課題研究報告書

情報検索のための自然言語処理ツール群の開発

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

関口 宏司

2014 年 3 月

課題研究報告書

情報検索のための自然言語処理ツール群の開発

指導教員　　白井清昭 准教授

審査委員主査　　白井清昭 准教授
審査委員　　島津明 教授
審査委員　　飯田弘之 教授

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

1210905 関口 宏司

提出年月：2014 年 2 月

概要

今日、情報検索はますます不可欠なツールとしてコンピュータユーザの日常に溶け込んでいる。しかしながら、現在の情報検索が抱える問題も大きい。情報検索システムがユーザの期待に応えられていない点は、(1)ユーザが求める情報を返せない、(2)ユーザが求めていない情報を返してしまう、という 2 点に集約される。前者の問題は「検索漏れ」、後者の問題は「検索誤り」と呼ばれる。検索漏れがあると、ユーザは必要な情報が得られるまで検索質問を少しずつ変えながら繰り返し検索を実行しなければならない。また検索誤りがあると、情報検索システムから返ってきた膨大な検索結果の中からユーザが求める文書を探す作業が困難になる。一般に検索漏れと検索誤りの問題はトレードオフの関係にある。検索漏れに対処するために検索システムの出力を大きくすると検索誤りが増えてしまい、検索誤りに対処するために検索システムの出力を小さくすると検索漏れが増えてしまうためである。そこで本課題研究では、両問題を同時に解決するのではなく、段階的に対処する方法を提案する。提案方法では、まず検索漏れを小さくし、次いで絞り込み検索によって漸次的に検索誤りを小さくする。

検索漏れを小さくするために、自然言語処理の既存研究を応用した「原型語とその省略語の自動抽出プログラム」「漢字送りがな表記揺れ知識の自動抽出プログラム」「N-best パス探索形態素解析プログラム」という 3 つのツールを開発した。

「原型語とその省略語の自動抽出プログラム」は、原型語と省略語候補の類似度計算に基づく酒井らの研究を辞書に応用したものである。開発したプログラムを Wikipedia に適用したところ、再現率は 90%を超えたが精度は 70%を下回った。しかし、省略語だけでなく関連語も正解とするように条件を緩めると、再現率はあまり変わらず、精度は 80%を超えた。「漢字送りがな表記揺れ知識の自動抽出プログラム」は、単語辞書から表記揺れがある漢字送りがなの組を自動抽出するものである。形態素数 392,126 件の単語辞書 IPAdic に本プログラムを適用したところ、9,449 件の漢字送りがな表記揺れ知識が抽出できた。「N-best

パス探索形態素解析プログラム」は、日本語における単語分割の曖昧性の問題に対処した形態素解析プログラムである。本プログラムは日本語テキストを形態素解析し、コスト最小のパス上の全トークンを出力した後、2 位以下 N 位までのパス上の名詞のみを出力する。2 位以下の重複するトークンや検索される可能性の低い名詞以外の語を出力しないことで情報検索における処理負荷を軽減できる。また、2 位以下の解からも名詞を出力することで、単語区切りの不一致による検索漏れを防ぐ効果が期待できる。本形態素解析プログラムは、永田の方法に基づき入力文の N-best パス解出力を行う。永田の方法では、最初に与えられた入力文を 1 文字ずつ前向きに解析し、単語ラティスを作成する。次に作成した単語ラティスからコストが小さい上位 N 個のパスを探索してパス上のトークンを順に出力する。単語辞書にはトライ構造をコンパクトにメモリに格納できるダブル配列を採用して実装した。さらに、「原型語とその省略語の自動抽出プログラム」および「漢字送りがな表記揺れ知識の自動抽出プログラム」の出力を利用できるようにした。実験により N-best パス解が出力され、「原型語とその省略語の自動抽出プログラム」および「漢字送りがな表記揺れ知識の自動抽出プログラム」の出力が適用されたことが確かめられた。このことから、この出力を使って転置インデックスを作成すれば、情報検索における再現率の向上が期待できることがわかった。

以上の検索漏れ対策により検索の精度は低下する。そこで、絞り込み検索により漸次的に検索誤りを小さくする。情報検索における絞り込み検索とは、ユーザが最初に実行したクエリに加えて新しい検索条件を追加して再検索することを指す。「絞り込み」検索のため、追加する検索条件は既存の検索条件に AND でつながれる。この操作により、精度を漸次的に向上させていくことが可能である。絞り込み検索を行うためには、検索対象文書レコードが、絞り込みを行うための構造を持っていなければならない。新聞記事などをはじめ、多くの文書はこのような構造を持っていないので絞り込み検索を容易に実行することができない。しかし、固有表現抽出技術を適用すると、絞り込み検索に適した構造を持たせることができる。固有表現抽出は品詞タグ付けなどと同様、系列ラベリング問題の一種である。固有表現タグ付きコーパスから、入力文における個々のトークンに対する固有表現タグを決定するモデルを機械学習した。機械学習アルゴリズムは条件付き確率場（Conditional Random Field; CRF）を採用した。訓練データとなる固有表現タグ付きコーパスとして、関根の固有表現タ

グ付きコーパスを用いた。情報検索をビジネスの場面で利用することを想定し、粒度の細かい関根の拡張固有表現階層のタグを 9 個の新たなタグにまとめ直した。関根の固有表現タグ付きコーパスから 12,000 文を抽出し、3 分割の交差検定によって本課題研究で実装した固有表現抽出ツールを評価したところ、固有表現タグの種類により F 値に差が出ることがわかった。そこで固有表現の品詞の出現頻度を調べたところ、成績の低い固有表現タグでは一般性の高い品詞の出現頻度が 1 位となっていた。逆に成績の高い固有表現タグでは固有表現の意味に沿った特徴的な品詞の出現頻度が高かった。また、別の有用な素性を追加することで F 値が改善されることを考察した。

目次

第1章　はじめに

1.1 研究の背景

コンピュータの高性能化とインターネットの発展に伴い、近年、膨大な電子化された情報がアクセス可能になってきている。そのため情報検索はますます不可欠なツールとしてコンピュータユーザの日常に溶け込んでいる。しかしながら、現在の情報検索が抱える問題も大きい。今日の情報検索システムがユーザの期待に応えられていない点は下記の 2 点に集約される。

- ユーザが求める情報を返せない
- ユーザが求めていない情報を返してしまう

前者の問題は「検索漏れ」、後者の問題は「検索誤り」と呼ばれる。検索漏れがあると、ユーザは必要な情報が得られるまで検索質問（クエリ）を少しずつ変えながら繰り返し検索を実行しなければならない。また検索誤りがあると、情報検索システムから返ってきた膨大な検索結果の中からユーザが求める文書を探す作業が困難になる。これらはユーザの作業負担が大きく、情報検索への依存度が高い分、問題解決への期待が大きい。そこで本課題研究では、自然言語処理の既存研究の成果を情報検索へ応用し、これらの問題を解決することを目指す。

1.2 研究の目的

本課題研究では、情報検索における前述の 2 つの問題に対応するため、自然言語処理の既存研究の成果を実際にプログラム開発して実行し、実行結果について考察を加えることを目的とする。具体的には以下のプログラムを開発する。

- 原型語とその省略語の自動抽出プログラム
- 漢字送りがな表記揺れ知識の自動抽出プログラム
- N-best パス探索形態素解析プログラム
- 固有表現抽出プログラム

なお本課題研究が対象とする情報は日本語テキストに限定し、日本語以外の外国語テキストおよび画像や音声は含まない。

1.3 本書の構成

本課題研究報告書の構成は以下の通りである。第 2 章で本課題研究の前提知識となる情報検索について、その概要を述べる。さらに前述の情報検索における 2 つの問題点が精度と再現率という定量的な指標で評価できることを示す。第 3 章では検索漏れの発生要因を分析し、検索漏れを低減するために実際に開発したプログラムについて説明する。開発したプログラムの実行結果を示し、考察を述べる。検索漏れを改善することで、すなわち再現率を改善することで、それとトレードオフの関係にある精度が低下してしまう問題は第 4 章で検討する。現実的な問題対策として固有表現抽出が有効であることを説明し、機械学習の手法を用いて固有表現抽出プログラムを実装する。また、このプログラムを実験により評価する。第 5 章で全体のまとめと今後の課題について述べる。

第2章　情報検索の概要

本課題研究の前提知識となる情報検索について、その概要を述べる。はじめに現在の標準的な情報検索システムが転置インデックスに基づくものであることを説明し、これにより情報検索システムの処理単位が単語であることを示す。次に情報検索の評価指標として一般に用いられている精度と再現率について説明し、これらの観点から情報検索システムの改善を目指す後章につなげる。最後に、転置インデックスを作成するために日本語テキストを単語に分割する方法について説明する。

2.1 情報検索システム

情報検索はユーザ（情報検索の利用者）からクエリ文字列を受け取り、該当する文書番号のリストをユーザに返すタスクである。このために、事前に転置インデックスと呼ばれるデータ構造を作成する。今、図 2.1 に示す 3 件の検索対象文書があるとする。

1: カツオはサザエの弟
2: サザエはワカメの姉
3: ワカメはカツオの妹

図 2.1 検索対象文書の例

図中の番号は文書番号である。検索対象文書から転置インデックスを作成する手順は以下の通りである。

(1) 図 2.2 のように、検索対象文書を単語に分割し、単語と文書番号のペアでまとめる。
(2) 手順(1)の結果得られたすべての単語のリストをソートし、同じ単語のものは文書番号をまとめる。

カツオ:1, は:1, サザエ:1, の:1, 弟:1
サザエ:2, は:2, ワカメ:2, の:2, 姉:2
ワカメ:3, は:3, カツオ:3, の:3, 妹:3

図 2.2 転置インデックス作成の途中図

以上の手順の結果、最終的に図 2.3 の転置インデックスが得られる。

の　　:1, 2, 3
は　　:1, 2, 3
カツオ　:1, 3
サザエ　:1, 2
ワカメ　:2, 3
弟　　:1
姉　　:2
妹　　:3

図 2.3 転置インデックス

転置インデックスが得られたら、これを用いて検索を行うことができる。たとえばユーザから「ワカメ OR 妹」というクエリが与えられたとする。すると情報検索システムは転置インデックスの単語リストをクエリの最初の単語である「ワカメ」で検索し、「ワカメ」を文書中に含む文書番号 2 と 3 を得る。続いてクエリの次の単語である「妹」で転置インデックスの単語リストを検索し、「妹」を文書中に含む文書番号 3 を得る[1]。ユーザのクエリはこれら 2 つの単語を OR で接続しているので、クエリの解としてシステムはユーザに文書番号 2 と 3 を提示すればよい。しかし、クエリの単語をより多く含む文書番号 3 を検索結果表示リストの先頭にし、次いでクエリの単語を 1 つしか含まない文書番号 2 を提示する方が、ユーザが求めているものにより合致する可能性が高い。この検索結果となる文書リストの表示順の操作をランキングと呼ぶ。ランキングはユーザクエリと各文書の類似度を降順に並び替えることで行える。情報検索にお

[1]単語をクエリとした転置インデックスの検索は、バイナリサーチやハッシュなどの高速なアルゴリズムが一般的に用いられる。

けるクエリと文書の類似度計算は、一般にベクトル空間モデルに基づく方法で行われる。ベクトル空間モデルではクエリと文書それぞれに対して単語の重みを要素とするベクトルを考え、これらのベクトルの近さを類似度と考える。一般にベクトル間類似度としてはコサイン類似度が用いられる。ベクトルの次元は転置インデックスに登場する全単語であるが、クエリベクトルにおいてはほとんどの単語がユーザクエリに含まれないために重みは 0 となり、コサイン類似度を測るときは無視できるため、類似度計算は高速に行える。

2.2 情報検索の評価指標

従来から用いられている情報検索の評価指標として精度と再現率がある。図 2.4 は情報検索システムに対してユーザがあるクエリを投げ、その応答をシステムから受け取ったときの、それぞれの文書集合に着目した図である。長方形の領域が転置インデックスに登録されている全文書集合、左の円の領域（A∪B）がユーザクエリに対してシステムが返した文書集合、右の円の領域（B∪C）がユーザが期待する文書集合（正解または適合文書ともいう）である。このとき、情報検索の精度 P（Precision）とは、システム応答の文書集合に含まれる正解文書の割合であり、式(2.1)で求められる。

$$P = \frac{|B|}{|A|+|B|} \tag{2.1}$$

また情報検索の再現率 R（Recall）とは、正解文書集合に含まれるシステム応答文書の割合であり、式(2.2)で求められる。

$$R = \frac{|B|}{|B|+|C|} \tag{2.2}$$

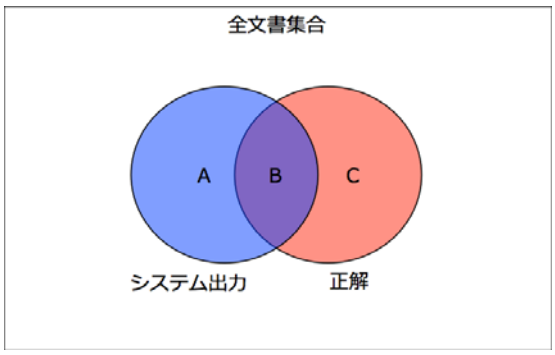

図 2.4 情報検索システムにおける出力文書と正解文書

情報検索システムをこれらの指標の下で評価するなら、システムの目標は左の円と右の円をなるべく近づけ、領域 A と領域 C を小さくすることである。領域 A はユーザが望まないシステム応答であり検索誤りと呼ばれるものである。|A|が 0 に近づくと式(2.1)から精度 P が 100%に近づく。一方領域 C はユーザが期待しているがシステムが返せなかった文書集合であり検索漏れと呼ばれるものである。|C|が 0 に近づくと式(2.2)から再現率 R が 100%に近づく。しかしながら精度と再現率はトレードオフの関係になることが知られており、精度向上を目指すと再現率が下がり、再現率向上を目指すと精度が下がる傾向がある。

本課題研究では、検索漏れの発生要因を分析し、検索漏れに対処するプログラムを開発することで領域 C を小さくすることを目指す。さらに領域 A の低減については絞り込み検索により対処することを考える。絞り込み検索とは、ユーザクエリに対してシステム応答が返された図 2.4 の状態で、システムから追加キーワードの候補をいくつか提示し、ユーザがその中から適切なものを選択すると、システムの出力文書集合をそのキーワードを含む文書のみに絞り込む手法である。絞り込み検索の一般的なインタフェースにおいては、検索システムは追加キーワードをリンクとして提示し、ユーザがリンクをひとつ選択してクリックすると絞り込み検索の結果を新たに提示する。システムから提示されるリンクラベルをヒントに、ユーザが適切なリンクを選択していくことで、領域 A を徐々に小さくすることを狙う。

