修 士 論 文

Webからの人物の属性情報抽出

平成 22 年 2 月 9 日提出

指導教員　　石塚 満　教 授

情報理工学系研究科 電子情報学専攻

48-086439　　渡 部　啓 吾

Abstract

Personal names are among one of the most frequently searched items in web search engines. Extracting information in the form of attributes and values for a particular person enables us to uniquely identify that person on the web. For example, although namesakes share the same name they usually have different date of births or affiliations. Given a set of documents retrieved for a particular person, we propose two stage approach to extract values for a set of attributes for that person. In the first stage we mark all potential attribute strings in a given text. The second stage then attempts to select the attribute values relevant to a person name. We use a named entity recognition tool to mark all occurrences of named entities in a given document. We then use a rule-based tagger to identify the variants of the given person name. Next, we employ a combination of rules and pre-compiled attribute value candidate lists to extract values for a given set of attributes. The candidate value lists are manually created using resources available on the web such as Wikipedia. Finally we select the attribute values by using Support Vector Machine (SVM). Features we use on learning SVM are specific expression, distance from target name, distance from other name, and n-gram. The proposed method is evaluated on the test data collection created for the attribute extraction subtask at the second Web People Search Evaluation Workshop (WePS2). According to the results in the evaluation, the proposed method improved f-measure by 62.3 persent from the top system among the 15 participating systems.

目 次

表 目 次

図 目 次

1 序論

1.1 人物の属性情報

まず属性とは，Wikipedia[1]によると，以下のような定義がなされている．属性（ぞくせい）とは，一般にあるものに共通して備わっているとされる性質や特徴のことである．例えば物体の色や形，人の能力，素性，社会的関係などである．人物に共通に備わっている性質が人物の属性であると考えると，それは身長や体重，生まれた日付（生年月日）などであると考えられるが，本論文ではもっと広く,「多くの人物に備わる，人物を特徴づける要素」を「人物の属性」と定義する．具体的には，住所や電話番号，あるいは電子メールのアドレスなども人物の属性と呼ぶこととする．

属性には属性自体を表す名前と，対応する属性の値を表す内容が存在する．本論文では以下，前者を「属性名」と呼び，後者を「属性値」と呼ぶ．例えば，“Barack Obama” の “Occupation” は “President” であるとしたとき，属性名は “Occupation” であり，属性値は “President” である．

1.2 研究の目標と背景

ウェブ上に存在するドキュメントの量は年々増えており，今日では有用な情報のリソースとして様々な場面で利用されている．実際，人が調べ物をするときには，検索エンジンを用いてウェブ上の情報を利用することが当たり前であるだけでなく，Wikipedia や Blog，SNS（Social Networking Service）などを通じて，一般のユーザから提供される情報も増え続けている．しかし，ウェブ上のドキュメントは新聞や雑誌などのように整ったものばかりではなく，内容や形式も多様であるため，自分の欲しい情報を手に入れるためには時間や手間といった大きなコストを支払う必要がある．そのため，より自分が欲しいと思った情報に簡単にアクセスできるように，ウェブ上から何らかの情報を自動的に抽出しようという試みが，これまで多く行われてきた．

その中でも人物に関する情報は特に重要な情報であり，実際検索エンジンに投げられるクエリには，約 30 %の割合で人名が含まれると言われている [12]．特定の人物について記述されたウェブ上のドキュメントでは，その人物が生年月日や職業といった多くの属性情報と結び付けられており，その情報を正確に抽出することで，後述するような様々なタスクに応用することが可能である．そこで，本研究では，入力として与えられた人名に対して，その人物に関するウェブ上のドキュメントから属性情報を抽出する事を大きな目標とする．

[1] http://ja.wikipedia.org/wiki/

図 1.1: 二人の “Jim Clark”

1.3 ウェブから人物の属性情報を抽出する際の問題点

ウェブ上のドキュメントから人物の属性情報を抽出する際にはいくつかの考慮すべき事項がある．以下，その事項について検討する．

1.3.1 同姓同名問題

ウェブ上のドキュメントで人名を扱う際に起こる問題の一つが同姓同名の問題である．例えば，“Jim Clark” というような一般的な名前の人物の属性情報を抽出しようとすると，集められたドキュメントの集合が，全て同じ人物について書かれているとは限らない．つまり，人名には曖昧性が存在するため，その曖昧性を解消しない限り，正確な属性の抽出を行うことが出来ない．

これは，固有表現でよく起こる問題であり，人名以外にも地名や組織名といったものでも，同様の問題が確認できる．このような問題は一般的には語義の曖昧性解消問題と呼ばれるものの一部である．ただし，語義の曖昧性については，予め辞書などに登録されているため，全体でどれくらい別の意味があるかといったことが容易に把握できる（ex. 「いらっしゃる」の意味は行く，来る，居るなど）．それに対し，固有表現の曖昧性解消の場合はいくつの異なる実体が存在するか把握することが，実質的に不可能であるため，より難しい問題となっている．

最近の同姓同名問題の解決の研究では，人物の属性情報を用いることでより精度を高めることが出来ることが示されている [10, 26]．例えば，“Jim Clark” という人名について調べてみると，ある一人はレーシングドライバー（図 1.1 左）であり，またある一人はネットスケープ社の創始者（図 1.1 右）であり，大学教授でもある．この二人は

名前という属性で言えば，全く同じであるが，職業という属性に着目することで，ドライバーの “Jim Clark” あるいは教授の “Jim Clark” であると判断することが出来る．

つまり，人物の属性情報が分かれば，同姓同名の解決に役立てることが出来ることが分かる．と同時に，同姓同名問題が上手く解決できることで，属性情報の抽出精度も向上する事が考えられる．そのため，これらの問題は相互に干渉しあう問題であると考えられる．

1.3.2 別名問題

また，同一人物でも別の名前を持っていることがあるので注意が必要である．例えば，“Jim Clark” は正式に “James H. Clark” と書かれるケースが考えられる．このケースの場合，“Jim Clark” が出現した文書から情報を抽出したとしても，それが十分な抽出にはなっていないことがある．

この問題を解決するために，外間らや本間らはウェブデータから，ある人物の別名（呼称）を抽出する研究を行ってきた [36, 37]．これらの手法は主に人名が出現する前後のパターンを利用して，その抽出を行っている．このとき，人物の属性情報が上手く抽出出来れば，名前という属性以外が共通する人物は同じ人物であると推定できるため，属性情報を利用することで，別名問題の解決にも役立てることが出来ると考えられる．同時に，別名問題が解決できることによって，属性情報の抽出も対象範囲を広げることが出来るため，再現率の向上に役立てることが出来る．そのため，この問題は同姓同名問題と同じように，属性情報抽出のタスクと相互に干渉しあう問題であると考えられる．

また，別名は人物の属性であるととらえることもできる．ある人物が「職業」や,「所属」と同じように「別名」という属性を持っているととらえる考え方である．本研究ではこの立場で，“Other name” という属性も抽出する事を目的としている．

1.3.3 属性の一意性と不変性

人物の属性情報の定義は前述であるが，属性の種類によって，一意性や不変性があるかどうか異なるので注意が必要である．例えば，“Barack Obama” の “Date of Birth” は “August 4, 1961” で一意に定まり，変動することも無い．その一方で，“Occupation” は現在は “President of the United States” であるが，任期が終われば “Occupation” も変わるはずである．また職業を二つ以上もつ人物も存在するので，“Occupation” という属性は一意に定まらず，変動する事もあると言える．このように，属性によって，その性質も異なり，その変動期間も異なる（例えば，職業は短期間では変わらない可能性が高いが，趣味などは短期間で変わる可能性がある.）と考えられるので注意する必要がある．

1.4 人物の属性情報を抽出する意義

前述した，同姓同名問題や別名問題の解決だけでなく，人物の属性情報を抽出出来ると，様々なタスクに応用する事が出来る．以下，具体的に応用できるタスクについて述べる．

1.4.1 質問応答タスク

質問応答タスクとは，ユーザが入力した自然言語での質問文に対して，何らかの手法で機械的に答えを作り，質問に応答するタスクである．その問いは例えば,「イグアナの全長はどのくらいですか」という問いであり，答えは「1.5〜1.8 メートルである」などという風になる．一般的な質問応答タスクにおける手法は，答えのタイプから推定しなければならないが，その推定は難度が高く，比較的簡単な問題の応答にも失敗するケースがある [33]．その一方，ある事柄の属性値が答えのタイプとなることが多い [39]．前述の問いもそのタイプであり，人物の属性情報を取得する事が出来れば，その質問に応答する事が出来る．

1.4.2 検索時のクエリ拡張

ウェブで検索を行う際に，ユーザが入力したクエリによっては，ユーザの求めていた情報が得られないことが多々ある．その際に関連するクエリを推薦することにより，ユーザが求めている情報が得られるようにするタスクがクエリ拡張である．Pasca などの研究 [23] でクエリログから属性情報が抽出できるように，ユーザが検索したい情報は，あるインスタンスの属性情報であることが多いため，属性情報を上手く抽出することで，クエリ拡張にも利用する事が可能である．

1.5 論文の構成

本論文の構成を述べる．以下，2 章で関連研究についてまとめ，3 章で目的と手法を整理し，構想を示す．続いて 4 章で提案手法について述べ，5 章でその評価を行う．6 章で本研究について議論を行い，7 章でまとめる．

2 関連研究

2.1 同姓同名問題に関する研究

人名の曖昧性解消タスクは，自然言語処理の分野では，重要な課題であり，以前から盛んに研究が行われてきた [4, 31, 41]．人名の曖昧性解消のタスクは一般的に，以下のようなステップに沿って行われる．以下，ステップに沿って，同姓同名問題の解決にどのような手法が用いられているか述べる．

人名の影響範囲の設定 人名の影響範囲の設定は，どの範囲で曖昧性解消を行うかの元となるため，問題の設定として非常に重要であり，ウェブから文書を取得し，検索エンジンでターゲットとなる人名を検索して，返ってきた結果を直接対象の文書とする研究 [6, 8, 10, 19, 26, 30, 42] や，新聞記事などの一記事を範囲とする研究 [1, 43] がある．

素性の選択 素性の選択は手法の差異を生み出す重要な部分である．単語を Boolean，または $tfidf$ で重みを付けて利用する研究 [5, 10, 26]，それを拡張した研究 [43] の他，特殊な Bigram を用いたもの [38]，固有表現の出現頻度を用いたもの [6, 30] がある．また，属性情報を用いた研究 [6, 10, 26] もあり，正確に属性情報を抽出する事が出来れば，強力な素性となる．

クラスタリング クラスタリングを行う際に最も重要なのが，対象となる文書間の類似度をどのように決定するかである．Cosine 類似度が最も多く利用され [1, 26]，他に相互情報量 [10] や，カルバック・ライブラー情報量などが用いられている [5, 8]．

このように，人名の曖昧性解消の研究の多くは，ドキュメントから単語単位で $tfidf$ を求め，それらをベクトルとしてコサイン類似度を利用してクラスタリングをするなどといった，単純な手法が中心であった．評価しているデータセットが必ずしも同じでは無いので，単純な比較は出来ないが，単純な語の $tfidf$ を用いるだけではある程度の精度までしか出すことは出来ず，人物の属性情報を用いることでより精度を高めることが出来ることが示されている [10, 26]．

それらの研究では，人名の曖昧性解消タスクのサブタスクとして，人物の属性情報の抽出が行われている．Mann らは名前と生年月日などの属性名－属性値のペアをシードとして用意し，ウェブページから言語的なパターンを抽出することで，対象とする人物の誕生日や出身地，職業，国籍，夫婦関係などを抽出した [10]．上田らは語尾や文字数に着目し，ヒューリスティックなルールを用いてウェブページから職業，所属，役職などの情報を抽出した [11]．また，木村らは予め職業なり，地名なりのリスト（辞書）を持ってきて，そのリストに載っているような語が出現した場合に，その人物の職業や出身地を決定し，その結果を用いて形態素解析器を学習させ，職業情報の抽出を行っている [26]．

このように，人名の曖昧性解消タスクでは属性情報を利用することで，精度を上げる研究が行われており，前述のように人名の曖昧性が解消できることによって，属性情報の抽出精度も上げることが出来るため，属性情報抽出の研究をする際には，同姓同名問題に関する研究について考慮する事が重要である．

2.2 情報抽出に関する研究

そもそも人物の属性情報の抽出は，情報抽出の研究の一部である．情報抽出の研究分野では，近年，二つのエンティティとその関係を組にした，三つ組構造の抽出に関する研究が盛んに行われている．これらの研究は，二つのエンティティを E1，E2 とし，その関係を R としたとき，何が入力として与えられ，何を抽出するかに着目する事で，分類する事が出来る．例えば，E1，E2，R の全てが未知であったときに，その三つ組構造を抽出する研究も行われている．Banko らは小規模のコーパスからルールベースを用いて，人手でアノテーションしたデータを用いることなく，分類器を作り，それを対象文書に適用することで，三つ組構造を抽出する手法を提案した [20]．また，Hasegawa らは二つのエンティティを NE タガーでタグ付けし，そのエンティティ間の関係をクラスタリングを利用して，ラベル付けする手法を提案している [35]．

これらの研究は 3 つ組構造が全く未知の場合に，抽出を行う研究であった．一方，R という関係を明確もしくは非明確に与え，その関係にあるエンティティを 2 つ抽出する研究も行われている．その分野のさきがけとなった有名な研究に Brin のものがある．Brin はある関係 R にある，二つのエンティティを正解例（シード）として数組与え，そのシードがウェブ上に出現したときにパターンを抽出し，今度はそのパターンを元にシードを拡張する手法（DIPRE システム）を提案し，実際に本の名前と著者の名前を正解例として，二つのエンティティの組を拡張する研究を行った [29]．Brin の研究に基づき，Agichtein らや Pantel らが手法の改良を行っている．Agichtein らは，DIPRE システムを改良し，“ORGANIZATION” と “LOCATION” の二つの組をシードとして，パターンを抽出し，シードの拡張を行った [9]．その際パターンの信頼性を評価し，信頼性の高いパターンを利用することで，繰り返しの抽出を可能にした．Pantel らは同様の手法をさらに改良し，パターンだけでなく，抽出したエンティティに対する信頼性も評価し，信頼性の高いパターンと，信頼は出来ないがエンティティのカバー率が高い一般的なパターンを使い分けて，ブートストラッピングを行い，シードの拡張を行った [25]．また，高橋らは抽象化したパターンを利用することで，日本語文書から，ドメインを限定せずに，対象物，属性名，属性値の三つ組構造を抽出する手法を提案した [39]．関根らは質問応答システムに利用するため，学習データを人手で作成し，単語をベースに作られたパターンの頻度を利用して，項目，属性名，属性値の三つ組構造を抽出した [33]．

関係を与え，エンティティの組を抽出する研究がある一方で，関係 R とエンティティを一つ（E1）を与え，エンティティと関係 R をもつ，新しいエンティティ（E2）を抽出する研究も行われている．クラスから属性名を抽出する研究も，E1 をクラス名，R

