2C2-4

検索エンジンを用いた人名読みの推定

Acquisition of Kana Person Names using Web Search Engine

酒巻 智宏[*1] 大向 一輝[*2] 丹 英之[*3] 武田英明[*2]
Tomohiro SAKAMAKI Ikki OHMUKAI Hideyuki TAN Hideaki TAKEDA

[*1]東京大学 University of Tokyo
[*2]国立情報学研究所 National Institute of Informatics
[*3]株式会社アルファシステムズ ALPHA SYSTEMS INC.

In this paper, we propose a method for acquiring the Japanese Kana name of a person from its Chinese character name by the search engine. Japanese Kana name is extracted from search results of Chinese character, and filtered by pattern matching and dictionary. If there are two or more Kana name in a certain Chinese character, our method separates candidates into multiple persons using their attribute such as occupations and belongings.

1. はじめに

近年、SNS やミニブログのように人にフォーカスしたサービスが注目されている。人物は、名前、出身地、所属、経歴など様々な属性を持っており、これらの属性を史料や Web といったドキュメントの中から抽出する研究は、多数行われている [3]。本研究では、人物に関する属性のうち、特に人名に焦点を当てる。日本人の人名については、漢字表記は Web 上で現れることが多いが、その読みがわからないことがある。人物の読みを調べる際には、漢字辞書や人名辞書を使うことが多い。漢字辞書を使う場合には、同じ漢字表記を持つ人物で、違う読み方をする場合に対応できない。また、人名辞書を使う場合には、人名辞書には著名な人物しか記載されていないために、調べたい読みが人名辞書中に存在しない場合がある。このため、辞書を用いるだけではある人名の読み方を一意に判定することは困難である。さらに近年、名前の多様化が進んでおり、一般常識で読むことができない名前が増加している。

そこで、本研究では、漢字表記と共に振り仮名が与えられているような情報を効率的に抽出する手法を提案する。

2. 関連研究

本節では本研究の目的と類似した研究について述べる。

Web 上のデータを用いて単語に対して読みを付与する方法として、三宅らの研究 [2] や本間らの研究 [6] があげられる。これらの研究は、形態素解析がうまく動作しないような未知語の読みや属性をいかに獲得するかの研究であり、すべての研究において、パターンを用いて読みの抽出を行っている。

三宅らの研究では、例えば、Yahoo!(やふー) のように、文章中において読みを付与したい単語の後にくる括弧内にその単語の読みが入る可能性が高いことを用いている。この研究では、単語の後の括弧を抜き出すという単純なパターンマッチングを用いているが、一定の精度で単語の読みを抜き出すことに成功している。日本語表現において、先行する文字列の後の括弧表現でその文字列の説明をする表現は一般的な表記方法であり、括弧表現による情報抽出は、自然言語処理の分野ではよく用いられている手法である。本研究では、この研究をさらに発展させ、読みを取得する対象を括弧内に限らず、より多くのパターンで読みを抜き出すことで、より一般性を高めて読みを付与することを目標とする。

また、本間らの研究では、別名と人物名がどのようなパターンでつながっているかを発見することで、Web 上のデータから別名を抽出する。「〜〜こと○○」のように、別名と人名をつなぐパターンを出現頻度が高いものから抽出し、得られた結果を SVM にかけることで候補の評価を行っている。この研究は、前述の三宅らの研究に対して、人物の別名の抽出という特定の目的に対してパターンマッチングを行ったものであり、高い精度で別名の付与が可能になっている。

3. 提案手法の概要

本研究では、以下の手順で人物の読みを抽出する。まず、Web 上から人物の読みを抽出するために、検索エンジンを用いて人物に関する情報を収集し、パターンマッチングにより読み部分を抜き出す。得られた読み候補にはノイズが含まれるため、読み候補をフィルタリングにかける。さらに、フィルタリングを通った読み候補が複数存在する場合には、同じ漢字表記をもち、違う読み方をする人物が存在すると考え、それらの人物の持つ属性を抽出し、その属性により複数の人物を分離し、それぞれの人物に対して読みを付与する。

4. 読み候補の取得

4.1 抽出パターンの生成

テストデータは、Wikipedia [*1] に存在する人物の漢字表記と読みのペアを 2817 件用意した。

どのようなパターンで検索結果スニペットから読みを抜き出すことができるのかを調べるため、テストデータを用いて、「漢字表記 and 読み」というクエリで検索を行い、スニペットを取得する。このとき、検索エンジンとして Google Ajax Search API [*2] を用い、29516 件のスニペットを取得した。