検索漏れの発生要因の分析とそれに対処するプログラム開発については第 3 章で述べる。絞り込み検索の詳細とそのためのプログラム開発については第 4 章で述べる。

2.3 日本語テキストの単語分割

検索対象文書から転置インデックスを作成するには、文書を単語に分割しなければならない。しかし本課題研究が対象とする日本語テキストは、英語等と異なり、単語と単語の間がスペースで区切られていないため、単語の範囲が明確ではない。

この問題に対し、日本語の情報検索システムでは従来より形態素解析により単語分割を行うことで対処している。形態素解析プログラムは単語辞書を参照しながら入力文書を単語分割するプログラムである。日本語の形態素解析につ

いては研究が進んでおり、一般公開されている形態素解析ツールもいくつか存在する。

一方、単語分割を行わずに、文中に含まれる全ての文字 N グラムを用いて転置インデックスを作成する手法もある。文字 N グラムとは連続する N 個の文字列を指す。たとえば「カツオはサザエの弟」という文における文字 2 グラム(N=2)は次のようになる。

カツ / ツオ / オは / はサ / サザ / ザエ / エの / の弟

転置インデックスはこのような文字 N グラムに対して作成される。すなわち、個々の文字 N グラムに対し、それを含む文書集合が転置インデックスに登録される。

検索時には、検索クエリの文から転置インデックスを検索するための検索キーを取得する。このとき、転置インデックス作成時に採用したものと同じ手法で検索キーを取得する必要がある。転置インデックスを形態素解析による単語分割を経て作成した場合、検索クエリ文を同様に形態素解析して単語に分割し、その単語を含む文書を検索する。一方、転置インデックス作成時に文字 2 グラムを用いた場合は、検索クエリの文に含まれる文字 2 グラム、たとえば「カツオ」がクエリのときにはそれに含まれる文字 2 グラム「カツ」「ツオ」を含む文書を検索する。

形態素解析による単語分割によって転置インデックスを作成する場合、単語を検索キーとして転置インデックスを検索するため、自然な検索が実現できる。しかしながら、文字 N グラムを用いて転置インデックスを作成する手法と比べて、精度が高く保てるというメリットがある反面、単語辞書に載っていない「未知語」に弱く、再現率が低くなるというデメリットがある。文書に未知語が存在すると、単語分割の失敗により未知語が単語として認識されず、転置インデックスに登録されないためである。一方、文字 N グラムでは全ての文字列が必ず転置インデックスに登録されるため、未知語のように誤って転置インデックスに登録されないために検索漏れを生じることがない。しかしながら、人間から見て不自然な文字列を検索キーとして転置インデックスを検索するため、検索の精度が低下するデメリットがある。

本課題研究では、転置インデックスを作成する際に、文字 N グラムと比較す

ると精度が高い形態素解析を用いて単語分割を行う方式を採用することとし、開発するツール群のひとつに形態素解析プログラムを含める。ただし、形態素解析によって転置インデックスを作成した場合、文字 N グラムよりも検索漏れが発生しやすい（再現率が低くなりやすい）ことから、検索漏れの具体的な発生要因を分析する。そしてそれらの問題への対応方法を検討し、実現可能なものについては形態素解析プログラムの補助ツールとしてプログラム開発を行う。

第3章　情報検索の検索漏れの低減

本章では、検索漏れの問題と対策について検討する。最初に検索漏れの発生要因を分析し、対策方法を項目ごとに述べる。次に本課題研究で実施する対策について開発する各プログラムの詳細を説明する。

3.1 検索漏れの要因と対策

検索漏れは、クエリに使用される文字列と検索対象文書に使用される文字列が一致しないときに発生する。両者の文字列が一致しない原因には全角・半角の違いなどの軽微なものから、「内閣総理大臣」と「首相」などの同義語のようなものまで多岐にわたるが、本課題研究では表 3.1 のように分類・整理した。

表 3.1 検索漏れの発生要因とその対策

要因	対策
ボキャブラリ・ ギャップ	シソーラスの利用、 原型語とその省略語の辞書からの自動獲得（※）
表記揺れ	正規化、漢字送りがな表記揺れ知識の自動獲得（※）
単語分割の曖昧性	文字 N グラムの併用、N-best パス探索形態素解析器（※）
未知語	文字 N グラムの併用、未知語の既知語化

（※印は本課題研究で開発するツール）

以下、それぞれの項目に対する詳細を具体例を交えて説明し、問題への対策を考察する。

3.1.1 ボキャブラリ・ギャップによる検索漏れとその対策

ボキャブラリ・ギャップによる検索漏れとは、検索対象文書執筆者が使う用語と検索者が使う用語の違いにより発生する検索漏れである。さらにボキャブラリ・ギャップは次のように細分化できると考えられる。

(1) 「内閣総理大臣」と「首相」など、語形が異なるが意味が同じであるもの。

同義語。

(2) 「言う」と「話す」など、語形が異なるが意味が似通っているもの。類義語。

(3) 「自動車損害賠償責任保険」と「自賠責保険」など、原型語とその省略語。

(1)(2)のパターンに関してはシソーラスを利用することで検索漏れを軽減することができるため、本課題研究の範囲外とする。(3)のパターンについては辞書から原型語とその省略語を自動的に抽出するプログラムを本課題研究にて開発する。その詳細は 3.2 節で述べる。

3.1.2 表記揺れによる検索漏れとその対策

表記揺れによる検索漏れとは、ボキャブラリ・ギャップほどの用語の隔たりはないものの、検索対象文書執筆者が使う文字と検索者が使う文字の一部が異なる場合を指す。用語の表記揺れはさらに以下のように細分化できる。

(1) 「ｺﾝﾋﾟｭｰﾀｰ」と「コンピューター」などの半角と全角の違いによるもの。

(2) 「ユーザー」と「ユーザ」などのカタカナ語末尾の長音記号の有無によるもの。

(3) 「慶応大学」と「慶應大学」など、漢字の新旧の違いによるもの。

(4) 「一二三」と「123」など、数字の表記方法の違いによるもの。

(5) 「2013 年」と「平成 25 年」など、西暦と和暦の違いによるもの。

(6) 「いすゞ」と「いすず」などの踊り字利用の有無によるもの。

(7) 「引っ越し」と「引越し」などの漢字送りがな表記の違いによるもの。

このうち(7)以外はルールベースのプログラムにより文字列の正規化を行うことで比較的簡単に対処できると考えられるため、本課題研究の範囲外とする。(7)については「引っ越し」に対して「引越し」以外にも「引越」があり、「旭ヶ丘」に対して「旭が丘」や「旭丘」などがあるように、送りがなの違いは多岐にわたるため、ルールベースのプログラムによる正規化では対処が難しい。この問題に対する対策として、単語辞書（形態素解析プログラムが参照する辞書）から漢字送りがなの表記揺れ知識を自動抽出する方法が考えられる。本課題研究ではそのプログラムを開発し、その詳細は 3.3 節で述べる。

3.1.3 単語分割の曖昧性による検索漏れとその対策

日本語は英語等と異なり単語と単語の間がスペースで分かち書きされていな

いことから単語分割に曖昧性がある。たとえば「ここではきものを脱ぐ」という文は、この文に埋め込まれている名詞を「着物」ととるか「履き物」ととるかで、次の 2 通りの単語分割の可能性がある。

(1) ここ／では／きもの／を／脱ぐ

(2) ここ／で／はきもの／を／脱ぐ

もしこの文の執筆者が「履き物」のつもりで書いたとしても、形態素解析プログラムが(1)のように単語を分割すると、転置インデックスには「はきもの」という単語は現れないので、「はきもの」をクエリとする検索時には検索漏れを起こす。

この問題は、最小コスト法に基づく形態素解析プログラムが入力文から単語ラティスを作成する際、最小コストのパス上のトークンのみを解として出力していることから発生していると考えられる。そのため本課題研究ではコストが小さい上位 N 個のパス上のトークンを出力する N-best パス探索形態素解析プログラムを開発し、単語分割の曖昧性による検索漏れの対策とする。このほかに文字 N グラムによる転置インデックスの作成も本検索漏れ対策として有効であるが、本課題研究の対象外とする。

3.1.4 未知語による検索漏れとその対策

単語辞書に載っていない単語を未知語という。未知語の存在もまた検索漏れの要因となる。たとえば未知語「アベノミクス」は単語辞書に載っていないため、本来は「アベノミクス」という 1 単語として解析されるべきであるが、誤って「アベ／ノ／ミクス」と 3 単語に分割される可能性があり得る。このため、単語区切りの不一致による検索漏れが発生する。

本検索漏れの対策としては以下のものが考えられる。

(1) 文字 N グラムによる転置インデックスの作成

(2) N-best パス探索形態素解析プログラムによる単語分割

(3) 未知語の辞書登録

(1)の方法で問題が解決することは明らかである。ただし、2.3 節で議論したように、形態素解析プログラムによる単語分割を経て単語を転置インデックスに登録する方法と比べて検索の精度が下がるという問題点もある。(2)の方法は本課題研究で開発する前述の N-best パス探索形態素解析プログラムにおいて、N を十分大きくとることで「アベノミクス」も「アベ／ノ／ミクス」も出力さ

れる可能性を高めようというものである。もっとも後述するように、開発する N-best パス探索形態素解析プログラムは、2 位以下のパスについては名詞のみを出力するので、名詞以外の未知語に対しては効果がなく、この問題への対策とするには不十分である。(3)は文書中の未知語を発見し自動的に単語辞書に登録することで未知語を未知語でなくしてしまう方法である。(3)の方法を採用する場合、「未知語の発見」と「発見した未知語の品詞とコスト推定」をどのように行うかが問題となる。本課題研究では、未知語の存在を要因とする検索漏れに対しては(2)による限定的な対策を実装するだけに留め、それ以外の対策の実現は将来の課題とする。

3.2 原型語とその省略語の自動抽出プログラム

本節では、辞書から原型語とその省略語のペアを自動抽出するプログラムについて説明する。また、開発したプログラムを Wikipedia[13]に適用して実行した結果および考察を述べる。

3.2.1 プログラムのアルゴリズム

酒井ら[4]は平文コーパスから原型語とその省略語を抽出するアルゴリズムを提案した。本課題研究ではこの方法を辞書に適用することで計算量を減らした。ここで辞書とは、見出し語とその説明からなるレコードの集合を指す。Wikipedia や現代用語の基礎知識[14]などが辞書の具体例である。本プログラムのアルゴリズムは以下の通りである。

(1) 辞書中の見出し語 t_A の説明から名詞または複合名詞（名詞の連続）を取り出し、それを省略語候補 t_B とする。なお未知語も名詞として扱い、複合名詞は名詞または未知語の連続とする。

(2) 各 t_B について後述する文字列条件 C_S、名詞条件 C_N が成立するか確認する。どちらも成立する場合はその t_B について(3)に進み、成立しない場合は当該 t_B は棄却する。すべての t_B が棄却されたら辞書から次の t_A を選んで(1)に行く。

(3) t_A と t_B の類似度を計算する。具体的には、t_A の説明 A_A から t_A の特徴ベクトル x_A を、t_B を使って書かれた説明の集合$\{A_B\}$（ただし$A_A \notin \{A_B\}$）から t_B の特徴ベクトル x_B を作成し、類似度 $S(x_A,x_B)$を計算する。特徴ベクトル x_A ならびに x_B の作成方法の詳細は後述する。S の計算にはコサイン類似度を

用いる。ただし、$\{A_B\}=\phi$ のときは $S(x_A,x_B)=1$ と定める。

(4) (3)で計算した類似度が閾値を超えた場合は t_A と t_B を原型語と省略語の組として出力する。すべての t_B について(3)(4)を実行したら、辞書から次の t_A を選んで(1)へ行く。

手順(2)の文字列条件 C_S について説明する。以下のすべてが成り立つとき，C_S が満たされる。

① t_A と t_B の最初の文字が一致する。

② t_B が t_A に完全に含まれない。

③ t_B を構成する文字がすべて t_A に含まれ、出現順序が一致する。

図 3.1 に示す具体例を用いて C_S を詳しく説明する。「原子力発電所」という見出し語とその説明から 3 つの省略語候補「原発」「原子力」「原電力」が得られたとする。

t_A=原子力発電所
t_{B1}=原発
t_{B2}=原子力
t_{B3}=原電力

図 3.1 見出し語「原子力発電所」とその省略語候補の例

まず C_S①の条件が成り立つか見てみると、t_A「原子力発電所」の最初の文字は「原」であり 3 つの省略語候補の最初の文字も「原」なので、すべての省略語候補について C_S①は成立する。次に C_S②についてであるが、t_{B2}「原子力」は t_A の文字列「原子力発電所」に完全に含まれるので t_{B2} は C_S②が不成立となる。最後に C_S③であるが、t_{B3}「原電力」の 3 つの文字「原」「電」「力」は t_A「原子力発電所」にすべて含まれるが出現順序が一致しないので t_{B3} は C_S③が不成立となる。したがってこの例の場合は省略語候補 t_{B1}「原発」だけが C_S の条件を満たす。

次に手順(2)の名詞条件 C_N について説明する。以下のすべてが成り立つとき、C_N 条件が満たされる。

① t_B は t_A から 2 つ以上の名詞を省略したものではない。

② t_B は t_A から最後の名詞を省略したものではない。

ここでの「名詞の省略」とは、t_A における名詞のすべての文字が t_B に出現し

ない場合、その名詞が省略されているとみなす。名詞の文字が 1 文字でも t_B に出現する場合はその名詞は省略されていない。

図 3.2 の具体例を用いて C_N を詳しく説明する。「山口福祉文化大学」という見出し語とその説明から 3 つの省略語候補「山福大」「山口大学」「山福文化」が得られたとする。

t_A=山口福祉文化大学
t_{B1}=山福大
t_{B2}=山口大学
t_{B3}=山福文化

図 3.2 見出し語「山口福祉文化大学」とその省略語候補の例

t_A「山口福祉文化大学」は形態素解析プログラムにより「山口／福祉／文化／大学」という 4 つの名詞に分割される。ここで t_{B1}「山福大」は「文化」という名詞が省略されているが、「山口」「福祉」「大学」は省略されておらず[2]、省略されている名詞は 1 つだけなので、C_N①が成立する。一方、t_{B2}「山口大学」は「福祉」と「文化」という 2 つの名詞が省略されているので C_N①が不成立となる。また t_{B3}「山福文化」は最後の名詞「大学」が省略されているので C_N②が不成立となる。よって t_{B1}「山福大」だけが C_N を満たすこととなる。

