自然言語処理 140−10
(2000. 11. 21)

固有表現抽出のための可読性の高い規則の自動生成

磯崎 秀樹
NTT コミュニケーション科学基礎研究所
〒691-0237 京都府相楽郡精華町光台2 − 4
isozaki@cslab.kecl.ntt.co.jp

あらまし　固有表現抽出は、地名や人名などの固有表現を文書から抜き出す処理であり、情報抽出や質問応答の重要な基礎技術である。英語ではすでに人間なみの精度で抽出する技術が確立しているが、日本語では、単語が分かち書きされていないことや、固有表現を大文字で始める習慣がないなどの理由により、著者が知る限りでは十分な精度がまだ達成できていない。人手で辞書や規則を作成していく方法とHMMや最大エントロピーモデルなどの統計的手法が有力だが、本稿では、人手による修正の容易な規則を正解コーパスから自動生成する方法を提案する。実験により、IREXで1位のシステムを超える成績を出すことができた。
キーワード　情報抽出、固有表現抽出、機械学習

Automatic Generation of Readable Rules for Named Entity Recognition

Hideki Isozaki
NTT Communication Science Laboratories
2-4 Hikari-dai, Seika-cho, Soraku-gun, Kyoto, 619-0237

Abstract　Named entity recognition is a task that extracts names of locations, persons, etc. from documents. It is a key technology for information extraction and question answering. Named entity recognition technology for English is already established. Japanese sentences, however, are not divided into words, and proper names are not capitalized. Hence, it is difficult to extract named entities with sufficient accuracy. Conventional systems use hand-crafted rules or statistical methods such as HMM. Here, we propose an alternative method that generates readable rules from a training corpus. Our experiments show that our system outperforms the first ranked system in IREX.
Keywords　information extraction, named entity recognition, machine learning

1 はじめに

固有表現（ＮＥ）抽出は、地名・人名・組織名・日時などの文章中の重要な要素（固有表現）を文書から抜き出す技術であり、大量の文書情報からの情報抽出 [1] や、質問応答システム [8, 6] などを構成するための基礎技術として重要である。英語では高精度で抽出する技術が確立しているが、日本語では、大文字のようなヒントがないため、一般的表現との区別が難しいことや、未知語の近辺での形態素解析の失敗により、著者の知る限り、まだ十分な精度が得られていない。

固有表現抽出を行う方法としては、規則を人手で書いていく方法 [4] と HMM や最大エントロピーモデルなどの統計的手法が使われる [11]。人手で書いていく手法は、加えた修正が意図しない悪影響を及ぼすことがあり、高度な推論能力と多大な労力を必要とする。一方、統計的手法は一般に大規模な正解コーパスを作成するのに手間と時間がかかる。しかし、正解コーパスを作る作業は、規則を管理する作業に比べれば簡単である。これまでに提案されている学習手法 [3, 11] は、学習結果が数値データで出力との因果関係が明確でないため、人間が修正を加えることは難しい。

本稿では、正解コーパスから、人間の理解できる固有表現抽出規則を自動生成する方法（[14] の改良）を提案する。提案手法では、図１に示すように各規則が独立に実行され、各規則の守備範囲が明確なため、システム全体の挙動が理解しやすく、不正解になった箇所はその原因を簡単につきとめることができる。著者の行った実験では、形態素解析に chasen2.02 を用い、正解コーパスとして 11.6 万のＮＥ（CRL 固有表現データの約６倍）を含む新聞記事を用いたところ、生成された規則群は、人手による修正を行わなくても、IREX 固有表現抽出タスク (以下 IREX-NE と略記)[3] 本試験で１位のシステム（辞書と規則を用いた人手によるシステム）の精度を超えることができた。

以下では、まず、規則の適用について説明し、次に規則の生成について説明する。その後、実験結果を示す。

2 規則の適用

本システムでは、図１に示すように各規則が「パターン」と呼ぶ部分と「付加制約」という部分から構成される。各パターンは互いに独立に文書に適用され、パターンにマッチした部分がＮＥの１次候補となる。この処理を１次選抜と呼ぶ。１次候補のうち、付加制約を満たしたものだけが２次候補となる。この処理を２次選抜と呼ぶ。こうして得られた全規則の２次候補同士が重ならないように、各候補の優先度を考慮して絞り込んだものがシステムの出力となる。