を属性関係とおけば，この分野に含まれると言える．その際，研究によって予め抽出対象となる E2 に対してシードとなる正解例を与える場合と，与えない場合がある．シードを与える研究として，Pasca，Wang ら，Reisinger らによるものある．Pasca はクラス名と少量のインスタンスを元に，検索エンジンのクエリログを利用して，属性名の抽出を行った [23]．Reisinger らはその手法を改良し，クエリログに加え，ウェブドキュメントを利用することで，精度の向上を行った [16]．また，Wang らはクラス名と，それに属するインスタンスを与え，そのインスタンスの拡張を行った．この研究は前 2 者の研究と多少異なり，前者はインスタンスを属性名抽出に利用したのに対し，後者はインスタンス集合の拡張を目的としている．クラスが与えられたときの属性名の抽出は質問応答の分野でも行われており，Tokunaga らは，単純な “[クラス名] の [属性名]” といったパターンを利用して，ヒューリスティックなルールを用いて属性名の抽出を行った [18]．

クラス名から属性名のみを抽出する研究だけでなく，属性名を抽出したあとに，その属性値を抽出する研究も行われている．Yoshinaga らはクラスとインスタンスが与えられたとき，ウェブを利用して，タグなどで強調された部分から属性名を抽出し，次に <table> タグなどの半構造化されたデータから属性値を抽出している [24]．また，Ravi らは，Pasca の手法 [23] を元に，HTML のタグ構造を利用して，属性名だけでなく，属性値の抽出を行っている [32]．

本研究は E1 を人物名，R を属性関係として与え，属性値となる E2 を抽出する研究である．

2.3 人物の情報抽出に関する研究

一般的なドメインに依存しない情報抽出の研究が広く行われている一方で，人物に関する情報抽出の研究もおこなわれている．その一部は節 2.1 で述べた通りである．森らは Jaccard 係数を用いて，ウェブから人物のキーワードを抽出する研究を行った [15]．その際，背景となる context ワードを設定し，それを用いて精度の向上を行っている．また，Dingli らはある部署に所属している人物に対して，人物の肩書きやメールアドレス，電話番号などを抽出し，一緒に働いている人物のグループを見つけだす研究を行った [2]．木村らはウェブから日付表現とその日付に対応する人物の行動を集め，年表化する研究を行った [27]．

3 構想

この章では取得する属性の定義，および属性情報の抽出手法の分類，その際に考慮すべき点などを説明し，どのような手法を選ぶべきか考察する．その上で，具体的な研究の目的を設定する．

3.1 取得する属性情報

人物の属性情報の定義は前述であるが，多くの属性がある中でどのような属性を取得すべきであるかを考える必要がある．ここで，その人物をより特徴づけるような属性を重要な属性であるとすると，不変性のある属性か，あるいは変わることがあるとしても頻繁には変わらない属性を選ぶべきだと考えられる．何故なら，頻繁に変わってしまう属性は，抽出時期に大きく依存し，応用的なタスクで利用する際にあまり役に立たないと考えられるためである．そこで，本研究では，上記のような特徴を持つ属性を抽出の対象とする．具体的には，表 3.1 のような 16 種類の属性を対象とする．これらの属性は第 2 回 Web People Search Evaluation Workshop(WePS)[2]のサブタスク（Attribution Extraction Task）で評価に利用されたものである [34].

3.2 属性情報の抽出手法

ウェブ上から属性情報を抽出する際には，いくつかのやり方があるが，大きく分けると二つのやり方がある．一つは構造的なデータからの抽出，もう一つは純粋な文章からの抽出である．

3.2.1 構造的データからの抽出

構造的データからの抽出とは，属性情報が何らかの構造をもって出現しており，その構造を利用して属性情報を抽出する手法である．ここでいう構造とは，HTML のタグ構造を指す．具体的には，テーブルタグ（<table>）や，列挙タグ（<li>）などが考えられる．人物のプロフィールについてまとめたページであれば，このようなタグ上の構造があることが多く，属性名と属性値（定義は前述）の組で出現する事が多いため，それを利用して属性情報を抽出する事が出来る．もともと構造的なデータを利用しているため，そこからデータを抽出する事はそれほど難しくは無いと考えられる．

[2]http://nlp.uned.es/weps/

表 3.1: 抽出する属性

属性名	内容
Affiliation	所属している組織・団体など
Award	受賞した賞など
Birthplace	生まれた場所，出身地
Date of birth	生年月日，またはその一部
Degree	取得した学位
Email	電子メールアドレス
FAX	ファックス番号
Major	専攻している学問
Mentor	大学などでのメンター
Nationality	国籍
Occupation	職業
Other name	その人の別の呼び名
Phone	電話番号
Relatives	親類や親戚など
School	所属，あるいは卒業した高校・大学など
Web site	その人物のもつウェブサイトのアドレス

3.2.2 テキストからの抽出

構造的なデータから属性情報を抽出する事はそれほど難しくは無いが，全ての属性情報がそのような構造的なデータとしてHTML上で表現されるとは限らない．実際にはその人物の紹介文だったり，略歴といった形で，普通のテキストとして表現されることも多い．この場合，構造が利用できない分，構造的データからの抽出よりも難度が高くなることに注意する必要がある．

構造的データからの抽出とテキストからの抽出はどちらも重要であり，実際にある人物の属性情報を得たいと思った場合は，それらを上手く組み合わせる必要があるが，本研究ではより難しいタスクである，テキストからの抽出を主な対象とする．なぜなら，構造的データはそれ自体がすでに利用価値が高く，バックグラウンドではデータベースのような形で保持されている可能性が高いため，改めて抽出する必要性が低いためである．その逆にテキストからの抽出は，必要性が高いだけでなく，難度が高いチャレンジングなタスクである．

3.3 対象言語

構造的なデータからの抽出では，対象とする言語はさほど問題にはならないが，テキストからの抽出では，属性情報の抽出手法は言語に依存する．例えば，文節ごとに区切りたいといった場合，英語であればスペースごとに区切ることである程度目的を達成する事が出来るが，日本語を用いた場合，mecab[3]や chasen[4]といった形態素解析機，もしくは cabocha[5]といった構文解析機を利用する必要がある．本研究では最も一般的に用いられている言語である，英語で書かれたテキストから，属性情報の抽出を行う．

3.4 対象人物の影響範囲

ウェブ上から人物の属性情報を抽出する場合，ウェブページ中のどの部分がその人物に対する記述なのか，その範囲を適切に設定する必要がある．最も単純な抽出範囲は一つの HTML ごとの抽出であるが，一つの HTML の記述全てが対象の人物について書いてあるとは限らない．そのため，どこからどこまでが対象の人物に対する記述であり，求める内容かどうかを上手く判断する必要がある．その範囲を判断するには，例えば HTML タグの木構造を利用する方法がある．木構造を利用して HTML をブロックごとに分け，それぞれのブロックに対して，その人物に対する記述であるかどうかの信頼度を計算し，その信頼度が最も高い部分を対象の人物の影響範囲とする手法などである．何れにせよ，適切に抽出範囲を設定できれば，属性情報の抽出精度も上がり，逆に属性情報の抽出精度を上げることで，より正確な影響範囲の切り出しが可能になると考えられる．本論文ではこの問題には触れていないが，より正確な人物の属性情報の抽出を考えるうえで，適切な影響範囲の切り出しは重要なタスクである．

3.5 目的と手法の整理

このように，適切な影響範囲の切り出しは重要なタスクであるが，考慮すべき点の多い，難度が高いタスクであり，現時点で属性情報の抽出と同時に行うことは煩雑であるため，本研究では抽出範囲を一つの HTML ページ単位とし，ウェブドキュメントは与えられたものとして，属性情報の抽出を行う．すなわち，対象となる人物名，および人物名が出現するウェブドキュメントを入力とし，その人物が持つ属性名と属性値のペアのリストを出力とする．また，構造的データからではなく，テキストからの抽出を目的とした研究を行う．

[3]http://mecab.sourceforge.net/
[4]http://chasen.naist.jp/hiki/ChaSen/
[5]http://chasen.org/ taku/software/cabocha/

4　提案手法

この章では提案手法について説明する．ウェブは非常に有用なリソースであるが，その文書には新聞や雑誌などに比べ整形されていない文章が多いため，そのまま属性情報を抽出しようとすると，意図しない無関係な語も多く含まれてしまう．そのため，選んだ属性値の中から適切な属性値を選ぶ操作が必要となる．そこで，本論文では以下のように二段階の手法を提案する．

Step1: 属性値の候補となる語を見つける
Step2: 候補からふさわしい属性値を選択する

つまり，Step1 で可能な限り属性値となりうる表現を集めて，その候補とし，Step2 において，集めた候補の中から正しい属性値を選別し，その他正しくない候補をふるいにかける．これは，Step1 はなるべく再現率を高める段階であり，Step2 は適合率を高める段階であることを示している．全体の流れを図 4.1 に示す．以下，各 Step の説明の前に，ウェブドキュメントに対して行う前処理について説明し，その後各 Step について詳細な説明を行う．

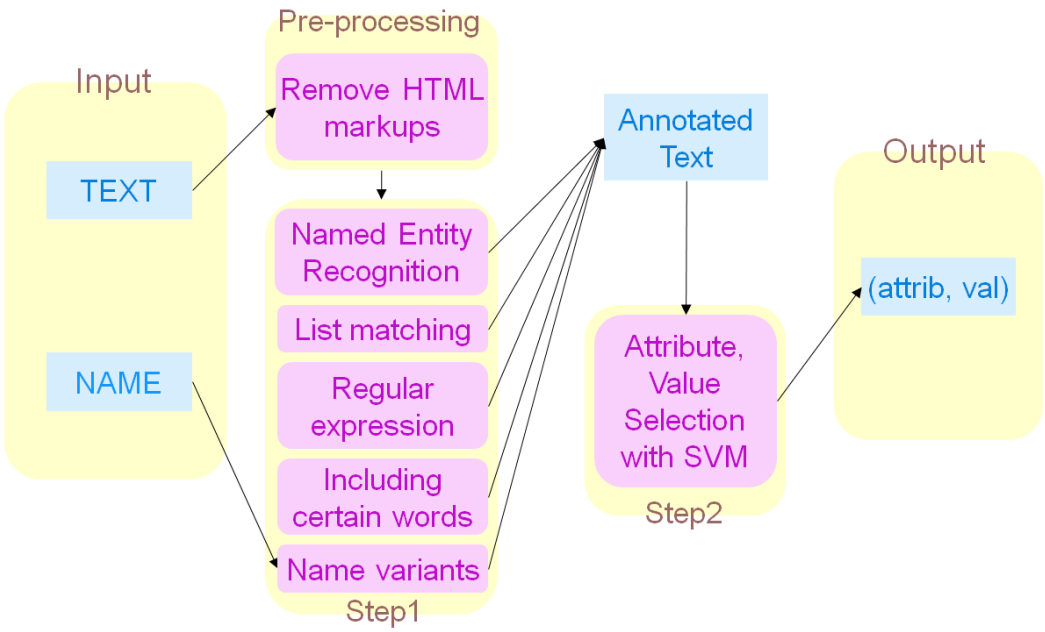

図 4.1: 全体の流れ

表 4.1: HTML の整形ルール

正規表現	内容
$s/<head>.*?</head>//gsi$	ヘッダ部分を削除
$s/<script[^<>]*?>.*?</script>//gsi$	JavaScript 部分を削除
$s/<style[^<>]*?>.*?</style>//gsi$	スタイルの指定を削除
$s/<!--.*?-->//gs$, $s/<!-->.*?<!-->//gs$	コメント部分を削除
$s/\&.\{1,6\};//g$	特殊文字の削除
$s/<[^<>]*?>//g$	その他タグの削除

4.1 前処理

本研究の入力は人名とウェブドキュメントである．ここでウェブドキュメントとはHTML で書かれたページとする（従って，PDF ファイルやパワーポイントなど，HTML 以外のページは対象外とし，入力として与えないこととする）．ここで，HTML ページには表示されているテキストだけでなく，様々なタグやスタイル，JavaScript などが含まれている．今回は構造的なデータからの抽出ではなく，テキストからの抽出を対象にしているため，これら余計な情報を取り除く必要がある．具体的な HTML の整形ルール[6]を表 4.1 に示す.

4.2 Step1: 属性値候補のマーク

Step1 では前処理の終ったテキストに対して，属性値候補の出現位置をマーク（ここでは，候補が出現した場所に何らかのしるしをつけて覚えておく，という意味で使用している．以下同様.）する．属性値の候補となりうるものは，それぞれの属性によって大きく異なる．例えば，生年月日を抽出しようと思えばその候補は数字などの日付表現となるが，所属を抽出しようと思えばその候補は大学や会社など，組織の名前になる．このとき数字などの日付表現であれば，正規表現で簡単に表すことができ，簡単に属性値の候補を見つけだすことが出来るが，大学や会社などの名前を見つけだすには正規表現などでは難しい．そのため，何らかの方法でその候補を見つけだす必要がある．提案手法では，リストを用いたマッチングや固有表現，特定の語を含む表現などを利用して属性値の候補を特定している．以下，正規表現を含めたそれぞれの手法について，どのような属性に用いるかも含め，説明する．

[6]プログラミング言語，Perl の正規表現に準拠.

表 4.2: 人名の表記ゆれの生成ルールと具体例

ルール	例: Barack Obama
（名前）＋（苗字）	Barack Obama
（苗字）＋（名前）	Obama Barack
間にカンマが入るもの	Obama, Barack
間に一単語入るもの	Barack Hussein Obama
（名前のイニシャル）＋（苗字）	B Obama
（苗字）＋（カンマ）＋（名前のイニシャル）	Obama, B
（名前のイニシャル）＋（一単語）＋（苗字）	B. H. Obama

表 4.3: 組み合わせる呼び名

Mr., Mrs. Miss., Ms, Rev., Prof., President, Minister, Prime Minister, General, Madame, Lady, Dr., King., Queen, Vice President, Senator, Lawyer, Major, Maj., General, Gen., Maj. Gen., Major General, Jr.