次に手順(3) における t_A, t_B の特徴ベクトル x_A, x_B の作成方法について説明する。なお、以下の方法は予備実験による検討を踏まえて定めている。

- 見出し語 t_A の説明 A_A に出現する単語のうち、重み w が大きいものから上位 40 個を選び、単語集合 W_A とする。単語の重み w は情報検索における TF・IDF 法に準じて式(3.1)のように定義する。

$$w = \sqrt{tf} \cdot (1 + \log(\frac{m}{df + 1})) \qquad (3.1)$$

  tf は単語が説明(A_A)の中で出現する回数、m は辞書における説明の総数、df は単語を含む説明の数である。
- 辞書から t_B を含む説明の集合$\{A_B\}$を取得する。この際、集合$\{A_B\}$の件数は最大で 10 件とした。t_B を含む説明が 10 件を超える場合は、t_B をなるべく多く含むものから上位 10 件の説明を$\{A_B\}$とした。

[2] 各名詞のうち 1 文字は t_{B1} に出現するため、省略されていないとみなす。

- $\{A_B\}$の各要素 A_B から、式(3.1)で計算した単語の重み w の大きいものから上位 10 件の単語を取り出す。それらの単語の和集合を W_B とする。
- W_A と W_B にともに含まれる単語で特徴ベクトル x_A, x_B を作成する。すなわち、特徴ベクトル x_A, x_B の次元は集合 $W_A \cap W_B$ に含まれる単語とする。特徴ベクトル x_A の各次元の重みは、説明 A_A における単語の重み w とする。一方、特徴ベクトル x_B の各次元の重みは、説明集合$\{A_B\}$内の全ての A_B における単語の重み w の和とする。

なお、本プログラムでは、見出し語の説明から省略語が抽出できなかった場合、見出し語の読みを使って手順(2)と手順(3)を実行し、省略語の抽出を試みる。たとえば、「堀内健」という見出し語は読みに変換すると「ホリウチケン」になり「ホリケン」が省略語として抽出できる。ただし、手順(2)では文字列条件 C_S のみをチェックする。読みに変換すると名詞か否かを判別できないので、名詞条件 C_N は使わない。

辞書の見出し語と説明の形態素解析には、本課題研究で開発する N-best パス探索形態素解析プログラム（3.4 節）を用いる。

3.2.2 プログラムの実行結果および考察

848,970 件の見出し語を持つ日本語 Wikipedia を辞書として本プログラムを実行した。Wikipedia には「曖昧さ回避」のページなど、一般の記事ではないページも多数存在する。これらのページは本プログラムの入力とするにはふさわしくないので、ヒューリスティクスによる除外ルールを設けた。付録 A に除外する見出し語の正規表現を掲載する。省略語として抽出するか否かを判定する類似度 $S(x_A,x_B)$の閾値は 0.2 と設定した。本プログラムの実行の結果、4,541 件の原型語とその省略語の組が得られた。

実際に得られたレコードの一部を図 3.3 に示す。各行はカンマで区切られた単語の列であり、先頭が原型語、2 番目以降が省略語を表す。図 3.3 は見やすさを考慮してグループ分けをしているが、グループ分けは人手で行ったものである。また、フジタク、シバレン、ホリケンなどは、見出し語の読みを使って獲得された省略語の例である。

図 3.4 に抽出誤り（抽出したが不正解のもの）・抽出失敗（抽出できなかったもの）の例を示す。左が原型語、右が誤抽出あるいは抽出に失敗した省略語である。

「十六進法」という Wikipedia の見出し語に対してその説明から「十進法」が省略語として抽出されているが、これは誤抽出である。「十六進法」と「十進法」は文字列条件 C_S、名詞条件 C_N が成立するので手順(2)では棄却できず、手順(3)で両者の類似度を計算した結果として棄却されるべきだが、実際には棄却できなかったことを示している。「準々決勝」と「準決勝」も同様である。原型語と省略語候補の類似度の閾値の調整で棄却することは可能であるが、実験による調整に頼るしかなく、適切な値とすることは困難な作業となる。一方、抽出失敗の例では、見出し語「スマートフォン」に対してその説明文中に出現する「スマホ」が抽出できなかった（「スマフォ」は抽出できた）。原型語中の文字列「フォ」が省略語では「ホ」に変わってしまっているため文字列条件 C_S が不成立となり抽出できない。見出し語「Mr.Children」は形態素解析プログラムで解析した結果未知語となり、読みがわからないので省略語候補「ミスチル」と文字列条件 C_S が不成立となる。見出し語「こちら葛飾区亀有公園前派出所」に対する省略語候補「こち亀」は、長い複合語から複数の名詞が省略された結果、名詞条件 C_N②が不成立となり抽出できなかった。

組織	生活
入国管理局, 入管	文房具, 文具
国際連合, 国連	ミスタードーナツ, ミスド
国土交通省, 国交省	通信販売, 通販
文部科学省, 文科省	投資信託, 投信
厚生労働省, 厚労省, 厚生省	セロハンテープ, セロテープ
経済産業省, 経産省	ビーフステーキ, ビステキ, ビフテキ
農林水産省, 農林省, 農水省	丸の内ビルディング, 丸ビル
テレビ朝日, テレ朝	新丸の内ビルディング, 新丸ビル
テレビ東京, テレ東	スポーツ／ゲームなど
テレビ埼玉, テレ玉	パシフィック・リーグ, パリーグ, パ・リーグ
マツモトキヨシ, マツキヨ	セントラル・リーグ, セ・リーグ
生活協同組合, 生協	マリオパーティ, マリパ
長期信用銀行, 長信銀, 長銀	一気通貫, 一通
東京電力, 東電	IT
日本弁護士連合会, 日弁連	ワードプロセッサ, ワープロ
人名	パーソナルコンピュータ, パソコン
藤岡琢也, フジタク	File Transfer Protocol, FTP
柴田錬三郎, シバレン	Domain Name System, DNS
浜田省吾, 浜省	Cascading Style Sheets, CSS
浜田幸一, 浜幸	User Datagram Protocol, UDP
松本潤, 松潤	Read Only Memory, ROM
堀内健, ホリケン	Random Access Memory, RAM
豊川悦司, トヨエツ	Document Object Model, DOM
松山ケンイチ, 松ケン	Local Area Network, LAN
松平健, マツケン	Common Gateway Interface, CGI

図 3.3 日本語 Wikipedia から抽出した原型語とその省略語の例

抽出誤り	
十六進法	十進法
準々決勝	準決勝
抽出失敗	
スマートフォン	スマホ
Mr.Children	ミスチル
こちら葛飾区亀有公園前派出所	こち亀

図 3.4 抽出誤りと抽出失敗

次に、本プログラムによる省略語抽出の精度と再現率を評価する。精度と再現率は省略語として抽出するための条件である類似度 $S(x_A,x_B)$の閾値に依存するため、ここでは閾値 0.2 と 0.1 の場合を比較した。閾値 0.2 と 0.1 の実行結果から式(2.1)の精度 P と式(2.2)の再現率 R を算出した。精度 P と再現率 R のパラメータである A,B,C は次のようにして求めた。ただし今回の実験では K=100 とした。

(a) 実行結果の CSV ファイルからランダムに$(t_A,\{t_B\})$のペアを K 個取り出す。初期値 0 の変数 A, B, C を用意する。

(b) すべての t_A について以下を繰り返す。

(ア) t_A の記事を Wikipedia から検索して記事を読む。後述の基準で省略語と認められる単語を集めて{t}とする。

(イ) $\{t_B\}$と{t}の要素を比較して、一致した要素の数を B に加算する。$\{t_B\}$には存在するが{t}には存在しない要素の数を A に加算する。逆に$\{t_B\}$には存在しないが{t}に存在する要素の数を C に加算する。

(a)のように実行結果ファイルからランダムに K 個取り出すため、そもそも実行結果ファイルに出力されない省略語が存在する場合は C がカウントされないことに注意する。上記手順(b)-(ア)では人手で Wikipedia の記事を読み、下記の基準で正解の省略語集合{t}を収集する。

1. 「略称は○○○である」「愛称は○○○である」「別称は○○○である」等と明記されている場合や括弧表現は正解として省略語集合{t}に要素を加える。
2. 文脈から明らかに同じ見出し語を指している省略語と思われる場合は正解

として省略語集合{t}に要素を加える。

3. 歴史的に旧名称（または現在の新名称）として使われていた場合は正解として省略語集合{t}に要素を加える。
4. 見出し語がカタカナ語でその説明に「発音はむしろ○○○に近い」という書き方で文字列条件 C_S が成り立つ○○○が記載されている場合は正解として省略語集合{t}に要素を加える。
5. 文字列条件 C_S が成立しない省略語は{t}に加えない。
6. 記事筆者のスペルミスは不正解として{t}に加えない。
7. 「よく○○○と呼ばれることもあるがそれは誤り」「よく○○○と混同されることがある」などと明記されている場合は不正解として{t}に加えない。

上記基準 2.は、Wikipedia の記事の筆者が執筆中に無意識に略称を作ってしまうことを考慮したものである。基準 3.は、歴史的な視点から説明文が書かれている場合に呼び名が変遷することを考慮したものである。見出し語が現代の名称の場合、説明文には過去から現在までの歴史が書かれ、過去の時代の名称が現代の名称の省略語に合致することがあり、この場合は正解とした。逆に見出し語が過去の名称の場合、説明文には現在から過去にさかのぼった歴史が書かれ、現代の名称が過去の名称の省略語に合致することがあり、この場合も正解とした。基準 4.は外来語がカタカナ表記される場合の表記揺れを考慮したものである。さらに基準 5.から 7.は不正解として{t}に加えない場合を明記した。基準 5.は、たとえば見出し語「国際通貨基金」の記事文中に「IMF」が略称として紹介されているが、文字列条件 C_S が成り立たないので{t}には加えないということを示している。

実験結果を表 3.2 に示す。また、100 個のエントリならびにそれらに対して算出したパラメータ A,B,C の値を付録 B に示す。再現率は高いが、精度は 70%以下にとどまった。また、一般に閾値を高く設定すると精度が向上するはずであるが、表 3.2 では閾値 0.2 よりも 0.1 の方が精度が高い。これは評価用のレコードをそれぞれでランダムに選んだためと考えられる。全レコードの人手によるチェックが難しい場合、今回のように原型語と省略語のペアをランダムに選ぶのではなく、Wikipedia のエントリをランダムに選択し、それに対して異なる閾値を設定してプログラムを適用した結果を評価・比較するべきである。このような評価は今後の課題としたい。

表 3.2 閾値別の原型語と省略語抽出の精度 P と再現率 R

	閾値=0.2		閾値=0.1	
	精度 P	再現率 R	精度 P	再現率 R
省略語	0.58	0.94	0.69	0.92
関連語	0.82	0.96	0.84	0.93

表 3.2 には省略語とは別に関連語の評価欄も設けている。ここで関連語とは、省略語ではないものの、広い意味で見出し語の別表記とみなすことで情報検索の観点から有益と認められるものである。関連語は省略語よりは条件が緩められている。たとえば、見出し語「ウェスリー・ブリスコ」に対して「ウェス・ブリスコ」は省略語ではないが関連語としている。その説明文には「ウェスリー・ブリスコ（Wesley Brisco、1983 年 2 月 21 日 - ）は、アメリカ合衆国のプロレスラー。現在はウェス・ブリスコ（Wes Brisco）のリングネームで WWE 傘下の FCW に所属。」とあり、情報検索の観点からは関連語として正解とみなすのがよいためである。関連語抽出の結果を省略語抽出の結果と比べると、再現率はあまり変わらないが、精度は大きく上回ることがわかる。

3.3 漢字送りがな表記揺れ知識の自動抽出プログラム

本節では、単語辞書から漢字送りがな表記揺れ知識を抽出するプログラムのアルゴリズムを説明し、次いで開発したプログラムの実行結果を示す。

3.3.1 漢字送りがな表記揺れ知識の抽出アルゴリズム

形態素解析プログラムが参照する単語辞書は、表層形である形態素文字列とその形態素の品詞と読みを含むレコードの集合になっているのが一般的である。本プログラムはこのレコード構造を利用する。アルゴリズムを以下に示す。

(1) 単語辞書から読みと品詞が同じ形態素の集合を 1 つのグループとして取り出す。
(2) そのグループから代表形態素を 1 つ選ぶ。代表形態素は漢字を最も多く含み、かつ最も長い文字列とする。
(3) グループの残りの要素のうち、代表形態素が持つすべての漢字を持っており、ひらがなは出現順序が一致するものを採用し、残りは棄却する。代表形態素とともに採用された形態素は CSV ファイル等に出力する。

(4) 単語辞書から取り出せるすべてのグループについて(1)～(3)を繰り返す。

具体例を用いて説明する。「ひっこし」という読みの名詞が単語辞書から図 3.5 のように 4 つ取り出されたとする。

引っ越し、引越し、引越、ひっ越し

図 3.5 単語辞書から取り出された 4 つの「ひっこし」

ここから最も漢字を多く含みかつ最も長い文字列を代表形態素として選ぶと「引っ越し」となる。残りの要素のうち、代表形態素「引っ越し」が持つすべての漢字を持っているのは「引越し」と「引越」である。これらについてはひらがなの出現順序も問題ない。「ひっ越し」は「引」という漢字を持たないので棄却される。よって CSV ファイルの 1 レコードとして「引っ越し, 引越し, 引越」が出力される。

3.3.2 プログラムの実行結果および考察

形態素数 392,126 件の単語辞書 IPAdic[15][3]に本プログラムを適用したところ、9,449 件の漢字送りがな表記揺れ知識が抽出できた。その一部を図 3.6 に示す。

お仕置,お仕置き	中高生,中・高生
下請け,下請	串焼き,串焼
不行き届き,不行届,不行届き	五重塔,五重の塔
並み大抵,並大抵	井の上,井ノ上,井上
冷や麦,冷麦	互い違い,互違い

図 3.6 IPAdic から抽出した漢字送りがな表記揺れ知識（一部）

ただし、使用した単語辞書では活用形も独立した見出し語として登録されているため、抽出した全 9,449 件の中には次のように原形と活用形のレコードがともに含まれる。

