修 士 論 文

WEB 上のテキストからの
表形式を出力とする情報抽出

指導教官　鳥澤健太郎 助教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

曽我部 泰正

2003 年月

要 旨

本論文では,web 上のテキストから表形式を出力とする情報抽出の手法を提案する. 表は文章の簡潔な要約とみなすことができ, 我々の情報抽出手法は一般のテキストを要約する一手法であると考えることができる. 近年 internet の普及により,web 上では様々な情報が公開されつつあり, その量は増加の一途をたどっている. ある事柄について web 上から情報を調べる際には, 主にサーチエンジンを用いた情報検索に頼っている. しかしながら, 通常のサーチエンジンは単にあるキーワードを含むサイトをリストアップするに過ぎず, その後にユーザー自身が検索結果のリスト中のサイトのそれぞれにアクセスして文書を読み, 情報を選別するという手間のかかる作業が必要となる. 現在, その手間を減らす一つの手法として自動要約システムが研究されているが, この手法を用いても結局は文章を読むことにはなんら変わりない. また, 上述の手間を減らすもう一つの手法として膨大な情報源の中から必要な情報のみを抜き出す「情報抽出」という技術について研究が行われている. しかしながら, 既存の情報抽出手法は抽出のパターンを人手で生成しなくてはならないという問題点がある. さらに手作業でのパターンの生成には時間がかかる上, 限られたトピックにしか対応できない.

この問題点を解決するため, 本研究では手作業によるパターンを必要としない情報抽出手法を提案する. 本研究では,web 上に存在する多数の表形式からの教師無し学習で得られた知識をもとに, 通常のテキストから表形式の構成要素となりうる語を抽出し, 表形式に変換する手法について研究を行う.

目 次

第 1 章

はじめに

1.1 研究の背景と目的

近年 internet の普及により,web 上では様々な情報が公開されつつあり, その量は増加の一途をたどっている. ある事柄について web 上から情報を調べる際には, 主にサーチエンジンを用いた情報検索に頼っている. この手法では単にあるキーワードを含むサイトをリストアップするに過ぎず, その後にユーザー自身が検索結果のリスト中のサイトのそれぞれにアクセスして文書を読み, 情報を得なくてはならず, 一般ユーザーにとって不便である. 現在, その手間を減らす手法として自動要約システムが研究されているが, この手法を用いても結局は文章を読むことにはなんら変わりない. そこで, 膨大な情報源の中から必要な情報のみを抜き出す「情報抽出」という技術について研究が行われている. しかしながら, 既存の情報抽出手法は抽出のパターンを人手で生成しなくてはならないという問題点がある. さらに手作業でのパターンの生成には時間がかかる上, 限られたトピックにしか対応できない. この問題点を解決するために本研究では,「人間はあるオブジェクトについての表を作成する際に, 常識に基づいてそのオブジェクトの記述に必要な情報だけを選んでいる.web 上にはこのように人間がなんらかを意図して作成した表が多数存在している.」という点に着目し,web 上に存在する不特定多数の表を収集し, 表をクラスタリング, 統合したものを入力データとして用い, それらの表の構成要素にタグ付けを行い,web 上に存在する通常テキスト中に表の構成要素と同じ記述があった場合に同様のタグを付与したものを学習データとみなし, 表の構成要素の推定の学習を行い, そのモデルをもとに通常のテキストから表の構成要素を抽出することにより, 幅広いトピッ

クに適用可能かつ手作業による抽出パターン作成を必要としない情報抽出の手法を提案する.

1.2 本論文の構成

2 章では情報抽出,web 上に存在する多数の表をクラスタリング, 統合するアルゴリズム及び SVM による機械学習を用いてテキストから固有表現を抽出するアルゴリズム, また固有表現を抽出する際のひとつの素性として用いる単語クラスについて説明する.3 章では web 上のテキストからの情報抽出システムについて,4 章では実験方法及び実験結果について,5 章では実験結果の考察,6 章では結論及び今後の課題について述べる.

第 2 章

関連研究

本章では, 吉田ら [1] による web 上に存在する多数の表を収集し, 表を種類ごとに分類, 統合するアルゴリズム及び山田らの提案した SVM を用いた固有表現抽出のアルゴリズム, 表の構成要素抽出の学習を行う際に素性として用いる単語クラスについて説明する.

2.1 情報抽出

近年テキストからの情報抽出という技術について活発に研究が行われている. その中心的存在となっているのが英語を対象とした情報抽出国際会議である MUC(Message Understanding Conference) [14] や日本語を対象に国内で開かれた IREX(Information Retrievaland Extraction Exercise) [15] などである.MUC における情報抽出とは新聞記事のようなテキストからあらかじめ指定されたイベントや事柄に関する情報を抽出し, その情報をデータベースに入力する, という技術である. その具体例として, 人事異動に関する新聞記事を基にした情報抽出結果をを図 2.1 に示す.

ここでは, 抽出したい情報は企業の重役の異動に関する情報であり, 抽出したい情報の内容としては, 該当者の氏名, 会社, 異動前役職名, 異動後役職名, 異動事由, 異動発生日というように与えられている.

情報抽出技術の適用により, 面倒な文章で書かれた人事異動の情報が視覚的にわかりやすい形に集約されている. このような技術は特定の情報を簡単に調べたいときなどに非常に役立つ. たとえば過去 10 年の新聞記事から, 企業の重役の異動に関する情報を得たいという場面を想定する. 現在よく用いられている情報検索に技術を利用すると, 適当

＜新聞記事＞
　ABC株式会社は十二日,臨時取締役会で,田中一郎社長が代表権のある会長に就任し,山田次郎副社長が社長に昇格する人事を内定したと発表した.鈴木三郎会長は代表取締役にとどまる.三月二十五日に開く株主総会後の取締役会議で正式決定する.田中社長は五期十年社長を務め,年齢も七十一歳と高齢になったため若返りを図る. ……

＜異動イベント情報＞
人名：　田中一郎
会社名：　ABC株式会社
異動前役職：　社長
異動後役職名：会長
異動理由：　昇格
異動発生日：　3月25日

人名：　山田次郎
会社名：　ABC株式会社
異動前役職：　副社長
異動後役職名：社長
異動理由：　昇格
異動発生日：　3月25日

人名：　鈴木三郎
会社名：　ABC株式会社
異動前役職：　会長
異動後役職名：代表取締役
異動理由：　降格
異動発生日：　3月25日

図 2.1 Sample Tables

な検索式を作成し, 異動に関連した記事へのインデクスを引き出すことはできる. しかし, 実際に必要な情報はユーザーがそれぞれの記事を読んで得なくてはならない. また自動要約という技術を利用してその手間を減らすことは可能であるが, それでも文章を読むことには変わりはない. ここでは一例として, 重役の人事異動をとりあげたが, 対象となる情報は, 前もって「抽出したい情報の型」が決められれるものであれば特にこだわりはなく, 新製品の情報, 合弁事業の情報など, 産業界に役立ちそうな内容や, 研究者向けには科学技術論文における技術内容の情報抽出や, 医療カルテ, ゲノムといった分野, またスポーツなどの特定イベントの情報など, 個人的な利用も含めて広く適用可能である.

次に情報抽出の手法について説明する. 初期の MUC では構文解析などの技術を用いる方式が主流であったが, パターンマッチングの方が性能的に優れていたために, 現在ではあまり研究されていない. パターンマッチングはその情報抽出の対象に関係する文や文の一部にマッチするパターンを用意しておいて, それを決まった順に適用し, 決定的に抽出する方法である. パタンマッチングに基づくシステムは多数のパターン (Pattern) 書き換え規則と, パターンを文書に照合するパターンマッチャ (Pattern Matcher) から構成される. 図 2.2 に簡略化した書き換え規則の例を示す.

```
[アーン]+ “株式会社”                    → /企業名/
/企業名/ “(本社・” (/名詞/+) ”)”        → /企業名/”(本社・” /地名/ ”)”
/企業名/ “(本社・” /地名/ ”)”           → /企業/
[1－3]*[0－9] “日”                      → /日付/
[0－9]+ “万” [0－9]+ “円”               → /金額/
「/名詞/+」”(” /金額/ “)”               → 「/製品名/」”(” /金額/ “)”
「/製品名/」”(” /金額/ “)”              → /製品/
/製品/ “と” /製品/ “の” [0－9]+ “機種”  → /製品/
/企業/ “は” /日付/ /製品/ “を発売した”  → /製品販売事象/
```

図 2.2 Sample Tables

パターンマッチングは難しい技術を使用せず, 深層的な理解を試みることなく情報抽出が実現できるという利点がある. しかし, 情報抽出にパターンマッチング技術を用いることによる問題点は, パターンを情報抽出のドメインごとに作成しなくてはならないという点である. 例えば,「人事異動」に関するパターンは「人事異動」にしか使用できず,

新たに「新製品の発売」に関する情報抽出を行いたい場合はそのためのパターンを新たに作成しなくてはならない. また, ドメインによって必要なパターンの数は異なるが, ある程度複雑なドメインの場合は 500 から 1000 近いパターンが必要となってくる. これらのパターンをドメインごとに全て手作業で作成していくのでは, コストが膨大となってしまう. そこでパターンを自動的に生成する方法について研究が進められている.

2.1.1 固有表現抽出

前述の IREX においては, 新聞記事を対象にした固有表現抽出が採用されている. 固有表現抽出とは文書中に含まれる, 組織名・地名・人名などの固有名詞や, 日付・時刻・金額などの数値表現を文書の中から発見し, タグを付加するタスクである. 固有表現抽出は, 情報抽出における重要な基礎技術であり, また形態素解析や構文解析などの処理に大きな影響を及ぼすため重要な問題とされている. 固有表現タグは以下に示す 8 種類となる.

- ⟨ORGANIZATION⟩ 組織名 ⟨/ORGANIZATION⟩ 複数の人間で構成され, 共通の目的を持った組織の名称
  例えば「日本銀行」や「自由民主党」

- ⟨PERSON⟩ 人名 ⟨/PERSON⟩ 固有の人を指す名前
  例えば「小泉純一郎」や「ブッシュ」

- ⟨LOCATION⟩ 地名 ⟨/LOCATION⟩ 固有の場所を指す名称
  例えば「石川県金沢市」や「国道２３号」

- ⟨ARTIFACT⟩ 固有物名 ⟨/ARTIFACT⟩ 人間の活動によって作られた固有物の名称
  例えば「ペンティアムプロセッサ」や「ノーベル化学賞」

- ⟨DATE⟩ 日付表現 ⟨/DATE⟩ 単位が２４時間以上のもの
  例えば「２００３年２月１日」や「春」

- ⟨TIME⟩ 時間表現 ⟨/TIME⟩ 単位が２４時間以下のもの
  例えば「午前９時」や「正午」

- ⟨MONEY⟩ 金額表現 ⟨/MONEY⟩ 金額をあらわす表現
  例えば「１２８円」や「９３ドル」

- ⟨PERCENT⟩ 割合表現 ⟨/PERCENT⟩ 割合をあらわす表現
  例えば「２４倍」や「６０%」

次に固有表現抽出の例を図 2.3 に示す.