4.2.1 人名のマーク

人名は属性抽出対象の人名と，その他の人名について分けて考える必要がある．例えば，“Other name” という属性の場合は，属性抽出対象の人名であるし，“Relatives” や “Mentor” に関しては，対象の人物以外の人名を利用する．対象の人物以外の人名のマークに関しては，後述の固有表現抽出やリストを用いるため，ここでは属性抽出対象の人名のマークについて説明する．

抽出対象の人名は入力として与えられるが，入力としては最も単純なファーストネームとラストネームの組み合わせを想定している．ただし文章中では，そのままファーストネーム＋ラストネームといった形で人名が表現されることはまれで，ファーストネームやラストネーム単体で出現するだけでなく，ミドルネームや，何らかの敬称（Mr., Mrs. など）や呼称（Dr., Prof. など）を伴っている場合が多い．これら抽出対象の人名を正確にマークするため，まず表 4.2 のようなルールで表記ゆれのパターンを生成し，表 4.3 に示すような呼び名を組み合わせることで，候補となる表現のリストを生成する．その後，リストにある表現とテキストで単純なマッチングを行い，抽出対象の人名をマークする．

4.2.2 リストを用いたマッチング

“Affiliation” などのように，属性値の候補が大学名や会社名などの組織名となる属性や，“Nationality” のような国名が候補となる属性，“Birthplace” のような地名が候

表 4.4: マッチングに用いたリスト

属性名	リスト内容
Affiliation	会社 (43047)，高校 (25271)，大学 (1726)，政府機関 (523)
Award	賞の名前 (459)
Degree	学位 (175)
Major	専攻名 (318)
Mentor & Relatives	人物名 (9987)
Nationality	国名 (444)
Occupation	職業名 (720)
School	高校名 (25271)，大学名 (1726)

補となる属性などは，表現のされかたに統一性はあまりないものの，その候補はある程度限定されている（例えば国名などはせいぜい 300 程度である）．このような特徴をもつ属性の，属性値候補は予め何らかの方法を用いて，候補のリストを作り，単純なマッチングを取る方法が効果的である．実際にマッチングに用いたリストを表 4.4 にまとめる．括弧内はそのリストに載っている要素の数を表す．

リストは Wikipedia から “list of . . .” などのページを利用して作成した．この抽出手法の対象となる属性は，“Affiliation”，“Award”，“Degree”，“Major”，“Mentor”，“Nationality”，“Occupation”，“Relatives”，“School” である．

4.2.3 特定の語を含む表現

“Award” に関する属性値の候補は，対象となる人物が取った賞などである．これらは上記のように，基本的には予め指定されたリストとのマッチングを行っている．しかし，それだけでなく，それらの賞の名前の多くは “Award” や “Prize” など特定の語を含み，なおかつそれらの名前は前置詞などを除くすべての単語が大文字で始まることが多いという特徴を持っている．そこで，“Award” や “Prize” を含み，大文字で始まる単語の組み合わせになっている表現を，属性 “Award” の候補とする[7]．

特定の語を含む表現は，現在 “Award” でのみ用いている．

4.2.4 固有表現抽出

“Relatives” や “Birthplace”，“Affiliation” などの属性は，属性値の候補となる値が，人名や地名，組織名であり，これらは固有表現[8]と呼ばれるものである．固有表現をテ

[7]正規表現で表すと，以下である．
$/(([A-Z][a-z\']*|of|the|for|\d+(st|nd|rd|th))\s)+?([Pp]rize|[Aa]wards?)/g$

[8]英語で Named Entity と呼ばれる，実体をもつもの．

表 4.5: 用いた正規表現

属性名	利用した正規表現
Email	$/[\backslash w \backslash - \backslash .] + \backslash @(\backslash w[\backslash w \backslash -] * \backslash .) + [\backslash w \backslash -] + /gi$
FAX & Phone	$/((\backslash + \backslash d\{1,3\}[-\backslash s]?\backslash(?\backslash d\backslash)?[-\backslash s]?\backslash d\{1,5\})\|(\backslash(?\backslash d\{2,6\}\backslash)?))$ $[-\backslash s]?\backslash d\{3,4\}[-\backslash s]?\backslash d\{4\}((\backslash sx\|\backslash sext)\backslash d\{1,5\})\{0,1\}/gi$
Web site	$/(\backslash d1,3\backslash .\backslash d1,3\backslash .\backslash d1,3\backslash .\backslash d1,3\|(https? : \backslash/\backslash/\|www[\backslash w \backslash -] * \backslash .)$ $[\backslash w\backslash .\backslash -]+)(\backslash : \backslash d*)?\backslash/?[\backslash w\backslash .\backslash - \backslash]* (\backslash?\backslash S+)?/gio)$

キスト中から抽出しようという研究は昔から行われており [14, 21]，現在では CRF などの機械学習を用いて，ある程度の精度で抽出に成功している．また，その手法を利用して，実際に固有表現抽出を行うツールも公開されているため，そのツールを用いて，固有表現の抽出を行う．本研究では，固有表現抽出を行うツールとしてポピュラーな Stanford Named Entity Recognizer [14] を用いた．このツールを利用すると，入力されたテキストに対して，“PERSON”，“ORGANIZATION”，“LOCATION” のタグをつけることが出来る．

本研究では，“PERSON” と判断されたものは，“Mentor”，“Relatives” に，“ORGANIZATION” と判断されたものは，“Affiliation” に，“LOCATION” と判断されたものは，“Birthplace” にそれぞれ利用した．

4.2.5 正規表現

“Date of birth” や “Email” などは，表現の方法が限られており，数字やアルファベットの単純な組み合わせとして，正規表現で簡単に表現できる．ただし，データベースや新聞，雑誌の記事と違い，ウェブ上に出現する属性値は，その表現方法が多様であることに注意が必要である．例えば，“Date of birth” の属性値の表現としては，“August 4, 1976” などが最も一般的であるが，“1976/8/4” や，“1976.8” などの表現も見受けられる．ある程度表現の仕方が定まっているとはいえ，このような表現の多様性を考慮に入れる必要がある．実際に提案手法で用いた正規表現を表 4.5 に示す．

正規表現による属性値候補のマークは，“Email”，“FAX”，“Phone”，“Web site” の各属性に対して行う．

4.3 Step1 のまとめ

Step1 では，上記のような手法を用いて，各属性の属性値の候補を見つけだし，その出現位置をマークする．具体的に Step1 を行った後のイメージが図 4.2 である．図 4.2 はある文章を入力したときに，それぞれどの部分がどのような属性値の候補としてマークされたかを示している．また，それぞれの属性に対して，どのような手法を用いて，属性値の候補をマークしたかを表 4.6 にまとめる．

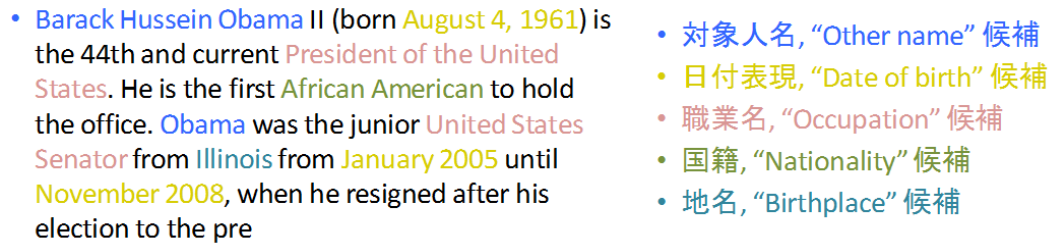

図 4.2: Step1 の処理例

表 4.6: Step1 での属性値候補のマーク手法

属性名	属性値候補のマーク手法
Affiliation	リストマッチング，固有表現抽出
Award	リストマッチング，特定表現の内包
Birthplace	リストマッチング，固有表現抽出
Date of birth	正規表現
Degree	リストマッチング
Email	正規表現
FAX	正規表現
Major	リストマッチング
Mentor	リストマッチング，固有表現抽出
Nationality	リストマッチング
Occupation	リストマッチング
Other name	別名生成によるマッチング
Phone	正規表現
Relatives	リストマッチング，固有表現抽出
School	リストマッチング
Web site	正規表現

4.4 Step2 の導入: 正しい属性値の選別

前述のように Step1 で選ばれた属性値の候補は，ウェブドキュメントから抽出されているため，ノイズが多く，適切な選別が必要になってくる．Step2 では，対象の人物の属性値としてふさわしいものを選び，それ以外のノイズをいかに綺麗に取り除けるかが重要である．ここで，どのような属性値の候補が正しい対象人物の属性値であるかを考える．実際に人が文章を見て，これがこの人の属性値だと判断する際に，どのような基準を持っているか考えると，以下のような仮説を立てることが出来る．

属性値になり易い表現 “Date of birth” という属性では，日付表現が属性値の候補となるが，表現によって，それが正しい属性値になる可能性が異なる．例えば，“August 4, 1976” という表現と，“76.8.4” という表現があったとき，人が見ると直感的に前者の方が日付表現らしいことが分かる．これは，前者の表現の場合，日付表現以外にはまずありえないが，後者の表現の場合は例えば IP アドレスの一部分などといったことがあり得るためである．このとき，前者の方が属性値の候補になりやすいと言える．

対象人名との距離が近い ウェブドキュメント中に属性値の候補が出現したとき，例えその表現が同じであったとしても，対象人物の属性値としてふさわしいかどうかには差が出てくる．ある属性値の候補が対象人物の人名とかけ離れて出現したとすると，それは属性値としてはふさわしくないことが多い．例えば “Date of birth” の場合を考えると，対象人名の近くに “August 4, 1976” という候補が出現し，遠くに “May 6, 1984” という候補が出現したとき，属性値の表現としてはどちらも同じように確からしいが，対象人名の遠くに出現した日付表現は，対象人名とかかわりの薄い日付である可能性が高い．つまり，対象人名が出現した位置に近ければ近いほど，属性値としてはふさわしいと考えることが出来る．

他の人名が近くに出てこない 対象人名との距離が近いほど，その人物の属性値としてふさわしいのと同様に，対象人物以外の人名が近くに出現する場合，その人物と結びついている属性値である可能性が高い．つまり，“Barack Obama” の “Date of birth” を求めたいと思ったときに，“Michelle Obama” という人名が近くに出現している日付表現は，“Michelle Obama” に関連する日付表現である可能性が高いため，対象人物以外の人名が近くに出現する場合は，対象人物の属性値としてはふさわしくないと考えることが出来る．

手掛かり表現の有無 例えば “Date of birth” という属性の場合，単純に日付表現だけが有るよりも，“Barack Obama was born in August 4, 1961” などのように，“was born in” といった表現が有る方が，より対象の人物の属性値である可能性が高い．それら手掛かり表現は各属性によって異なるだけでなく，属性値候補の手前に出現しやすいパターン，属性値候補の後ろに出現しやすいパターンがあることに注意が必要である．このとき，単純な “Place of birth: Tokyo, Japan” といった表現の場合でも，“Place of birth” といったフレーズは “Birthplace” という属性の手掛かり表現の一種と考えることが出来る．

以上のような仮説に基づき，属性値候補の選別を行う．その際に，それぞれの仮説によるスコアの重みづけや，手掛かり表現の取得，その閾値の設定などをヒューリスティックに行った場合と，それぞれの仮説を表現する素性をつくり，機械学習によって選別を行う場合の 2 パターンの手法を検討し，実際に評価する．以下，ヒューリスティック，機械学習，それぞれの詳細な手法について説明する．

表 4.7: 日付表現のスコア

スコア	正規表現	例
高	$/(MONTH\_EXP)[\backslash.,]?\backslash s*\backslash d\{1,2\}[\backslash.,]?\backslash s+\backslash d\{1,4\}/gi$	August 4, 1976
∧	$/\backslash d\{1,2\}[\backslash.,]?\backslash s+(MONTH\_EXP)[\backslash.,]?\backslash s*\backslash d\{1,4\}/gi$	4 Augsut, 1976
‖	$/(MONTH\_EXP)[\backslash.,]?\backslash s*\backslash d\{3,4\}/gi$	Augst, 1976
‖	$/\backslash d\{1,2\}[\backslash s\backslash/]+\backslash d\{1,2\}[\backslash s\backslash/]+\backslash d\{3,4\}/g$	8 4 1976
‖	$/\backslash d\{1,4\}\backslash/\backslash d\{1,2\}\backslash/\backslash d\{1,2\}/g$	1976/8/4
∨	$/\backslash d\{1,4\}\backslash.\backslash d\{1,2\}\backslash.\backslash d\{1,2\}/g$	1976.8.4
低	$/\backslash d\{1,4\}\backslash/\backslash d\{1,2\}/g$	1976/8

4.5 Step2: ヒューリスティックな手法

属性値候補の選別を行うために，それぞれの属性の属性値候補ごとに,「どれだけ属性値としてふさわしいか」を表すスコアを付け，そのスコアが一定の閾値を上回ったものを属性値として採用する．具体的には上述の，4 つの仮説をもとにつけたスコアを式 4.1 のように掛け合わせて，スコアとする．ここで，$S_{Attribute}$ はどれくらい属性値になりやすい表現か，$S_{TarDist}$ は対象人名との距離に基づくスコア，$S_{OthDist}$ は他の人名との距離に基づくスコア，S_{Cue} は手掛かり表現の有無およびその距離に基づくスコアを表す．

$$Score = S_{Attribute} \times S_{TarDist} \times S_{OthDist} \times S_{Cue} \tag{4.1}$$

以下，それぞれのスコアの求め方について説明する．

4.5.1 属性値になり易い表現

属性値になりやすいかどうか表現によって差をつけている属性は，“Affiliation” と “Date of birth” である．“Affiliation” ではリストマッチングによるもののスコアを高くし，固有表現抽出によるスコアを低く設定している．これは，固有表現抽出器によって抽出された “ORGANIZATION” というタグの信頼性が低い（大文字から始まる単語が連続すると，組織であるなしに関わらず “ORGANIZATION” というタグが振られてしまう可能性が高い）ためである．

“Date of birth” では，その表現によって表 4.7 のように，スコアの高低を定めており，一般的に日付表現として用いられる可能性が高いもののスコアを高く，用いられる可能性が低いもののスコアを低く設定している．ここで，$(MONTH\_EXP)$ は表 4.8 のような表現を表す．このようにして，$S_{Attribute}$ を定める．

表 4.8: 月の名前の表現

January, February, March, April, May, June, July, August, September, October, November, December, Jan, Feb, Mar, Apr, Jun, Jul, Aug, Sep, Sept, Oct, Nov, Dec

4.5.2 対象人名との距離

対象人名との距離が近ければ近いほど高く，遠くなるにつれて低いスコアになるように，式 4.2 のようにスコアを設定する．ここで，Dec_{Tar} は 0 から 1 の値を取る減衰定数（属性によって異なる），$Dist_{Tar}$ は属性値候補と対象人名との距離とする．距離は単語数で表す．つまり，”Barack Obama was born in August 4, 1961”であれば，属性値候補 “August 4, 1961””と対象人名 “Barack Obama” との距離は 4 となり，このとき減衰定数 Dec_{Tar} が 0.99 であれば，$S_{TarDist}$ は約 0.961 となる．

$$S_{TarDist} = Dec_{Tar}^{Dist_{Tar}} \tag{4.2}$$

4.5.3 他の人名との距離

他の人名との距離が近ければ近いほど低く，遠くなるにつれて高いスコアになるように，式 4.3 のようにスコアを設定する．ここで，Dec_{Oth} は 0 から 1 の値を取る減衰定数（属性によって異なる），$Dist_{Oth}$ は属性値候補と対象人名以外の人名との距離，$OthRate$ は，0 から 1 の値を取り，他の人名が出現した場合にどのくらいスコアを下げるかを決める値である．距離は対象人名の場合と同じように，単語数で表す．つまり，対象人名が “Barack Obama” であるとき，“Michelle Obama was born in January 17, 1964” という文章の，属性値候補 “January 17, 1964” と，他の人名である “Michelle Obama” との距離は 4 となり，このとき，減衰定数 Dec_{Oth} が 0.99，$OthRate$ が 0.05 であったとすると，$S_{OthDist}$ は約 0.952 となる．

$$S_{OthDist} = 1 - OthRate \times Dec_{Oth}^{Dist_{Oth}} \tag{4.3}$$

4.5.4 手掛かり表現の有無

手掛かり表現があれば，表現が無い属性値候補よりも高くなるようにスコアを設定する．この際，手掛かり表現が属性値候補の前にあるか，後ろにあるかも考慮する．また，手掛かり表現と属性値候補の距離が近ければ近いほど，スコアを高く設定し，遠くなるにつれて，スコアを低くする．手掛かり表現は一様な重要度ではなく，その重要度によって重みを付ける．例えば，“was born in” のあとには，“born” よりも “Date of birth” が来る可能性が高いため，前者が出現したほうがスコアを高く設定する．

表 4.9: 属性ごとの手掛かり表現

属性名	手掛かり表現の例
Affiliation	enter*, member of*, work for* ...
Award	win*, cop* ...
Birthplace	birthplace*, born*, native* ...
Date of birth	born in*, birth* ...
Degree	recieve*, get*, degree ...
Email	E-mail*, mail* ...
FAX	fax*
Major	degree, major in* ...
Mentor	work with*, coach, trainer ...
Nationality	nationality*
Occupation	position*, serve*, title* ...
Other name	as known as*, other name* ...
Phone	tel*, phone*, mobile* ...
Relatives	spouse, brother, sister, wife ...
School	school*, graduate ...
Web site	web*, url*, website ...