切かか,切りかか
切かかっ,切りかかっ

[3] 正確には、形態素解析ツール MeCab の付属辞書として配布された IPAdic を用いた。

切かから,切りかから
切かかり,切りかかり
切かかりゃ,切りかかりゃ
切かかる,切りかかる
切かかれ,切りかかれ
切かかろ,切りかかろ
切かかん,切りかかん
吸い付け,吸付け
吸い付けよ,吸付けよ
吸い付けりゃ,吸付けりゃ
吸い付ける,吸付ける
吸い付けれ,吸付けれ
吸い付けろ,吸付けろ
吸い付けん,吸付けん
　　：

図 3.6 より、漢字熟語のうしろにひらがなが送られる場合と送られない場合の表記揺れパターンである「下請け」と「下請」や「串焼き」と「串焼」、漢字熟語の中間にひらがなが送られる場合と送られない場合の表記揺れパターンである「並み大抵」と「並大抵」や「冷や麦」と「冷麦」、それら両者の組み合わせである「不行き届き」と「不行届」と「不行届き」、さらにはカタカナや中黒（・）の有無による表記揺れパターンである「井の上」と「井ノ上」と「井上」、「中高生」と「中・高生」が抽出できている。

こうして得られた漢字送りがな表記揺れ知識を 3.4 節で後述する N-best パス探索形態素解析プログラムに適用し、情報検索の転置インデックスを作成すれば、「冷や麦」という単語を使って書かれた文書は「冷や麦」と「冷麦」の両方の単語が転置インデックス上に展開されるので、検索者が「冷麦」という単語を使って検索しても、「冷や麦」という単語を使って書かれた文書が検索にヒットし、検索漏れが低減できる。

しかし「お祭り騒ぎ」という単語は、 辞書（たとえば JUMAN[16]の単語辞書）によっては「騒ぎ」が「さわぎ」とひらがな表記される場合があり、このときは本プログラムのアルゴリズムでは次のように 2 行に展開されてしまう。

お祭り騒ぎ,お祭騒ぎ
お祭りさわぎ,お祭さわぎ

情報検索の検索漏れを低減する目的に照らし合わせれば、この 2 行は 1 行で出力したい（同じ単語の表記揺れとして取り扱いたい)。しかしこのプログラムは漢字のうしろに位置する送りがなの有無による表記揺れを吸収しようとするものであるため、漢字自体がひらがな表記されてしまう場合を元のものと同一であると判断することができない。「騒ぎ」以外に漢字がひらがな表記されるものを探してみると、「回り」や「降りる」がある。

1. 立ち回り,立回り → 立ちまわり,立まわり
2. 触れ回り,触回り → 触れまわり,触まわり
3. 飛び回り,飛回り → 飛びまわり,飛まわり
4. 駆け回り,駆回り → 駆けまわり,駆まわり
5. 舞い降りる,舞降りる → 舞いおりる,舞おりる
6. 飛び降りる,飛降りる → 飛びおりる,飛おりる

「回り」は他の単語である「立ち」「触れ」「飛び」「駆け」と接続して複合語を形成し、その上で「まわり」とひらがな表記されることがあることを示している。同様に「降りる」も他の単語である「舞い」「飛び」と接続して複合語を形成し、その上で「おりる」とひらがな表記されることがあることを示している。このように、「回り」と「まわり」が使われている複数のサンプルを集め、読みや品詞が異なる 1.〜4.を横断的に見ることで「回り」が「まわり」と書かれることを認識できると「立ち回り,立回り」と「立ちまわり,立まわり」は 1 行に出力され、さらなる検索漏れの低減に寄与するものと考えられる。

IPAdic 以外にも JUMAN と UniDic[17]の単語辞書に対し本プログラムを適用した。抽出した件数を表 3.3 に示す。ただし、JUMAN の辞書はいくつかのサブ辞書から構成されているが、Wikipedia の見出し語を収録しているサブ辞書 Wikipedia.csv は除外した。

表 3.3 漢字送りがな表記揺れ知識の辞書別抽出件数

	形態素数 (a)	抽出件数 (b)	割合 (b/a)
IPAdic	392,126	9,449	0.024
JUMAN	583,476	31,105	0.053
UniDic	756,463	27,599	0.036

表 3.3 より、JUMAN からは最も多くの漢字送りがな表記揺れ知識を抽出でき、形態素数との比でも IPAdic の 2 倍以上となっている。IPAdic、JUMAN とも長い複合語が 1 形態素として登録されているが、特に JUMAN では複合語を形成している単語が漢字になる場合とひらがなの場合という組み合わせが比較的多いことが原因のひとつと考えられる。たとえば、「色取り取り」という 3 単語からなる複合語は、IPAdic では「色とりどり」だけが登録されている。これに対し、JUMAN では漢字送りがな表記揺れと漢字のひらがな表記のさまざまな組み合わせパターンも展開され、次のように 20 個もの形態素が登録されている。

色取り取り、色取り取、色取取り、色取取、色取々、いろ取り取り、
色取りどり、色とり取り、いろ取り取、いろ取取り、色取どり、
色とり取、いろ取取、いろ取々、いろ取りどり、いろとり取り、
色とりどり、いろ取どり、いろとり取、いろとりどり

3.4 N-best パス探索形態素解析プログラム

本節では、最小コスト法[6]に基づく N-best パス探索形態素解析プログラムについて述べる。本プログラムは MeCab[11]が公開している単語辞書を用いているため、最初に単語辞書について説明する。次いで本プログラムのアルゴリズムについて説明し、最後に実行結果と考察を述べる。

MeCab は N-best 解を出力する最小コスト法に基づく形態素解析の先行研究のプログラムである。MeCab は入力文から単語ラティスを作成し、コスト最小のものから上位 N 個のパスを取り出し、N 個のパス上にあるトークンをすべて出力する。形態素解析→構文解析→意味解析→文脈解析と解析の深度を進めていく自然言語処理において、MeCab の N-best 解出力は後段の解析への有用な情報である。一方本課題研究で開発する N-best パス探索形態素解析プログラム

が出力するトークンは、情報検索で使用するという目的に特化しているという点で MeCab の出力とは異なる。本プログラムはコスト最小のパス上の全トークンを出力した後、2 位以下 N 位までのパス上のユニークな名詞のみを出力する。ここでユニークな名詞とは、品詞が名詞であり、表層形の文字列と文書中での出現位置がそれまでに出力したトークンとは異なるものを指す。2 位以下の重複するトークンや検索される可能性の低い名詞以外の語を出力しないことで情報検索における処理負荷を軽減できる。また、2 位以下の解からも名詞を出力することで、単語区切りの不一致による検索漏れを防ぐ効果が期待できる。

3.4.1 使用する単語辞書

本プログラムは MeCab が公開している辞書を用いる。MeCab では IPAdic[15]、JUMAN[16]、UniDic[17]の各品詞体系に従った 3 つの独立した辞書が用意されている。3 つの辞書はそれぞれ CSV 形式の単語辞書ファイルと連接コスト表ファイルからなっている。単語辞書ファイルのレコード形式を図 3.7 に示す。同図の通り第 4 カラムまではすべての辞書で共通の内容である。第 5 カラム以降は辞書により記載内容が異なる。3 つの辞書とも第 5 カラム以降に形態素の品詞と読みの情報を持つが、カラム位置は辞書ごとに異なっている。

表層形, 左文脈 ID, 右文脈 ID, 生起コスト, …

図 3.7 MeCab が公開する単語辞書 CSV ファイルのレコード形式

表層形は形態素の文字列である。左文脈 ID・右文脈 ID は形態素の左右それぞれの文脈 ID であり、1 から始まる整数となっている。コーパスから単語の出現確率と品詞の連接確率を学習する際に機械的に割り振る ID であり、辞書により ID の総数は異なる。文脈 ID は後述する連接コスト表を引く際に参照する。生起コストは形態素の生起コストであり、単語ラティスにおけるノードのコストに対応する。一方、連接コスト表ファイルは図 3.8 に示すような 2 次元テーブル形式のファイルである。行方向に左文脈 ID、列方向に右文脈 ID を 1, 2, 3, …と並べていき、左右文脈 ID が交差する位置に連接コストの整数値が書かれたファイルとなっている。

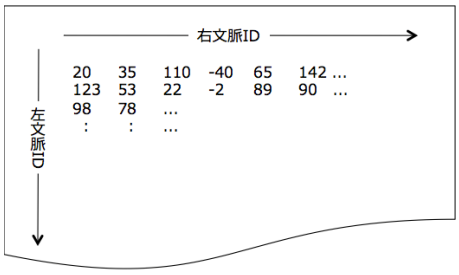

図 3.8 MeCab が公開する連接コスト表ファイルの形式

単語辞書における生起コストと連接コスト表における連接コストは、コーパスから条件付き確率場（Conditional Random Field; CRF）[9]で学習した確率から求めたコストであり、最小コスト法に基づく形態素解析プログラムが参照するコストとして利用できる。

本プログラムは前述の 3 つの辞書を使えるように開発した。N-best パス探索機能により 2 位以下のパス上のトークンは名詞しか出力しないという仕様のため、プログラムは単語辞書の CSV ファイルから品詞情報も読み込む必要がある。しかし品詞情報は辞書により CSV ファイル内でのカラム位置が異なるため、プログラム実行時に使用する辞書種別を引数として与えるようにした。プログラムは引数で与えられた辞書種別を参照し、フォーマットの異なる CSV ファイルを読み込むサブモジュールを切り替えることで品詞情報を読み込む。

3.4.2 単語ラティスの作成アルゴリズム

本形態素解析プログラムは、永田[5]の方法に基づき入力文の N-best パス解出力を行う。永田の方法では、最初に与えられた入力文を 1 文字ずつ前向きに解析し、図 3.9 に示すような単語をノードとするグラフ構造（単語ラティスと呼ぶ）を作成する。次に作成した単語ラティスからコストが小さい上位 N 個のパスを探索してパス上のトークンを順に出力する。よって、ここではまず、入力文から単語ラティスを作成する手順を説明する。

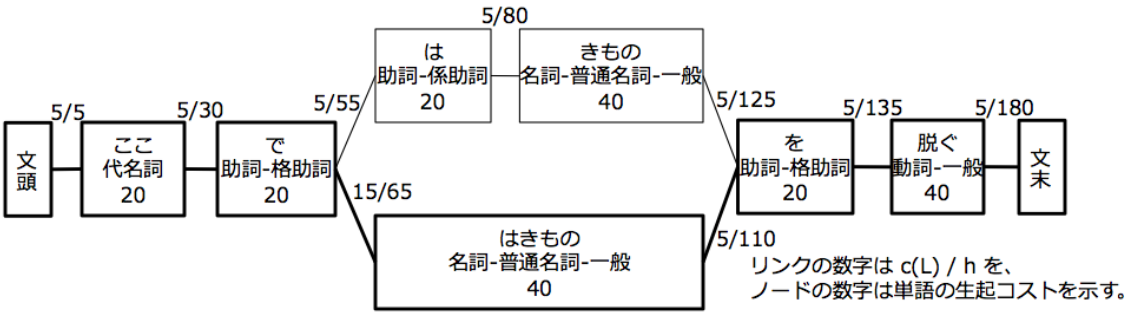

図 3.9 単語ラティスの例

単語ラティスを作成する手順は次の通りである。

(1) 入力文 S を受け取る。

(2) 文頭ノードと文末ノードを用意する。文頭ノードは位置 1 で終わるノードとし、文末ノードは位置|S|で始まるノードとする。ただし|S|は S の長さ（文字数）を表す。

(3) S の文字位置を示す変数 i を用意し、i=1…|S|の範囲で以下の処理を繰り返す。

① 位置 i の文字で始まる単語を単語辞書で検索する。単語がなければ i をインクリメントして再度検索する。

② 該当する単語をノードとして追加する。

③ 位置 i で終わるノード N_L の右文脈 ID と位置 i で始まるノード N_R の左文脈 ID を使って連接コスト表から連接コスト c(L)を求める。それらのノードの間にリンク L を張り、リンク L にひもづくパラメータ h に以下のように文頭からそこに至るまでのコストを割り当てる。

$$h = \min H(N_L) + c(N_L) + c(L) \qquad (3.2)$$

ここで $\min H(N_L)$はノード N_L に接続する左リンクのうち最小の h であり、$c(N_L)$はノード N_L の生起コストを示す。

単語ラティスの作成手順(3)を具体例を用いて詳述する。S=“ここではきものを脱ぐ”という入力文を、図 3.10 の単語辞書と図 3.11 の連接コスト表を使って形態素解析を行い、単語ラティスを作成する。まず文頭ノードと文末ノードを作成し、文頭ノードの右文脈 ID を 1、文末ノードの左文脈 ID を 7 とする。

まず i=1 のとき、「こ」（その次の文字は「こ」）で始まる単語を単語辞書で確認すると、「ここ」があるのでノードとして追加する。位置 1 で終わるノードは文頭ノードでありその右文脈 ID は 1、そして位置 1 で始まるノードは「ここ」

でありその左文脈 ID は 3 である。文頭ノードと「ここ」ノードの間にリンク L を張り、その連接コストは連接コスト表から c(L)=5 となる。このリンクの h を式(3.2)を使って求めると、N_L は文頭ノードだから $\min H(N_L)=0$, $c(N_L)=0$ であり h = 0 + 0 + 5 = 5 である。次に i=2 のとき、「こ」（その次の文字は「で」）で始まる単語は単語辞書にないので i をインクリメントする。すると i=3 となり、「で」を単語辞書で確認すると単語があるので「で」のノードを追加する。位置 3 で終わるノードは「ここ」でありその右文脈 ID は 3、位置 3 で始まるノードは「で」でありその左文脈 ID は 4 である。連接コスト表から両者のリンク L の連接コストは c(L)=5 である。このリンクの h を式(3.2)を使って求めると、N_L は「ここ」ノードだから $\min H(N_L)=5$, $c(N_L)=20$ であり h = 5 + 20 + 5 = 30 である。次に i=4 のとき「は」で始まる単語を単語辞書で確認すると「は」と「はきもの」が見つかるのでこれらをノードとして追加する。位置 4 で終わるノードは「で」でありその右文脈 ID は 4、位置 4 で始まるノードは 2 つあり、1 つは「は」でありその左文脈 ID は 5、もう 1 つは「はきもの」でありその左文脈 ID は 2 である。「で」と「は」は連接コスト表よりコスト 5、h は h = 30 + 20 + 5 = 55 となる。一方、「で」と「はきもの」は連接コスト表よりコスト 15、h は h = 30 + 20 + 15 = 65 となる。以下説明を省略するが、i=|S| まで同様の手順を繰り返す。