<DOC>
<DOCNO>940413099</DOCNO>
<SECTION>経済</SECTION>
<AE>無</AE>
<WORDS>147</WORDS>
<HEADLINE><ORGANIZATION>タカラ</ORGANIZATION>社長に副社長
の<PERSON>佐藤
博久</PERSON>氏が昇格</HEADLINE>
<TEXT>
おもちゃ大手の＜ORGANIZATION＞タカラ＜/ORGANIZATION＞は＜
DATE＞十二日＜/DATE＞＜PERSON＞佐藤博久＜/PERSON＞副社長が
社長に昇格する人事を内定した。創業者の＜PERSON＞佐藤安太＜
/PERSON＞社長は会長に就任する見通し。＜PERSON＞博久＜/PERSON
＞氏は＜PERSON＞安太＜/PERSON＞氏の長男。正式決定は＜DATE＞六
月下旬＜/DATE＞。
＜PERSON＞佐藤博久＜/PERSON＞氏(さとう・ひろひさ) ＜DATE＞1979
年＜/DATE＞慶大法卒、＜DATE＞80年＜/DATE＞＜ORGANIZATION＞タ
カラ＜/ORGANIZATION＞入社。常務などを経て＜DATE＞92年4月＜
/DATE＞から副社長。＜LOCATION＞東京都＜/LOCATION＞出身、38歳。
＜/TEXT＞
＜/DOC＞

図 2.3 Sample Tables

固有表現は多様性に富み, また次々と新たに生み出されるため, 辞書に登録することは不可能である. そのため辞書だけを手がかりにして固有表現を同定することは不可能である.

固有表現抽出に用いられている方法はパターン駆動型, 自動学習型の 2 つに分類される.

パターン駆動型

パターン駆動型は前述の情報抽出の主要技術であるパターン書き換え規則を固有表現抽出に用いるものである. パターン書き換え規則の作成は人手によるもので, 数百の規則が用いられるのが普通である.

自動学習型

自動学習型はラベル付与を行う書き換え規則を正解付きコーパスから機械学習により自動的に獲得するものである. この方法では, 学習用に正解コーパスを作る必要があるが, データスパースネスに強い学習モデルを使えばそれほど大量のコーパスがなくても高い精度を得ることができる. 学習方法としては, 関根ら [8] による決定木を用いたもの, 内元ら [10] による, 最大エントロピー法とパターン書き換え規則とを用いるもの, 山田ら [4] による Support Vector Machines を使用する方法等がある.

- 決定木を用いた手法
  関根ら [8] は決定木を用いて自動的にパターンを学習する方法をとった. 従来の学習システムの問題点は部分的に手作業のルールを使用すること, 手動で調節しなくてはならないパラメータを持つこと, 自動的な手段では性能がよくないことが挙げられている. また, 決定木は決定的であるため, 固有表現の種類や範囲に矛盾が生じるなどの問題も挙げられている. しかし, 関根らのシステムは JUMAN が出力する品詞, 文字型 (漢字, ひらがな, アルファベット, 数, 記号, それらの組み合わせ) 辞書, 固有表現の開始位置, 中間位置, 終了位置, を表すタグを決定木の入力に与えること, テキスト内で最も確率の高い首尾一貫したタグを選び出すことで, これらの問題点

を解決している.

- 最大エントロピー法とパターン書き換え規則を用いた手法
  内元ら [10] は最大エントロピー法とパターン書き換え規則を用いて固有表現抽出を行っている. 固有表現にはひとつあるいは複数の形態素からなるものと, 形態素単位より短い部分文字列の 2 種類が存在する. 前者の固有表現に対しては固有表現の始まり, 中間, 終わりなどを表すラベルを 40 個用意し, そのラベルを最大エントロピーモデルによって推定することによって抽出を行う. 最大エントロピーモデルはデータスパースネスに強いため, 大量の学習データがなくても高い精度が得られる. 後者の固有表現は, 学習コーパスに対するシステムの解析結果と正解データとの差異から自動獲得した書き換え規則によって抽出する. 実験により, 着目している形態素の前後 2 形態素に関する見出し語及びその品詞情報が素性として有効であるとしている. また固有名詞辞書を利用して,IREX-NE の本試験データに対して行った実験では, 一般ドメインに対し F 値で 80.17 の精度を得ている.

- Support Vector Machines を用いた手法
  山田ら [4] は Support Vector Machines を用いて固有表現抽出を行っている. 固有表現抽出規則を学習する場合には, 学習に用いる素性は前述のように, 語彙, 文字種, 品詞などを用いるため, その素性空間は数万以上の高次元空間となるため, 過学習に頑強なアルゴリズムが求められる.Support Vector Machines はその汎化誤差が素性空間の次元に依存しないことが理論的に証明されており, 実験的にも,Chunking, 文書分類など, 高次元素性空間での学習の必要な様々な問題に対して適応され, 他の学習アルゴリズムと比べて良い成績を収めている. 山田らは固有表現抽出を行うに際して, 二値分類器である Support Vector Machines を多値分類に対応できるように拡張した. 様々な素性を組み合わせて比較実験を行ったところ, 素性として, 語彙, 品詞細分類, 文字種を用いて学習した結果が良いことがわかった. また, 固有表現抽出を文頭から文末に向かって行う右向き解析と文末から文頭に向かって行う左向き解析を比較したところ, 左向き解析のほうが精度が良かったとしている. 山田らの実験は,IREX-NE 本試験データに類似した CRL(郵政省通信総合研究所) 固有表現データを用いて行い,F 値が 83.01 と, 前述の関根ら [8] や, 内元ら [10] と同等以上の精度が得られたと報告している.

2.1.2 本研究との関連

このようにして得られた固有表現抽出の結果を考慮して, 情報抽出を行う際のパターン書き換え規則を生成することにより, 情報抽出システムの精度向上につながる.

固有表現抽出タスクでは先程紹介した,8 種類のクラスに属するか否かという情報のみを付与している. 本研究ではこの固有表現抽出の技術を応用し, 固有表現抽出の際のクラスの数をさらに細かく設定することにより, パターン書き換え規則を用いることなく, 単独で情報抽出もしくはそれに近い結果を得るようなシステムを構築することを試みる.

2.2 表のクラスタリングに関する研究

吉田ら [1] は WWW ページ上に存在する表を収集し, 教師無し学習を用いて表を種類ごとに分類, 統合して出力するための手法を提案している. 本研究ではこの手法により得られた知識をもとに通常のテキストから表形式の構成要素となりうる語を抽出する. 本節では吉田ら [1] の研究について説明する.

2.2.1 概要

WWW の魅力は, 世界中に存在する多様な情報に容易に触れることが出来る点にあるが, その多様さゆえに, ユーザーが欲する情報にアクセスするためには何らかの形でこれらの情報を整理する必要がある. 情報をどのようにして整理するかについては様々な方法が考えられるが, 吉田らは表, すなわち,HTML の TABLE タグ ⟨TABLE⟩,⟨/TABLE⟩ で囲まれた部分に着目し, それぞれの表の内容により多数の表を分類, 統合する手法についての研究を行っている.

図 2.4 に吉田らのシステムの動作例を示す. この例では自己紹介の表,PC スペックの表の集合から, 自己紹介の表同士,PC スペックの表同士をそれぞれクラスタにまとめ, 同一クラスタ内の表を統合して出力している.

吉田らの手法は大きく分けて以下の 2 つのプロセスから構成される.

- *Table Structure Recognition* (表の構造の認識)
- *Table Integration* (表の統合)

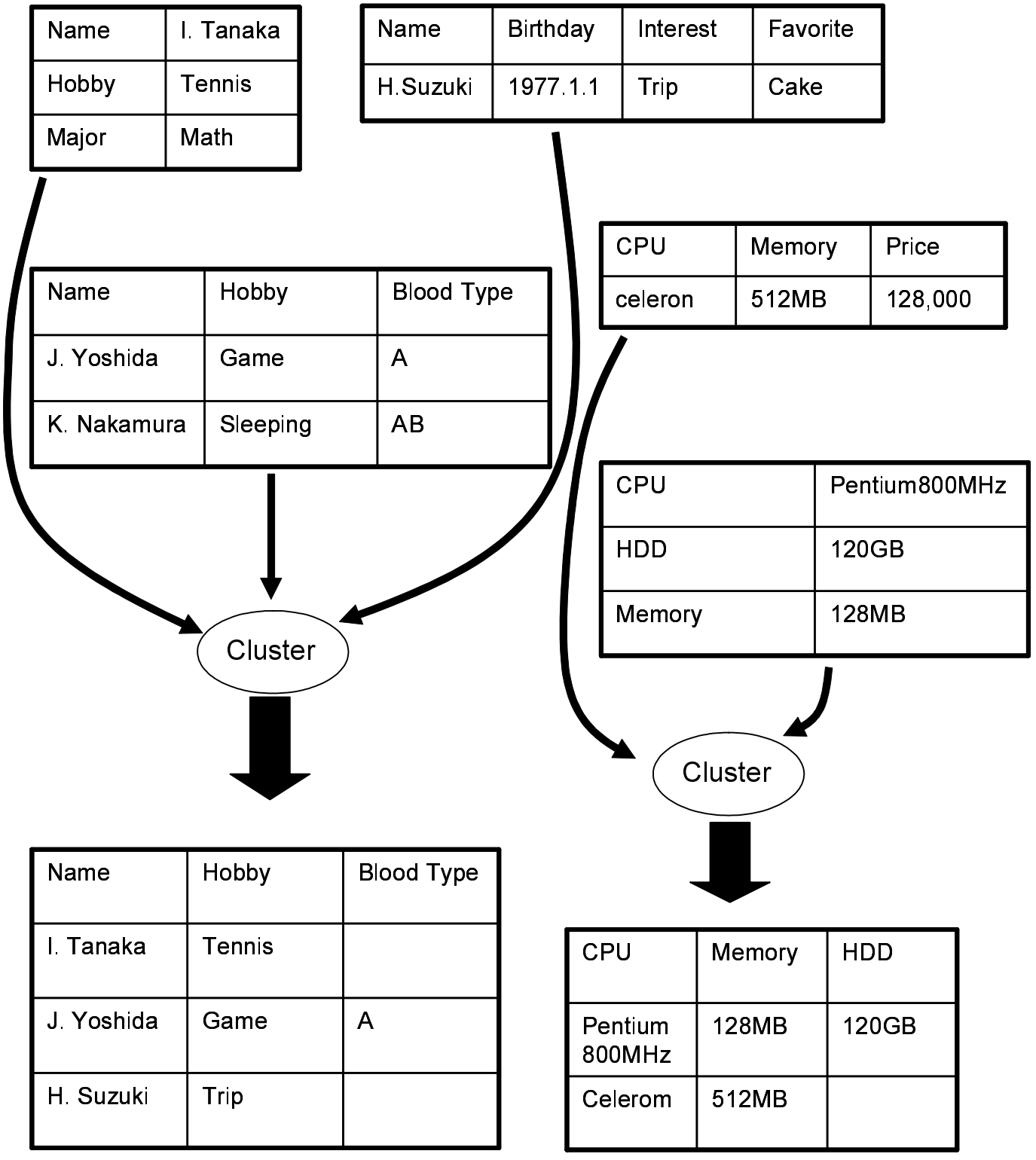

図 2.4 吉田らのシステムの動作例

この手法では,表は実世界に存在するオブジェクトを属性(attribute)とその属性値(value)の対によって表現しているという仮説に基づいている.例えば図 2.5 の表 (b) は”Name”という属性と”Hanako”という属性値を持っている.このような属性と属性値のレイアウトのことを *table structure* と呼ぶ.