これらの要素を加味し，式 4.4 に従ってスコアを計算する．ここで，Dec_{Cue} は 0 から 1 の値を取る減衰定数（属性によって異なる），$Dist_{Cue}$ は属性値候補と手掛かり表現との距離，$CueScr$ は，0 から 1 の値を取り，手掛かり表現自体の重要度を示す値である．距離は対象人名の場合と同じように，単語数で表す．つまり，対象人名が “Barack Obama” であるとき，”Barack Obama was born in August 4, 1961” という文章の，属性値候補 “August 4, 1961” と，手掛かり表現 “was born in” との距離は 1 となり，このとき，減衰定数 Dec_{Cue} が 0.99，$CueScr$ が 0.9 であったとすると，S_{Cue} は 1.891 となる．計算式で 1 を足しているのは，手掛かり表現が無かったとき，$S_{Cue} = 1$ とし，手掛かり表現があることによってスコアが上がるようにするためである．つまり，手掛かり表現が無くてもスコアは上がらないが，手掛かり表現があるとスコアが上がるような仕組みである．

$$S_{Cue} = 1 + CueScr \times Dec_{Cue}^{Dist_{Cue}} \tag{4.4}$$

表 4.9 に属性ごとの手掛かり表現の例を示す．ここで，手掛かり表現に ‘*’ が付いているものは，手掛かり表現が属性値候補の前に置かれた場合のみ有効．無印のものは，前に置かれても，後ろに置かれても有効な手掛かり表現である．手掛かり表現は，実際に学習データの正解データを用いて，正しい属性値の前後に多く出現した表現を人

手で選別した．

4.6 Step2: 機械学習を用いた手法

前節で説明した，ヒューリスティックな手法では，各仮説に基づくスコアの重みや，手掛かり表現のリストなど，全ての要素を人手を用いて決定し，属性値の選別を行った．しかし，ヒューリスティックな手法は，人手を用いているため，正確性が高く，間違いが少ないものの，適切な重みや閾値を設定したり，手掛かり表現のリストなどを作るのには莫大なコストがかかる．今回のように，それほど大きくない規模での実験であれば大きな問題は起きないが，実際に属性抽出システムを運用しようと考えると，作成コストもさることながら，その維持にも大きなコストがかかる．そのため，出来るだけ人手による関与を減らし，自動的に属性値候補の選別が出来るようなシステムを考える必要がある．具体的には，機械学習を利用して属性値候補の選別を行う手法を提案する．

機械学習を利用して属性値の候補を選別するために，予め問題を整理し，正確に定義しておく必要がある．Step1 を終えた段階では，属性ごとに属性値の候補が複数存在する状態である．この属性値候補それぞれに対し，正しいか正しくないか，という 2 値分類を行うタスクを想定する．ここで 2 値分類を利用せず，属性をひとまとめにして扱い，属性値の候補を各属性ごとに振り分ける多値分類のタスクとして扱うことも可能であると考えられるが，提案手法では 2 値分類を用いる．機械学習を行うために考慮すべき要件は，以下の通りである．

1. 学習データの用意
2. 素性の選択
3. 学習モデルの選択
4. パラメータの設定

以下，それぞれの要素について説明する．ただし，属性ごとに分類器を作成する事を想定している．

4.6.1 学習データの用意

学習データとしては，人名とウェブドキュメントがあり，そのウェブドキュメント中のどの部分がそれぞれの属性の正解となる値か示されているデータが必要となる．幸いにも，WePS というワークショップが開催されており，学習データはそれを利用する事が可能である．

学習データの作成は，以下のように行う．まず，提供されているウェブドキュメントに対して Step1 による属性値候補の抽出を行うのと同時に，与えられている正解の

属性値の表現も候補に加える．次に得られた候補の中から，正解とされているものを正例とし，候補として選ばれてはいるが，正解とされていないものを負例とする．

4.6.2 素性の選択

機械学習の肝となる部分である．前述の 4 つの仮説に基づき，その仮説を表現できるような素性を選択する必要がある．ここでは，利用する素性を説明し，その素性を用いて作る特徴ベクトルの作り方について説明する．

属性値になり易い表現 属性値になり易い表現かどうかは，その属性値の表現を素性として入れることによって対応する事が出来る．このとき，単に属性値の表現が “August 4, 1976” であるという素性を入れても良いが，その場合同じ表現が出てこない限り学習の意味を成さなくなってしまうので，ある程度表現を抽象化する必要がある．例えば，ヒューリスティックな手法で用いた，表 4.7 のような正規表現で表されるパターンであるかどうかを素性とし，抽象化することで機械学習に利用する事が出来る．何れの場合であっても，特徴ベクトルの値はそれぞれの表現やパターンが出現したか出現しないかの 0 または 1 で表される．

対象及びその他の人名との距離 対象人名との距離はそのまま素性に入れることが可能である．それによって，人名との距離が近ければ近いほど正しい属性値になりやすいという仮説が正しいかどうか，学習することが出来る．また，その他の人名も同様であり，距離をそのまま素性に入れる．距離は単語数に基づく距離を用いる．このとき，特徴ベクトルの値を 0 から 1 で正規化するため，距離を 1000 で割り，距離が 1000 以上のものは，特徴ベクトルの値を 1 で統一している．

手掛かり表現の有無 手掛かり表現の有無を素性として表すには，前後の n グラムに注目すればよい．例えば，“was born in August 4, 1976” であれば，属性値の候補である “August 4, 1976” の前にあるトリグラムを取れば，“was born in” というパターンが抽出されることになる．これを「直前のトリグラムが “was born in” である.」という素性にすれば，手掛かり表現となるべきものが機械学習によってはっきりするはずである．本研究では，属性値候補の前後それぞれ 5 単語分のテキストを抽出し，前後それぞれに対して取りうる可能な，ユニグラム，バイグラム，トリグラムを素性とする．すなわち，前後 5 単語分から n グラム（n=1,2,3）を取って，素性とする．このとき，対象の人名のファーストネームは “[First Name]”，ラストネームは “[Last Name]” と置き換えることで，パターンの抽象化を行う．

上記で説明した，学習に利用する素性と特徴ベクトルでの値をを表 4.10 にまとめる．ここで，“attribute” は属性値のなりやすさを表現する抽象化されたパターン，“tar_dist” は対象人名との距離，“oth_dist” は他の人名との距離，“pre” は前置されるパターン，

表 4.10: 利用する素性とその例

対象人名	Barack Obama
属性値候補	August 4, 1961 (Date of birth)
テキスト	Barack Obama was born in August 4, 1961. His father, Barack Obama Sr., was born and raised in a small village in Kenya, where he grew up herding goats with his own father, who was a domestic servant to the British.

素性名	テキストから作られる素性
attribute	$/(MONTH\_EXP)[\backslash.,]?\backslash s*\backslash d\{1,2\}[\backslash.,]?\backslash s+\backslash d\{1,4\}/gi$ （表 4.7）
tar_dist	0.004 （“Barack Obama” との距離 / 1000）
oth_dist	0.003 （“Barack Obama Sr.,” との距離 / 1000）
pre_uni_1	in （“in” という表現が発火，以下同様）
pre_uni_2	born
pre_uni_3	was
pre_uni_4	[Last Name]
pre_uni_5	[First Name]
pre_bi_1	born in
pre_bi_2	was born
pre_bi_3	[Last Name] was
pre_bi_4	[First Name] [Last Name]
pre_tri_1	was born in
pre_tri_2	[Last Name] was born
pre_tri_3	[First Name] [Last Name] was
post_uni_1	His
post_uni_2	father,
post_uni_3	[First Name]
post_uni_4	[Last Name]
post_uni_5	Sr.,
post_bi_1	His father,
post_bi_2	father, [First Name]
post_bi_3	[First Name] [Last Name]
post_bi_4	[Last Name] Sr.,
post_tri_1	His father, [First Name]
post_tri_2	father, [First Name] [Last Name]
post_tri_3	[First Name] [Last Name] Sr.,

“post” は後置されるパターン，“uni” はユニグラム，“bi” はバイグラム，“tri” トリグラムを表す．また，n グラムの素性についている数字は，属性値候補からの距離を表している．つまり，“pre_bi_2” であれば，前方向に距離 2 離れたバイグラムを示しており，実際に特徴ベクトルを作る際には，“pre_bi_2” が “was born” である要素が発火したとして，その要素を 1 とおく．ベクトルの値は 0 から 1 である．

4.6.3 学習モデルの選択

機械学習によって 2 値分類を行うことの出来る学習モデルは Naive Bayes や Neural Network，Support Vector Machine[3]，C4.5 Decision Tree，k-Nearest Neighbors，など様々なモデルが考えられるが，本研究では，これらの手法の中でも分類精度が高く，汎化性能が高いと言われている Support Vector Machine(SVM)[9]を用いる．また，その際にカーネルはガウシアンカーネルを利用する．

4.6.4 パラメータの設定

SVM で学習を行う際，カーネルの設定よりも重要になると言われているのが，コストやガンマ値[10]，といったパラメータの設定である．一般的に SVM のパラメータの最適化は交差検定を用いて行われるが，2 値分類の SVM では分類精度をなるべく高めるように学習が行われる．本研究では，属性値候補が正しいか，正しくないかの判定に SVM を用いており，その精度が最大になったときに，必ずしも評価実験の F 値が最大になるわけではない．実際，トレーニングデータの一部を用いて SVM の学習を行い，残りのデータで評価実験を行った場合，適合率は高いが，再現率が低く，トータルの F 値は低い事が確認できた．

この問題を解決するため，本研究では，SVM として，ϵ -SVR を用いて，サポートベクトル回帰を行った．これは単純な 2 値分類ではなく，正解の場合を 1，不正解の場合を-1 として，回帰分析を行い，一つの属性値候補に対して-1 から 1 の実数値を返すものである．このとき，1 に近ければ近いほど，正解である確率が高く，-1 に近ければ近いほど不正解である確率が高い．ここで，0 を閾値としてそれ以上の値を返す物を取得すれば，単純な 2 値分類の SVM と同じになるが，本研究では，閾値を-1 から 1 まで動かし，トレーニングデータ上での F 値の推移をみることによって，F 値が最大になったときの閾値を利用する．これによって，テストデータにおいても最大の F 値が得られるように工夫する．

[9]一般的な SVM の実装ライブラリである libsvm を用いた．
[10]ガウシアンカーネルを用いる場合

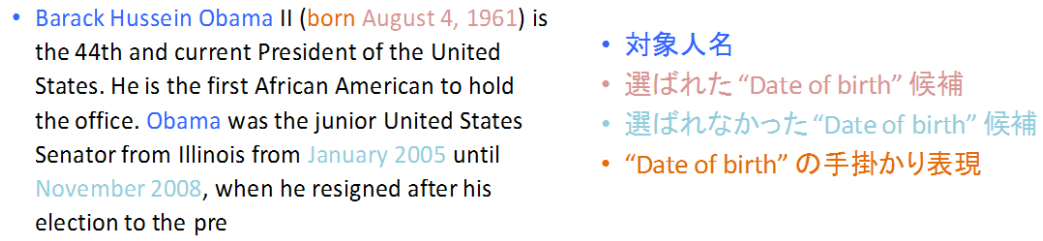

図 4.3: Step2 の処理例

4.7 Step2 のまとめ

Step2 では，上記のような手法を用いて，Step1 で得られた属性値の候補を選別する．具体的に Step2 を行った後のイメージが図 4.3 である．図 4.3 はある文章を入力とし，Step1 を行ったあとの，“Date of birth” という属性の抽出について示している．このとき，“Date of birth” として，“August 4, 1961” が選ばれているのは，対象人名に近く，手掛かり表現が近くに出現しているためである．

5 評価

この章では，提案手法の評価を行う．評価実験では，まず Step1 でどれだけの候補を集めることが出来たかを評価する．そして，集められた候補の中で，どれだけ精度よく Step2 において属性表現を抽出できたかどうかをヒューリスティックな手法，機械学習を用いた手法，それぞれに対して評価する．テストデータ，トレーニングデータはともに，Web People Search Evaluation Workshop（WePS）で提供されているものを用いた．以下，まず WePS について説明し，利用したデータ，抽出のルールについて詳しく述べる．その後，実験の設定，手順を示し，最後にその結果をまとめる．

5.1 Web People Search Evaluation Workshop

WePS ではこれまで，第 1 回が 2007 年 6 月，第 2 回が 2009 年 4 月に開かれており，2010 年 9 月には第 3 回が開催される．そのタスクは回毎に多少異なっているが，大きな目的はウェブ上に出現する人名の曖昧性を解消することである．すなわち，入力として人物の名前と，検索エンジンによって集められたドキュメントが与えられ，そのドキュメント集合を，同姓同名ではあるが異なる人物ごとにクラスタリングを行い，同姓同名の人物が何人居て，それぞれのドキュメントはどの人物に属するかを当てることが目的である．第 1 回は，人名の曖昧性解消タスクのみ，第 2 回は，人名の曖昧性解消タスクと人物の属性情報抽出のタスクが別々に行われ，第 3 回は人物の属性情報抽出と曖昧性解消のタスクが一つになり，さらにエンティティを人名に限らず，組織名を対象とした曖昧性解消タスクが行われる予定である．

本研究の評価実験では，トレーニングデータとして，WePS1 におけるデータを，テストデータとして，WePS2 におけるデータを利用した．また，評価に際しては，WePS2 の抽出ルールに従った．以下，そのデータの作成手法，抽出ルール，評価指標について詳しく述べる．

5.1.1 テストデータ

Step1 の評価実験，Step2 の評価実験はともに，WePS2 の評価データセット [34] を用いた．このデータセットでは，曖昧性を持つ 30 の人名（表 5.1）が用いられている．そのうち，10 人の名前は 1990 年のアメリカにおける人口調査[11]で得られた人名からファーストネームとラストネームをランダムに組み合わせたもの，またある 10 人の名前は 2008 年の “The Annual Meeting of the Association for Computational Linguistics”（ACL-08）の参加者[12]，残りの 10 人の名前は Wikipedia[13]からそれぞれランダムに選

[11] http://www.census.gov/main/www/cen1990.html
[12] http://www.ling.ohio-state.edu/acl08/
[13] http://en.wikipedia.org/wiki/