表層形	左文脈 ID	右文脈 ID	生起コスト	品詞
きもの	2	2	40	名詞-普通名詞-一般
ここ	3	3	20	代名詞
で	4	4	20	助詞-格助詞
は	5	5	20	助詞-係助詞
はきもの	2	2	40	名詞-普通名詞-一般
を	4	4	20	助詞-格助詞
脱ぐ	6	6	40	動詞-一般

図 3.10 単語辞書（例）

		右文脈 ID					
		1	2	3	4	5	6
左文脈 ID	2	5	5	10	15	5	10
	3	5	15	10	5	5	10
	4	100	5	5	20	20	20
	5	100	5	5	5	20	20
	6	20	10	10	5	5	10
	7	100	5	5	5	5	5

図 3.11 連接コスト表（例）

前述のアルゴリズムでは、文字位置 i の文字で始まる単語が単語辞書にないと位置 i で終わるノードとの間のリンクが途切れる場合がある。そこで位置 i の文字で始まる単語が単語辞書にないときは未知語として単語ノードを追加する。未知語ノードの単語生起コストは単語辞書中の最大の生起コストと同じ値を割り当てる。同じく未知語と他の単語（未知語を含む）との連接コストは連接コスト表中の最大コストを割り当てる。これらの処置により、未知語を含むパスはのちの探索処理にて選択されにくくなる。また、未知語の単語ノードを追加するときは、未知語が始まった位置の文字と同じ字種が続くまでを 1 単語とする。たとえば S=“ポテンシャルがある”という入力文において、i=1 のとき辞書引きに失敗したとすると、位置 1 で始まる文字「ポ」はカタカナなので、カタカナが連続する範囲の i=6 までの「ポテンシャル」が未知語として 1 単語に同定される。

3.4.3 単語ラティスの後向き N-best パス探索

本項では、作成した単語ラティスを後向きに探索して N-best パス解を出力する方法を説明する。後向き探索の手順は次の通りである。

(1) リンクの集合Ωを用意し、初期値Ω＝φ（空集合）とする。また第 N 位解を表す変数 n を用意し、初期値 n=1 とする。

(2) 文末ノードに接続するリンクをすべてΩに追加する。

(3) Ωから f が最小のリンク L を取り出す（Ω＝φのときは終了）。ただし、f は次のように計算される。

$f = g + h$ (3.3)

ここで g は L にひもづくパラメータであり、文末ノードから L までのコストを表す。文末ノードに接続するリンクの g はすべて 0 とする。

(4) (3)で取り出したリンク L に接続する左ノード N_L を調べる。N_L が文頭ノードならそれまでにたどったノード N_L を逆向きに出力すると第 n 位のパス上のトークンとなる。ただし、n>1 の場合はユニークな名詞のみ出力する。n<N なら n をインクリメントして(3)に戻り、そうでないなら第 N 位までの結果を出力したので終了する。

(5) N_L が文頭ノードでなければ N_L の左に接続する全リンクを Ω に追加する。なお Ω にリンクを追加する際は、追加するリンクにひもづく g を次のように計算する。

$$g = g(L) + c(L) + c(N_L) \quad (3.4)$$

ここで g(L)はリンク L にひもづく g である。

(6) (3)に戻る。

次に図 3.12 の具体例を用いて後向き探索を説明する。図 3.12 は図 3.9 の単語ラティスを文末から文頭に向かって探索する 2 つのパスを表示したものである。まず初期値 φ の Ω に文末ノードに接続する 1 つのリンクを追加する。次に Ω から f が最小のリンクを取り出すが、Ω には今追加したリンクがあるだけなのでそのリンクを取り出す。このときの f は式(3.3)より、f = 0 + 180 = 180 である。このリンクの左ノード N_L は「脱ぐ」なので、「脱ぐ」に接続する左リンクを Ω に追加する。ただしこのときの g は式(3.4)において g(L)=0, c(L)=5, $c(N_L)$=40 から g = 0 + 5 + 40 = 45 である。そして手順(3)に戻って Ω から f が最小のリンクを取り出すが、やはり Ω には先ほど追加したリンクがあるだけなのでそのリンクを取り出す。このときの f は式(3.3)より、f = 45 + 135 = 180 である。このリンクの左ノード N_L は「を」なので、「を」に接続する左リンクを Ω に追加する。「を」に接続する左リンクは 2 つある。2 つのリンクとも g は式(3.4)において g(L)=45, c(L)=5, $c(N_L)$=20 から g = 45 + 5 + 20 = 70 である。ただし、「を」から「はきもの」に延びるリンクの h は 110 であり、「きもの」に延びるリンクの h は 125 であため、それぞれの f は 70 + 110 = 180 と 70 + 125 = 195 である。そして手順(3)に戻って Ω から f が最小のリンクを取り出すが、このとき Ω には先ほど追加した 2 つのリンクがある。Ω から取り出される f が最小なリンクは「を」から「はきもの」に延びる f=180 のリンクであるから、これを取り出す。

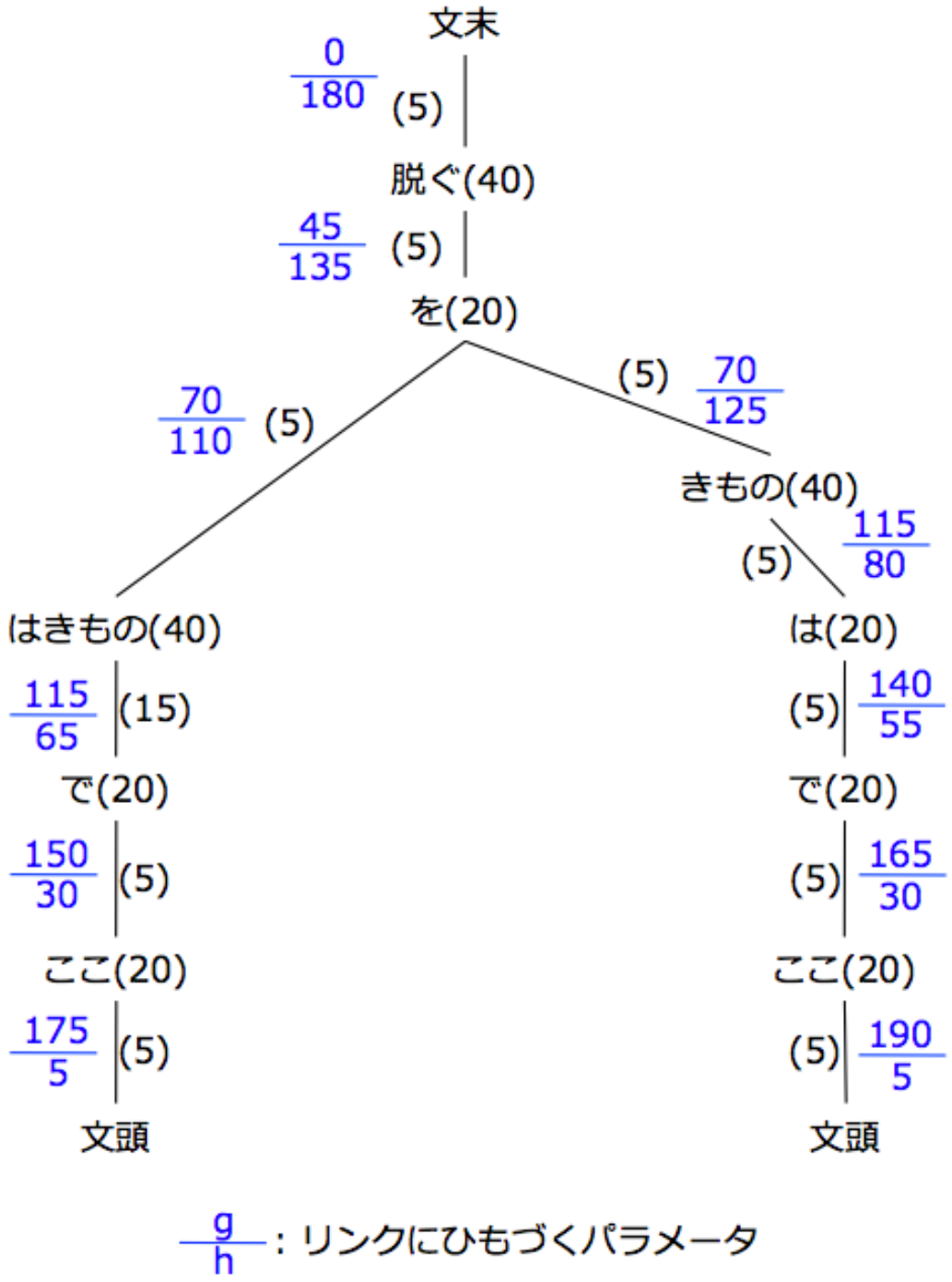

図 3.12 後向き N-best パス探索

このリンクの左ノード N_L は「はきもの」なので、「はきもの」に接続する左リンクを Ω に追加する。そのときの g は式(3.4)において g(L)=70, c(L)=5, $c(N_L)$=40 から g = 70 + 5 + 40 = 115 である。また h は 65 なので、f は式(3.3)

より f = 115 + 65 = 180 である。そして手順(3)に戻ってΩから f が最小のリンクを取り出す。このときΩには f が 195 と 180 の 2 つのリンクがある。f が最小なリンクは「はきもの」から「で」に延びる f=180 のリンクであるのでこれを取り出す。

この処理を繰り返して図 3.12 の文末からノード「脱ぐ」、ノード「を」そして左側のノード「はきもの」へと延びるリンク（すべて f=180 である）に連なるノードが文頭までたどられ、第 1 位のパス上のトークンとして出力される。そしてΩからその次に小さい f=195 のリンク（図 3.12 のノード「を」から右側に延びるリンク）が取り出され、以下同様にして文頭までたどられる。ただし、第 2 位以下のパスからはユニークな名詞のみ出力する。

3.4.4 ダブル配列による辞書実装と CSV ファイルの利用

形態素解析プログラムが使用する単語辞書の構成について考える。日本語の形態素解析では、入力文の各文字位置においてその文字位置から始まる文字列と一致するすべての単語を辞書から検索する必要がある。たとえば入力文 S=“きもの”をカナ漢字変換のために形態素解析する場合、1 文字目から始まる単語として「き（木／器／奇／…)」「きも（肝)」「きもの（着物)」、2 文字目から始まる単語として「も（藻／喪／模／…)」「もの（者／物)」、3 文字目から始まる単語として「の（野／乃／之／…)」を検索する。形態素解析プログラムにおいて処理時間の大半は辞書の検索に費やされ、使用メモリの大半は辞書の格納に使われる。そのため単語辞書は効率的にメモリに格納され、登録単語数に依存せず高速に検索できることが望ましい。

トライは日本語の形態素解析のための単語辞書に適したデータ構造であり、そのコンパクトな実装としてはダブル配列[1]が知られている。本プログラムではダブル配列を実装し、MeCab の単語辞書と「原型語とその省略語の自動抽出プログラム」および「漢字送りがな表記揺れ知識の自動抽出プログラム」が出力した CSV ファイルの内容を格納して利用できるようにした。本節ではその実装について述べる。

§ トライ

トライはキー集合 K の共通接頭辞を併合して作られる木構造である[2]。たとえば次のキー集合 K1 に対するトライを図 3.13 に示す。ただし#はキーの終端記

号を示す。

K1 = { babar#, baby#, bear#, beard#, bee# }

トライはオートマトンの一種である。よって、状態 n のときに文字 a を受け取って状態 m に遷移することを動作関数 g(n,a)=m と表すこととする。また初期状態 q_0 を持つ。たとえば図 3.13 のトライは g(1,b)=2, g(2,a)=3, g(2,e)=10, … という動作関数があり、初期状態は q_0=1 である。またこのオートマトンは終端記号#を受け取ると受理状態集合 F の 1 つの状態に遷移する。図 3.13 のトライの受理状態集合は F={7, 9, 13, 15, 17}である。受理状態は、単語辞書においてはその単語が辞書に掲載されていることを示す。

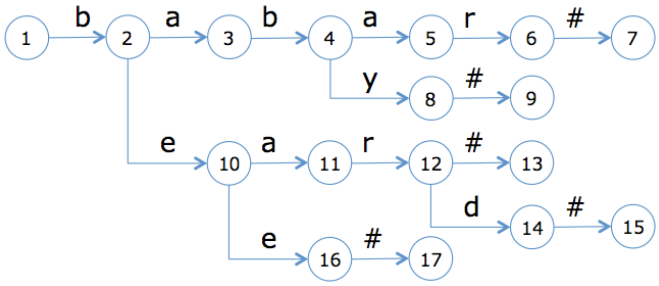

図 3.13 トライによる単語辞書

試しに S=“baby#”を図 3.13 のトライで検索すると、初期状態 q_0=1 から、g(1,b)=2, g(2,a)=3, g(3,b)=4, g(4,y)=8, g(8,#)=9 となり、9∈F のため受理状態となる。一方 S=“be#”では同じく初期状態 q_0=1 から、g(1,b)=2, g(2,e)=10 と遷移するが、g(10,#)は未定義なので“be#”は受理されず、“be”は辞書に掲載されていないことがわかる。

トライの特徴のひとつは、入力文字列の左端より始まるすべての接頭辞（最左部分文字列）が 1 回の走査で探索できることである。たとえば前述の S=“きもの”のすべての最左部分文字列「き」「きも」「きもの」が 1 回で探索できる。トライのもうひとつの特徴は、節から出る枝が一定時間で探索できるならば、キー検索の計算量はキーの総数に関係なく、キーの長さに比例することである。

§ トライのダブル配列表現

トライはダブル配列によってコンパクトに表現できる。ダブル配列は 2 つの配列 base と check を用いてトライの遷移関数 g(n,a)=m を次のように表現する。

m = base[n] + a
check[m] = n

つまり、状態 n のときに文字 a を受け取った場合は base 配列の n 番目の要素の値に文字 a の内部コードを加算して m を得て、check 配列の m 番目の要素の値が n と等しいかチェックする。等しい場合は状態遷移が成功したことを示し、等しくない場合は状態遷移が失敗したことを示す。また base[n]=0 のときは終了状態に到達したことを示し、これ以上状態遷移を行わない。この動作を図 3.14 の例を用いて説明する。