様々なフォーマットで書かれた表を 1 つに統合するためにはまずはじめにそれぞれの表の

Name	Tel.	Recommendation
Lake Restaurant	31-2456	special Lunch
Café Bonne	12-3456	chicken curry
Metro Restaurant	56-7890	fried rice

(a) A table representing restaurants

Name	Hanako	Blood Type	A
Gender	Female	Birthday	22 Feb.
Nationality	Japanese	Tel.	12-3456

(b) A table representing a person

John	Richard	Tom
Jude	Mary	Bill

(c) A list of names

図 2.5 Sample Tables

table structure を認識する必要がある.このタスクのことを *Table structure recognition* と呼ぶ.*Table integration* タスクは認識された *table structure* に基づいて行われる.次節か

らはこれらのタスクの詳細について説明する.

2.2.2 Table Strucuture Recognition

Table Structure Recognition タスクとは表のどの部分が属性でどの部分が属性値なのかを認識するものである.

このタスクに関してはいくつかの関連研究が存在するが, ほとんどの手法が数字で表された語の数や文字列の長さや HTML タグ, 又は線の太さなどの表層から汲み取ることの出来る素性を用いて認識している. それらの手法により, 意義深い結果が得られているが, 吉田らの手法では異なった見地からこのタスクを見ている. 人間による表の解釈には概念的知識 (ontological knowledge) が必要である. 例えばもし, ある人についての表を作りたいとした場合, ふさわしい表の属性として,"名前"や"誕生日"や"趣味"などを選び, さらにそれらの属性にふさわしい属性値をチョイスするであろう. 吉田らは人間のこのような判断は一般的な概念的知識にのっとって行われていると考えている. 吉田らの手法ではこれらの概念的知識を様々なオブジェクトについて様々なフォーマットで書かれた多数の表から抽出し, その知識を表の構造の認識に用いる. これを Expectation Maximization Algorithm [?] を用いて実装する. このアルゴリズムにより文字列が属性 (または属性値) として現れる確率を推定し, その確率分布をもとに table structure を決定する.

定義

吉田らは表 T を

$$(\langle s_1, s_2, ..., s_{xy} \rangle, x, y)$$

と表現する.

T: 表

x: 表の中の列

y: 表の中の行

$\langle s_1, s_2, ..., s_{xy} \rangle$: 表中の連続した文字列

- 表中のそれぞれの文字列は属性又は属性値のいずれかに分類されると仮定する.table

structure は属性とその値が表中のどこに現れているのかを特定する. また, 表中のそれぞれの文字列に att または val のいずれかのラベルを与えこの構造を表現する.

$$\langle (s_1, l_1), (s_2, l_2), ..., (s_n, l_n) \rangle$$

l_i は s_i に対応するラベルである.

- table structure は図 2.6 に示す 9 種類に分類されるものと仮定する.
  今後は簡素化のため table structure のことを type と呼ぶこととする.

table structure 決定のアルゴリズム

このアルゴリズムは次の式に示すような推定確率に基づいてタイプ

$$M = \langle m_1, \ldots, m_n \rangle$$

の中から尤もらしいシーケンスを表$\mathcal{T} = \langle T_1, ..., T_n \rangle$ の入力シーケンスとして選択する.

$$\begin{aligned} \mathcal{M} &= \arg \max_{\mathcal{M}=\langle m_i \rangle_{i=1}^n} \prod_i P(m_i|T_i)\} \\ &= \arg \max_{\mathcal{M}=\langle m_i \rangle_{i=1}^n \prod_i P(m_i,T_i)} \end{aligned}$$

ここで確率 $P(m_i, T_i)$ をパラメータ θ を用いて表現する. $P_\theta(m_i, T_i)$ は以下のようになる.

$$\theta = \{P_\theta(m|x, y)\} \cup \{P_\theta(s|l)\}$$

$$\begin{aligned} P_\theta(m, T) &= P_\theta(m, \{\langle s_[i] \rangle_i = 1^n, x, y\}) \\ &= P(x, y) P_\theta(m|x, y) P_\theta(\langle s_[i] \rangle_{i=1^n} | m, x, y) \\ &\approx P(x, y) P_\theta(m|x, y) P_\theta(\langle s_[i] \rangle_{i=1^n} | m) \\ &\approx P(x, y) P_\theta(m|x, y) \prod_{(s,l) \in m(T)} P_\theta(s|l) \end{aligned}$$

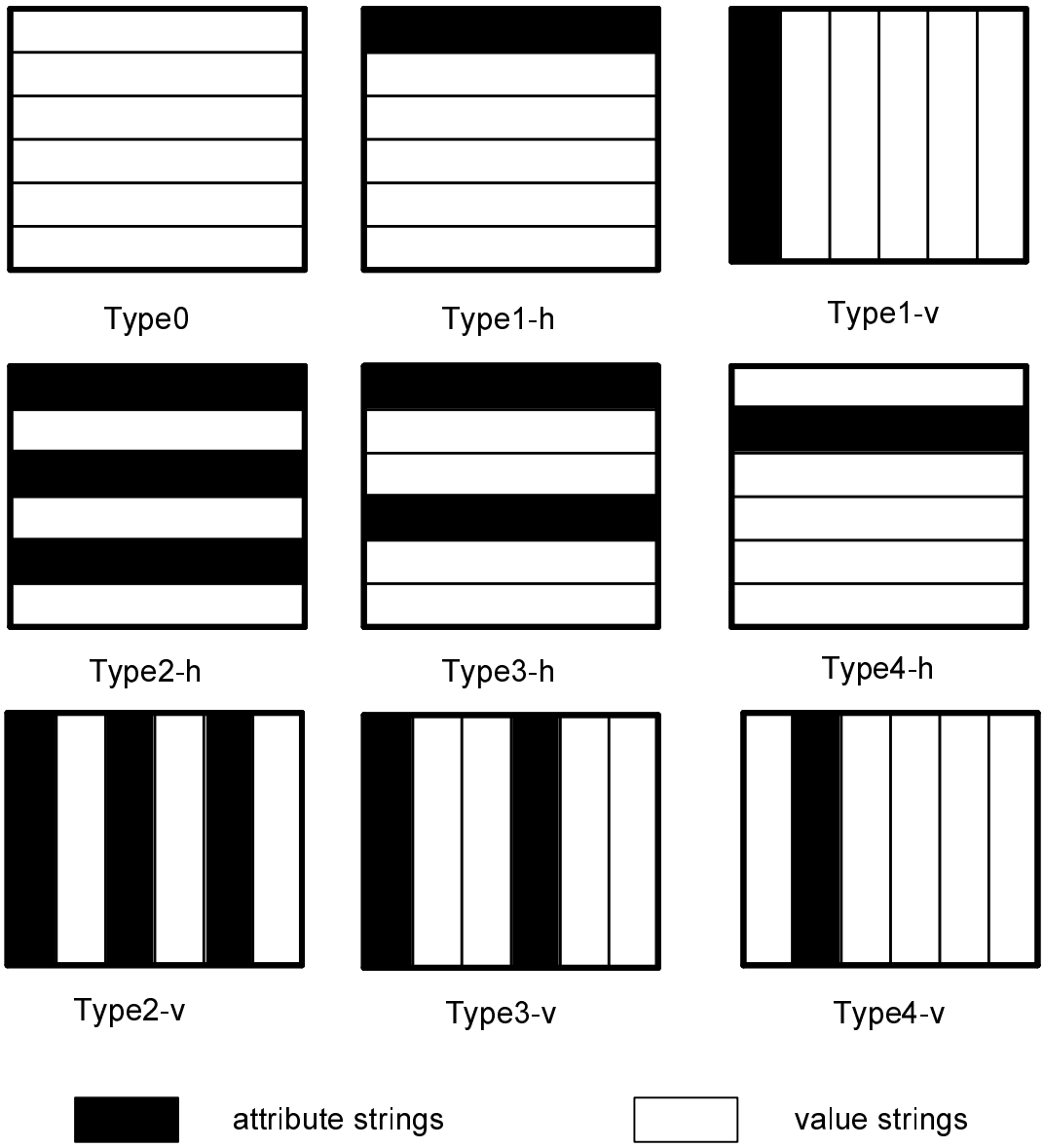

図 2.6 table structure の種類

最後の式変形において

$$P_\theta(\langle s_{[}i]\rangle_{i=1^n}|m)$$

は表のタイプ $m(T)$ において全ての (s,l) の組についてラベルが l であったとき, その文字列が s である確率

$$P(s|l)$$

の積によって近似した.

EM アルゴリズムでは次の式によってパラメータ θ を繰り返し調整することにより

$$\prod_i P(m_i|T_i)$$

の値を改善している.

$$P_\theta(m|x,y) = \frac{1}{|\mathcal{T}_{\S\dagger}|}\sum_{T\in\mathcal{T}_{\S\dagger}} P_{\theta'}(m|T)$$

$$P_\theta(s|l) = \frac{1}{Z_{\theta'}(l)}\cdot\sum_{i,k}\sum_{k(m(T_i))=(s,l)} P_{\theta'}(m|T_i)$$

$k(m(T))$: $m(T)$ の k 番目の要素

$\sum_{k(m(T_i))=(s,l)}$: $m(T)$ の k 番目の要素が (s,l) となるような m の値の総和

θ' : θ の一つ前の状態

$Z_{\theta'}(l)$: $\sum_s P_{\theta'}(s|l) = 1$ とするための正規化項

$\mathcal{T}_{xy}$: サイズが (x,y) の表の系列

2.2.3 Table Integration

Table Integration タスクは

- Table Clustering
- Table Merging

という 2 つのプロセスに分かれている. 本小節ではこれらのプロセスの詳細について説明する.

Table Clustering

Table Clustering とは, 様々な web サイトに存在する表の中から, よく似た物事について書かれた表のクラスタを生成するプロセスである. 表のクラスタリングを行うために用いる, 特徴属性という概念について説明する.

特徴属性とはある物事について書かれた表の属するクラスに特有な属性のことである. 以下に例を示す.

- 「Hobby」は自己紹介に関する表に特有の属性である
- 「CPU」はコンピュータに関する表に特有の属性である

ある属性 a の特有性の度合いを関数 $uniq(a)$ として以下のように表現する.

$$uniq(a) =_{def} cooc(a)\dot{e}xcl(a)$$

ここで,

$$cooc(a) =_{def} \frac{1}{V(a)} \sum_{b \in V(a)} Freq(b, a)$$

$$excl(a) =_{def} \frac{1}{V(a)} \sum_{b \in V(a)} \frac{Freq(b,a)}{|U(b)|}$$

$U(a)$: 属性 a の現れている表の集合

$V(a)$: $U(a)$ 中に現れた属性の集合

$Freq(b,a)$: 表 $U(a)$ 中に現れている属性 b の頻度

$cooc(a)$: どれぐらい首尾一貫して $V(a)$ 中の属性 a が繰り返し起こっているかという共起頻度を表している.

$excl(a)$: $V(a)$ 中の属性の排他的さの度合いを示す指標

$uniq(a)$ が大きければ $U(a)$ はよく似た物事について書かれた表の集合となる.

吉田らのアルゴリズムでは $uniq(a)$ が大きくなるような属性 a をとり, $U(a)$ をクラスタとする.

Table Merging

次に Table Merging について説明する.

Table Merging タスクはよく似た属性を 1 つの集団（クラスタ）として分類する属性クラスタリングを実行することにより実現される.