2.1 １次選抜

１次選抜で用いられるパターンは、ＮＥがどのような単語列から構成されているべきかを指定する。各単語に関する条件を以下の３つ組で表す。

（単語の出現形，構成文字種，品詞）

ここで構成文字種はその単語を構成する文字種を表し、次のように分類する。記号、ひらがな、単漢字、複数漢字、単カタカナ、複数カタカナ、ギリシャ文字、単大文字（例：Ａ）、複数大文字（例：ＡＢＣ）、小文字、先頭だけ大文字（例：Ａｂｃ）、１万未満の小さな整数、浮動小数、大きな整数、数字ラベル（０１２０のように数値の表記にはあまり用いないパターンの数字）、その他の英数、その他。単漢字は形態素解析の失敗時に、単一のカタカナは組織などの略称（例：ア社）に、単一の大文字は人名の略称（例：Ｒ．スミス）に、それぞれ発生しやすく、特徴的なので、２字以上の場合と分けた。複数大文字は組織名に、先頭だけ大文字は人名に用いられる傾向がある。そこでこれらを区別した。また、１万未満の整数は日時や割合の表記に、大きな整数は金額の表記に用いられる傾向があるので分けた。

漢字の固有名詞であれば何でもよいという条件は、以下のように表すことができる。

（＿，漢字，固有名詞）

最初の「＿」は Prolog の匿名変数に相当する記号で、単語の出現形が何でもよいことを示している。このような単語に関するパターンを以下では「単語パターン」と呼ぶ。

この記法を用いると、「岡山県倉敷市」のような地名のパターンは、次のように書ける。

パターン 245 地名
（＿，漢字，固有名詞）（県，漢字，接尾語）
（＿，漢字，固有名詞）（市，漢字，接尾語）

システムはこのようなパターンが与えられると、それが文書中のどの部分にマッチするかを調べ、１次候補として出力する。

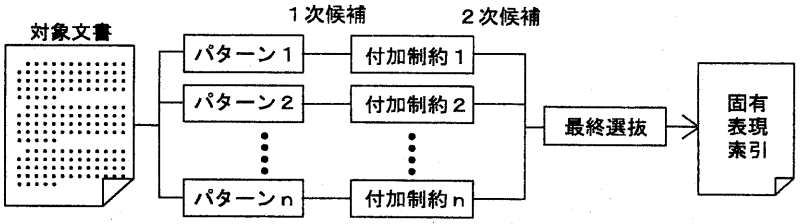

図 1: 提案システムの構成

ただし、日本語特有の問題として、単語分割の問題がある。たとえば、「新大阪府知事」という文字列の中の「大阪府」を地名として抽出したいが、形態素解析システムが「新大阪」と「府知事」に分けてしまうという場合である。最初の単語の先頭から 1 文字、最後の単語の末尾から 2 文字は不要なので削除したい。そこで、パターンの形式を拡張し、以下のように先頭と末尾の削除文字数を指定できるようにする。

パターン 319 地名 1,2

（新大阪，漢字，固有名詞）（府知事，漢字，一般名詞）

削除文字数は、1 次候補の段階では利用されず、最終出力を決定するときに利用される。

2.2 2 次選抜

2 次選抜では、付加制約を利用して、1 次候補の中から可能性の低そうな候補を取り除く。たとえば、次のようなパターンがあったとする。

パターン 701 地名 0,0（__，カタカナ，未知語）

これは、カタカナ未知語を地名とするパターンである。しかし、直後に「監督」や「社長」などがあれば、このカタカナ語は人名か組織名の可能性が高く、地名である可能性は低い。2 次選抜では、候補自身とその周囲に関してあらかじめ与えられた付加制約を調べる。候補を含む文の単語列を $(w_0, c_0, p_0), \ldots, (w_m, c_m, p_m)$ とし、そのうち、候補を構成する単語列を $(w_b, c_b, p_b), \ldots, (w_e, c_e, p_e)$ とする。ただし、w_i は単語の出現形、c_i は構成文字種、p_i は品詞である。

上記のパターン 701 にたとえば次のような付加制約があれば、可能性の低い候補が排除できる。

制約 701

$w_{e+1} \neq$ '監督' and $w_{e+1} \neq$ '社長'

図 2: 最長一致に基づく候補の最終選抜

ルール	田中太郎賞選考委理事中野元氏	
人名 (姓)	田中	中野
人名 (名)	太郎	元
人名 (姓名)	田中太郎	**中野元**
地名		中野
人工物名	田中太郎賞	
組織名	**田中太郎賞選考委**	