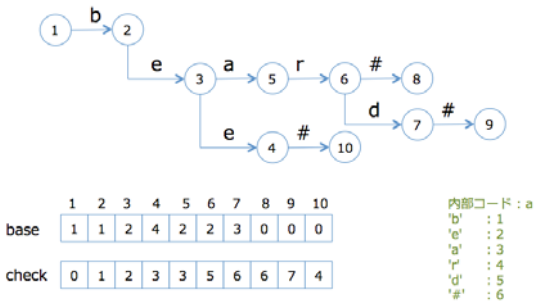

図 3.14 トライのダブル配列表現の例

S=“bee#” を初期状態 q_0=1 からダブル配列上で確認すると次のようになり、最終状態 10 において base[10]=0 となるのでこの文字列は受理される。

base[1] + ‘b’ = 1 + 1 = 2, check[2] = 1 : OK
base[2] + ‘e’ = 1 + 2 = 3, check[3] = 2 : OK
base[3] + ‘e’ = 2 + 2 = 4, check[4] = 3 : OK
base[4] + ‘#’ = 4 + 6 = 10, check[10] = 4 : OK
base[10] = 0 : ”bee#”を受理

別の文字列 S= “bed#” では次のようになり、この文字列は受理されないことがわかる。

base[1] + ‘b’ = 1 + 1 = 2,　check[2] = 1　　: OK
base[2] + ‘e’ = 1 + 2 = 3,　check[3] = 2　　: OK
base[3] + ‘d’ = 2 + 5 = 7,　check[7] = 6　　: NG（∵ 3≠6）

なお、トライ構造の辞書から base 関数と check 関数を求めるアルゴリズムの説明は省略する。詳細は文献[1]を参照していただきたい。

§ 単語辞書と CSV ファイル内容の格納方法

本課題研究で開発する形態素解析プログラムは、MeCab の単語辞書に加え「原型語とその省略語の自動抽出プログラム」および「漢字送りがな表記揺れ知識の自動抽出プログラム」が出力した CSV ファイルの内容をトライに格納して参照できなければならない。しかし単語辞書レコードと CSV ファイルレコードは情報が異なるため工夫が必要である。そこで本プログラムでは図 3.15 のように任意の情報（可変長レコード）を格納する data ファイルに単語辞書や CSV ファイルの情報を保存する。そして別途 data 内のレコードの先頭アドレスを引く辞書を用意し、トライの最終状態には 0 の代わりにこの辞書のインデックス k（k>0）を格納するようにした。ただし、正の k の値をこのままトライに格納すると遷移先状態番号と区別がつかなくなるので、k ではなく負号をつけて-k を格納した。

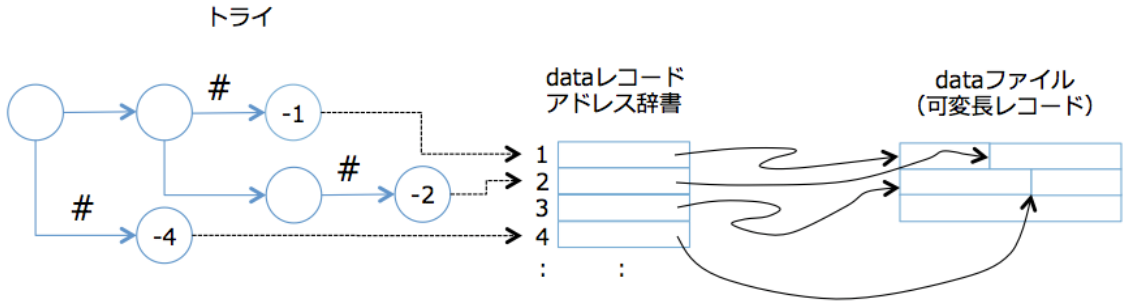

図 3.15 任意の辞書エントリを格納するためのデータ構造

3.4.5 プログラムの実行結果および考察

開発した形態素解析プログラムによる単語分割の例を以下に示す。

入力文：ここではきものを脱いでください。

ここ	代名詞
で	助詞-格助詞
は	助詞-係助詞
きもの	名詞-普通名詞-一般
を	助詞-格助詞
脱い	動詞-一般
で	助詞-接続助詞
ください	動詞-非自立可能
。	補助記号-句点

上は「ここではきものを脱いでください。」という文を N=1 で形態素解析した結果である。名詞としては「きもの」のみが同定されていることがわかる。N=2 で形態素解析すると、次のように「きもの」に加え「はきもの」が出力される。

ここ	代名詞
で	助詞-格助詞
はきもの	名詞-普通名詞-一般
は	助詞-係助詞
きもの	名詞-普通名詞-一般
を	助詞-格助詞
脱い	動詞-一般
で	助詞-接続助詞
ください	動詞-非自立可能
。	補助記号-句点

なお辞書はひらがなの「きもの」と「はきもの」両方の名詞が含まれる UniDic を用いた。

入力文：旭が丘へ引っ越しました。

旭が丘　　名詞-固有名詞-地域-一般
へ　　助詞-格助詞-一般
引っ越し　　動詞-自立
まし　　助動詞
た　　助動詞
。　　記号-句点

上は「旭が丘へ引っ越しました。」という文を、N=1 で形態素解析した結果である。辞書は IPAdic を用いた。同じ条件でさらに漢字送りがな表記揺れ知識の CSV ファイルを適用すると次のようになる。

旭が丘　　名詞-固有名詞-地域-一般
旭丘　　名詞-固有名詞-地域-一般
へ　　助詞-格助詞-一般
引っ越し　　動詞-自立
引越し　　動詞-自立
まし　　助動詞
た　　助動詞
。　　記号-句点

元の入力文の「旭が丘」「引っ越し」とは表記の異なる「旭丘」「引越し」が出力されている。この出力を使って転置インデックスを作成すれば、情報検索における再現率の向上が期待できる。

第4章　情報検索の絞り込み検索による検索誤りの漸次的低減

前章では検索漏れの問題と対策について検討し、実際に検索漏れを発生しにくくするためのプログラムを開発した。検索漏れを発生しにくくするということは再現率の向上につながる。しかし既に述べたとおり、再現率と精度はトレードオフの関係があるため、検索漏れ対策により検索精度は一般的に低下する。この問題に対し、本課題研究では検索精度向上を直截的に追求することは避け、情報検索の絞り込み検索機能を活用することでユーザの総合的な利便性向上を目指す。

そこで本章では最初に絞り込み検索について説明し、次に固有表現抽出により抽出した固有名詞が絞り込み検索の情報として有効に働くことを示す。最後に開発した固有表現抽出プログラムとその実行結果および考察を述べる。

4.1 固有表現抽出と情報検索システムへの応用

自然言語処理の基礎技術である固有表現抽出を情報検索に適用することで、検索誤りを漸次的に低減させることができることを示す。

4.1.1 情報検索の絞り込み検索機能

情報検索における絞り込み検索とは、ユーザが最初に実行したクエリに加えて新しい検索条件を追加して再検索することを指す。「絞り込み」検索のため、追加する検索条件は既存の検索条件に AND でつながれる。簡単に使えるように、Web システムであればサーバ側からユーザの GUI に絞り込み条件がリンク等で提示されるのが一般的である。例として、パッケージツアーの販売サイトでユーザが「ハワイ」と検索したときにシステム側から提示される絞り込み検索用のリンクを図 4.1 に示す。

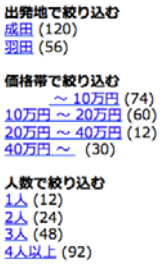

図 4.1 Web システムにおける絞り込み検索用リンクの例

ユーザは検索結果を見ながら必要なだけ絞り込み検索を実行できる。たとえば、最初に「ハワイ」というクエリを実行して次の S_1 という結果を得る。

S_1 = {“ハワイ”で検索した結果の文書集合}

ユーザが $|S_1|$ が大きいと感じると、次に自身の希望と照らし合わせて「出発地＝成田」の絞り込みリンクをクリックして次の S_2 を得る。

S_2 = {“ハワイ”で検索した結果の文書集合} ∩
{出発地が成田の文書集合}

ユーザがまだ $|S_2|$ が大きいと感じれば、さらに「価格帯＝10 万円以下」の絞り込みリンクをクリックして次の S_3 を得る。

S_3 = {“ハワイ”で検索した結果の文書集合} ∩
{出発地が成田の文書集合} ∩
{価格帯が 10 万円以下の文書集合}

この一連の操作は、式(2.1)においてユーザ自らが $|A|$ を小さくし、精度を 100%に近づけようとすることに等しい。3 章で論じた検索漏れ対策を行うことによりそれとトレードオフの関係にある検索誤りが増えるが、このような絞り込み検索機能を組み合わせることで簡単なユーザ操作により検索誤りを漸次的に小さくしていくことが可能である。

このように便利な絞り込み検索機能であるが、適用するためには検索対象文書レコードが図 4.2 のような構造を持っていなければならない。

ツアー名	出発地	価格	人数
ハワイ通の旅	成田	150,000 円	1
ハワイ島一周	羽田	240,000 円	2

図 4.2 絞り込み検索に適した構造を持つ文書の例

図 4.2 の最初のカラム「ツアー名」がユーザのクエリ（先の例では「ハワイ」）の検索対象となるカラムである。残りのカラム「出発地」「価格」「人数」が図 4.1 のリンクと対応しており、絞り込み検索に利用される。

新聞記事などをはじめ、多くの文書はこのような構造を持っていないので絞り込み検索を容易に実行することができない。しかし、固有表現抽出技術を適用すると、絞り込み検索に適した構造を持たせることができる。次項でこのことを示す。

4.1.2 固有表現抽出と絞り込み検索への適用

固有表現抽出とは、自然言語で書かれた文書から固有名詞を属性つきで抽出するタスクである。図 4.3 に固有表現抽出の例を示す。この例では、与えられた入力文に固有表現抽出を施すと、属性「人名」の値として「安倍晋三」という固有名詞が、属性「組織名」の値として「自民党」という固有名詞が、属性「地名」の値として「山梨県鳴沢村」という固有名詞が、そして属性「時刻」の値として「10 日」という固有名詞がそれぞれ抽出されている。

図 4.3 固有表現抽出の例

したがって、固有表現抽出で抽出可能な属性を文書レコードのカラムとして用意しておき、記事カラムの文書に対して固有表現抽出を適用すれば、前述の絞り込み検索が可能な構造を持った文書レコードを得られる（図 4.4）。

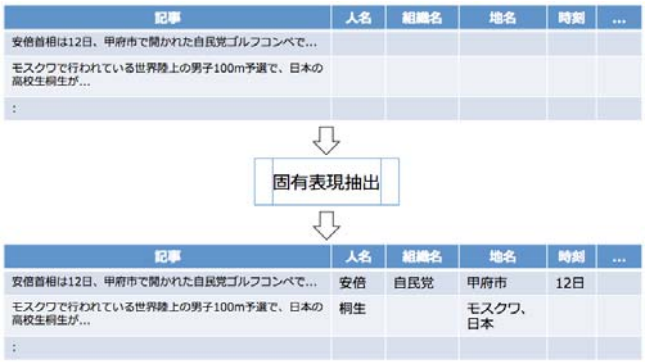

図 4.4 絞り込み検索のための固有表現抽出技術の適用

固有表現抽出は品詞タグ付けなどと同様、系列ラベリング問題の一種ととらえることができる。系列ラベリングとは、複数の要素から構成される系列 $\mathbf{x}_i$ (i=1,2,...)にラベルの系列 y_i を付与することを指す。たとえば、「安倍首相は自民党本部へ向かった」という文の品詞タグ付けは次のようになる。

安倍　　名詞-固有名詞-人名-姓
首相　　名詞-普通名詞-一般
は　　助詞-係助詞
自民　　名詞-固有名詞-一般
党　　接尾辞-名詞的-一般
本部　　名詞-普通名詞-一般
へ　　助詞-格助詞
向かっ　動詞-一般
た　　助動詞

これは「安倍／首相／は／...」という 1 つの単語を要素とする素性ベクトルの系列に「名詞-固有名詞-人名-姓／名詞-普通名詞-一般／助詞-係助詞／...」というラベルを割り当てる系列ラベリングとみなすことができる。同様に、固有表現抽出も系列ラベリングとみなせる。同じ文に対する固有表現タグの例を以下に示す。

安倍　　S-人名
首相　　O

は　　O
自民　　B-組織名
党　　E-組織名
本部　　O
へ　　O
向かっ　O
た　　O

固有表現は一般に複数の単語から構成されるため、ラベルの定義に工夫が必要である。IREX 日本語固有表現抽出タスク[8]では IOB1, IOB2, IOE1, IOE2 および SE と呼ばれるチャンクタグ手法が提案されている。チャンクタグ I はチャンクの内部、チャンクタグ B はチャンクの開始位置、チャンクタグ E はチャンクの終了位置、チャンクタグ O はチャンク外を表す。IOB1 ではチャンクタグ I, O, B を用いるがチャンクタグ B はチャンクが連続する際のチャンク境界におけるチャンクの開始位置にのみ付与する。IOB2 ではチャンクタグ I, O, B を用いるがチャンクタグ B はすべてのチャンクの開始位置に付与する。同様に IOE1 ではチャンクタグ I, O, E を用いるがチャンクタグ E はチャンクが連続する際のチャンク境界におけるチャンクの終了位置にのみ付与する。IOE2 ではチャンクタグ I, O, E を用いるがチャンクタグ E はすべてのチャンクの終了位置に付与する。SE ではチャンクタグ I, O, B, E, S を用い、1 トークンでチャンクになるものにチャンクタグ S を割り当てる。1 トークンでチャンクにならない場合にそのチャンクの開始位置にチャンクタグ B を、終了位置にチャンクタグ E を割り当てる。固有表現抽出ではチャンクは固有表現に対応する。上の例は SE 方式による固有表現タグである。この文では「自民党」の 2 語で 1 つの固有表現（組織名）を表すが、「自民」に B-組織名、「党」に E-組織名を割り当てることで「自民／党」がひとつのチャンク（固有表現）であることを示している。

4.2 固有表現抽出手法

本課題研究における固有表現抽出の手法を述べる。予備実験から、トークナイズの最小単位を前章で作成した N-best パス探索形態素解析プログラムにて単語分割した形態素とする。そこで、前処理として N-best パス探索形態素解析プログラムで入力文の単語分割を行う。ただし、N=1、すなわちベストの解のみ