スニペットから読み候補を抜き出すためには、計 4 つのパターンが必要になる。

$$\text{田中}\ (a)\ \text{太郎}\ (b)\ \text{たなか}\ (c)\ \text{たろう}\ (d) \qquad (1)$$

式 1 で、それぞれ (a) が「漢字の苗字と名前間のパターン」、(b) が「漢字と読み間のパターン」、(c) が「読みの苗字と名前

連絡先: 酒巻智宏，東京大学大学院新領域創成科学研究科，〒277-8561 千葉県柏市柏の葉 5-1-5，TEL:04-7136-4003，Email:sakamaki@mcl.iis.u-tokyo.ac.jp

∗1 http://ja.wikipedia.org/
∗2 http://code.google.com/intl/ja/apis/ajaxsearch/

間のパターン」、(d) が「読み終端のパターン」となる。これらの位置に、どのような記号や文字列が現れるかを出現頻度順に並べたものを表 1 に示す。

表 1: 出現するパターンの例

(a)	出現数	(c)	出現数
なし	16395	space	12592
space	12474	なし	12157
space3	248	・	3287
…	…	…	…
(b)	出現数	(d)	出現数
（	11284	）	10821
,	3184	)	3931
(	2812	、	3677
…	…	…	…

(a)(c) は、何も入らない場合やスペースが一つ入る場合が上位に、(b) は、「(」や「（」といった記号が、 (d) は、「)」や「、」といった記号が上位に現れるという結果になった。

ただし、パターンを多く使うほど、読み以外の文字列を抜き出す可能性も高くなる。そこで、パターン数による読み抽出の性能評価を行う。なお、パターンは、出現頻度順に並べたものを用いる。評価基準は適合率を用いた。適合率は読み候補が抽出できたキーワードに対してどれだけの精度で正しい読み付与ができたかを示す。

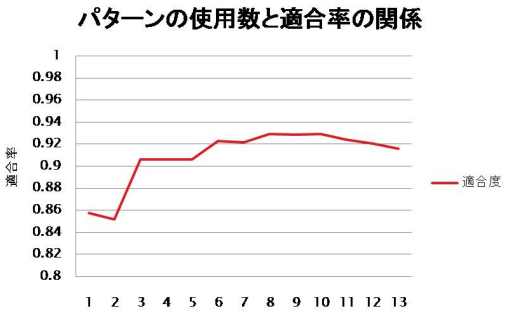

図 1: パターンの使用数と適合率の関係

図 1 によれば、パターン数が 10 の時が最も高い適合率で読みを抜き出すことができることがわかる。

4.2 スニペットからの読み候補の抽出

人物の読みを取得する最初の段階として、人物の漢字表記をクエリとして検索エンジンで検索を行う。検索エンジンには、Yahoo! Search BOSS [*3]を使用した。本研究では、検索結果の上位 500 件を解析対象とする。

得られた検索結果から、表 1 で得られた、漢字の苗字と名前の間、漢字と読みの間、読みの苗字と名前の間、読みの終端の各パターンを用いることで、読み候補を抜き出していく。

5. 読み候補の絞り込み

取得された読み候補の中には、明らかに人物の読みでないものが含まれる。例えば、人物の所属 (「とうきょうだいがく」「しすてむそうせいがっか」といったもの) や、出身地 (「さいたまけん」「とうきょうと」といったもの) などは、明らかに読みでないものとして候補から除外するべきである。

本研究では、これらをフィルタリングするための手法として、DP マッチング [4] を用いる。DP マッチングは、動的計画法を用いたマッチング手法であり、2 つの要素のずれや伸縮に対応しやすい、計算量が少ないといった特徴を持つ。先行研究 [2] を参考に、DP マッチングのパラメータの決定やスコア付けを行った。

はじめに、表 2 のように、人物の漢字表記から、漢和辞書、人名辞書を用いて辞書的に可能な読み方を生成する。まず、各漢字ごとに漢和辞書を用いてすべての読みを列挙し、組み合わせることによりすべての可能な読みを生成する。次に、苗字と名前の一覧が記載された人名辞書を用い、苗字、名前ごとにすべての読みを列挙し、漢和辞書で得られた読みと組み合わせて可能な読みを生成する。