2.2.4 実験方法及びその結果

吉田らの手法を用いて web 上から集めてきた 35232 個の表を含む html(このデータセットを S とする) に対して実験を行った. なお, これらの表のほとんどは日本語によって書かれていた.

Table Structure Recognition

S 全体からパラメータ推定を行った後, 表にどの type が割り当てられるかを S からランダムに 175 個の表を選んで前述のアルゴリズムを適用精度の評価はクローズドテス

トで行った. 精度は $n/175$ として計算された. n は正しい type が割り当てられた表の数である.

- 認識された表の種類及び個数

Type0	76
Type1-h	61
Type1-v	35
Others	3

- iteration と精度の相関
  iteration を行うことにより表の構造認識の精度が 0.66 から 0.78 に向上しており,iteration の有用性が示された.

- 先行研究との比較
  次に,html から表を抽出するタスクについて, 先行研究である Chen [16] らの手法との性能比較を行った.Chen [16] らの研究ではあらかじめ表ではないものすなわち（type0 に値するもの）を除去した上での抽出の精度は precision が 92.92Chen [16] らの研究の評価基準にのっとって吉田らの手法の精度を評価すると 79.44

  Chen [16] らの研究においては HTML のジャンルを航空会社のページだけに限定していたが, 吉田らの手法は様々なジャンルの多数の表に対して適用可能であることを考慮すれば十分な精度であるといえる.

Table Integration

- Table Clustering の結果
  $|V(a)|$ の値の大きい上位 15 個のクラスタを選んだ. それぞれのクラスタ中の無作為に選んだ１０の項目についてチェックすることによりクラスタリング結果を調査した. その結果考察として,”Hobby” ”CPU” ”Capital Money” などの属性は特徴属性としてふさわしいと言える. 一方で,”contents”や”Title”などは多義に取ることができるため, 特徴属性としてはふさわしくない. クラスタリング結果を改善する

ために, $uniq$ の定義を変更することや, 別のクラスタリング手法を採用することなどが考えられている.

- Table Merging の結果
  次に table merging の性能を示す.

  吉田らのアルゴリズムは頻度が上位 7 番目までの属性を統合された表に採用している. これらの属性をそれぞれのクラスタの main attribute と呼ぶ. 例えば,"Hobby" クラスタにおける main attribute は,Name,Birthday,Address,Bloodtype,Job,Hobby,Foods の 7 つである. 特徴属性が"Hobby" "CPU" "Capital Money" の 3 つのクラスタについて table merging の評価行った. それぞれのクラスタについて, 無作為に 10 のオブジェクトを選び, オリジナルの表中の値が統合された表中に出現しているかどうか (recall) 及び, 統合された表中に出現している値は正しい物かどうか (precision) の測定結果を表 2.1 に示す.

表 2.1 table merging の結果

Cluster	Precision	Recall	F-measure
SHUMI	0.98	0.79	0.87
CPU	0.90	0.82	0.86
SIHONKIN	0.94	0.77	0.85

2.2.5 本研究との関連

以上のように, 吉田らはパラメータ推定にトレーニングサンプルを与えない確率モデルに基づいた表の統合の手法を確立した. 本研究では吉田らの手法を用いて教師無し学習によって自動的に統合された表から得られる知識に基づき, 表形式ではない一般の web 上のテキストから表の構成要素となる語を抽出することを目標とする. 次節では, テキストからの表の要素の抽出のために用いるアルゴリズムの元となった山田ら [4] による Support Vector Machines を用いた固有表現抽出について説明する.

2.3 Support Vector Machines を用いた日本語固有表現抽出

情報抽出の節でも述べたが, 固有表現抽出規則を学習する場合には, 学習に用いる素性として, 語彙, 文字種, 品詞などを用いるため, その素性空間は数万以上の高次元空間となるため, 過学習に頑強なアルゴリズムが求められる.Support Vector Machines はその汎化誤差が素性空間の次元に依存しないことが理論的に証明されており, 実験的にも,Chunking, 文書分類など, 高次元素性空間での学習の必要な様々な問題に対して適応され, 他の学習アルゴリズムと比べて良い成績を収めている.

本節では, 日本語の固有表現抽出タスクにおいて, 現時点で最も高い精度を誇っている山田ら [4] による Support Vector Machines を用いた固有表現抽出について説明する.

2.3.1 固有表現の表現方法

IREX 日本語固有表現タスクにおいて定義されている固有表現については情報抽出の節で述べた通りである.

固有表現抽出は, ある単語が固有表現か否かを識別する Chunk 同定問題とみなすことができる. Chunk 同定問題においては,1 つ以上の形態素からなる Chunk を IOB1,IOB2,IOE1,IOE2 という 4 種類のタグを使用して表記する手法が提案されている. [17] 図 2.7 に 4 つの記法を用いて”小泉首相は七日午前零時を期して”という文に対してタグを付与したものを示す.

	小泉	首相	は	五	日	,	日	米	両国
IOB1	I-PERSON	O	O	I-DATE	I-DATE	O	I-LOCATION	B-LOCATION	O
IOB2	B-PERSON	O	O	B-DATE	I-DATE	O	B-LOCATION	B-LOCATION	O
IOE1	I-PERSON	O	O	I-DATE	I-DATE	O	I-LOCATION	E-LOCATION	O
IOE2	E-PERSON	O	O	B-DATE	E-DATE	O	E-LOCATION	E-LOCATION	O
SE	S-PERSON	O	O	B-DATE	E-DATE	O	S-LOCATION	S-LOCATION	O

図 2.7 固有表現識別のためのタグ表現

- IOB1
  固有表現である単語にIというタグを付与し, 同種類で別の表現が連続した場合には後続する固有表現の開始位置にBというタグを付ける. 固有表現以外の単語にはOというタグを付与する.

- IOB2
  IOB1とは違い, 固有表現の開始位置には必ずBというタグを付与する.

- IOE1
  固有表現である単語にIというタグを付与し, 同種類で別の表現が連続した場合には後続する固有表現の終了位置にEというタグを付ける. 固有表現以外の単語にはOというタグを付与する.

- IOE2
  IOE1とは違い, 固有表現の終了位置には必ずEというタグを付与する.

- SE
  １つの単語からなる固有表現にSというタグを付与. 複数の単語からなる固有表現にはその開始単語にB, 終了単語にE, 固有表現内の単語にI, それ以外の単語にOというタグを付与する.

ここからは, 固有表現の開始終了位置をあらわすB,I,O,E,Sの表記のことをChunkタグと呼び,IREXで定義した８つの固有表現のことを固有表現の種類と呼ぶ. また,Chunkタグと固有表現の種類が一つになったB-DATEのような表記のことを固有表現タグと呼ぶ. 固有表現タグを使用することで固有表現抽出規則の学習は入力文中の各単語を固有表現タグに分ける分類規則の学習として扱うことができる.

2.3.2 SVMの日本語固有表現抽出への適用

固有表現抽出規則の学習とは入力された文の各単語に対し固有表現タグに分類する規則を学習することである. また, 固有表現抽出とは未知の文に対して各単語の固有表現タグを推定することである. そのため, まず文の形態素解析を行い, 単語列に分割する. 固有表現タグの学習及び推定を行う単位は単語単位であり, １つの事例は1単語に対応する.

このとき, 文頭から順に固有表現タグを推定する方法を右向き解析と呼び, 逆に文末から順に固有表現タグを推定する方法を左向き解析という.

- 固有表現抽出規則の学習右向き解析を行う場合, 文頭から i 番目の単語に関する素性は $i-2$ から $i+2$ 番目までの各単語の単語自身, 品詞, 文字種を使用する. また, 複数の単語からなる固有表現を考慮するために, $i-2$ と $i-1$ 番目の固有表現タグも素性として使用する. これらの素性を要素とするベクトル x と i 番目の固有表現タグを分類すべきクラスを y とすれば (x, y) という組が 1 つの事例となる.

  左向き解析を行う場合, 使用する素性の種類は右向きと同様である. しかし, 文末を始点とするため に対する位置が右向き解析とは逆になる.$i-n(n\rangle 0)$ は i に対して文末側の n 個隣の単語を表す. また,$i+n$ は i に対して文頭側の n 個隣の単語を表す.

- 固有表現タグの推定 i 番目の固有表現タグの推定には学習時と同様に $i-2$ から $i+2$ 番目までの各単語の単語自身, 品詞, 文字種を素性として使用する. 未知の文に対する固有表現抽出では解析の最初において固有表現タグは未知であるため, $i-2$ と $i+2$ 番目の固有表現タグは各位置で推定した結果をそのまま使用する. その例を図 2.8 に示す.

位置	$i-2$	$i-1$	i	$i+1$	$i+2$
入力文	首相	は	五	日	午前
品詞	名詞	助詞	名詞	名詞	名詞
文字種	漢字	平仮名	漢字	漢字	漢字
固有表現タグ	O	O	B-DATE	I-DATE	B-TIME

図 2.8 固有表現タグ推定に使用する素性

図 2.8 において, 学習時における単語”五”に関する事例は, 分類するクラスは B-DATE で素性は枠内の要素全てを使用する.

同様の文をテストデータとし, 固有表現抽出を行う際は, 単語”五”の素性は学習時と同様に枠内の要素全てを使用する. $i-2$ と $i-1$ 番目の固有表現タグはそれぞれ

の位置で SVM によって推定した結果をそのまま使用する.

2.3.3 実験

実験には CRL(郵政総合通信所) 固有表現データを使用した. CRL 固有表現データは毎日新聞 95 年版 1,174 記事, 約 11,000 文に対して IREX で定義された固有表現がタグ付けされたものである. データ中の固有表現の総数は 19262 個であった. 形態素解析には茶筌を使用し, 評価は CRL 固有表現データを 5 等分に分割し, 訓練 4, テスト 1 の比率で交差検定を行い, それらの総合の $F_{\beta=1}$ 値を使用した. また, 実験では Kernel 関数として,d 次の多項式関数 $K(x_i, x_j) = (x_i \dot{x}_j + 1)^d$ を使用した.

素性に違いによる精度の比較

Chunk タグを IOB2, 解析方向を右向き,Kernel 関数は 2 次の多項式関数に固定して, 使用する素性が以下に示す 4 種類の場合について抽出精度を調査した.

(1) 単語自身, 品詞大分類

(2) 単語自身, 品詞細分類

(3) 単語自身, 品詞大分類, 文字種

(4) 単語自身, 品詞細分類, 文字種

品詞大分類とは名詞, 動詞, 助詞などの分類で, 品詞細分類とは名詞-普通名詞, 動詞-自立-サ変などの細かい分類のことである. 文字種とはカタカナ, 平仮名, 漢字, 記号, 数字, アルファベットの 6 種類とし, 単語に含まれる文字種全てを素性とした. その結果, 素性空間の次元数の最も高い (4) について最良の結果が得られた. また,LOCATION,ORGANIZATION,PERSON の 3 つの固有表現は, 品詞細分類情報を使用することにより, 大幅な精度の向上が見られた.

Chunk タグと解析方向の違いによる精度の比較

素性を単語自身, 品詞細分類, および文字種に固定し,Chunk タグと解析方向の違いによる抽出精度を調査した.