を用いる。辞書は UniDic を利用した。

次に、各単語（トークン）にチャンクタグを割り当てる。タグチャンク方式は IOB2 を用いる。固有表現タグ付きコーパスから、入力文における個々のトークンに対する固有表現タグを決定するモデルを機械学習する。機械学習アルゴリズムは CRF[9]を採用した。CRF の学習には工藤の開発した CRF++[10]を用いた。

訓練データとなる固有表現タグ付きコーパスとして、関根の固有表現タグ付きコーパス[12]を用いた。このコーパスにおける固有表現タグのセットは関根の拡張固有表現階層の定義に従っており、およそ 200 種類の固有表現タグから構成される。本課題研究では粒度の細かい関根の拡張固有表現階層のタグを以下に示す 9 個の新たなタグにまとめ直した。なおこれらのタグは、情報検索をビジネスの場面で利用することを想定して選定した。

ORGANIZATION	（組織）
FACILITY	（施設）
PERSON	（人名）
TITLE	（肩書き）
LOCATION	（場所）
PRODUCT	（製品）
EVENT	（イベント）
PLAN	（計画・政策・構想）
DATETIME	（日時）

これらのタグと関根のタグとの対応は付録 C に示す。

CRF の学習に用いた素性を以下に示す。

- トークン及びその前後 2 つのトークンの出現形
- トークン及びその前後 2 つのトークンの文字種

  なお、文字種は表 4.1 のように定義した。
- トークン及びその前後 2 つのトークンの品詞
- 1 つ前のトークンの固有表現タグ

上記の素性を利用するため、関根の固有表現タグ付きコーパスを以下のようなフォーマットに変換した。第 2 カラムはトークンの文字種、第 3 カラムは品

詞、第 4 カラムは固有表現タグである。固有表現タグは 9 種類の粗い分類に変換されている。なお、参考のため、学習に使用した CRF++のテンプレートを付録 D に示す。

```
「          OTHER 補助記号-括弧開          O
創造        OTHER 名詞-普通名詞-サ変可能    B-PLAN
と          HIRA  助詞-格助詞              I-PLAN
やさし      HIRA  形容詞-一般              I-PLAN
さ          HIRA  接尾辞-名詞的-一般        I-PLAN
の          HIRA  助詞-格助詞              I-PLAN
国造り      OTHER 名詞-普通名詞-一般        I-PLAN
の          HIRA  助詞-格助詞              I-PLAN
ビジョン    KATA  名詞-普通名詞-一般        I-PLAN
」          OTHER 補助記号-括弧閉          O
―          OTHER 補助記号-一般            O
―          OTHER 補助記号-一般            O
村山        OTHER 名詞-固有名詞-人名-姓    B-PERSON
首相        OTHER 名詞-普通名詞-一般        B-TITLE
、          OTHER 補助記号-読点            O
年頭        OTHER 名詞-普通名詞-一般        B-DATETIME
所感        OTHER 名詞-普通名詞-一般        O
```

表 4.1 形態素の文字種

文字種	意味
DIGIT	形態素がすべて算用数字であることを示す。
HIRA	形態素がすべてひらがなであることを示す。
KATA	形態素がすべてカタカナであることを示す。
ALPHA	形態素がすべてアルファベットであることを示す。
OTHER	上記以外（漢字や記号など）。

4.3 プログラムの実行結果および考察

関根の固有表現タグ付きコーパスから 12,000 文を抽出し、3 分割の交差検定

によって本課題研究で実装した固有表現抽出ツールを評価した。表 4.2 ならびに図 4.5 は、固有表現のタイプ毎の精度(PRECISION)、再現率(RECALL)、F 値(F)を示している。これは 3 分割交差検定の 3 回の試行の平均である。また、全ての固有表現に対する評価を「Total」に示した。

表 4.2 固有表現抽出の評価

	PRECISION	RECALL	F
DATETIME	0.9263	0.8941	0.9099
EVENT	0.6932	0.4496	0.5452
FACILITY	0.6527	0.4732	0.5483
LOCATION	0.8322	0.8758	0.8534
ORGANIZATION	0.7580	0.6570	0.7037
PERSON	0.8764	0.8245	0.8495
PLAN	0.7185	0.3362	0.4567
PRODUCT	0.5432	0.3226	0.4036
TITLE	0.8833	0.7342	0.8010
Total	0.8301	0.7333	0.7786

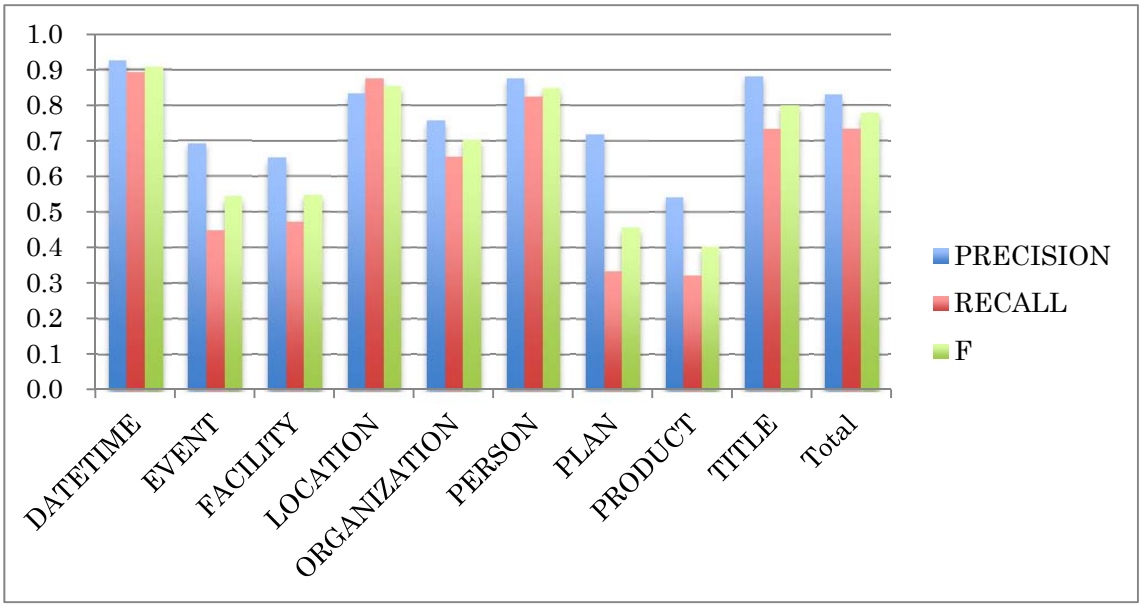

図 4.5 固有表現抽出の評価（表 4.2 のグラフ表示）

F 値で比較すれば、EVENT, FACILITY, ORGANIZATION, PLAN, PRODUCT, TITLE が他のタグに比べて値が低い。この原因の探究を試みた。まず、タグの種別毎に固有表現の出現頻度の分布を調べた。その結果を図 4.6 に示す。

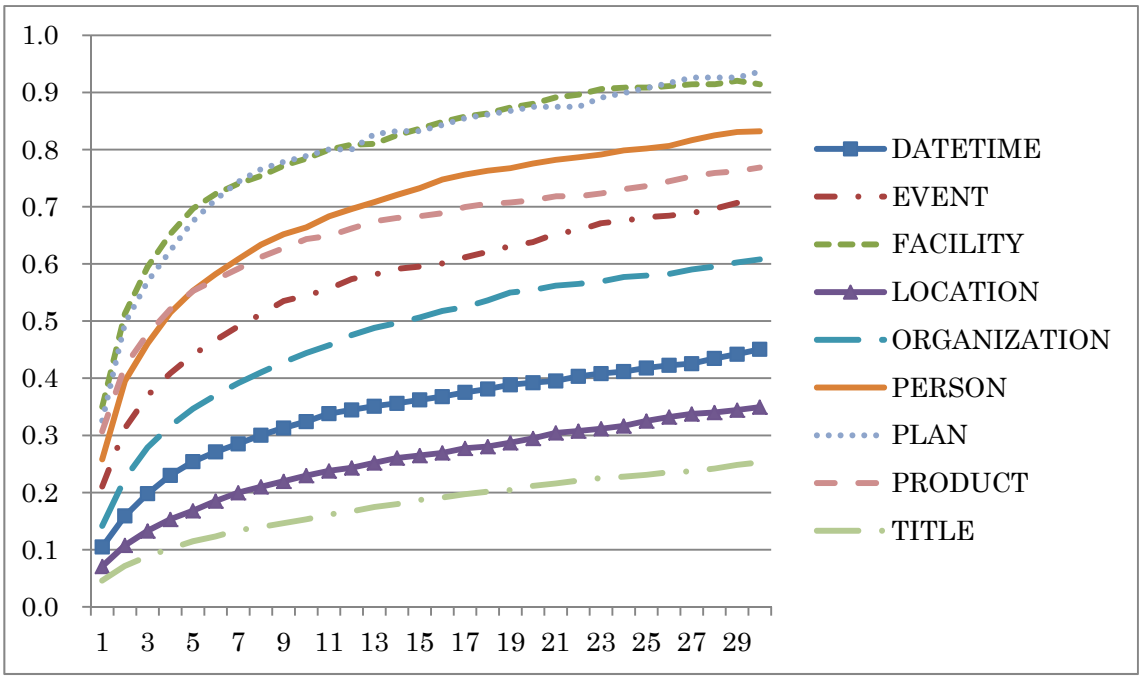

図 4.6 タグ種別ごとの固有表現の相対累積出現頻度

図 4.6 において破線で示したグラフは F 値が低かったタグである。横軸は固有表現の出現頻度 x を示し、縦軸は固有表現の累積出現頻度(出現頻度が x 以下の固有表現の出現頻度の和)を示す。ただし縦軸は相対値(固有表現の頻度の総和に対する割合)を示している。したがって、このグラフで上側にある線ほど、また立ち上がりが速い線ほど、出現頻度が低い固有表現が多い、言い換えれば多様な固有表現がコーパス中に出現していることを示している。このグラフによれば、EVENT, FACILITY, ORGANIZATION, PLAN, PRODUCT に関しては出現頻度が低い固有表現が多いことが F 値が低い原因のひとつと考えられる。しかし同じく出現頻度が低い固有表現の多い PERSON は F 値が高く、F 値が低い TITLE は出現頻度が低いキーワードは多いとはいえない。したがって固有表現の出現頻度の分布は F 値を左右する決定的な要因ではない。

次に、タグ別に固有表現の品詞の出現頻度をカウントした(詳細は付録 E を参

照)。すると F 値が低かった EVENT, FACILITY, ORGANIZATION, PLAN, PRODUCT, TITLE についてはいずれも品詞「名詞–普通名詞–一般」の出現頻度の順位が 1 位であったが、F 値が高かった DATETIME、LOCATION、PERSON は品詞「名詞–普通名詞–一般」は第 2 位以下となっていることがわかった。「名詞–普通名詞–一般」という品詞は一般性の高い品詞と考えられ、この品詞が出現頻度第 1 位となっているタグについてはよい成績をおさめることが難しいと考えられる。一方、PERSON は「名詞-固有名詞-人名-姓」「名詞-固有名詞-人名-名」「名詞-固有名詞-人名-一般」など、DATETIME では「名詞-数詞」「名詞-普通名詞-助数詞可能」「接尾辞-名詞的-助数詞」などの固有表現タグの意味に沿った特徴的な品詞の出現頻度が高いことがわかった。LOCATION に関しては品詞の出現頻度の 1 位から順に「名詞-固有名詞-地名-一般」「名詞–普通名詞–一般」「名詞-固有名詞-地名-国」となっており、やはり特徴的な品詞が多いことがわかる。

現在使用しているトークンの出現形、文字種、品詞、直前のトークンの固有表現タグ以外の素性を加えることも F 値を改善するための有効な手段と考えられる。先行研究として、あらかじめ構築した上位下位関係辞書を参照し、トークンの上位語を素性として利用する福島ら[7]、風間ら[3]の方法が提案されている。福島らの研究ではコーパスから次のカギ括弧表現を見つけ、concept を上位語、instance を下位語とする辞書を自動構築した。

concept「instance」

そして固有表現抽出のモデルを学習するときに、トークンが instance にマッチしたときにその concept を素性として用いた。同様に風間らの研究では、Wikipedia の見出し語を下位語に、その見出し語の説明文の第一文の最初の助詞「は」の後の名詞（「は」の後に複数の名詞が続く場合はその最後の名詞）を上位語として抽出して同様の上位下位関係辞書を構築した。特に福島らの研究ではテストデータの固有表現が学習データにも現われているかどうかの詳細な分析を行っており、上位下位関係辞書の効果が未知 NE（テストデータのみに現れ学習データには現れない固有表現）に対しては特に ARTIFACT において再現率が 5%以上も向上するなど効果が大きいという結果を示した。ただし既知 NE（学習データにもテストデータにも現れる固有表現）に対しては再現率を下げ

てしまう場合もある。これについて福島らは、上位下位関係辞書のノイズが原因だと考えており、辞書の精度を高めることで既知 NE の再現率を下げないようにすることができるかもしれないとしている。

第5章　おわりに

本課題研究では、情報検索のための自然言語処理ツール群を開発するために、最初に現在の情報検索が抱える問題点について整理した。情報検索の性能は精度と再現率という定量的な指標で評価できるが、両者はトレードオフの関係にあるので、両指標を同時に向上させることは困難である。そこで本課題研究ではまず再現率を向上させ、次いで絞り込み検索による精度の漸次的向上を達成することで、情報検索ユーザの総合的な利便性向上を目指すことを提案した。

再現率を向上させるということは検索漏れを低減させることと同義である。そこで検索漏れの要因を分析し、その対策として(1)原型語とその省略語の自動抽出プログラム、(2)漢字送りがな表記揺れ知識の自動抽出プログラム、(3)N-best パス探索形態素解析プログラムという 3 つのプログラムを開発した。(1)と(2)で抽出した知識を(3)の形態素解析プログラムで用いることで、検索漏れの要因をある程度取り除くことができた。しかし、未知語の存在を要因とする検索漏れへの対策は本研究課題では取り扱わなかった。未知語の同定および未知語の品詞（または文脈 ID）とコスト推定が今後の課題となる。