なお、本研究では、漢和辞書として「Infoseek マルチ辞書 漢和辞典」 [*4]。人名事典として日外アソシエイツ[*5]の「苗字 8 万よみかた辞書」と「名前 10 万よみかた辞典」を用いた。

表 2: 辞書によって漢字表記から生成される読み候補

	田	中	太	郎
1	た	なか	た	ろう
2	でん	なか	た	ろう
3	た	ちゅう	た	ろう
…	…	…	…	…
25	たなか		ふと	ろう
…	…	…	…	…
32	でん	なか	たろう	
…	…	…	…	…

次に、表 2 で得られたすべての可能な読みとスニペットから得られた読み候補に対して DP マッチングを適用する。一つの読み候補について、辞書的に可能な読み方すべてと DP マッチングを行い、そのうち最もスコアが良かったものをその読み候補のスコアとする。

また、DP マッチングによるフィルタリングを行う際に、スコアの閾値を決める必要がある。

図 2(上) は閾値を変化させた際の、閾値以上の DP マッチングのスコアを持つ読み候補数の変化をグラフにしたものである。また、図 2(下) は閾値以下の DP マッチングのスコアを持つ読み候補中に正解読み候補が含まれている数の変化をグラフにしたものである。図 2 を見ると、閾値が 2 の時がフィルタリングの性能、正解データの損失ともに最もバランスのとれる結果を得られるといえる。

6. 複数の読み候補への属性付与

複数の読み候補がある場合は、同じ漢字を持ち、違う読み方をする複数の人物が存在すると考えられる。ここで、人物の持つ属性を用いて、その人物を同定することで、それぞれの人物に対して読みを付与する。

本研究では、スニペット中に現れる人物の読み候補と共起する人物の属性を抽出し、それぞれの人物がもつ属性とする。人名の曖昧性解消の手法について調査した [1] によれば、人物同

*3 http://developer.yahoo.com/search/boss/

*4 http://dictionary.infoseek.co.jp/

*5 http://www.nichigai.co.jp/

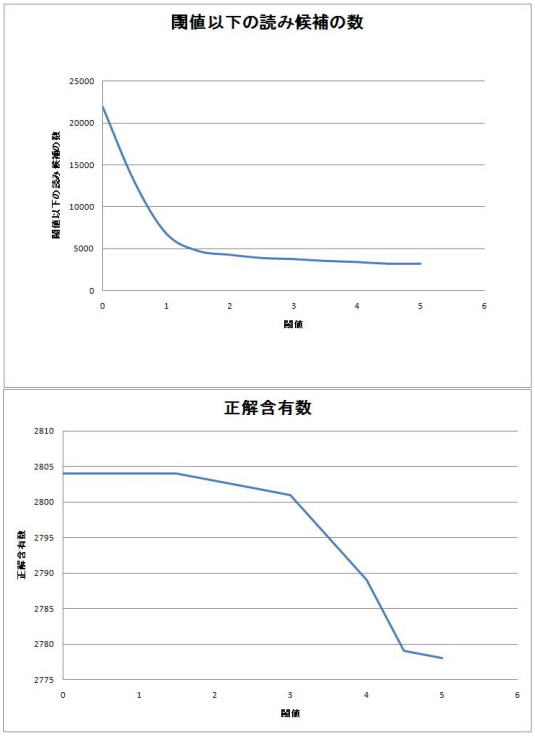

図 2: 閾値による DP マッチングの結果の変化

定を行う際に適した属性として、職業、所属、同僚などが挙げられている。また、スニペット中に現れる人物の属性を用いて同姓同名人物の特定を行った [7] では、使用する属性を職業、地位、役割としている。本研究では、これらの先行研究を踏まえ、人物同定を行うにあたって、職業、地位、役割を用いることとし、日本語語彙体系 [5] の内で職業、地位、役割カテゴリ以下に分類される 2 文字以上の単語を取得し、形態素解析器 MeCab *6 の辞書に追加し、スニペットを形態素解析することで職業、地位、役割を人物の属性として抽出した。日本語語彙体系において職業、地位、役割カテゴリ以下に所属する単語の例を表 3 に示す。