表 5.1: テストデータの人名

Amanda Lentz, Benjamin Snyder, Bertram Brooker, Cheng Niu, David Tua, David Weir, Emily Bender, Franz Masereel, Gideon Mann, Hao Zhang, Helen Thomas, Herb Ritts, Hui Fang, Ivan Titov, James Patterson, Janelle Lee, Jason Hart, Jonathan Shaw, Judith Schwartz, Louis Lowe, Mike Roberts, Mirella Lapa, Nicholas Maw, Otis Lee, Rita Fisher

ばれている．それぞれの名前は 2 つのトークン，つまりファーストネームとラストネームから構成されている．

それらの人名に対して Yahoo!の API によって検索を行い，それぞれ検索結果 Top150 件のページから，“.txt” と “.html” で表されるページ，合計 3468 ページを取得する．この中に評価に用いるのに適切でないと判断されたページが 585 ページあり，評価には残された 2883 ページを用いる [34]．以下に，適切では無いと判断される条件を示す．

- タグや，ヘッダを除く，HTML のテキスト内に検索に用いた名前が含まれていない．例えば，検索に用いた名前が “Barack Obama” である場合，“Barack H. Obama” のみしか出現しないようなページは対象外とする．
- 属性抽出の対象人名に対して，同姓同名な 2 人以上の人物に関する記述がある場合．具体的には，大学教授の “Jim Clark” と，レーシングドライバーの “Jim Clark” が同時に出現する場合などがある．
- DBLP や CiteSeer などのデータベースから得られたページや，amazon.com，Yahoo Shopping などのショッピングサイトのようなページ．
- ベースが英語で書かれていない．
- 対象人名が架空のキャラクターを示している場合．
- ページの内容が創作であった場合（対象人名の指す人物が実在の人物である場合も含む）．

作成されたデータセットのうち，2421 ページは少なくとも 1 つ以上の属性情報の記述が有り，残りの 462 ページは 1 つも属性情報の記述が無いページである．表 5.2 にテストデータにおける，属性ごとの出現頻度を示す．

5.1.2 トレーニングデータ

Step2 におけるトレーニングデータは WePS2 でトレーニング用に公開された，WePS1 のデータ [13] を用いた．このトレーニングセットは，基本的に前述のテストデータと同

表 5.2: テストデータにおける属性ごとの出現頻度

属性名	合計	1 ページ平均数	1 ページ最大数
Affiliation	3,105	1.03	19
Award	264	0.09	14
Birthplace	301	0.10	4
Date of birth	370	0.12	4
Degree	335	0.11	6
Email	209	0.07	5
FAX	65	0.02	2
Major	173	0.06	6
Mentor	343	0.11	12
Nationality	250	0.08	2
Occupation	3,292	1.10	20
Other name	797	0.27	6
Phone	219	0.07	5
Relatives	914	0.30	29
School	494	0.16	10
Web site	154	0.05	4

表 5.3: トレーニングデータの人名

Alexander Macomb, Allan Hanbury, Andrew Powell, Anita Coleman, Christine Borgman, David Lodge, Donna Harman, Edward Fox, George Clinton, Gregory Crane, Jane Hunter, John Kennedy, Michael Howard, Paul Clough, Paul Collins, Thomas Baker, Tony Abbott

様の手法で作られており，表 5.3 に表す 17 人の人物の名前が用いられている．そのうち，7 人の名前は Wikipedia から，残り 10 人の名前は 2006 年の “European Conference on Research and Advanced Technology for Digital Libraries”（ECDL-06）[14]のプログラム委員からランダムに選ばれている．それぞれの名前はテストデータと同様，2 つのトークン，つまりファーストネームとラストネームから構成されている．

それらの人名に対して Yahoo!の API によって検索を行い，それぞれ検索結果 Top100 件のページから，既に存在しないページを除き，合計 1683 ページを取得する．この中に前述したような，学習に用いるのに適切でないと判断されたページ，および 1 つも属性情報の記述が無いページが合わせて 708 ページあり，学習には残された 975 ペー

[14]http://www.ecdl2006.org/

表 5.4: トレーニングデータにおける属性ごとの出現頻度

属性名	合計	1 ページ平均数	1 ページ最大数
Affiliation	1398	1.43	24
Award	177	0.18	12
Birthplace	167	0.17	2
Date of birth	202	0.21	3
Degree	180	0.18	6
Email	90	0.09	2
FAX	35	0.04	2
Major	108	0.11	6
Mentor	26	0.03	4
Nationality	61	0.06	2
Occupation	1567	1.61	19
Other name	397	0.41	7
Phone	85	0.09	2
Relatives	583	0.60	19
School	286	0.29	12
Web site	45	0.05	2

ジを用いる [13]．表 5.4 にテストデータにおける，属性ごとの出現頻度を示す．

5.1.3 抽出ルール

ウェブを対象とした属性情報抽出のタスクでは，どの属性値を正解とするべきか，判断が難しいことが多い．WePS2 では，その問題を解決するため，抽出に際してのルールを規定している．その抽出ルールを簡単に説明する．属性ごとに定められた細かな抽出ルールは WePS2 のタスクガイドライン[15]に詳しい．ここでは，属性全体に共通する一般的な抽出ルールについて説明する．

一般的な抽出ルールは以下に示す通りである．

- 属性値はページに書かれている “そのまま” 抽出し，そのページ上に存在しない属性値は抽出しない．また，HTML のリンク先にあるページからの抽出も行わない．
- 一つの属性に対して，二つ以上の正しいと思われる属性値がある場合は，その全てを抽出する．例えば “Birthplace” の候補として，“Japan” と “Tokyo” が，“He

[15]http://nlp.uned.es/weps/weps2/WePS2_Attribute_Extraction.pdf

was born in Japan, in the city of Tokyo.” という文章で表れた場合，その両方を抽出する．ただし，その二つの候補が，“He was born in Tokyo, Japan” のように，一つのフレーズとして出現した場合には，一つの属性値 “Tokyo, Japan” として抽出する．

- 属性値の内容が例え事実と異なっていたとしても，抽出する．例えば，“Macomb, Alexander (1782-1841) General: Alexander Macomb was born on Detroit, Michigan, on June 25, 1841” という文章から “Date of birth” を抽出する場合，“1782” と，“June 25, 1841” の両方を抽出しなければならない．
- 内容としては同じ属性値が，一つのページ内で，異なった表現で二つ以上出現する場合，その全てを抽出する．例えば，“Date of birth” に対して，“June 25, 1841” と “1841/6/25” という候補が出現する場合，その両方を抽出する．
- 英語以外の言語で書かれた属性値は抽出しない．
- 一つの属性値が，カンマやピリオド，大文字小文字などの違いで複数表現されていた場合，複数抽出しても，一つだけ抽出しても，どちらでも良い．それらに対してペナルティやアドバンテージは一切与えない．
- 抽出した属性値にスペースや改行が複数含まれていたり，連続していても問題はない．評価の際に，連続したスペース，改行は一つのスペースに置き換えられる．
- “the” などの冠詞が属性値の頭に来る場合は，つけてもつけなくてもどちらでも良い．それらに対してペナルティやアドバンテージは一切与えない．
- アスキー文字で無いものは評価時に ‘?’ に置き換えるので，その処理の際にペナルティやアドバンテージが発生する事は無い．

本研究で行う評価実験は，以上のようなルールに沿って行う．

5.1.4 評価指標

評価指標は情報抽出で一般的に用いられている，適合率，再現率，F 値を用いる．その定義は以下の式 5.1，式 5.2，式 5.3 の通りである．

$$\text{適合率} = \frac{\text{選別した中の正しい属性値の数}}{\text{選別した属性値の数}} \tag{5.1}$$

$$\text{再現率} = \frac{\text{選別した中の正しい属性値の数}}{\text{正しい属性値の数}} \tag{5.2}$$

$$F\text{ 値} = \frac{2 \times \text{適合率} \times \text{再現率}}{\text{適合率} + \text{再現率}} \tag{5.3}$$

表 5.5: 交差検定のためのグループ分け

Group1	Edward Fox, Allan Hanbury, Alexander Macomb, John Kennedy
Group2	Donna Harman, Andrew Powell, David Lodge, George Clinton
Group3	Gregory Crane, Jane Hunter, Michael Howard
Group4	Paul Clough, Thomas Baker, Paul Collins
Group5	Christine Borgman, Anita Coleman, Tony Abbott

5.2 評価実験

評価実験は前述のように，Step1 に対する評価実験，Step2 の 2 つの手法に対する評価実験を行う．以下，3 種の実験の設定について述べる．評価は全て WePS2 の基準に沿って行う．

Step1 Step1 でどれだけ多くの正しい属性値を候補として見つけることが出来たかを評価する．この評価実験では前述のテストデータを用いて，Step1 による属性値候補の抽出のみを行い，Step2 による属性値候補の選別は行わない．すなわち，Step2 を終えた後に取得可能な最大の属性値の数を評価する．この実験は後述する再現率の上限を調べるためのものである．

ヒューリスティックな手法 提案手法のシステムによってどの程度正確に属性情報の抽出が行えるかどうか評価する．Step1 によって属性値の候補を抽出し，Step2 ではヒューリスティックに設定した手法を用いて選別を行う．

機械学習を用いた手法 提案手法のシステムによってどの程度正確に属性情報の抽出が行えるかどうか評価する．Step1 によって属性値の候補を抽出し，Step2 では機械学習を用いた手法を用いて選別を行う．パラメータの設定は，トレーニングデータを人名ごとに表 5.5 のような 5 グループに分け，4 グループのデータを用いて学習し，残りの 1 グループのデータで評価し，最も F 値が高くなるようなパラメータの設定を用いる．

5.3 結果

この節では評価実験の結果を示し，その結果について考察する．特に，機械学習を用いた手法に重点をおいて，属性ごと，人名ごとの適合率，再現率，F 値について評価する．

表 5.6: 属性ごとの結果（Step1 のみ）

Attribute	Match	OVG	Miss	Precision	Recall	F-measure
Affiliation	1439	58884	1648	0.024	0.466	0.045
Award	56	1113	208	0.003	0.304	0.007
Birthplace	91	27191	208	0.003	0.304	0.007
Date of birth	117	16719	252	0.007	0.317	0.014
Degree	84	794	251	0.096	0.251	0.138
Email	177	1685	32	0.095	0.847	0.171
FAX	31	1154	34	0.026	0.477	0.050
Major	95	8370	78	0.011	0.549	0.022
Mentor	261	118805	82	0.002	0.761	0.004
Nationality	214	10916	36	0.019	0.856	0.038
Occupation	1081	13622	2211	0.074	0.328	0.120
Other name	87	218	702	0.285	0.110	0.159
Phone	92	1093	127	0.078	0.420	0.131
Relatives	578	118488	335	0.005	0.633	0.010
School	211	2221	281	0.087	0.429	0.144
Web site	128	726	26	0.150	0.831	0.254
Total	4742	381999	6511	0.012	0.421	0.023

5.3.1 Step1 のみの評価

Step1 に対する評価実験に対しての，属性ごとのマッチ数（Match），超過して生成した数（OVG），取れなかった数（Miss），適合率（Precision），再現率（Recall），F 値（F-measure）を表 5.6 に示す．また，同様の結果を人名ごとに評価したものを表 5.7 に示す．これは，全く選別をしておらず，Step1 で候補が得られたものをそのまま抽出したシステムであるため，この評価よりも Match 数が上がることは無く，この再現率が，提案手法の限界の再現率である．

属性ごとの結果を詳細に見てみると，再現率が高い属性は “Nationality”，“Email”，“Web site”，“Mentor”，“Relatives” の順である．これらを見ると，“Nationality” は属性値となる候補自体が限られているため，“Email” と “Web site” は正規表現を利用することで候補を上手く抽出できるためであると考えれれる．“Mentor”，“Relatives” の再現率が高いのは，人名候補の抽出が上手くいっているためであるが，これは固有表現抽出器の精度が高く，人名を漏らさず抽出できているからだと考えられる．反対に，“Other name” や “Degree” の精度は低くなってしまっている．“Other name” は別名の抽出問題であり，難しいタスクである．提案手法では，単純なヒューリスティックなルールを用いて，本人の人名が一部含まれているような表現しか，属性値の候補とし

表 5.7: 人名ごとの結果（Step1 のみ）

Name	Match	OVG	Miss	Precision	Recall	F-measure
Benjamin Snyder	134	28320	176	0.005	0.432	0.009
Cheng Niu	16	7647	38	0.002	0.296	0.004
David Weir	333	7107	542	0.045	0.381	0.080
Emily Bender	110	11066	216	0.010	0.337	0.019
Gideon Mann	78	13094	169	0.006	0.316	0.012
Hao Zhang	142	6419	203	0.022	0.412	0.041
Hui Fang	60	5212	126	0.011	0.323	0.022
Ivan Titov	15	8793	51	0.002	0.227	0.003
Mirella Lapata	33	10340	47	0.003	0.412	0.006
Tamer Elsayed	43	9200	103	0.005	0.295	0.009
Amanda Lentz	142	41315	170	0.003	0.455	0.007
Helen Thomas	522	17015	500	0.030	0.511	0.056
Janelle Lee	55	28298	144	0.002	0.276	0.004
Jonathan Shaw	156	5437	249	0.028	0.385	0.052
Judith Schwartz	173	7622	236	0.022	0.423	0.042
Otis Lee	147	19565	183	0.007	0.445	0.015
Rita Fisher	200	3533	210	0.054	0.488	0.097
Sharon Cummings	88	4702	158	0.018	0.358	0.035
Susan Jones	215	3108	300	0.065	0.417	0.112
Theodore Smith	211	6299	329	0.032	0.391	0.060
Bertram Brooker	221	28343	215	0.008	0.507	0.015
David Tua	113	8511	242	0.013	0.318	0.025
Franz Masereel	75	18287	53	0.004	0.586	0.008
Herb Ritts	208	8266	262	0.025	0.443	0.047
James Patterson	198	7541	216	0.026	0.478	0.049
Jason Hart	331	6773	435	0.047	0.432	0.084
Louis Lowe	82	35440	188	0.002	0.304	0.005
Mike Robertson	265	5644	393	0.045	0.403	0.081
Nicholas Maw	199	12637	205	0.015	0.493	0.030
Tom Linton	177	6465	152	0.027	0.538	0.051

表 5.8: 属性ごとの結果（ヒューリスティック）

Attribute	Match	OVG	Miss	Precision	Recall	F-measure
Affiliation	709	14151	2378	0.048	0.230	0.079
Award	0	371	264	0.000	0.000	0.000
Birthplace	145	374	154	0.279	0.485	0.355
Date of birth	118	239	251	0.331	0.320	0.325
Degree	18	445	317	0.039	0.054	0.045
Email	132	66	77	0.667	0.632	0.649
FAX	0	1	65	0.000	0.000	0.000
Major	8	113	165	0.066	0.046	0.054
Mentor	1	31	342	0.031	0.003	0.005
Nationality	86	660	164	0.115	0.344	0.173
Occupation	260	7009	3032	0.036	0.079	0.049
Other name	37	2647	752	0.014	0.047	0.021
Phone	128	933	91	0.121	0.584	0.200
Relatives	7	989	906	0.007	0.008	0.007
School	74	312	418	0.192	0.150	0.169
Web site	20	723	134	0.027	0.130	0.045
Total	1743	29064	9510	0.057	0.155	0.083

ていないため，全く別の “Other name”，例えば “Hideki Matsui” に対する “godzilla’ のような別名は取得する事が出来ない[16]．これは，他の属性に比べても，候補が一定では無いため，特に難しいタスクであると言える．実際にこの別名抽出のみをターゲットとした研究も行われている [36, 37]．“Degree” の候補は基本的にはリストによるマッチングで対応できると考えられるが，表現のされ方が多様で，表記揺れが大きいため，再現率が低くなってしまっていると考えられる．この再現率を高めるためには，表記揺れを吸収するか，リスト拡張などを行う必要がある．詳細は後に議論する．

人名ごとの結果を詳細に見てみると，最も高い再現率は “Franz Masereel” で 0.586，最も低い再現率は “Ivan Titov” で 0.227 であった．他の人名はおおよそ 0.3 から 0.5 の範囲内であり，人名によって大きな差は見られなかった．これは，提案手法における Step1 が人名の属するドメインによらず，属性値の候補を取得する事が出来，手法が適用可能であることを示している．

[16] “Hideki Matsui” に対する “H. Matsui” などは取得可能.