その結果, 最良の精度は IOB2 の左向き解析の場合で 83.2d であった.IOB1 を除く全ての Chunk タグで, 右向きよりも左向き解析のほうがよい結果が得られた. 考えられる理

由として, 複数の単語からなる固有表現はその後方の単語によって種類が決定されることが多い.「野村証券」という固有名詞が組織名であることは「野村」という語ではなく「証券」という語に強く起因する. 本研究では解析方向で決定的な固有表現タグを推定するため左向き解析では「証券」の固有表現タグを先に推定しその推定結果を「野村」の固有表現タグ推定に素性として使用する. このため後方の単語の推定結果が固有表現全体の推定に大きく影響した.

Kernel の違いによる精度の比較

適用する Kernel 関数の次数 d を 1 から 4 まで変化させ, 素性の組み合わせを考慮した学習が固有表現抽出にどれだけ重要であるかを調査した. 用いた素性は, 単語自身, 品詞細分類, 文字種で,Chunk タグとして IOB2, 解析方向は左向きとして精度を測定した.

その結果 $d = 1$ (素性の組み合わせを考慮しない) 場合は考慮した場合と比べ大きく劣る.2 つの素性の組み合わせを考慮した 2 次の多項式の場合が最良の精度だった.3 次以上の多項式では訓練事例数に対し必要以上に素性空間の次元数が増加したため抽出精度が低下したと思われる.

関連研究との比較

IREX 固有表現タスクにおける他手法との比較を行う. 比較の対象としたのは, 内元ら [10] の最大エントロピー法と書き換え規則による手法と, 颯々野ら [7] による, 可変長文脈を考慮した手法である. その結果を表 2.2 に示す.

SVMs(山田)	ME(内元)	ME(颯々野)
83.01	80.17	82.8

表 2.2 固有表現抽出の精度比較

内元らの手法では, 素性として, 単語自身, 品詞, 文字種を使用している. 形態素解析は JUMAN [18] を使用し, 素性として使用する文脈長は前後 2 単語の固定長としている.Chunk タグは Start/End 法を使用し, 形態素解析の単語分割が固有表現の開始位置直

前, 及び終了直後で分割されない問題には, 誤り駆動による書き換え規則の自動抽出により対処している. 学習アルゴリズムには最大エントロピー法を用いて各単語に対して固有表現タグ付与確率を推定し, ビタビアルゴリズムにより文全体で最適な固有表現タグを推定する. さらに内元らの手法では低頻度素性の使用による過学習を回避するために頻度による素性選択を行っている. 訓練データには CRL 固有表現データ,IREX-NE 予備試験トレーニングデータ,IREX-NE 予備試験データの約 12,000 文を使用し, さらに精度向上のために固有表現に関する辞書情報を使用している. 最良の結果は IREX 本試験 GENERAL データに対して F 値で 80.17 と報告されている.

颯々野らは, 素性としては内元らと同様で, 形態素解析には BREAKFAST [19] を用いている. 素性として使用する文脈長は固有表現の長さに応じて可変長に拡張し,Chunk タグは Start/End 法と,IOB2 の両方について実験している. 学習アルゴリズムは最大エントロピー法と決定リストの 2 つについて実験を行っている. 最大エントロピー法を用いた場合, 内元らと同様の制約を用いてあらかじめ素性選択を行っている. 颯々野らの手法による最良の結果は,Start/End タグを使用し, 最大エントロピー法により学習を行った場合で,IREX 本試験 GENERAL データに対して F 値で 82.8 と報告されている. 山田らは IREX 本試験データを使用できなかったため, 交差検定を行った. 結果は最高の精度を得た前後 2 単語の単語自身, 品詞細分類, 文字種を素性とし,Chunk タグは IOB2, 二次の多項式 Kernel 関数で左向き解析を行った場合である. 山田らの手法の最低精度は 81.6 であり, 内元らの手法と同等以上に精度が期待できる. 総合での精度は 83.2 であった. また, 比較した 2 つの手法では, 過学習を回避するためにあらかじめ制約により, 低頻度の素性を排除し, 全体の素性数を制限している. このことは訓練データに対する精度を犠牲にしてテストデータの精度を上げることを意味している. 内元らの訓練データに対する精度は F 値で約 85 と報告している. 一方山田らの手法では, 素性数を減少させる制約を使用していない. 山田らの手法による訓練データに対する精度は F 値で約 99 であり, 訓練データに対してほぼ完全に固有表現を抽出できている.

2.3.4 本研究との関連

本研究では固有表現抽出タスクを応用し, 吉田らの手法を用いて教師無し学習によって自動的に統合された表から得られる知識に基づき, 表形式ではない一般の web 上のテ

キストから表の構成要素となる語を抽出することを目標としている. 固有表現抽出タスクにおいて, 現時点で最も良い精度を誇っている山田らの手法を応用することにより表の構成要素となる語を抽出する.

2.4 教師無し学習による単語クラスタリング

2.4.1 後置詞の標準化についての研究

鳥澤による, 後置詞の標準化についての研究 [11] において Expectation Maximization{ EM } アルゴリズムを用いて意味的な格フレームの学習及び, 単語のクラスタリングを行っている. これは Rooth [12] 及び,Hofmann and Puzicha [?] の手法を元にいくつかの拡張を施したものである. 鳥澤の手法では学習データとして, $\langle v, rel, n\rangle$ という形の 3 つ組素性を用いる. v は動詞であり, n は先行詞, rel は v と n との間の関係を表している. 鳥澤の研究では, rel のことを $cooccurrencerelation$ (共起関係) と呼び, それらはケースマーカーや非ケースマーカーなどを含む後置詞となる. $cooccurrencetriple$ と呼ばれる, $\langle v, rel, n\rangle$ の三つ組は 2 つに分割される. 一つ目は $\langle v, rel\rangle$ で, 二つ目は n である. 鳥澤は $\langle v, rel\rangle$ のことを $argumentposition$ と呼ぶ. ここで, $cooccurrencetriple$ $\langle v, rel, n\rangle$ の確率は次のように定義される.

$$P\left(\langle v, rel, n\rangle\right) =_{def} \sum_{a \in A} P\left(\langle v, rel\rangle | a\right) P\left(n|a\right) P\left(a\right)$$

a は共起のクラスを表している. 直感的には, 名詞 n の意味クラス及び $\langle v, rel\rangle$ で示された $argumentposition$ に一致する. ここで,A は $\{1, 2, 3, \cdots, k\}$ で表される k 個の整数の組であると仮定する. EM に基づいたクラスタリング手法では, 動詞 v , 先行詞 n , 共起関係 rel 及びクラス a の確率はそれぞれ $P\left(\langle v, rel\rangle | a\right)$, $P\left(\langle n|a\rangle\right)$, $P\left(a\right)$ の形で推定される.

ここで, 確率 $P\left(\langle n|a\rangle\right)$ は単語のクラス分類とみなすことができる. ここで確率 $P\left(\langle a|n\rangle\right)$ は $P\left(\langle n|a\rangle\right)$ からベイズの定理を適用することにより得られる. 確率 $P\left(\langle a|n\rangle\right)$ は n という語が出現したときにその語がクラス a に属する確率であると捉えることができる. 同様にして,$P\left(a|\left\langle v, rel\right\rangle\right)$ は $P\left(\langle v, rel\rangle | a\right)$ から求められる. しかし, 与えられたコーパス中にクラス a が存在しないため, $P\left(\langle v, rel\rangle | a\right)$, $P\left(\langle n|a\rangle\right)$ 及び $P\left(a\right)$ は簡単には推定を行え

ない. そこでEMに基づいたクラスタリング手法を用いて, 観測されていないデータ a の確率の推定を次のようなiterationの手順により行う. まずはじめに与えられたコーパス中に見られる, 共起関係のリストについて考える. 実際にはこれらは静的構文解析器による構文木のことである.

$$L = \langle \langle v_0, rel_0, n_0 \rangle, \langle v_1, rel_1, n_1 \rangle, \cdots, \langle v_m, rel_m, n_m \rangle \rangle$$

ここで,L の観測される可能性は次の式によって定義される.

$$\prod_{\langle v_i, rel_i, n_i \rangle \in L} P(\langle v, rel, n \rangle) = \prod_{\langle v_i, rel_i, n_i \rangle \in L} \left\{ \sum_{a \in A} P(\langle v_i, rel_i \rangle | a) P(n_i | a) P(a) \right\}$$

EMアルゴリズムは最尤推定法の一種であり, 尤度がより高くなるように,iteration中にパラメータ

$$\{P(\langle v, rel \rangle | a) | v \in V, rel \in Rel, a \in A\} \cup \{P(n|a) | n \in N, a \in A\} \cup \{P(a) | a \in A\}$$

を調整する.V は観測された全ての動詞の集合,N は観測された全ての名詞の集合,Rel は可能性のある関係の集合である.iterationは尤度がある閾値に到達する, もしくは, 決められたiterationの回数をこなしたときに終了する.

以上のようにして生成された単語クラスの例を表 2.3 に示す.

CLASS-869

word	classification
エアーニッポン	0.562
全日空	0.354
JAS	0.246
JAL	0.232
ノースウエスト	0.199
ユナイテッド	0.198

表 2.3 単語クラスの例

2.4.2 本研究との関連

本研究では鳥澤の手法によって生成された単語クラスを表の要素となる語をテキスト中から抽出する際の素性として加えることによって精度の向上を図る.

第 3 章

web 上のテキストからの情報抽出システム

3.1 処理の流れ

本システムの実装までの処理の流れをを図 3.1 に示す.

本研究では吉田ら [1] の手法を用いて教師無し学習によって自動的に統合された表から得られる知識に基づき, 表形式ではない一般の web 上のテキストから表の構成要素となる語を抽出する. 特に属性値の抽出を行い, 情報抽出の結果とする. なぜなら, 属性値及びその属性値の属するクラス名が判れば, その属性値の属する属性は一意に特定できるからである.

3.2 web 上の表のクラスタリング, 統合

吉田ら [1] の手法を用いて表のクラスタリング, よく似た事柄について書かれた表の統合を行う.

web 上からランダムに html を取得した. その総容量は 8GB で,html 中に含まれる表の数は 213600 個であった. これらの html をもとに, 吉田ら [1] の手法を用いて表のクラスタリング, 統合を行った.