表 3: 職業、地位、役割カテゴリの単語の例

単語の例
スペシャリスト、ドクター、法律家、プログラマー 水先案内、会計士、学者、科学者 魔法使い、組長、代表取締役

得られた属性は、TF-IDF の値により重み付けをする。

$$TFIDF(d,t) = tf(d,m) \cdot log\frac{N}{df(m)} \qquad (2)$$

- $tf(d,m)$: 文章 d 中での単語 m の出現回数
- N: 文章の総数
- $df(m)$: 単語 m が含まれる文章の数

*6 http://mecab.sourceforge.net/

TF-IDF の値は、単語の重み付けを行う際に一般的に用いられる値である。日本語語彙体系から得られた辞書の中には、「一人前」「まとめ役」といったように、文章中によく現れる一般的な単語のために、人物を特定するのにあまり役に立たない単語が含まれる。式 2 を見るとわかるように、このような単語の $df(m)$ の値は大きくなり、結果として TF-IDF 値は低くなるために、重要度を低く設定することができる。

抽出された属性を、TF-IDF の値により並び替え、最大で上位 5 件をその読みを持つ人物の属性とする。実際に抽出された属性の例を表 4 に示す。

表 4: 抽出された人物の属性の例

漢字表記	読み候補	属性 1	属性 2	属性 3
井上昌己	いのうえ まさき	選手	レーサー	-
	いのうえ しょうこ	歌手	シンガー	ライター

表 4 の例では、「井上昌己」という漢字表記をもつ人物の内、「選手」「レーサー」であるのは「いのうえまさき」、「歌手」「シンガー」であるのは「いのうえしょうこ」といったように、それぞれの人物に対して読みを付与することができた。

7. 評価実験

7.1 評価方法

提案手法の有効性を検証するために、大規模なデータでの評価実験を行う。実験対象は、NII *7 が運営する論文データベース「CiNii *8」である。CiNii が保持する著者データのうち、30704 件を抽出し元データとする。CiNii では著者データが漢字表記と人名のローマ字表記のペアで格納されている。本研究では、ローマ字を平仮名に変換して正解データとした。

7.2 実験結果

実験結果を表 5 に示す。

表 5: 実験結果

実験結果	値
元データの数	30704
なんらかの読みを付与できた候補の数	28461
スニペット中に正解が含まれる候補の数	24235
正しい読みを付与できた候補の数	21237

本手法でなんらかの読みを付与できた候補の数は、28461 件であった。本研究の手法では、そもそも検索結果のスニペットに正しい読みが存在しない場合は読みを抽出することができない。スニペット中に正しい読みが含まれるものは、24235 件という結果であった。この中から、21237 件の人物について正しい読みを付与することができた。また、先行研究 [2] 参考にして、式 3〜5 で示される一般性 (generality) と適合率 (precision)、再現率 (recall) を評価尺度として用いた。結果は、一般性が 69.2%、適合率が 74.6%、再現率が 80.9%となった。なお、一

*7 http://www.nii.ac.jp
*8 http://ci.nii.ac.jp

般性と適合率、再現率は以下の式で表される値である。

$$\text{一般性} = \frac{\text{正しい読みが得られた元データの数}}{\text{すべての元データの数}} \quad (3)$$

$$\text{適合率} = \frac{\text{正しい読みが得られた元データの数}}{\text{読み候補が抽出された元データの数}} \quad (4)$$

$$\text{再現率} = \frac{\text{正しい読みが得られた元データの数}}{\text{スニペット中に正解読みがある元データの数}} \quad (5)$$

また、同じ漢字を持ち、違う読みをする複数の人物が存在する場合に、それぞれの人物にどのような属性が付与されたかを表 6 に示す。

表 6: 各人物に付与された属性

漢字表記	付与された読み	属性 1	属性 2	属性 3
中島洋	なかじまひろし	教職員	児童	役員
	なかじまよう	院長	医学博士	職人
中村俊也	なかむらしゅんや	仏子	演出家	俳優
	なかむらとしや	仏子	演出家	俳優
山田良治	やまだよしはる	教授	社長	委員長
	やまだりょうじ	住職	俳人	-
…	…	…	…	…

表 6 をみると「中島洋」という漢字を持つ人物の中で、「なかじまひろし」という読みを持つ人物に対して抽出された属性は、「教職員」「児童」などであり、「なかじまよう」という読みを持つ人物に対して抽出された属性は、「院長」「医学博士」であった。これらより、教育関係者の「なかじまひろし」と、医療関係者の「なかじまよう」という人物が存在することがわかり、それぞれの人物に対して読みを付与することができた。