2.3 最終選抜

最終選抜では、重なりのある 2 次候補同士を調整し、出力する候補を決定する。選択にあたっては、図 2 に示すように最長一致を基本とする優先順位を利用する [5, 14]。「田中太郎賞選考委理事中野元氏」という文に対して、「田中」「田中太郎」「太郎」等の人名、「田中太郎賞」という人工物名、「田中太郎賞選考委」という組織名、「中野」という地名が 2 次候補として残っているとしよう。すると最長一致により、「田中太郎賞選考委」が選出され、これと重なる「田中」などの候補は出力されないことが決定される。次に、出力の決定した「田中太郎賞選考委」と重ならない候補の中から最長一致により「中野元」が選出される。

しかし、文が「中野」だけで、地名の「中野」と人名の「中野」の両候補がともに 2 次候補として残っている場合、長さだけでは判断できない。この場合、正解コーパス中での正解数の多い方を優先する。つまり、NE の開始位置を `tb`、終了位置を `te`、正解数を `freq` としたとき、(`tb`, −`te`, −`freq`) の辞書式順序が最小となる候補を選ぶ。

3 規則の生成

以上説明した枠組みのための規則の自動生成方法を説明する。以下の自動生成方法は一見単純であるが、以前に人手で作成したとき [5] に得られた

以下のような知見に基づいている。

- 規則を一般化するとprecisionが下がる。precisionを上げるためには規則に複雑な制約を付加しなければならなくなり、保守が困難になる。
- 同じ種類（たとえば地名）に分類されていても、実際には様々なもの（住所表示、施設名、地方名など）が含まれており、ひとまとめにして考えると問題が生じる。
- 最長一致による選択は間違いが少ない。

まず、パターンは以下で説明する一般化手続きに従い、各正解から1つずつ作られる。この一般化手続きは、「日本語では、被修飾語が修飾語よりあとに来るので、最後の単語が重要である」という事実にに基づいてデザインされたものである。また、各正解に対応するパターンをひとつだけにすることで、各パターンの分担範囲を明確にすることができる。

まず、正解コーパスからNEタグを削除したものを形態素解析して、各文ごとに単語列を作る。

$$(w_0,c_0,p_0)(w_1,c_1,p_1)...(w_m,c_m,p_m)$$

ここでw_iはi番目の単語の出現形、c_iはその構成文字種、p_iは品詞である。この結果とNEタグの位置を比較して各NEをちょうどカバーするリストを取り出す。

たとえば、「彼は<PERSON>中野次郎</PERSON>に会った。」という文なら、「(中野, 漢字, 人名・姓)(次郎, 漢字, 人名・名)」というリストが、「それは<LOCATION>奈良市</LOCATION>内で買った。」なら、「(奈良, 漢字, 地名)(市内, 漢字, 一般名詞)」が、「彼は新<LOCATION>大阪府</LOCATION>知事に会った。」なら、「(新大阪, 漢字, 地名)(府知事, 漢字, 一般名詞)」が得られる。

得られたリストに削除すべき前後の文字数の情報を付加し、各要素を以下の単語パターンで置き換えたパターンを作る。

1. その単語がNEの最初あるいは最後の単語で、その一部が削除される場合はそのまま。
2. その単語がNEの最後の単語（1語の場合も）で、数字や固有名詞でないならそのまま。
3. それ以外の単語は出現形を「__」にした単語パターン。

以上の処理により、たとえば「奈良市内」からは次のようなパターンが得られる。

パターン 324 地名 0,1

（__, 漢字, 地名）（市内, 漢字, 一般名詞）

あるいは、「アンカーに福島を起用」という文で、「福島」が形態素解析により「地名」と判断された場合、次のようなパターンが得られる。

パターン 778 人名 0,0 （__, 漢字, 地名）

こうして得られた規則群を、NEタグのない訓練用文書に適用し、1次候補の集合を得る。この中には多数の間違いが含まれている。たとえば「奈良の鹿」という文章の「奈良」が人名の候補として選ばれる。そこで、この1次候補の中から正解だけが選ばれるよう、各パターンに制約を付加する。この制約を自動生成するため、C4.5を利用する。