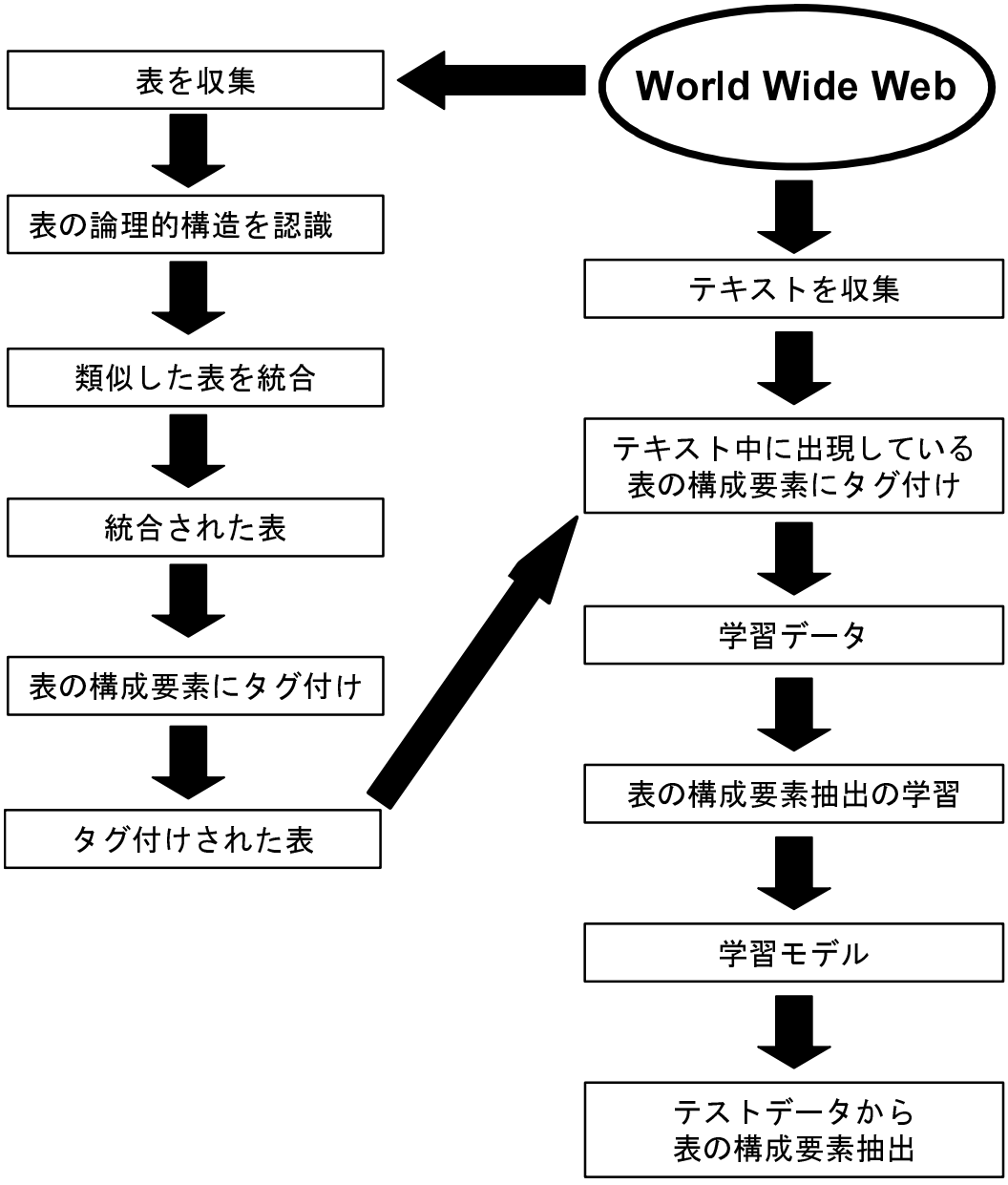

図 3.1 本研究における処理の流れ

3.2.1 表の定義

表とは属性とその属性に対応する属性値をもったものである. その例を図 3.2 に示す.

図 3.2 表の定義

3.3 表の構成要素へのタグの付与

まず, 表中の全てのアルファベット及び数字を半角文字に変換する. これは形態素解析器による形態素への分割方法が全角文字に対する場合と半角文字に対する場合とで異なるためである. 次に, 表の各構成要素の形態素解析を行う. 解析には JUMAN [18] を用いた. 表の属性となっている語には $\langle ATTRn \rangle$, また, その値となっている語に対しては $\langle VALn \rangle$ というタグを付与する. (n) また, 表の構成要素は単独の形態素で構成されている場合と複数の形態素から構成されている場合があり, 特に複数の形態素から構成されている場合, タグを付与する際に注目している形態素が構成要素のどの部分であるかを識別する必要がある. 以下に識別方法を示す.

- ある表の構成要素 (属性または属性値) が複数の形態素から構成される場合, その先頭の形態素には $\langle ATTRn \rangle$ 又は $\langle VALn \rangle$ の後に表の構成要素の開始位置を意味する BEGIN というタグを付与する.
  例: 6 ⟨VAL7⟩BEGIN 倍 ⟨VAL7⟩MID 速 ⟨VAL7⟩END

- ある表の構成要素 (属性または属性値) が複数の形態素から構成される場合, その末尾の形態素には $\langle ATTRn \rangle$ 又は $\langle VALn \rangle$ の後に表の構成要素の開始位置を意味する END というタグを付与する.
  例: $6\langle VAL7 \rangle BEGIN$ 倍 $\langle VAL7 \rangle MID$ 速 $\langle VAL7 \rangle END$

- ある表の構成要素 (属性または属性値) が複数の形態素から構成される場合, その中間の形態素には $\langle ATTRn \rangle$ 又は $\langle VALn \rangle$ の後に表の構成要素の開始位置を意味する BEGIN というタグを付与する.
  例: $6\langle VAL7 \rangle BEGIN$ 倍 $\langle VAL7 \rangle MID$ 速 $\langle VAL7 \rangle END$

- ある表の構成要素 (属性または属性値) が単独の形態素から構成される場合, その形態素には $\langle ATTRn \rangle$ 又は $\langle VALn \rangle$ の後に表の構成要素の終了位置を意味する END タグを付与する.
  例: $6\langle VAL7 \rangle BEGIN$ 倍 $\langle VAL7 \rangle MID$ 速 $\langle VAL7 \rangle END$

3.4 テキストへのタグ付与

前小節で述べた表の構成要素がテキスト中に出現している箇所にタグを付与する.

まず, 収集した web 上のテキストに対し, 前述の表の構成要素に対して行ったように, アルファベット及び数字を全て半角文字に変換した後, 形態素解析を行う. 解析には JUMAN [18] を用いた. 次に表の構成要素と同じ語が出現している箇所に同様のタグを付与する.

3.5 学習に用いる素性

本研究では表の構成要素の学習の際に用いる素性として, 以下のものを利用する.

- (1) 語彙
- (2) 読み
- (3) 標準形
- (4) 品詞細分類
- (5) 形態素の先頭 4 バイト
- (6) 形態素の末尾 4 バイト

- (7) 形態素の属する単語クラス

この他に, 語彙として数字が出現した際にはその語の標準形のカラムの数字を num に置き換え, 素性とした. これは, 例に示したように, 数字とアルファベットによって 1 つの形態素が構成されている場合, 数字の後に続く語が何らかの単位を表すことがしばしばあるため, これらの数字を “num” に置き換えたものを素性として加えることにより抽出を行う際に有効であると考える.

例） 128MB 128MB numMB

3.6 単語クラスの導入

本研究では, 表の構成要素の学習を行う際に, 素性として, 形態素の属する単語クラスを用いた. 単語クラスは新聞記事 33 年分をもとに, 前述の鳥澤 [11] の手法により, 生成されたものを用いる.⟨ クラス数は 2500 で, 単語数は 37638⟩ 導入例を図 3.1 に示す.

野球	やきゅう	野球	名詞	普通名詞	⟨ CLASS1063P0.5 ⟩
テニス	テニス	テニス	未定義語	カタカナ	⟨ CLASS1063P0.25 ⟩

表 3.1 単語クラスの導入例

3.7 表の構成要素抽出の学習

本研究では, 工藤 [6] により提供されている,Support Vector Machines に基づく汎用的な Chunker である **YamCha** を用いて表の構成要素抽出の学習を行う.Chunking($Chunk$ 同定問題) とは, 与えられた文を適当な解析単位に分割し, その分割した各要素に名前を付与することである.Chunking は自然言語処理において, 最も基本的な処理の 1 つとして認識されている. 文節切り, 英語の基本句同定 ($BasePhraseChunking$), 形態素解析, 分かち書き, 固有名詞抽出などが Chunking の範疇に入る.**YamCha** は Support Vector Machines を学習アルゴリズムとし, 統一的な枠組みでこれらの処理を行う. 解析精度に関しても,BaseNP

Chunking において 2001 年 6 月の時点で最も高い精度を示している. また 2000 年 9 月に行われた CoNLL2000 Shared Task, Chunking においては参加 11 チーム中 1 位の成績を収めている.

本研究における表の構成要素の抽出も Chunking の範疇に入ると考える. 本研究では前小節で定義した素性を用いて表の構成要素抽出の学習を行う.

解析方向は文頭から文末の方向に, 順に表の構成要素を表すタグを推定する右向き解析を用いた. 訓練事例は解析方向順に,i 番目の表の構成要素を表すタグを分類クラスとし, 素性は $i-2$ から $i+2$ までの単語の語彙, 読み, 標準形, 品詞細分類, 単語クラス, 先頭 4 バイト, 末尾 4 バイト,$i-2$ から $i-1$ の表の構成要素を表すタグを使用し生成する.

テスト事例は, 語彙, 読み, 標準形, 品詞細分類, 先頭 4 バイト, 末尾 4 バイト, 単語クラスに関しては既知であるが, 表の構成要素を表すタグに関しては未定であるため, 推定したタグを順次動的に追加し, 以降の解析の素性として利用する. 即ち, 解析方向順に i 番目の単語の素性は $i-2$ から $i+2$ までの単語の語彙, 読み, 標準形, 品詞細分類, 単語クラス, 先頭 4 バイト, 末尾 4 バイト, $i-2$ から $i-1$ の表の構成要素を表すタグは,$i-2$ から $i-1$ 番目の解析で推定したタグを使用する. 図 3.2 に, 入力文“ CPU は Celeron800MHz を ”に対して右向き解析した場合の 3 番目の単語“ Celeron ”の素性を示す.

位置	語彙	$\cdots$	表の要素タグ
1	CPU	$\cdots$	$\langle ATTR2 \rangle$ END
2	は	$\cdots$	$\langle$o$\rangle$
3	Celeron	$\cdots$	$\langle$VAL2$\rangle$END
4	800MHz	$\cdots$	
5	で	$\cdots$	

表 3.2 学習に使用する素性

入力文が訓練データの場合, 表の構成要素を表すタグ $\langle$ VAL2 $\rangle$END が分類するクラスであり, 素性として,1 から 5 までの語彙, 読み, 標準形, 品詞細分類, 単語クラス, 先頭 4 バイト, 末尾 4 バイト,1 から 2 の表の構成要素を表すタグを使用する. 同一の文がテストデータであれば,3 番目の表の構成要素を表すタグを推定するために,1 から 5 までの語彙,

読み, 標準形, 品詞細分類, 単語クラス, 先頭 4 バイト, 末尾 4 バイト,1 から 2 番目で推定した表の構成要素を表すタグを素性とする.

第 4 章

実験

本節では, 表の構成要素の抽出実験の方法について述べる. 本研究では特に, 属性値の抽出を行い, その精度を測定する. なぜなら, 属性値及びその属性値の属するクラス名が判れば, その属性値の属する属性は一意に特定できるからである. 本研究では, まず, 単語クラスを素性とすることの有用性を調べるために自己紹介のトピックにおいて単語クラスを導入しない場合の抽出を行った.

次に単語クラスを導入した上で,

- 「自己紹介」

- 「PC のスペック」

の 2 つのトピックに関する表の構成要素をテキストから抽出する実験を行った. Support Vector Machines による学習時に用いる Kernel は Polynomial Kernel で, 次数は 2 次,Slack 変数は 1 に設定した.Chunking の解析方向は右向き, 考慮する素性は前後 2 単語とした.

4.1 自己紹介に関するトピック

4.1.1 実験に用いるデータ

実験に用いたデータは以下の 3 つである.