表 5.9: 人名ごとの結果（ヒューリスティック）

Name	Match	OVG	Miss	Precision	Recall	F-measure
Benjamin Snyder	42	771	268	0.052	0.135	0.075
Cheng Niu	7	759	47	0.009	0.130	0.017
David Weir	132	1180	743	0.101	0.151	0.121
Emily Bender	42	1005	284	0.040	0.129	0.061
Gideon Mann	27	852	220	0.031	0.109	0.048
Hao Zhang	66	747	279	0.081	0.191	0.114
Hui Fang	31	518	155	0.056	0.167	0.084
Ivan Titov	8	725	58	0.011	0.121	0.020
Mirella Lapata	10	775	70	0.013	0.125	0.023
Tamer Elsayed	13	700	133	0.018	0.089	0.030
Amanda Lentz	33	1185	279	0.027	0.106	0.043
Helen Thomas	189	1456	833	0.115	0.185	0.142
Janelle Lee	21	859	178	0.024	0.106	0.039
Jonathan Shaw	70	911	335	0.071	0.173	0.101
Judith Schwartz	62	1083	347	0.054	0.152	0.080
Otis Lee	26	495	304	0.050	0.079	0.061
Rita Fisher	100	859	310	0.104	0.244	0.146
Sharon Cummings	35	855	211	0.039	0.142	0.062
Susan Jones	83	711	432	0.105	0.161	0.127
Theodore Smith	65	828	475	0.073	0.120	0.091
Bertram Brooker	56	1247	380	0.043	0.128	0.064
David Tua	28	1474	327	0.019	0.079	0.030
Franz Masereel	28	1086	100	0.025	0.219	0.045
Herb Ritts	64	1137	406	0.053	0.136	0.077
James Patterson	53	1279	361	0.040	0.128	0.061
Jason Hart	174	1015	592	0.146	0.227	0.178
Louis Lowe	44	981	226	0.043	0.163	0.068
Mike Robertson	105	1052	553	0.091	0.160	0.116
Nicholas Maw	58	1419	346	0.039	0.144	0.062
Tom Linton	71	1100	258	0.061	0.216	0.095

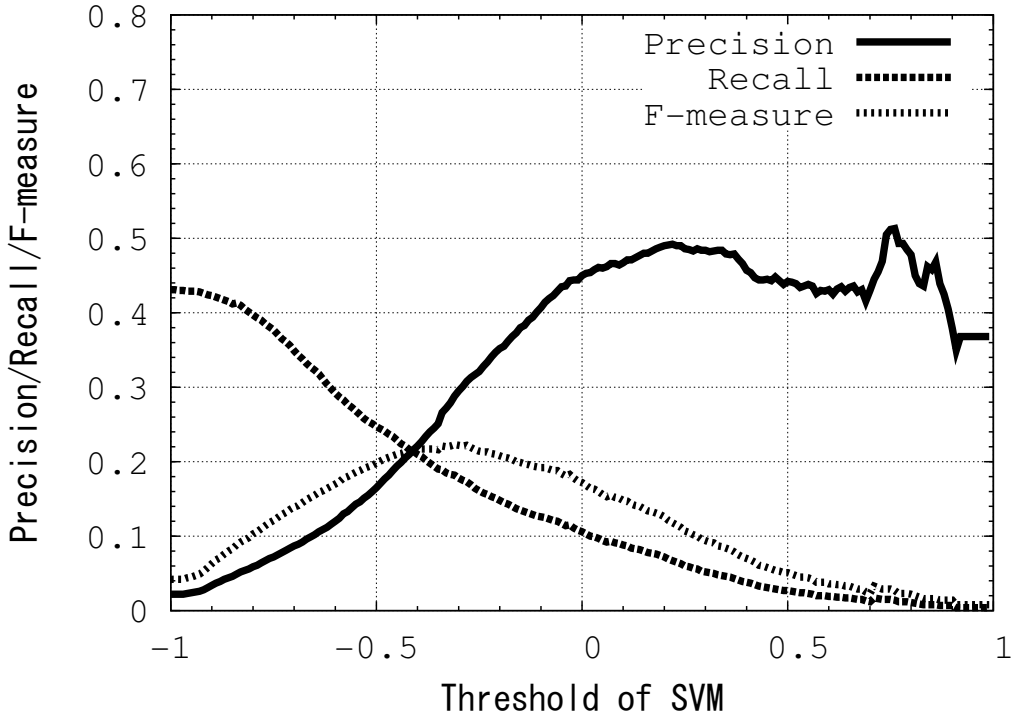

図 5.1: SVM の閾値に対する F 値の変化

5.3.2 ヒューリスティックな手法の評価

ヒューリスティックな手法を用いたシステムの，属性ごとのマッチ数（Match），超過して生成した数（OVG），取れなかった数（Miss），適合率（Precision），再現率（Recall），F 値（F-measure）を表 5.8 に示す．また，同様の結果を人名ごとに評価したものを表 5.9 に示す．

F 値が高くなっているのは，“Email”，“Birthplace”，“Nationality”，“Date of birth” の順である．逆に，“Award”，“Degree”，“FAX”，“Major”，“Mentor”，“Relatives”，などのように，そもそもマッチ数が無いかあるいは殆ど無い属性も存在する．これは，ヒューリスティックな手法においては，各属性に対して，適切な手掛かりとなる表現を設定し，対象人名との距離や他の人名との距離などに対する重みづけを適切に行わなければならないが，それには大きなコストがかかり，十分な設定が行えていないためと考えられる．“Email” や “Date of birth” などは，比較的手掛かり表現が分かり易く，決まったコンテキストで出現する事が多いため，F 値が大きくなったのでは無いかと考えられる．

人名ごとの結果を見ると，“Jason Hart”，“Rita Fisher”，“Helen Thomas” に関しては，比較的 F 値が高くなっているが，他の人名に関しては高くない値となっており，

表 5.10: 属性ごとの結果（機械学習）

Attribute	Match	OVG	Miss	Precision	Recall	F-measure
Affiliation	431	1343	2656	0.243	0.140	0.177
Award	7	58	257	0.108	0.027	0.043
Birthplace	45	283	254	0.137	0.151	0.144
Date of birth	64	319	305	0.167	0.173	0.170
Degree	53	181	282	0.226	0.158	0.186
Email	26	7	183	0.788	0.124	0.215
FAX	24	25	41	0.490	0.369	0.421
Major	50	226	123	0.181	0.289	0.223
Mentor	0	2	343	0.000	0.000	0.000
Nationality	17	3	233	0.850	0.068	0.126
Occupation	905	2968	2387	0.234	0.275	0.253
Other name	86	143	703	0.376	0.109	0.169
Phone	45	87	174	0.341	0.205	0.256
Relatives	114	1124	799	0.092	0.125	0.106
School	133	228	359	0.368	0.270	0.312
Web site	5	3	149	0.625	0.032	0.062
Total	2005	7000	9248	0.223	0.178	0.198

3 者を除くと，個々の人名ごとで大きな差は見らない.

5.3.3 機械学習を用いた手法の評価

トレーニングデータに対して，サポートベクトル回帰を用いた際の閾値を-1 から 1 まで変化させ，その際の適合率，再現率，F 値の推移を図 5.1 に示す．その結果，閾値が-0.31 のときに，F 値が最大の 0.223（適合率は 0.288，再現率は 0.182）となった．以下は，サポートベクトル回帰の閾値を-0.31 としたときの結果である．

テストデータに対して，機械学習を用いたシステムの，属性ごとのマッチ数（Match），超過して生成した数（OVG），取れなかった数（Miss），適合率（Precision），再現率（Recall），F 値（F-measure）を表 5.10 に示す．また，同様の結果を人名ごとに評価したものを表 5.11 に示す．それぞれの属性の適合率，再現率，F 値について，Step1 を行った状態からどのように変化したかを表すグラフを図 5.2，図 5.3，図 5.4 に示す．

属性ごとの結果を詳細にみると，F 値が高い属性は順に，“FAX”，“School”，“Phone”，“Occupation”，“Major” である．このうち，“FAX”，“School”，“Occupation” は Step1 で集めた候補から選別を行う際に，殆ど再現率を下げることなく，適合率が上がって

表 5.11: 人名ごとの結果（機械学習）

Name	Match	OVG	Miss	Precision	Recall	F-measure
Benjamin Snyder	43	243	267	0.150	0.139	0.144
Cheng Niu	4	45	50	0.082	0.074	0.078
David Weir	115	240	760	0.324	0.131	0.187
Emily Bender	48	251	278	0.161	0.147	0.154
Gideon Mann	48	162	199	0.229	0.194	0.210
Hao Zhang	50	122	295	0.291	0.145	0.193
Hui Fang	26	80	160	0.245	0.140	0.178
Ivan Titov	5	245	61	0.020	0.076	0.032
Mirella Lapata	25	41	55	0.379	0.312	0.342
Tamer Elsayed	12	72	134	0.143	0.082	0.104
Amanda Lentz	18	301	294	0.056	0.058	0.057
Helen Thomas	280	677	742	0.293	0.274	0.283
Janelle Lee	11	328	188	0.032	0.055	0.041
Jonathan Shaw	77	175	328	0.306	0.190	0.234
Judith Schwartz	78	299	331	0.207	0.191	0.198
Otis Lee	37	321	293	0.103	0.112	0.108
Rita Fisher	78	160	332	0.328	0.190	0.241
Sharon Cummings	31	135	215	0.187	0.126	0.150
Susan Jones	87	138	428	0.387	0.169	0.235
Theodore Smith	93	286	447	0.245	0.172	0.202
Bertram Brooker	112	194	324	0.366	0.257	0.302
David Tua	13	187	342	0.065	0.037	0.047
Franz Masereel	47	227	81	0.172	0.367	0.234
Herb Ritts	123	323	347	0.276	0.262	0.269
James Patterson	123	305	291	0.287	0.297	0.292
Jason Hart	119	204	647	0.368	0.155	0.219
Louis Lowe	43	557	227	0.072	0.159	0.099
Mike Robertson	91	194	567	0.319	0.138	0.193
Nicholas Maw	84	296	320	0.221	0.208	0.214
Tom Linton	84	192	245	0.304	0.255	0.278

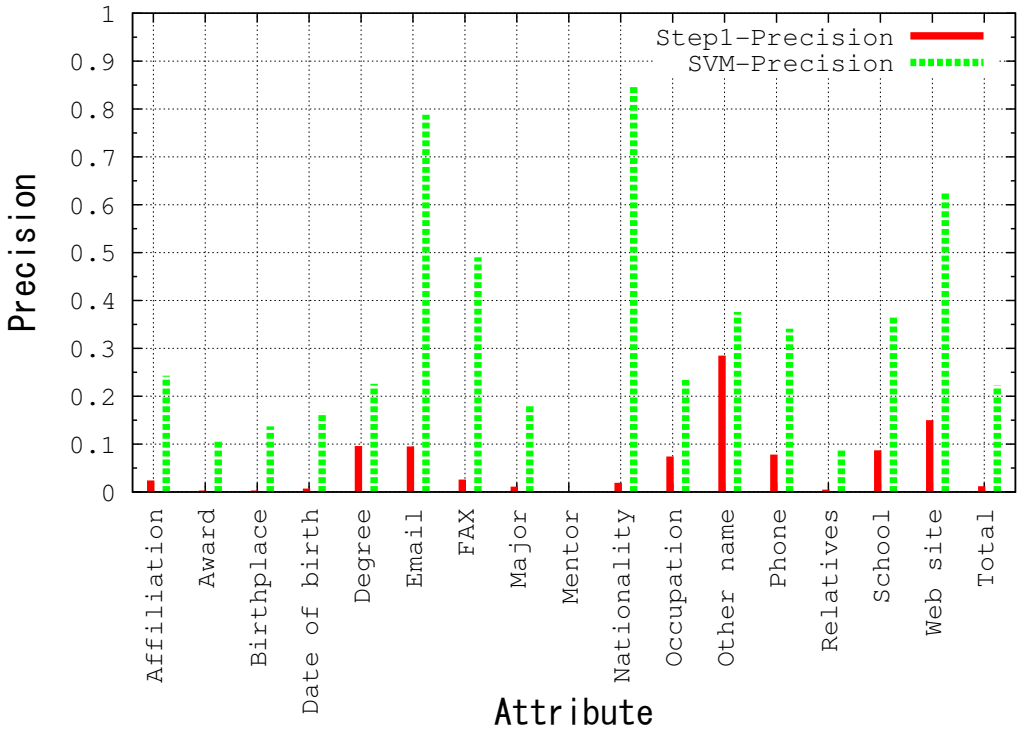

図 5.2: 属性ごとの適合率の比較（Step1 vs SVM）

いる．また，“Phone”，“Major” では Step1 に比べると，再現率は半分ほどに下がってしまっているが，適合率が大幅に上がっているため，結果的に F 値が高くなっている．

一方で，“Mentor”，“Award”，“Web site”，“Relatives”，“Nationality” は F 値が低くなっている．特に，“Mentor” と “Web site” に関しては，Step1 のみを行った際よりも F 値が低くなってしまっている．“Mentor” の F 値が低くなっているのは，学習器によって正しいと判断された属性値が 2 つ[17] しか無かったためである．これは，学習データにおいて，負例の数が多く，殆どの候補を負例と判断するような学習器になってしまったためだと考えられる．この問題を解決するには，Step1 の手法に対して，学習に入る前に，もう少し適合率を上げるような工夫が必要である．“Web site” に関しては，Step1 である程度精度よくマーク出来る一方で，その出現場所や前後のパターンにはあまり統一性が無く，人名との距離や，n グラムなどの素性が上手く機能しなかったためであると考えられる．将来的には，属性ごとに異なる素性を用いるような手法を利用する事も考えられる．また，“Web site” と同様に，“Award”，“Relatives” の F 値が低くなっているのも，出現場所や前後のパターンに統一性が無いためと考えられる．“Nationality” の F 値が低くなっているのは，殆どの候補を不正解と学習器が