この学習にあたって、どのような属性の組を学習例として与えるかが問題となる。ここでは単純に1次候補を構成する単語列$(w_b,c_b,p_b)\ldots(w_e,c_e,p_e)$と、その前の$k$単語$(w_{b-k},c_{b-k},p_{b-k})\ldots(w_{b-1},c_{b-1},p_{b-1})$、その後ろの$h$単語$(w_{e+1},c_{e+1},p_{e+1})\ldots(w_{e+h},c_{e+h},p_{e+h})$をそのまま並べた$3(k+l+e-b+1)$個の属性値$w_{b-k},c_{b-k},p_{b-k},\ldots,w_{e+h},c_{e+h},p_{e+h}$の組を学習例として与える。ただし、3回以上出現しない属性値は「その他」にまとめた。そして正解に対応する1次候補を正例とし、それ以外を負例とする。ただし、正解と重なって正例より優先順序の低いものは負例として採用しない[14]。たとえば、「田中太郎賞選考委」の「田中太郎」は人名として正解NEに含まれていないが、「田中太郎」が人名でないと学習する必要はないからである。

なお、決定木学習が陥りやすい過学習を避けるため、以下の処理を追加する。数字の値がいくつかということはNEを決定するうえであまり大きく影響しないので、数字はw_iとしてダミーをかわりに入れておく。また、「▽」のような特殊な記号が句点のかわりに使われることがある。そこで、正解固有表現の外側に記号（読点、括弧、中黒を除き、句点を含む）が現れた場合は、その記号自身と、そこから外側の単語すべてを学習データから削除し、空を表すダミーの記号で埋める。

可読性を考え、C4.5の出力をc4.5rulesで規則化する。これをPerlのコードに翻訳し、2次選抜を実行するプログラムの一部とする。たとえば付加制約778として以下のようなコードが得られる。

```
sub cns778 {
   (略)
  if ($W[2] eq '氏') { return 1; }
```

```
  (略)
if ($W[1] eq '大阪') { return 0; }
  (略)
return 1; }
```

このコードは「パターン 778 の 1 次候補は、後ろに「氏」が付いていれば 2 次候補であり、その単語自身が「大阪」であれば 2 次候補でなく、デフォールトでは 2 次候補である」という判断を表している。

4 改良

上記で述べた本システムの成績をさらに向上させるため、以下の 3 つの改良を加える。

4.1 括弧の処理

上記の枠組みでは、カギ括弧により組織名や書籍名などの範囲が明示されていても、それを一体として扱うことができない。これは人工物名の場合、致命的である。そこで、固有表現をちょうど囲むカギ括弧の処理を付加する。

カギ括弧内の不定長の単語列を扱うため、規則の生成と適用において、カギ括弧の中の単語をまとめ、見かけ上 1 語とする。その品詞は特殊な値とし、カギ括弧でくくられていない単語列とはマッチしないようにする。なお、カギ括弧の中身をまとめただけでは、「～を応援する会」のように内部の重要なヒントを見逃してしまうので、カギ括弧の中の最初と最後の単語・文字種・品詞を学習の属性として加え、固有表現の各種類の付加制約を作る。一体になった単語の前後には必ずカギ括弧があり冗長なので、学習データを作るときはカギ括弧を除く。

さらに、カギ括弧の前には、「英字紙『～』」「レストラン『～』」のように後続のカギ括弧中の言葉を説明する言葉が付くことが多い。ここではカギ括弧の直前が一般名詞か接尾語の場合に、それを予告語と呼ぼう。著者の経験では、予告語はきわめて有効なヒントであり、訓練データ中に 1 回しか登場しなくても利用したい。日本語語彙大系 [13] を用いて予告語のカテゴリーを調べるのも一つの手であるが、ここでは、訓練データ中に現れる予告語の辞書を自動作成する。また、予告語のある候補を優先するため、優先順位を (tb, − te, tp, − freq) に変更し、tp はその候補のカギ括弧の直前に、対応する種別の予告語があるときに 0、それ以外で 1 になるものとする。

なお、「奈良には鹿がいるんですよね」のような発言まで前処理で機械的に 1 語にすると、その中の「奈良」を抽出できない。そこで前処理はせず、これまでの 1 次選抜の処理の結果に影響を与えないようにカギ括弧の処理を付加する。カギ括弧はネストして用いられるので、スタックを用意し、それぞれの深さに対して制約を調べる。