1. web 上からランダムに取得した 213600 個の表を含む総容量 8GB の html をもとに, 吉田ら [1] の手法を用いて複数の表をクラスタリング, 統合した結果のうち, 自己紹介に関するもの. この表は 499 個の異なる表を統合したものである.
2. 新聞記事 33 年分をもとに, 前述の鳥澤 [11] の手法により生成された単語クラス ⟨ クラス数 2500, 単語数 37638⟩
3. テキストからの表の構成要素抽出のためにサーチエンジン google を用いて, 次のような検索式で自己紹介に関する記事を検索し, 上位 500 件の記事を収集し, そのうち 470 記事を訓練データとした. テストデータとして残りの 30 記事の中から, 表の属性値となる語が 1 語しか含まれていない記事を除いた 28 記事を採用した.
「プロフィール 名前 趣味 血液型 年齢 出身」

4.1.2 分類すべきクラス

吉田ら [1] の手法により統合された自己紹介に関する表は図 4.1 に示す 10 の属性及びその属性値から構成されている. この表の属性はその属性値を要素とするクラスのラベルであるとみなす事ができる.

図 4.1 のように key 属性を除いて表の左から順にクラス名を n ($2 \leq n \leq 11$; n は整数) とすると, それぞれのクラスの属性を表現するタグは

$$\langle ATTR2\rangle \cdots \langle ATTR11\rangle$$

で表現できる. 分類すべきタグの種類はこれらの属性及びその属性値

$$\langle ATTR2\rangle \cdots \langle ATTR11\rangle$$

及び

$$\langle VAL2\rangle \cdots \langle VAL11\rangle$$

である.

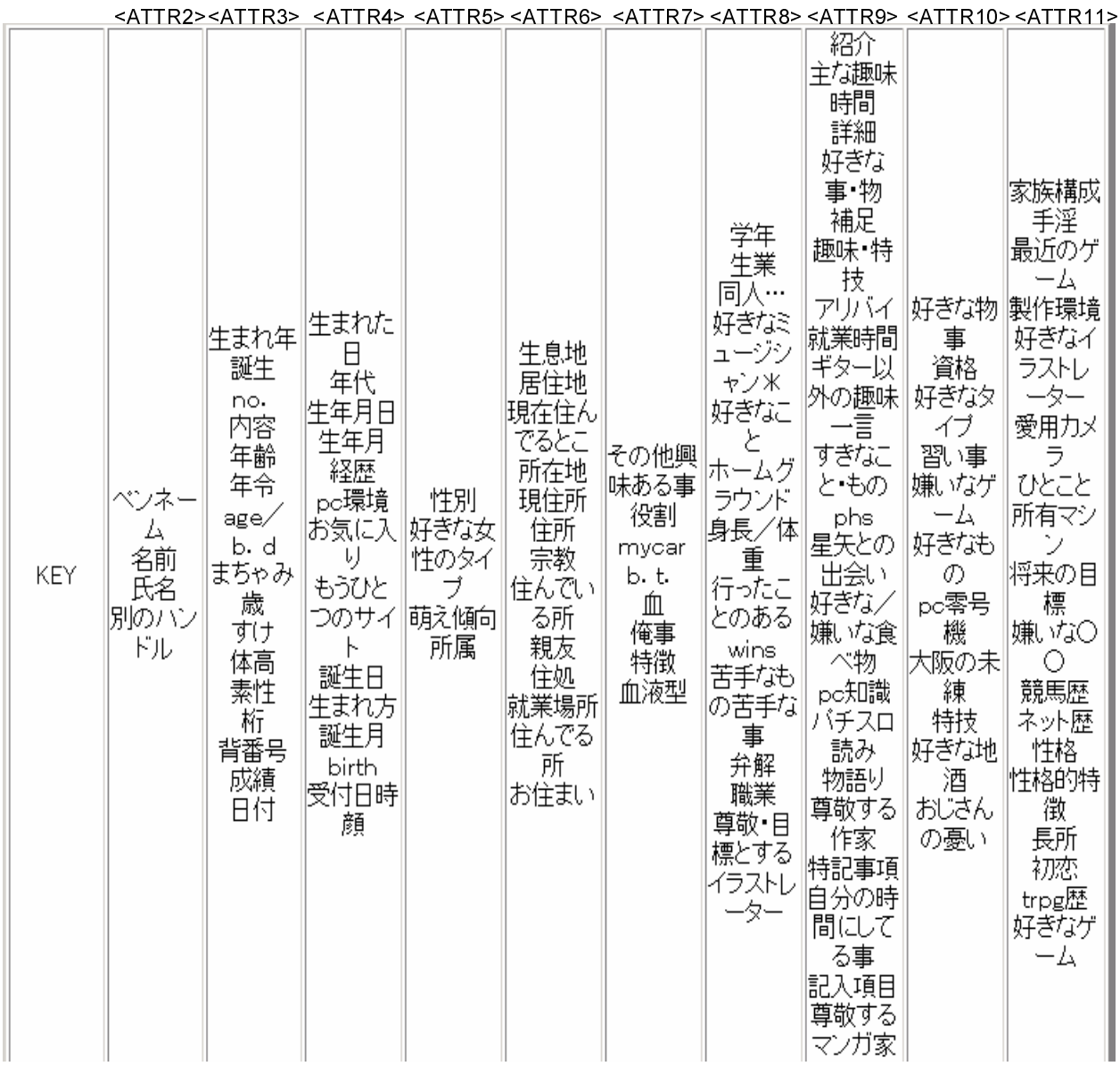

図 4.1 自己紹介に関する表の属性

4.2 PCのスペックに関するトピック

4.2.1 実験に用いたデータ

実験に用いたデータは以下の 3 つである.

1. web 上からランダムに取得した 213600 個の表を含む総容量 8GB の html をもとに, 吉田ら [1] の手法を用いて複数の表をクラスタリング, 統合した結果のうち,PC のスペックに関する表. この表は 264 個の異なる表を統合したものである.
2. 新聞記事 33 年分をもとに, 前述の鳥澤 [11] の手法により生成された単語クラス ⟨ クラス数 2500, 単語数 37638⟩
3. テキストからの表の構成要素抽出のためにサーチエンジン google を用いて, 次のような検索式で自己紹介に関する記事を検索し, それぞれの検索結果の上位 250 件の計 1000 記事を収集し, そのうち 970 件を訓練データ, テストデータとして残りの 30 記事の中から, 表の属性値となる語が 1 語しか含まれていない記事を除いた 23 記事を採用した.
   (a) PC のスペックに関する記事の検索式は以下の通りである. それぞれの検索結果の上位 250 記事, 計 1000 記事を収集した.
      i. 「スペックは PC」
      ii. 「スペックは CPU」
      iii. 「スペックは HDD」
      iv. 「スペックは メモリ」

4.2.2 分類すべきクラス

吉田ら [1] の手法により統合された PC のスペックに関する表は図 4.2 に示す 10 の属性及びその属性値から構成されている. この表の属性はその属性値を要素とするクラスのラベルであるとみなす事ができる.

図 4.2 のように key 属性を除いて表の左から順にクラス名を n ($2 \leq n \leq 11$; n は整数) とすると, それぞれのクラスの属性を表現するタグは

$$\langle ATTR2 \rangle \cdots \langle ATTR11 \rangle$$

で表現できる. 分類すべきタグの種類はこれらの属性及びその属性値

$$\langle ATTR2\rangle \cdots \langle ATTR11\rangle$$

及び

$$\langle VAL2\rangle \cdots \langle VAL11\rangle$$

である.

	<ATTR2>	<ATTR3>	<ATTR4>	<ATTR5>	<ATTR6>	<ATTR7>	<ATTR8>	<ATTR9>	<ATTR10>	<ATTR11>
KEY	実家用マシン コンピュータ本体 cpu／clock pentium ii0mhz pcm dos／v機 cpu cpucpuクーラー	pcカードスロット mother メモリースロット motherboard m／b 入力方式 使用マシン m／bchipset コンピュータ名 マザー マザーボード マザーボード mb	メモリ memry mainmemory cpuの数 メモリ mem メモリー メインメモリ パソコン本体 メインメモリー memory メモリ容量 memolyram	ハードディスク アドレシング 内臓ハードディスク x?window コメント harddisk system?hdd hd hddhdd クーラー センサ(ジャンパ切り外し可) 動画入力ボード ディスク hdd	ビデオカード ビデオ ビデオボード software グラフィックカード グラフィック graphics グラフィックボード ビデオメモリ videocard ディスプレイアダプタ ビデオチップ ビデオコントローラ vram vga motherbord graphics card ga	c／d cdrom cd－rw cd?rom ドライブ 出力 cd?rom cd－rom ドライブ cd?r／rw drive cd?rom dvd?rom cdドライブ cd?rom 情報 ドライブ cd?rom?d 周辺機器 othereq uipment cd?rdrive 画像編集 cd－r／w cd－r	flopy disk 昔のpc フロッピーディスクドライブ インターフェイス fd フロッピー fdd teacfd?0hg idehdd フロッピードライブ	オーディオ機能 carddrive オプション サウンドボード soundboard サウンドカード 音源ボード aux soundcard pci0:sound 編集機材 サウンド sound?board 音源カード sound	ソフト sequenser turobolinux0.0 preload software os 使用os 導入理由 お絵かきソフト extension	monitor display ディスクプレイ モニター モニタ ディスプレイ crt ディスプレー ディスプレイモニタ resolution moniter グラフィックアダプタ ディズプレイ

図 4.2 PC スペックに関する表の属性

4.3 評価基準

抽出実験は学習データとは異なるテストデータに対して抽出を行うオープンテストとする. 以下の評価基準に基づいて抽出実験の評価とした.

- 属性値及びその属性値の属するクラス名が判れば, その属性値の属する属性は一意に特定できるため, テストにおいては, テキストから表の構成要素となるべき属性値がふさわしいクラスとして（正しく）抽出できたかどうかを判定することにより, 本研究における情報抽出の精度とする.

- 統合された表の構成要素に特殊記号が含まれているため, テキストに対してタグ付けを行う際に, 特殊記号に対しても表の構成要素を示すタグが付与されている. 評価を行うにあたり, テストを行った際に “\”, “?”, “.” 及び “:” などの特殊記号に対して表の構成要素を示すタグが推定された場合, 判定の対象から除いた.

- 数字は複数の属性に対する属性値となる可能性があるため, 数字のみから構成される形態素が含まれる表の構成要素に関しては評価の対象から除いた.

- 図 4.2 及び図 4.1 に出現しているキー属性を除く属性のうち, その属性値の表中での出現頻度が非常に低いものについては, 属性から排除した. 以下に示す属性に対する属性値と判定された場合に正解とみなす.