[17] しかもそれらは正解で無い．

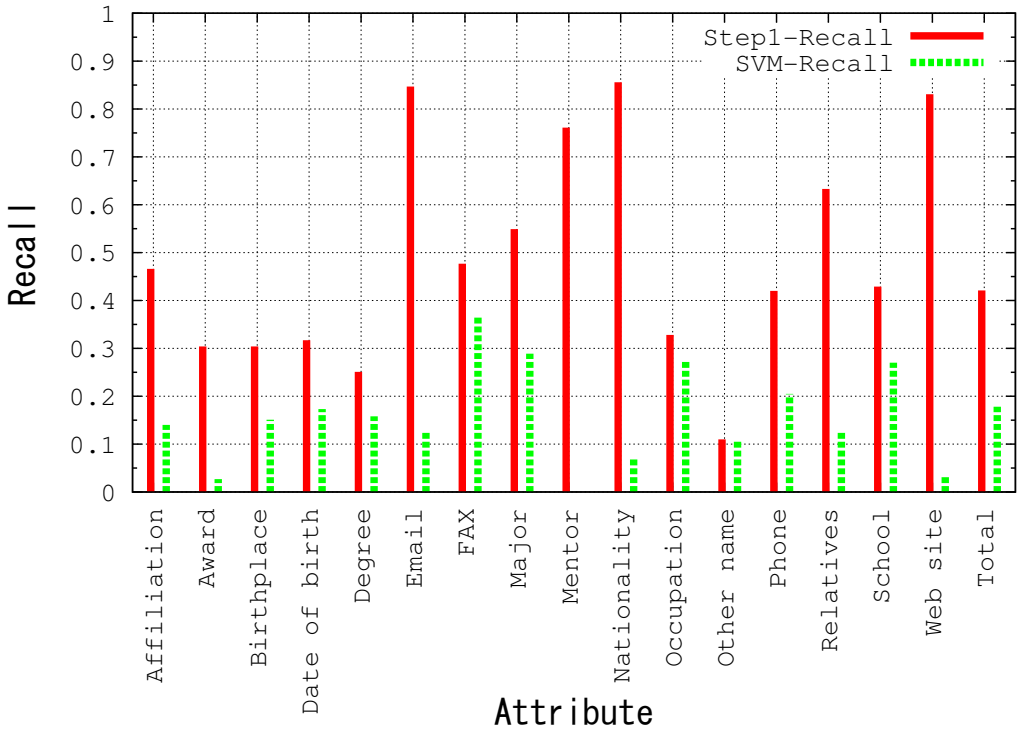

図 5.3: 属性ごとの再現率の比較（Step1 vs SVM）

判断しているためである．適合率自体は低くないので，サポートベクトル回帰の閾値を下げると，F 値が上がる可能性がある．全体としては学習した-0.31 という閾値が最適なので，属性別に閾値を設定することで，“Nationality” の F 値を上げられる可能性があるが，同時に，学習データがそれほど多くない属性は過学習されてしまう危険もある．

人名ごとの F 値を詳細に見ると，“Janelle Lee”，“David Tua”，“Cheng Niu” の 3 人の F 値が 0.1 を切っており，低い値である．“Janelle Lee” に関しては，“Affiliation”，“Relatives” について，大きく再現率を下げてしまったこと，“David Tua” に関しては，“Mentor” において，選別の際に再現率が大きく下がってしまったこと，“Cheng Niu” に関しては，もともと再現率が低かったことと，“Affiliation” の再現率が下がってしまったことが原因であると考えられる．

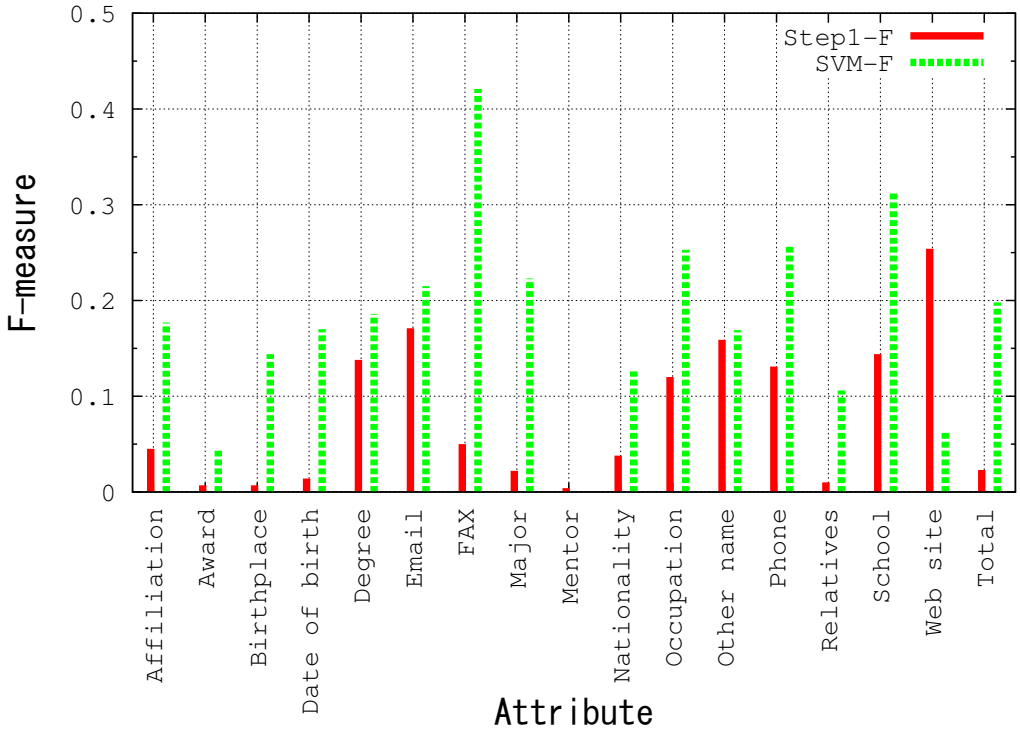

図 5.4: 属性ごとの F 値の比較（Step1 vs SVM）

5.4 全体的な評価

提案手法のヒューリスティックな手法は WePS2 に投稿されたものである．実際に WePS2 に参加した 15 システムの適合率，再現率，F 値を表 5.12[18]に示す．提案手法のトータルの F 値は 0.083 と低い値であったが，評価に参加した 15 システムの中で 5 番目の成績であった．F 値が最も高いシステム [44] で 0.122，適合率が最も高いシステム [44] で 0.304，再現率が最も高いシステム [17] で 27.4 であるため，属性抽出というタスクは非常に難しいタスクであると言える．また，用いられた手法ら [7, 17, 22, 40, 44] に大きな部分での差は無く，ヒューリスティックに抽出ルールなどを設定した部分で，精度に差がついたと考えられる．

その結果を踏まえ，機械学習による手法を提案し，表 5.10 に示したように，適合率が 0.223，再現率が 0.178，F 値が 0.198 と，既存研究で最も良かった F 値 0.122 よりも 0.076 ポイント，62.3%の向上が見られた．

[18] この表は，WePS2 の結果として公開されたものであり，適合率，再現率，F 値は最大を 100 としたパーセンテージで表されている．

表 5.12: WePS 全体の結果

Rank	System	Precision	Recall	F-measure
1	PolyUHK	30.4	7.6	12.2
2	CASIANED	8.5	19.0	11.7
3	ECNU_2	8.0	17.6	11.0
4	ECNU_1	6.8	18.8	10.0
5	MIVTU	5.7	15.5	8.3
6	UvA_2	4.4	27.4	7.6
7	UvA_1	2.7	27.3	5.0
8	UC3M_5	8.0	3.6	5.0
9	UvA_5	3.3	2.8	3.1
10	UC3M_1	2.5	2.2	2.3
11	UC3M_2	2.4	2.2	2.3
12	UC3M_3	2.2	2.0	2.1
13	UC3M_4	2.2	2.0	2.1
14	UvA_3	0.7	0.2	0.2
15	UvA_5	0.2	0.0	0.0

6　議論

この章では，本研究について議論を行う．まず，精度向上のための提案手法の改良について述べる．その後，本研究の将来の展望について述べる．

6.1　Step1 の改良

本研究の Step1 は，属性値の候補を抽出するタスクである．このタスクは当然，正しいと思われる属性値の候補全てにマークを行い，再現率を出来るだけ高めるべきであるが，本当に一から全てをマークしてしまうと，Step2 での選別も上手くいかなくなってしまう．そのため，適合率をある程度に保ちつつ，再現率を出来るだけ上げるような工夫が必要となる．そのような目的のもと，以下，Step1 の改良手法について議論する．

6.1.1　属性値候補の拡張

本研究の Step1 では，リストを用いたマッチング，特定の語を含む表現，固有表現抽出（NER），正規表現などを用いて抽出を行った．これらは固有表現抽出を除いて，全てヒューリスティックに作成したものであり，それを作ったり，維持をするためには大きなコストがかかる．そのため，その作業を自動的に行えないかどうか検討する必要がある．

リストを用いたマッチングでは，Wikipedia を利用してリストを作っているが，このリストの作成は，あるクラスに属するインスタンスの拡張問題を応用できる．すなわち，会社名のリストを作ろうと思えば，予め “Google” や “Microsoft” などのシードとなるインスタンスを用意しておき，ウェブ上などからパターンを抽出し，そのパターンを元に “Apple” などの新たなインスタンスを獲得する手法である．インスタンスの拡張の研究には Wang ら [28] によるものなどがある．

特定の語を含む表現を用いた手法は，現在 “Award” でのみ利用しているが，他の属性にも応用が可能である．例えば会社名であれば，“Corp.” や “Inc.” などの表現が含まれる可能性が高い．これらは例えば，前述のような手法でリストを拡張し，拡張されたリストの中で出現頻度の高い単語やパターン，或いはその抽象化されたものなりを抽出することでルールを自動的に作ることが出来る．このとき，属性値候補に含まれる表現から，その属性値候補がどの程度属性リストにふさわしいか（会社名としてふさわしいか），重み（信頼度）を付けることが出来ると考えられる．上手く属性値候補に重みをつけることが出来れば，Step2 の属性値の選別にも活かせると考えられる．

固有表現抽出では，現在既存の固有表現抽出器を用いて，一部の固有表現の抽出しか行っていないが，そもそも固有表現抽出器は正解例を用いた CRF などの機械学習によって作られており，他の固有表現であっても正解例を学習することでその識別が可能である．これは，リストの中のシードを利用して，そのインスタンスが出現する

ドキュメントを集め，それを学習データとすることで，実現できると考えられる．前述のリストを用いたマッチングや．特定の語を含む表現を用いた手法についても，学習によってその要素を取り入れることが出来ると考えられる．

6.2 Step2 の改良

本研究における Step2 は，Step1 においてマークされた候補を選別するタスクである．このタスクでは基本的には正しいものを選別し，適合率を上げることが大きな目標となるが，その際に出来るだけ再現率を下げないようにすることが肝要である．以下，Step2 の改良手法について議論する．

6.2.1 属性値になる確率

節 6.1.1 において，属性値候補の重みについて述べたが，それは，属性値候補の表現によるものである．属性値候補の表現だけでなく，属性値候補自体をその属性値になり易いかなり難いか評価出来ると考えられる．例えば，ウェブ上で “Google” や “Amazon” という語が出現したとき，それが勤めている会社の文脈で使われることは非常に少ない．一般的な場合では，それは検索結果の画面や，ショッピングサイトの画面である．つまり，“Google” や “Amazon” といったインスタンス自体が，“Affiliation” という属性の候補になる確率が低いということである．逆に “The University of Tokyo” といったインスタンスは出現すれば，それが “Affiliation” の候補である可能性が高い．このように出現頻度によって，属性値の候補になる確率が異なると考えられるため，そのような確率モデルを利用することで，属性値候補の重みを正しく設定できると考えられる．

具体的には，学習データの中で，“Google” という語が出現した回数で，“Google” が実際に “Affiliation” の意味で使われた回数を割ることで，その確率を推定することが出来ると考えられる．しかし，そのためには大量の学習データが必要となってしまう．その問題を回避するためには，検索エンジンを用いて “Google” と “Affiliation” の共起ヒット件数を “Google” のヒット件数で割ることで，近似するなどの手法が考えられる．

6.2.2 学習データの自動生成

本研究では，ヒューリスティックな手法，機械学習を用いた手法の二つを提案したが，緻密にやればヒューリスティックな手法も良い精度になることが期待できるが，ルールの作成やその維持など限界があることが想定できる．そのため，実際の実験結果のように機械学習による手法が高い精度を得られることが多い．機械学習では学習データの量が重要だが，本研究のように，16 もの正解の属性を人手でアノテーションするのは非常に大変でコストがかかる．これを自動的に生成できるようになれば，より精度

の高い学習が可能となる．それだけでなく，学習データの数が少なかったために，データがスパースで素性に入れることが出来なかったものも，学習データの量を増やすことが出来れば，その素性も学習に組み込むことが出来るようになる．

具体的には，Wikipedia の Infobox[19]などを用いて，人名とその属性情報の正解データを作成し，その人名と正解データをクエリとして検索エンジンに投げることで，その二つが出現するドキュメント集合を得ることが出来る．そのドキュメント集合の中から，検索に用いた属性情報の正解データをマークすることで，学習データを大量に収集する事が出来ると考えられる．

6.2.3 利用出来る可能性のある素性

前述のような手法で，大量の学習データが収集できたとすると，それまでは使うことの出来なかった素性が利用できる可能性がある．例えば，HTML のタグ情報などがそれにあたり，学習データが少ないときにはタグ構造が同一のものは少ないため，学習に用いることは難しいが，学習データを増やすことによって，Ravi の研究 [32] のように，ドメインと HTML タグを組み合わせることによって学習に用いることが出来ると考えられる．他にも，今回の実験では試していないが，属性値候補の単語数や文字数，形態素情報など利用できる可能性があるものは多いので，今後の研究の対象としたい．

6.2.4 パターンの抽象化

今回，機械学習を行う際の n グラムの素性として，パターンの対象人名部分を [First Name] や [Last Name] などで置き換え，その抽象化を行ったが，もっと大きな視点でパターンの抽象化を行う手法が考えられる．パターンの抽象化を行うことで，特定の表現に偏った学習を防ぐことが出来，Step2 における選別精度の向上に繋げることが出来る．最も単純には，数字部分を “\d” に置き換えたり，過去形や複数形を原形にする，形態素を用いるなどが考えられる．ただし，あまりに抽象化してしまうと，特徴的なパターンが抽出できなくなる可能性もあるので，そのバランスを取ることが重要である．

6.3 将来の展望

本研究の大きな目的は，人名を入力として入れると，属性情報を表示してくれるようなシステムを作ることである．今回は，人名だけでなくテキストが与えられたものとして，属性情報の抽出を行ったが，将来的にはテキストを抽出するシステムを組み合わせることで，人名から自動的に属性情報を抽出するようなシステムの構築を目指

[19]Wikipedia 上の右上部分にあり，テンプレートを用いて，テーブル構造上に属性情報などが表示されている．

していきたい．その際には，どのようなテキストが属性情報を含む，プロフィールのようなテキスト判断する事が肝要となる．

そのようなシステムを作る際に，工夫する事が出来るポイントとして，人名の曖昧性解消と組み合わせること，属性ごとの関連性を用いること，が考えられる．以下，その 2 点について述べる．