なお、カギ括弧中に句読点がある場合は、そのカギ括弧中の単語列を発言と見なし、これらの規則とのマッチングは行なわない。実際には句読点を含む人工物名が存在するが、これらについては通常の処理のみ行なう。

4.2 規則の融合と末尾語辞書の自動作成

上記の一般化手続きによれば、たとえば次のようなパターンが得られる。

パターン 1022 組織名 0,0
（__，漢字，人名）（__，漢字，一般名詞）
（研究所，漢字，一般名詞）
パターン 1028 組織名 0,0
（__，漢字，地名）（__，漢字，一般名詞）
（高校，漢字，一般名詞）

しかし、これだけでは一般性が低すぎる。そこで、組織名・人名・地名の 2 語以上からなるパターンは最後の単語を規則間で共有することで適用範囲を広げたい。しかし、任意の一般名詞まで広げると広げすぎであり、学習で有効な末尾語を学習するには膨大なデータを必要とする。そこで、以下のようにパターンから最後の単語を分離して末尾語を共有する。この末尾語がこの辞書に属さなければ、パターンにマッチしなかったものとする。

パターン 10001 組織名 0,0
（__，漢字，人名）（__，漢字，一般名詞）
（__，漢字，一般名詞）
パターン 10002 組織名 0,0
（__，漢字，地名）（__，漢字，一般名詞）
（__，漢字，一般名詞）
末尾語辞書 組織名
（研究所，漢字，一般名詞）
（高校，漢字，一般名詞）

これにより、「人名＋一般名詞＋『高校』」もカバーできるようになる。同様に、文字種と品詞が一致するもの同士で末尾語を共有する。ただし、単漢字の場合は、未知語が過分割された可能性が大きい。そこで最後の 2 単語が単漢字の場合は辞書に登録しない。

4.3 文脈の考慮

周辺数単語を見ただけでは、ある程度以上に精度を向上させることは難しい。そこで文脈を考慮するために2つの工夫を加える。

ひとつは、記事のトピックの判断である。スポーツ記事では「横浜」や「広島」などが組織名としてしばしば用いられるので、記事の先頭４０文字以内に「野球」または「サッカー」という文字列があればスポーツ記事、なければ一般記事と判断し、この判断結果を学習時に属性のひとつとして追加した。

もうひとつは、ＩＡＴＡなどの略称の処理である。新聞記事の場合、略称と正式名が記事のどこかに「国際航空運送協会（ＩＡＴＡ）」のように隣接して書かれていることが多い。そこで、略称の抽出精度を向上させるため、ここでは「正式名(略称)」「略称（正式名)」のようなパターンを手がかりにして略称の抽出精度を上げるため、１次選抜と２次選抜の間に以下の処理を追加する。

「組織名（組織名」のように、同じ種類の１次候補同士が「(」をはさんで隣接し、一方がカタカナ語か大文字語の場合、その語を略称と考え、(記事番号，規則番号，略称）を「略称ファイル」に書き出す。一方がカタカナ語で他方が大文字語なら大文字語を略称とする。

２次選抜では、１次候補が略称ファイルに含まれていたら、対応する規則の付加制約のチェックを行わず、ただちに２次候補として出力する。

5 実験と結果

実装はPerlで行なった。なお、形態素解析の結果に間違いが多いと成績が下がるので、以下のような修正を加えた。「１，２，３」や「・１，２００」のように単一の数字とは考えられない単語は、「１/，/２/，/３」や「・/１，２００」のように分割する。また、「３月」のような月名も数字を分離する。「Ａ、Ｂ両党」や「Ａ，Ｂ、Ｃ各省」のようなパターンの「両党」や「各省」は１語のものとそうでないものがあるので、「両」「同」「全」「各」から始まる２～３文字の一般名詞は、最初の文字を接頭語、それ以外を一般名詞として分割する。

また、chasenのデフォルトの設定ではカタカナ語が過分割されやすいので、未知語の品詞コストを2000、接続コストを5000まで下げ、過分割されにくくした。さらに、未知語は「サ変名詞」として出力されるが、「固有名詞一般」として出力されるよう設定した。一方、jumanは括弧と記号が連続している場合にまとめてひとつの記号として扱うので、括弧と記号を切り離す前処理を付加する。また、「＝」はJISコードにない漢字の代わりに用いられるので、単漢字の固有名詞一般とした。辞書の間違いなどについては、今回とくに修正しなかった。