自己紹介に関するトピックにおいて採用した属性を示す

- ⟨ATTR2 ⟩
  ペンネーム 名前 氏名 別のハンドル

- ⟨ATTR3 ⟩
  生まれ年 誕生 年齢 年令 age/b.d 歳 背番号 成績 日付

- ⟨ATTR4 ⟩
  生まれた日 年代 生年月日 生年月 経歴 誕生日 誕生月 birth 受付日時

- ⟨ATTR5 ⟩
  性別 好きな女性のタイプ

- ⟨ATTR6 ⟩
  生息地 居住地 現在住んでるとこ 所在地 現住所 住所 宗教 住んでいる所 住処 就業場所 住んでる所 お住まい

- ⟨ATTR7 ⟩
  その他興味ある事 役割 mycar 血 特徴 血液型

- ⟨ATTR8 ⟩
  学年 生業 好きなミュージシャン 好きなこと ホームグラウンド 苦手なもの苦手な事 職業

- ⟨ATTR9 ⟩
  紹介 主な趣味 好きな事・物 趣味・特技 すきなこと・もの 好きな/嫌いな食べ物 尊敬する作家 特記事項 自分の時間にしてる事 尊敬するマンガ家さん 趣味

- ⟨ATTR10 ⟩
  好きな物事 資格 好きなタイプ 習い事 好きなもの 特技 好きな地酒

- ⟨ATTR11 ⟩
  家族構成 最近のゲーム ひとこと 将来の目標 嫌いなもの 競馬歴 性格 性格的特徴 長所 初恋 好きなゲーム

PC のスペックに関するトピックにおいて採用した属性を示す

- ⟨ATTR2 ⟩
  CPU CPU/CLOCK

- ⟨ATTR3 ⟩
  mother motherboard m/b m/bchipset マザー マザーボード mb

- ⟨ATTR4 ⟩
  メモリ mainmemory mem メモリー メインメモリ メインメモリー memory メモリ容量 ram

- ⟨ATTR5 ⟩
  ハードディスク 内臓ハ-ドディスク harddisk hdd hd ディスク

- ⟨ATTR6 ⟩
  ビデオカード ビデオ ビデオボード グラフィックカード グラフィック graphics グラフィックボード ビデオメモリ videocard ディスプレイアダプタ ビデオチップ ビデオコントローラ vram vga graphicscard

- ⟨ATTR7 ⟩
  c/d cdrom cd-rw cdrom ドライブ cdrom cd-rom ドライブ cdr/rwdrive cdromdvdrom cd ドライブ cdrom 情報 ドライブ otherequipment cdrdrive cd-r/w cd-r

- ⟨ATTR8 ⟩
  flopydisk 昔の pc フロッピーディスクドライブ インターフェイス fd フロッピー fdd idehdd フロッピードライブ

- ⟨ATTR9 ⟩
  オーディオ機能 carddrive サウンドボード soundboard サウンドカード 音源ボード aux soundcard pci0:sound 編集機材 サウンド 音源カード sound

- ⟨ATTR10 ⟩
  ソフト sequenser turobolinux preloadsoftware os 使用 os お絵かきソフト

- ⟨ATTR11 ⟩
  monitor display ディスクプレイ モニター モニタ ディスプレイ crt ディスプレー

ディスプレイモニタ resolution moniter ディズプレイ

4.4 実験結果

4.4.1 自己紹介に関する抽出結果 (単語クラス非導入)

自己紹介のトピックに関して, 素性として単語クラスを導入しなかった場合の抽出精度を表 4.1 に示す.

Precision	0.8571
Recall	0.3803
$F_{\beta=1}$measure	0.5268

表 4.1 単語クラス非導入での自己紹介に関する抽出結果

4.4.2 自己紹介に関する抽出結果

自己紹介に関する抽出結果を表 4.2 に示す.

Precision	0.8857
Recall	0.3865
$F_{\beta=1}$measure	0.5381

表 4.2 自己紹介に関する抽出結果

4.4.3 PC のスペックに関する抽出結果

PC スペックに関する抽出結果を表 4.3 に示す.

Precision	0.4135
Recall	0.7237
$F_{\beta=1}$measure	0.5263

表 4.3 PC スペックに関する抽出結果

第 5 章

考察

5.1 単語クラスの導入に関する考察

表 4.1 の単語クラスを素性として導入しない場合の結果と表 4.2 の単語クラスを素性として導入した結果との間で精度を比較すると, 単語クラスを導入したほうが僅かではあるが,recall,precision 共に向上している. これは単語クラスを未導入だった際に表の属性値ではないと判定されていた語が, 単語クラスを導入したことにより, 正しく抽出されたことによるものであった.

5.2 自己紹介のトピックに関する考察

自己紹介のトピックにおける抽出結果は recall が 38.65 と低い値になっている. これは, 名前や現住所など, その属性値となる候補数が膨大である場合に抽出がなされていないためである. また名前の属性値として抽出されるべき語である,「ミツヒコ」という語があった場合,JUMAN による形態素解析結果において, 品詞細分類は「人名」となるべきであるが, それが「未定義語」として定義されることも, 人名の抽出がうまくいかない原因のひとつではないかと考える.

precision が 88.57 と高いのは, 次の 2 つの要因によると考える.

- (1) 今回の評価では, 数字だけで 1 つの形態素を構成しており, 属性値と推定されているものは, 評価の対象としなかった

- （2）性別や血液型のように, 学習データ中にその属性に対する候補が全て出ている場合にほぼ全て抽出に成功している

5.3 PC スペックのトピックに関する考察

PC スペックのトピックにおける抽出結果は recall が 72.37 と比較的高い精度が出ている. 今回の評価基準では数値が単独で形態素を構成しているものが表の属性値となっている場合評価の対象外としたため何の値かわからないものは入っていない. “128MB” のように, 数字とアルファベットから形態素が構成されている場合, 本手法の前処理の過程で標準形の素性における数字を “num” に置き換えたことにより, 学習データに出現していないが, 表の構成要素となるべき属性値の抽出が正しく行われていた.

precision が低かった原因として考えられるのが,PC スペックに関するテストデータ記事の中に属性 ⟨ATTR10 ⟩（ OS やソフトウエアに関する属性 ）に対する属性値である,“Linux” という語が頻繁に出現し, 人がテストデータを判断しすると,PC スペックに関するものではなく,“Linux マガジン”,“Linux today” のように, このケースで抽出対象となる, ソフトウエアとしての “Linux” ではなく, 別の固有名詞に対する語の一部としての “Linux” を誤って認識してしまったケースが多かったためである.

第 6 章

結論

本稿では,web 上に存在する多数の表から得られる知識を用いて,web 上のテキストから表の構成要素となりうる属性値を抽出することにより, 幅広いトピックに適用可能かつ手作業による抽出パターン作成を必要としない情報抽出の手法を提案した.

従来の情報抽出の手法では, 抽出したいトピックごとに手作業により抽出パターンを作成して抽出を行ってきた. この手法ではパターンの作成に多大な労力がかかる上, 限られたトピックにしか対応できない.

人間はあるオブジェクトについての表を作成する際に, 常識に基づいてそのオブジェクトの記述に必要な情報だけを選んでいる.web 上にはこのように人間がなんらかを意図して作成した表が多数存在している. 本研究ではこの点に着目し,web 上に存在する不特定多数の表を収集し, 表をクラスタリング, 統合したものを入力データとして用い, それらの表の構成要素にタグ付けを行い,web 上に存在する通常テキスト中に表の構成要素と同じ記述があった場合に同様のタグを付与したものを学習データとみなし, 表の構成要素の推定の学習を行い, そのモデルをもとに通常のテキストから表の構成要素を抽出する実験を行った.PC スペックと自己紹介のトピックについて抽出実験を行い, それぞれ F 値で 56.0 及び 53.8 という精度を得た.

最後に今後の課題として, 以下の 5 つを挙げる.

1. さらに多数の表をもとに, 表の構成要素を収集,
2. 学習データとして用いる web 上のテキストの量を増やす
3. 単語クラスの適用方法を改める

4. 学習に用いる素性の組み合わせを検討
5. 今回評価対象から除いた “数字のみから構成される形態素が含まれる表の構成要素” の抽出を可能とするためのアルゴリズムの検討

謝辞

本研究を進めるにあたり, 日頃から方針, 内容について熱心にご指導を賜りました鳥澤健太郎助教授に厚く御礼申し上げます.

そして, 機械学習アルゴリズムについて細やかなご助言を頂いた自然言語処理学講座の山田寛康助手,web 上の表の統合に関して様々なアドバイスをいただきました東京大学大学院情報理工学系研究科の吉田稔氏に深く感謝いたします.

本研究にご理解と多大なご協力を賜りました, 東条敏教授をはじめとする知識工学講座の皆様に感謝いたします.

参考文献

[1] Minoru Yoshida, Kentaro Torisawa and Jun'ichi Tsujii. (2001). Extracting ontologies from World Wide Web via HTML tables. In the Proceedings of the Pacific Association for Computational Linguistics (PACLING 2001). pp. 332-341.

[2] Minoru Yoshida, Kentaro Torisawa and Jun'ichi Tsujii. (2001). A method to integrate tables of the World Wide Web. In the Proceedings of the first International Workshop on Web Document Analysis (WDA 2001). pp. 31-34.

[3] Minoru Yoshida. (2002). Extracting Attributes and Their Values from Web Pages. In the Proceedings of the ACL 2002 Student Research Workshop. pp. 72–77.

[4] 山田 寛康, 工藤 拓, 松本裕治 (2002) Support Vector Machine を用いた日本語固有表現抽出 情報処理学会論文誌 , Vol 43, No. 1, pp.43-53

[5] 工藤 拓, 松本 裕治 (2002) Support Vector Machine を用いた Chunk 同定, 自然言語処理, Vol.9, No, 5 pp 3-22

[6] 工藤 拓. YamCha: Yet Another Multipurpose CHunk Annotator . http://cl.aist-nara.ac.jp/ taku-ku/software/yamcha/

[7] Manabu Sassano and Takehito Utsuro (2000). "Named Entity Chunking Techniques in Supervised Learning for Japanese Named Entity Recognition," COLING 2000, pp. 705 - 711.

[8] Satoshi Sekine, Ralph Grishman and Hiroyuki Shinnou. A Decision Tree Method for Finding and Classifying Names in Japanese Texts In Proceedings of the the Sixth Workshop on Very Large Corpora 1998.

[9] 関根 聡. テキストからの情報抽出 情報処理,Vol40,No.4,pp370-373, 情報処理学会 1999.

[10] 最大エントロピーモデルと書き換え規則に基づく固有表現抽出内元 清貴, 馬 青, 村田 真樹, 小作 浩美, 内山 将夫, 井佐原 均 自然言語処理 (言語処理学会誌) 2000 年 4 月,7 巻,2 号, p.63 〜 p.90

[11] Kentaro Torisawa. An Unsupervised Method for Canonicalization of Japanese Postpositions in Proceedings of the 6th Natural Language Processing Pacific Rim Symposium (NLPRS 2001), pp. 211-218, December, 2001.

[12] Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. Inducing a Semantically Annotated Lexicon via EM-Based Clustering. In 37th Annual Meeting of the ACL, 1999

[13] Thomas Hofmann and Jan Puzicha. Unsupervised Learning from Dyadic Data. Technical Report ICSI-TR-98-042, International Computer Science Institue (ICSI), Berkeley. December, 1998.

[14] MUC Homepage, http://www.muc.saic.com

[15] IREX Homepage, http://nlp.cs.nyu.edu/irex/index-j.html, 1999.

[16] H.Chen,S.Tsai,and J.Tsai. Mining tables from large scale HTML texts. 18th International Conference on Computational Linguistics (COLING) pages 166-172,2000.

[17] Erik F.Tjong Kim Sang and Jorn Veenstra. Representing test chunks. in Proceedings of EACL'99, pp. 173-179, 1999.

[18] 松本 裕治, 黒橋 禎夫, 妙木 裕, 新保 仁, 長尾 眞, "利用者定義可能な日本語形態素解析システム JUMAN 使用説明書", 京都大学工学部長尾研究室, 1991.

[19] 颯々野学, 斉藤由香梨, 松井くにお. アプリケーションのための日本語形態素解析システム, 言語処理学会第 3 回年次大会論文集,pp.441-444 1997.