を組み合わせるような手法に関しては，そのクラスタリングの部分に本研究で提案する手法を用いることで，本研究の提案手法と組み合わせて用いることができる．そのような，従来研究の手法との組み合せに関する検証実験は今後の課題である．また，本研究の提案手法では，同姓同名人物のどちらかが圧倒的に有名（ページ数が非常に多い）な場合には，ページが少ない人物の固有名詞句を抽出できなかった．今後，このような場合にでも固有名詞句を抽出できるようにアルゴリズムを改良するのも重要な今後の課題である．

参 考 文 献

1) Hui Han, Lee Giles, Hongyuan Zah, Cheng Li, Kotas Tsioutsiouliklis: "Two Supervised Learning Approaches for Name Disambiguation in Author Citations," In Proc.of JCDL, pp.296-305, 2004
2) Xin Dong, Alon Halevy, Jayant Madhavan: "Reference Reconciliation in Complex Information Spaces," In Proc.of SIGMOD, 2005
3) Ron Bekkerman, Andrew McCallum: "Disambiguating Web Appearances of People in a Social Network," In Proc.of WWW Conference, pp.463-470, 2005
4) X. Wan, J. Gao, M. Li, and B. Ding: "Person resolution in person search results: WebHawk," In Proc.of the 14th ACM CIKM, pp.163-170, 2005
5) 佐藤進也，風間一洋，福田健介，村上健一郎：「実世界指向 Web マイニングの提案とその同姓同名人物分離問題への適用」，日本データベース学会 Letters Vol.3, No.4, pp.21-24, 2005
6) 相澤彰子「テキストコーパスにおける特徴語抽出のための分析ツール」，情報処理学会研究会研究報告, 2000-FI-061, 2001
7) 相澤彰子「テキストからの再利用文字列の抽出と分析」，情報処理学会研究会研究報告, 2003-DBS-130, 2003
8) 向田直生「Web ページ上に現れる固有名詞の同定手法」，修士論文, 北陸先端科学技術大学院大学, 2006 年 2 月
9) フリー百科事典「ウィキペディア」, http://ja.wikipedia.org/wiki
10) 形態素解析システム「茶筅」, http://chasen.naist.jp/hiki/ChaSen/
11) KH Coder, http://khc.sourceforge.net/