正解コーパスとしては、毎日新聞の94～95年の6087記事を用いた。その内訳は、IREXで配布されたCRL固有表現データ1174記事（以下CRLと略称)、分野特定課題用訓練データ23記事（以下NERTと略称）に加え、佐々木の拡張タグデータ[15]を最新のIREXの定義に即して修正した4890記事である。また、内元ら[11]の実験条件と合わせ、CRL固有表現データ、IREX-NE本試験逮捕トレーニングデータに加え、古い定義によるIREX-NE予備試験トレーニングデータとIREX-NE予備試験データ用い、形態素解析としてjumanを用いた場合の本システムの成績も求めた。実験条件を合わせるため、公開データのタグの誤りは修正していない。なお、CRLとNERTには、正解作成者が判断を迷ったときにOPTIONALというタグが付けられている。たとえば以下の最初の例は、「成田空港」が組織名か地名かで迷ったことを表す。

```
<OPTIONAL POSSIBILITY=ORGANIZATION,
 LOCATION TYPE=0>成田空港</OPTIONAL>
```

今回は、付けなくてもよいことを表すNONEが含まれる場合はタグを削除し、ない場合は列挙されている種別の中で、CRL中で一番頻度の多い種別を機械的に選んだ。

これら全部の訓練データを用いると、１万近い数の規則ができるが、それでも効率よく実行できるように最適化している。訓練データとしてCRLの各単語に文字種と品詞の入ったデータを用いた場合、規則生成に要したCPU時間は、Pentium III 866MHz, 256MBのlinuxマシンで約3分であり、この規則をIREX-NEの一般課題（以下general）に適用した場合の１記事あたりのCPU時間は0.1秒である。

制約を学習する時には前１語、後ろ２語を与えた。C4.5はデフォールトで「利得比基準」により属性を選択するが、-gオプションを付けると「利得基準」により属性が選択される。通常、「利得基準」は過学習をおこしやすいとされているが、今回の結果では、「利得基準」を用いた方が若干よい

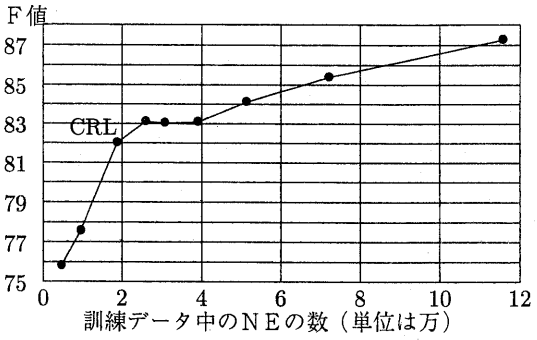

図 3: 訓練データのサイズと一般課題のF値の変化

結果が得られている。これは、「利得比基準」を用いた場合、品詞や文字種が選ばれやすく、抽象的な誤りやすい規則ができやすいのに対し、「利得基準」を用いると単語が選ばれやすいためと考えられる。

本手法のスコアを IREX-NE 本試験に参加した１５システム中の上位６システムや、本試験後に公表された文献で学習を用い比較的よい成績のものと比較したのが表１である。arrest は正解ＮＥの数が general の約 1/4 と小さいので、general のスコアを優先して並べてある。また、chasen2.02 で正解コーパスを増やしていった時の一般課題のＦ値の変化を図３に示す。ただし、データ量が増えた場合、末尾語辞書への登録を２回以上出現した場合に限った方が成績が若干よいので、ここでは２回以上出現した末尾語のみ、規則の融合・辞書登録を行なっている。これを見ると、ＮＥの数が３万のあたりで成績が足踏みしている。これは、たまたまそのあたりに、一般課題に不利なケースがあり、その影響を受けている。また、まだ飽和していないように見える。

本手法では、品詞と文字種の両方の制約を規則の各単語に課しているので、少量のデータでは十分な recall が達成できないのではないかと危惧されたが、内元らとの結果と比較すると、本手法の recall は general 75.03/arrest 80.72、内元らの recall は general 74.50/arrest 81.75 と大差ない。

6 おわりに

本手法の特長は以下の２点である。

- 入手しやすい形態素解析と決定木学習のシステムを用いて、比較的高精度の固有表現抽出システムが簡単に構築できる。
- 生成された規則は人間が読んで理解・修正できる。