「中村俊也」という漢字を持つ人物には、「なかむらしゅんや」「なかむらとしや」という複数の読みが抽出されたが、どちらに対しても「仏子」「演出家」「俳優」という共通の属性が抽出された。

8. 考察と今後の課題

評価実験では全体の 69.2%に対して読みの付与に成功した。読みの付与に失敗した理由としては、「スニペット中に正解読みがない」「パターンによって正解読みを取得できない」「DP マッチングで正しい読みが除外される」という 3 点が挙げられる。また、全体のうち 1 割はローマ字を平仮名に変換する時点で正しく変換されなかったことが原因である。精度のさらなる向上のためにはこれらの箇所でパラメータの調整を行うことが考えられる。

表 6 の「中島俊也」の例のように、複数の読みが抽出された場合の一部で、どちらの読みに対しても同様の属性が与えられた結果が存在した。実験結果からは同様の属性が与えられた原因の特定までは困難であったため、それぞれがなぜ失敗したかを特定することはできなかった。原因としては、そもそもひとりの人物に対して 2 つの読みが付与されている可能性が考えられる。これは、誤植や、不慮により、その人物の読み方をドキュメント作成者が間違えた際に起こり得る。

また、同じ漢字を持ち、同じ読みをする複数の人物が存在した場合には、この手法だけでは人物を完全に分離することは難しい。このような人物をどう扱うかについては今後検討していく必要がある。

また、今回の研究ではパターンマッチングのみを用いて読みの抽出を行ったが、HTML や XML の構造を用いて読みの抽出を行ったり、機械学習を用いることで読みとそれ以外を分類したりすることでさらなる精度の向上が期待できる。これらは、今後の課題である。

最後に、本手法を用いた名前読み付与システムの紹介をする。

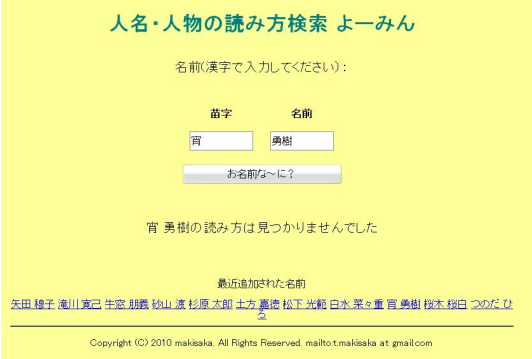

図 3: 本手法を用いた名前読み付与システム

図 3 は、名前読み付与システム「よーみん」のトップページである。なお、「よーみん」は Web アプリケーションとして実装されており、`http://cyprus.ex.nii.ac.jp/~sakamaki/grad/` の URL 上に配置されている。

参考文献

[1] 関根聡. Web 検索における人名の曖昧性解消技術の動向：同姓同名のクラスタリング. 情報処理, Vol. 49, No. 5, pp. 573–578, 2008.

[2] 三宅純平, 竹内翔大, 川波弘道, 猿渡洋, 鹿野清宏. 括弧表現に基づく web テキストマイニングを用いた流行語への自動読み付与の提案. 電子情報通信学会技術研究報告. SP, 音声, Vol. 108, No. 422, pp. 1–6, 2009.

[3] 石川徹也, 北内啓, 城塚音也. 歴史オントロジー構築のための史料からの人物情報抽出. 自然言語処理 = Journal of natural language processing, Vol. 15, No. 4, pp. 3–18, 2008.

[4] 内田誠一. Dp マッチング概説：基本と様々な拡張. 電子情報通信学会技術研究報告. PRMU, パターン認識・メディア理解, Vol. 106, No. 428, pp. 31–36, 2006.

[5] 白井諭, 大山芳史, 池原悟, 宮崎正弘, 横尾昭男. 日本語語彙大系について. 情報処理学会研究報告.IM, [情報メディア], Vol. 98, No. 106, pp. 47–52, 1998.

[6] 本間大輝, Bollegala Danushka, 松尾豊, 石塚満. 3zk-3 web を用いた人物の別名抽出. 全国大会講演論文集, Vol. 70, No. 5, pp. 1–6, 2008.

[7] 木村塁, 戸田浩之, 田中克己. 検索結果スニペットのクラスタリングによる同姓同名人物の特定. データ工学ワークショップ (DEWS2006) 論文集, 2006.