6.3.1 人名の曖昧性解消との組み合わせ

提案手法のような，テキストのみに依存して属性情報を抽出するシステムが出来れば，その属性情報を元に，より正確な人名の曖昧性解消が可能である．そうすると，今度は同じ人物を指している複数のドキュメントを用いて，その頻度情報などを利用することで，提案手法による属性情報の抽出精度を高めることが出来る．このようなブートストラップ的な手法を利用することで，二つのタスクでより高い精度を得ることが出来ると考えられる．

6.3.2 属性ごとの関連性

現段階では，属性ごとの関連性を一切考慮していない．「職業が大学教授であれば，賞としてオリンピックの金メダルをもらう可能性は低い」であるとか，「生まれた場所が東京であれば，国籍は日本の可能性が高い」といった，人間が知識として常識的に持っている，属性ごとの関連性を利用することで，精度が挙げられる可能性がある．それには，二つの属性が同じ人物のものである（曖昧性解消が行われている）必要があるが，前述のように人名の曖昧性解消が行われている環境では，このような属性ごとの関連性が利用できると考えられる．

7　結論

本論文では，ウェブ上の文書から人物の属性情報を抽出する手法を提案した．提案手法は二つの Step から成り，Step1 ではリストや NER，正規表現などを用いて属性値となりうる候補にマークをし，Step2 ではその候補の中から対象人名との距離，手がかり表現の有無などを元に適切な属性値を選別するヒューリスティックな手法と，対象人名との距離や，n グラムなどを素性として SVM で学習することで適切な属性値を選別する機械学習を用いた手法の二つを提案した．提案手法はヒューリスティックな手法では，全体の F 値が 0.083 であり，WePS2 参加 15 システムの中で 5 番目の成績であったが，機械学習を用いた手法では，参加 15 システムのうちトップのシステムによる F 値に対して，62.3%向上させる事が出来た．しかし，機械学習を用いた手法においても F 値は 0.198 であり，まだまだ高い値とは言えないため，今後もより精度向上のための工夫が必要である．

このように属性情報の抽出は難しいタスクであるが，人名の曖昧性解消や，別名問題，質問応答，クエリ拡張など様々なタスクにも応用可能であり，非常に挑戦しがいのあるタスクである．今後もより精度を高めると共に，議論で述べたような，人名を入力しただけで，その属性情報が抽出可能なシステムを構築出来るように，今後も研究を行っていきたい．

謝辞

本研究を進めるにあたって，多くの方々の多大なご協力を頂きました．指導教官の石塚満教授には，研究の方向性に対する指導だけでなく，進捗に対しても気を配って頂き，たいへん有益な助言や支援をいただきました．深く感謝の意を申し上げます．

石塚研の秘書である藤田さんには，学会やチューターの際の事務手続きだけでなく，研究室生活が快適に過ごせるよう気を配っていただきました．また，助教である土肥さんには，ネットワークや研究環境の整備から石塚研での生活において，様々な場面で助けて頂きました．お二人のおかげがあったからこそ，充実した研究室生活が送れたと思っております．

松尾さん，石田さん，友部さん，岡崎さん，森さん，中山さん，岡さん，ダヌシカさん，榊さんを初めとした石塚研 OB の方々や松尾ぐみの皆さまには，色々と助言や支援を頂き，本当にお世話になりました．松尾さんには，研究内容の指導にとどまらず，研究とはどのように進めていくべきか，論文の書き方など，非常に色々なことを学ばせて頂きました．岡崎さんには毎回の松尾ぐみで，的確な助言を頂くだけでなく，修士論文の進捗も気にかけてくださり大変お世話になりました．ダヌシカさんには，研究に行き詰ったときに相談に乗っていただき，色々なアイデアを提案して頂きました．石塚研 OB の方々や松尾ぐみの皆さまのおかげで，研究分野についての理解を深めることができ，そのディスカッションから多くのことを学ばせて頂きました．深く感謝致します．

去年石塚研を卒業された金さん，古川さん，大根さん，また研究室は異なりますが，同じく去年修士を卒業された広畑さんには，自分が卒論生の頃から身近な相談相手になって頂いたり，仲良くさせて頂き，非常に楽しい研究室生活を過ごすことが出来ました．研究室の先輩として一番身近な存在であり，様々なことを学び，また教えて頂きました，ここに感謝の意を申し上げます．

石塚研のミーティングや，研究室生活などで，博士課程の皆さまには本当にお世話になりました．顔さんには研究のお話を伺うだけでなく，初の海外での学会でスペインに行った時には，英語の拙い自分の代わりにホテルなどで色々と話して頂く機会もあり，本当にお世話になりました．李さんには研究室生活を送る上で欠かせない留学生とのコミュニケーションを手伝っていただいたり，研究のディスカッションをさせて頂きました．ドゥクさんには研究についてディスカッションするだけでなく，パソコンやネットワークに関して，色々と教えてもらい大変お世話になりました．石塚研のネットワークが繋がらなければ，研究自体が立ち行かなくなるので，大変感謝しております．チューターを務めさせて頂いたマムドゥーヘさんとは，研究の話や身の回りの話を英語で話すことが殆どで，拙いながらも若干の英語力を身につけることが出来ました．お世話になった博士課程の皆さまに，深く感謝致します．

修士から石塚研に入ってきた，伊藤くん，後藤くん，竹澤くん，谷くん，アンドレくん，昨年度から持ちあがりの井口くんら，修士 1 年の皆には研究のディスカッションから，雑談相手，一緒にスポーツをしたり，旅行に行ったりと色々とお世話になり

ました．修士１年の皆は研究に対する姿勢も素晴らしく，非常に良い刺激を受けました．それだけでなく，研究に関しても本来教える立場にあるはずが，逆に学ばせてもらえることも多く，研究室生活を送る上で修士１年の皆の存在は非常に大きいものでした．ここに深く感謝致します．

今年度から研究室に入ってきた卒論生の，田中くん，白石くんは研究の意識も高く，非常にまじめで，良い刺激を受けました．白石くんは自分の研究を引き継いで，取り組んでいてくれたこともあって，ディスカッションをする中で，貴重なアイデアや，意見に触れることが出来ました．深く感謝致します．

最後に同期である，菅原くん，滝さん，タコアさんには，研究に関しては，お互いに刺激しあうことの出来るライバルであり，研究外では非常に仲の良い友達として付き合う事が出来，充実した研究室生活を送ることが出来ました．

石塚研や松尾ぐみを中心として，非常に優秀でやる気に溢れる仲間に恵まれ，共に研究室生活を送ることが出来たことに深く感謝し，またその他，支えて頂いた全ての方々に感謝して，謝辞に代えさせて頂きたく思います．

2010 年 2 月 9 日

渡 部　啓 吾

発表文献

- 渡部啓吾, Danushka Bollegala, 松尾豊, 石塚満. 検索エンジンを用いた関連語の自動抽出. フレッシュマンのための人工知能研究交流会, 2008
- 渡部啓吾, Danushka Bollegala, 松尾豊, 石塚満. 検索エンジンを用いた関連語の自動抽出. 第 23 回人工知能学会全国大会 (JSAI-08), 3B1-4, 2008.
- K. Watanabe, D. Bollegala, Y. Matsuo, and M. Ishizuka. A Two-Step Approach to Extracting Attributes for People on the Web. In *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS-09) at the 18th International World Wide Web Conference*, 2009.
- 渡部啓吾, Danushka Bollegala, 松尾豊, 石塚満. Web からの人物の属性情報抽出. 第 23 回人工知能学会全国大会 (JSAI-09), 3B2-2, 2009.
- 渡部啓吾, Danushka Bollegala, 松尾豊, 石塚満. Web からの人物の属性情報抽出. 情報処理学会創立 50 周年記念（第 72 回）全国大会 (IPSJ-10), 2010. （発表予定）

参考文献

[1] A. Bagga and B. Baldwin. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In *Proceedings of the 17th International Conference on Computational Linguistics (COLING-98)*, pages 79-85, 1998.

[2] A. Dingli, F. Ciravegna, D. Guthrie, and Y. Wilks. Mining Web Sites Using Unsupervised Adaptive Information Extraction. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*, 2003.

[3] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT-92)*, pages 144-152, 1992.

[4] B. Malin. Unsupervised Name Disambiguation via Social Network Similarity. In *Proceedings of the Workshop on Link Analysis, Counterterrorism, and Security, in conjuction with the SIAM International Conference on Data Mining (SIAM/ICDM-05)*, pages 93-102, 2005.

[5] C. Gooi and J. Allan. Cross-Document Coreference on a Large Scale Corpus. In *Proceedings of the Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics (HLT/NAACL-04)*, 2004.

[6] C. Niu, W. Li, and R.Srihari. Weakly Supervised Learning for Cross-document Person Name Disambiguation Supported by Information Extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL-04)*, 2004.

[7] C. Sanchez and P. Martinez. UC3M at WePS2-AE: Acquiring Patterns for People Attribute Extraction from Webpages. In *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS-09) at the 18th International World Wide Web Conference*, 2009.

[8] D. Bollegala, Y. Matsuo, and M. Ishizuka. Disambiguating Personal Names on the Web using Automatically Extracted Key Phrases. In *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI-06)*, 2006.

[9] E. Agichtein and L. Gravano. Snowball: Extracting Relations from Large Plain-Text Collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries (ICDL-00)*, 2000.

[10] G. Mann and D. Yarowsky. Unsupervised Personal Name Disambiguation. In *Proceedings of Conference on Natural Language Learning (CoNLL-03)*, pages 33-40, 2003.

[11] 上田洋, 村上晴美, 辰巳昭治. Web 上の同姓同名人物識別のための職業関連情報の抽出. 第 22 回人工知能学会全国大会, 2D2-3, 2008.

[12] J. Artiles, J. Gonzalo, F. Verdejo. A Testbed for People Searching Strategies in the WWW. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-05)*, pages 569-570, 2005.

[13] J. Artiles, J. Gonzalo, and S. Sekine. The SemEval-2007WePS Evaluation: Establishing a benchmark for the Web People Search Task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-07) at the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, pages 64-69, 2007.

[14] J. Finkel, T. Grenager, and C. Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, pages 363-370, 2005.

[15] 森純一郎, 松尾豊, 石塚満. Web からの人物に関するキーワード抽出. 人工知能学会論文誌, Vol.20, No.5, pages 337-345, 2005.

[16] J. Reisinger and M. Pasca. Bootstrapped Extraction of Class Attributes. In *Proceedings of the 18th International World Wide Web Conference (WWW-09)*, pages 1235-1236, 2009.

[17] K. Balog, J. He, K. Hofmann, V. Jijkoun, C. Monz, M. Tsagkias, W. Weerkamp, and M. Rijke. The University of Amsterdam at WePS2. In *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS-09) at the 18th International World Wide Web Conference*, 2009.

[18] K. Tokunaga, J. Kazama, and K. Torisawa. Automatic Discovery of Attribute Words from Web Documents. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05)*, pages 106-118, 2005.

[19] 白砂健一, 小山聡, 田島敬史, 田中克己. Web の構造情報とプロファイル抽出を用いたオブジェクト識別, DEWS2006, 2C-i7, 2006.

[20] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open Information Extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, 2007.

[21] M. Fleischman and E. Hovy. Fine Grained Classification of Named Entities. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING-02)*, pages 1-7, 2002.

[22] M. Lan, Y. Zhang, Y. Lu, J. Su, and C. Tan. Which Who are They? People Attribute Extraction and Disambiguation in Web Search Results. In *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS -09) at the 18th International World Wide Web Conference*, 2009.

[23] M. Pasca. Organizing and searching the World Wide Web of facts - step two: harnessing the wisdom of the crowds. In *Proceedings of the 16th International Conference on World Wide Web (WWW-07)*, pages 101-110, 2007.

[24] N. Yoshinaga and K. Torisawa. Open-Domain Attribute-Value Acquisition from Semi-Structured Texts. In *Proceedings of OntoLex Workshop at the 6th International Semantic Web Conference (ISWC-OntoLex-07)*, 2007.

[25] P. Pantel and M. Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, pages 113-120, 2006.

[26] 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, DEWS-06，2Ci11，2006.

[27] 木村塁, 小山聡, 田中克己. Web からの人物事典生成のための経歴情報の自動収集. 日本データベース学会 letters, Vol.5, No.2, pages 29-32, 2006.

[28] R. Wang and W. Cohen. Automatic Set Instance Extraction using the Web. In *Proceedings of the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP-09)*, 2009.

[29] S. Brin. Extracting Patterns and Relations from the World Wide Web. In *Proceedings of the 1998 International Workshop on the Web and Databases (WebDB-98)*, 1998.

[30] 小野真吾, 佐藤一誠, 吉田稔, 中川裕志. 重要語抽出を用いた Web 文書上の同姓同名の曖昧さ解消, DEWS-08, A7-3, 2008.

[31] 佐藤進也, 風間一洋, 福田健介, 村上健一郎. 実世界指向 Web マイニングによる同姓同名人物の分離, 情報処理学会論文誌：データベース, Vol. 46, No. SIG 8 (TOD26), pages. 26-36, 2005.

[32] S. Ravi and M. Pasca. Using Structured Text for Large-Scale Attribute Extraction. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM-08)*, 2008.

[33] 関根聡, 須藤清, 安藤まや. 属性値の自動抽出と質問文パターンを使った百科事典質問応答システム. 言語処理学会第 11 回年次大会, 2005.

[34] S. Sekine and J. Artiles. Weps 2 evaluation campaign: overview of the web people search attribute extraction task. In *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS-09) at the 18th International World Wide Web Conference*, 2009.

[35] T. Hasegawa, S. Sekine, and R. Grishman. Discovering Rleations among Named Entities from Large Corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL-04)*, 2004.

[36] 外間智子, 北川博之. Web データを用いた人物の呼称抽出. 日本データベース学会 letters, Vol.5, No.2, pages 49-52, 2006.

[37] 本間大輝, Danushka Bollegala, 松尾豊, 石塚満. Web を用いた人物の別名抽出. NLP 若手の会第 2 回シンポジウム, 発表 12, 2007.

[38] T. Pedersen, A. Purandare, and A. Kulkarni. Name Discrimination by Clustering Similar Contexts. In *Proceedings of the 6th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-05)*, pages 220-231, 2005.

[39] 高橋哲朗, 乾健太郎, 松本裕治. テキストから属性関係を抽出する. 情報処理学会研究報告, 自然言語処理研究会報告, pages 19-24, 2004.

[40] X. Han and J. Zhao. CASIANED: People Attribute Extraction based on Information Extraction. In *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS-09) at the 18th International World Wide Web Conference*, 2009.

[41] X. Li, P. Morie, and D. Roth. Robust Reading: Identification and Tracing of Ambiguous Names. In *Proceedings of the Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics (HLT/NAACL-04)*, pages 17-24, 2004.

[42] 吉田康浩, 鍛治伸裕, 喜連川優. 同姓同名人物情報の分類に関する一考察, DEWS-07, B9-7, 2007.

[43] Y. Chen and J. Martin. Towards Robust Unsupervised Personal Name Disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-07)*, 2007.

[44] Y. Chen, S. Lee, and C. Huang. PolyUHK: A Robust Information Extraction System for Web Personal Names. In *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS-09) at the 18th International World Wide Web Conference*, 2009.