今回、可読性を重視した構成にしたので、間違いがある場合に、どこでなぜ間違えたかを、正解コーパスにさかのぼって調べるプログラムを作成するのも比較的容易であった。これにより、コーパスの誤りや固有表現の定義の問題が多数発見できた。関根ら [10] や野畑 [7] も決定木を用いた固有表現抽出法を示しているが、ルールを確率的に解釈しているので、原因の追求や処理の修正は本手法ほど単純ではないと思われる。可読性を問わない場合に、どこまで成績が上がるかなども調べてみたい。ＱＡシステムへの応用を考えると、もっと細かい分類を行ないたいので、少ない正解データから固有表現抽出規則を学習する方法 [2, 12] についても考えてみたい。

最後に、本研究を支援してくださった勝野裕文、加藤恒昭両博士と固有表現の正解データを提供してくれた佐々木裕博士に感謝します。

参考文献

[1] *Proceedings of the Sixth Message Understanding Conference*. Morgan Kaufmann, 1996.

[2] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of EMNLP/VLC*, 2000.

[3] IREX 実行委員会（編）. IREX ワークショップ予稿集, 1999.

[4] J. R. Hobbs, D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, and M. Tyson. FASTUS: A cascaded finite-state transducer for extracting information from natural-language text. In *Finite-State Language Processing*, pp. 383–406. MIT Press, 1997.

[5] 磯崎. 辞書式優先順位に基づく日本語固有表現抽出. 情報処理学会研究報告 NL-135-5, pp. 33–40, 2000.

[6] 賀沢, 加藤. 意味制約を用いた日本語質問解答システム. 情報処理学会研究報告, 2000. NL-140-26.

[7] 野畑. 決定木を用いた学習に基づく固有表現抽出システム. IREX ワークショップ予稿集, pp. 201–206, 1999.

[8] 佐々木, 磯崎, 平, 廣田, 賀沢, 平尾, 中島, 加藤. 質問応答システムの比較と評価. 電子情

表 1: IREX-NE の課題に対する成績の比較

「固有名詞数」は辞書登録されている固有名詞の数で単位は千語。
「訓練コーパス」は訓練に使ったコーパスのサイズで単位は千文。
HC は手書き、ME は最大エントロピーモデル、TBL は変換に基づく誤り駆動学習、DT は決定木、DL は決定リスト、NA は不明を表す。(2) は 2 回以上出現した末尾語のみ辞書登録した場合。

システム	general	arrest	形態素解析	固有名詞数	訓練コーパス	手法
IREX1247	83.86	87.43	NA	89	5	HC [3]
IREX1205	80.05	78.08	breakfast	105	11	HC + TBL [3]
IREX1227	77.37	85.02	juman3.3	43	11	ME [3]
IREX1224	75.30	77.61	sumomo	NA	10	HC [3]
IREX1231	74.82	81.94	NA	NA	2	HC [3]
IREX1223	72.18	74.90	juman3.6	34	11	ME [3]
内元ら	79.42	83.91	juman3.6	34	13	ME+TBL [11]
本手法	77.92	82.63	juman3.61	34	13	DT
内元ら	80.17	85.75	juman3.6	36	13	ME+TBL [11]
颯々野ら	80.0	NA	breakfast	NA	11	DL [9]
本手法	87.25	87.79	chasen2.02	126	70	DT(2)

報通信学会技術研究報告 NLC, 10 2000.

[9] M. Sassano and T. Utsuro. Named entity chunking techniques in supervised learning for Japanese named entity recognition. In *Proceedings of the International Conference on Computational Linguistics*, pp. 705–711, 2000.

[10] S. Sekine. NYU: Description of the Japanese NE system used for MET-2. In *Message Understanding Conference*, 1998.

[11] 内元, 馬, 村田, 小作, 内山, 井佐原. 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. 自然言語処理, 7(2):63–90, 2000.

[12] T. Utsuro and M. Sassano. Minimally supervised Japanese named entity recognition: Resources and evaluation. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, pp. 1229–1236, 2000.

[13] 池原, 宮崎, 白井, 横尾, 中岩, 大山, 林. 日本語語彙大系. 岩波書店, 1997.

[14] 磯崎. 並行実行される固有表現抽出規則の一括生成. 人工知能学会全国大会論文集, pp. 142–144, 2000.

[15] 佐々木. トランスデューサによる日本語固有表現抽出. 言語処理学会第 5 回年次大会発表論文集, pp. 108–111, 1999.