再現率を向上させたため、それとトレードオフの関係にある精度は低下する。そこで絞り込み検索を用いた精度の漸次的向上を行うことを提案し、固有表現抽出の技術を適用して検索対象文書に絞り込み検索のための情報を付加することを示した。固有表現抽出の学習とテストは既存のプログラムを用いたが、結果の評価と分析のためのプログラムを開発し、エラーの分析を行った。その結果、固有表現と関連の深い品詞タグを素性として利用できる固有表現タグに対しては比較的高い F 値が得られ、逆に「名詞-普通名詞-一般」という一般的な品詞の出現頻度が高い固有表現タグに対する F 値は低かった。また、他の素性の発見が有用な手がかりとなる可能性を示した。今後は先行研究が提案している他の素性を実際に試していきたいと考えている。

謝辞

本課題研究を進めるにあたり、日頃から研究方針および研究内容について的確なご指導を賜りました白井清昭准教授にこの場を借りて厚く御礼を申し上げます。また本課題研究を温かく見守ってくださいました島津明教授および機械学習全般について貴重な時間を割いてくださいました小谷一孔准教授、池田心准教授に厚く御礼申し上げます。

文献

[1] 青江 順一, “ダブル配列による高速ディジタル検索アルゴリズム,” 電子情報通信学会論文誌, J71-D(9) 1988, pp. 1592-1600.
[2] 青江 順一, “トライとその応用,” 情報処理, 34(2) 1993, pp. 244-251.
[3] 風間 淳一, 鳥澤 健太郎, “Web 上の資源から構築した複数の固有表現辞書を用いた日本語固有表現認識,” 言語処理学会 第 14 回年次大会発表論文集, 2008 年 3 月, pp. 813-816.
[4] 酒井 浩之, 増山 繁, “略語とその原型語との対応関係のコーパスからの自動獲得手法の改良, ” 自然言語処理, Vol. 12 (2005) No. 5, pp. 207-231, 2005.
[5] 永田 昌明, “前向き DP 後向き A*アルゴリズムを用いた確率的日本語形態素解析システム,” 情報処理学会研究報告, Vol.94, No.47 1994, pp. 73-80.
[6] 久光 徹, 新田 義彦, “ゆう度付き形態素解析用の汎用アルゴリズムとそれを利用したゆう度基準の比較,” 電子情報通信学会論文誌, D-II, J77-D-II(5) 1994, pp. 959-969.
[7] 福島 健一, 鍛治 伸裕, 喜連川 優, “コーパスからの固有表現辞書の自動構築,” 知識ベースシステム研究会 人工知能学会, 2007-12-03, 79, pp. 19-24.
[8] IREX 実行委員会, “IREX ワークショップ予稿集,” 1999.
[9] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data, In Proc. of ICML, pp.282-289, 2001.
[10] http://crfpp.googlecode.com/svn/trunk/doc/index.html
[11] https://code.google.com/p/mecab/
[12] http://www.nlp.cs.nyu.edu/ene/
[13] http://ja.wikipedia.org/
[14] http://gendaiyougo.jp/
[15] http://sourceforge.jp/projects/ipadic/
[16] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN
[17] http://sourceforge.jp/projects/unidic/

付録

A. 日本語 Wikipedia における不適切なエントリの除外のためのルール

原型語とその省略語の自動抽出プログラムにおいて、日本語 Wikipedia の不適切なエントリを除外するヒューリスティクス（正規表現による除外ルール）を以下に示す。^は文字列の先頭、$は文字列の末尾、.は任意の文字、+は 1 回以上の繰り返し、¥d は数字を表わす。

一覧	^Help:	の首相$	^明治.+年
における	^Category:	の国王$	^大正.+年
において	^Template:	形電車$	^昭和.+年
についての	^Portal:	系電車$	^平成.+年
に関する	^プロジェクト:	駅$	^¥d+年代$
の登場人物	^ファイル:	高等学校$	^¥d+月¥d+日$
の歴史	^日本の	中学校$	決議¥d+
県立	県$	小学校$	¥d+条
都立	市$	幼稚園$	^第¥d+
道立	区$	方法$	第.+回
府立	町$	事件$	第.+期
市立	村$	の旗$	第.+次
区立	郡$	行政区画$	道.+号.*線
町立	州$	^¥d+$	^オリンピック.+選手団$
村立	出入口$	^¥d+世紀$	^全国高等学校野球選手権.+大会$
曖昧さ回避	の大統領$	^¥d+年	

B. 原型語とその省略語の自動抽出プログラムの評価

原型語とその省略語の自動抽出プログラムにて、閾値が 0.2 のときの結果からランダムに 100 レコード選び評価した結果を下表に示す。数字は式(2.1)と式(2.2)における領域 A,B,C のカウント、括弧つきの数字は以下の基準（再掲）の番号である。

1. 「略称は○○○である」「愛称は○○○である」「別称は○○○である」等と明記されている場合や括弧表現は正解として省略語集合{t}に要素を加える。
2. 文脈から明らかに同じ見出し語を指している省略語と思われる場合は正解

として省略語集合{t}に要素を加える。

3. 歴史的に旧名称（または現在の新名称）として使われていた場合は正解として省略語集合{t}に要素を加える。
4. 見出し語がカタカナ語でその説明に「発音はむしろ○○○に近い」という書き方で名詞条件 C_N が成り立つ○○○が記載されている場合は正解として省略語集合{t}に要素を加える。
5. 名詞条件 C_N が成立しない省略語は{t}に加えない。
6. 記事筆者のスペルミスは不正解として{t}に加えない。
7. 「よく○○○と呼ばれることもあるがそれは誤り」「よく○○○と混同されることがある」などと明記されている場合は不正解として{t}に加えない。

必要に応じて実際の Wikipedia の説明文の抜粋または注釈を加えているものもある。

見出し語と省略語のレコード	省略語			関連語		
	A	B	C	A	B	C
スーパーマリン スピットファイア, スーパースピットファイア	1				1 (3)	
小松簡易裁判所, 小松簡裁		1			1	
フィリピン女子バスケットボールリーグ, フィリピンバスケットボールリーグ (※)「1998 年にフィリピンバスケットボールリーグ（PBL）の女子版として発足。」	1				1 (※)	
ウェスリー・ブリスコ, ウェス・ブリスコ (※)「ウェスリー・ブリスコ（Wesley Brisco、1983 年 2 月 21 日 - ）は、アメリカ合衆国のプロレスラー。現在はウェス・ブリスコ（Wes Brisco）のリングネームで WWE 傘下の FCW に所属。」	1				1 (※)	
Cryptographic API, CryptoAPI		1			1	
オーストラリアン・ブルドッグ, オールド	1			1		
フランシスコ, フランコ		1			1	
ケロロ RPG 騎士と武者と伝説の海賊, ケロン	1			1		
アイリッシュ・レッド・アンド・ホワイト・セッター, アイリッシュ・セッター (※)「アイリッシュ・レッド・アンド・ホワイト・セッター（英：Irish Red and White Setter）とは、アイルランド原産のセッター犬種である。日本でもよく知られている、アイリッシ	1				1 (※)	

ュ・セッターの原種でもある。」						
ドアをノックするのは誰だ?, ドアノック	1			1		
無伴奏チェロ組曲, 無伴奏曲 (※)「ヴァイオリンの無伴奏曲と同様にバッハ自身が弾くために書かれたとする説も浮上している。」	1				1 (※)	
中部電力, 中電		1			1	
日本税理士会連合会, 日税連		1			1	
宝永永字丁銀, 宝永永字銀	1			1		
紀宝町町民バス, 紀宝町民バス		1 (3)			1 (3)	
THE WAY HOPE GOES, THA	1			1		
ヒンドゥスタン・ユニリーバ, ヒンドゥスタン・リーバ		1 (3)			1 (3)	
汐留パートナーズグループ, 汐留グループ		1 (2)			1 (2)	
友松円諦, 友松諦 誤抽出「叔父の友松諦常に師事。」	1			1		
高級白板紙, 高板		1			1	
古典ラテン語, 古ラテン語	1				1	
森永北陸乳業, 森永乳	1 (6)			1 (6)		
辻堂西海岸, 辻堂海岸 (※)「南に辻堂海岸がある。」	1				1 (※)	
LISTSERV, LSTSRV	1				1	
雑誌記事索引, 雑索		1			1	
専当一心流, 専当流		1			1	
東京三菱銀行, 東銀	1		(5)	1		(5)
竹内畝流, 竹内流 (7)「竹内流と混同されることもある。」	1 (7)			1 (7)		
総武快速線, 総武線	1				1	
千賀ノ浦, 千賀浦	1			1		
アラスカ航空軍団, アラスカ軍	1				1	
アラスカ航空軍団, アラスカ軍団		1 (2)			1 (2)	
揚げ出し豆腐, 揚出し	1			1		
留萌簡易裁判所, 留萌簡裁		1			1	
アメリカ合衆国鉄道管理局, アメリカ鉄道管理局		1 (2)	(5)		1 (2)	(5)
国立環境研究所, 国環研		1			1	
気筒休止エンジン, 気筒エンジン	1			1		

ラストストーリー, ラスト		1			1	
みずほフィナンシャルグループ, みずほグループ	1				1	
キン肉マン, キマン	1			1		
門戸開放政策, 門戸開放策		1 (2)			1 (2)	
広州恒大足球倶楽部, 広州足球倶楽部		1 (3)			1 (3)	
鎌倉市指定景観重要建築物, 鎌倉市景観重要建築物		1 (2)			1 (2)	
ヘブライ文字, ヘブライ字	1				1	
レットキス, レトキ	1			1		
技術・家庭, 技家		1			1	
伊勢湾岸自動車道, 伊勢湾岸道		1			1	
名古屋短期大学, 名短		1			1	
銃器対策部隊, 銃対		1			1	
米沢市市民バス, 米沢市民バス	1				1	
日本報道写真連盟, 日報連		1			1	
延岡簡易裁判所, 延岡簡裁		1			1	
八戸短期大学, 八戸大学	1				1	
郵便認証司, 郵便司	1				1	
国際標準レコーディングコード, 国際標準コード	1				1	
ジョシュア・コックス, ジョシュ・コックス		1			1	
明姫幹線, 明幹		1			1	
三笠市民生活協同組合, 三笠市民生協		1 (2)			1 (2)	
どんど晴れ, どん晴		1			1	
且テイ侯, 且侯	1			1		
川崎中央郵便局, 川崎郵便局		1 (3)			1 (3)	
ばんえつ物語, ばん物 (※) カタカナのバンモノがとれなかった		1	1 (※)		1	1 (※)
パナソニック, パナック	1				1	
Crank'n Power, Cranker	1			1		
積乱雲, 積雲	1				1	
是貞親王, 是貞王		1			1	
神波憲人, 神憲	1				1	
柿崎憲家, 柿崎家	1				1	
田老テレビ中継局, 田老中継局		1 (2)			1 (2)	

盛岡西警察署, 盛岡警察署		1 (3)			1 (3)	
鋼の錬金術師の主要な登場人物, ハガレン	1				1 (2)	
秋田公立美術工芸短期大学, 秋田公立美術大学 (※) この仮称への改名計画があるということ	1				1 (※)	
中村簡易裁判所, 中村簡裁		1			1	
網走本線, 網走線		1 (3)			1 (3)	
イージートゥーン, イジトン		1			1	
天竹神社, 天竹社	1				1	
宝永正字丁銀, 宝丁銀	1			1		
宝永正字丁銀, 宝永丁銀	1				1 (3)	
労働金庫連合会, 労金連		1			1	
パンダコパンダ, パパンダ	1				1	
ブリーフ&トランクス, ブリトラ		1			1	
小樽市指定歴史的建造物, 小樽市歴史的建造物		1 (3)			1 (3)	
女神転生, メガテン		1			1	
大垣簡易裁判所, 大垣簡裁		1			1	
カザフの新疆脱出, カザフ脱出		1 (2)			1 (2)	
骨髄芽球, 骨髄球	1			1		
那覇簡易裁判所, 那覇簡裁		1			1	
File Transfer Protocol, FTP		1			1	
チチェスター, チチスター		1 (4)			1 (4)	
蛯原友里, エビユリ (※) エビが拾えなかった		1	1 (※)		1	1 (※)
水沢簡易裁判所, 水沢簡裁		1			1	
スーパーマリオブラザーズ, スーマリ (※) スーパーマリオが拾えなかった		1	1 (※)		1	1 (※)
軍事用ロボット, 軍用ロボット (※) 軍事ロボットが拾えなかった		1	1 (※)		1	1 (※)
自動アンケート作成, 自アン		1			1	
フランス国立宇宙研究センター, フランス宇宙研究センター		1 (2)			1 (2)	
全米知事協会, 全米知事会		1			1	
世界征服彼女, セカジョ		1			1	
DIGITAL MORNING DRIVE, DMD		1			1	

ジーザス&メリーチェイン, ジザメリ		1			1	
神はサイコロを振らない, 神サイ		1			1	
神戸電鉄粟生線, 神鉄粟生線		1 (2)			1 (2)	
Indoor MEssaging System, IMES		1			1	
合計	43	59	4	18	84	4

原型語とその省略語の自動抽出プログラムにて、閾値が 0.1 のときの結果からランダムに 100 レコード選び評価した結果を下表に示す。数字と括弧つきの数字は前述と同じである。

見出し語と省略語のレコード	省略語			関連語		
	A	B	C	A	B	C
マアレ・アドゥンミーム, マアレ・アドゥミーム		1 (2)			1 (2)	
五箇条の御誓文, 五箇条誓文		1 (3)			1 (3)	
西仙北スマートインターチェンジ, 西仙北インターチェンジ		1 (3)			1 (3)	
福山多度津フェリー, 福山フェリー	1				1	
キットピーク国立天文台, キットピーク天文台		1 (3)			1 (3)	
マリオパーティ 3, マリパ	1		1	1		1
日本映画・テレビスクリプター協会, 日本映画スクリプター協会		1 (3)			1 (3)	
木曽三川, 木曽川	1				1	
防衛医科大学校, 防医 (※) 防衛医大が拾えなかった		1	1 (※)		1	1 (※)
片山伯耆流, 片山流		1			1	
鰍沢警察署, 鰍沢署		1 (2)			1 (2)	
ドゥルミトル, ドルミト	1			1		
伊豆佐比売神社, 伊豆佐売神社	1				1	
農政全書, 農書	1				1	
城崎町文芸館, 城崎文芸館		1 (2)			1 (2)	
Men's Street, Mst		1 (2)			1 (2)	
地方入国管理局, 地方入管局		1			1	
実験動物中央研究所, 実中研		1			1	

聴覚脳幹誘発電位, 聴覚誘発電位	1				1	
アジア銀行家協会, アジア銀行協会		1			1	
バスケットボールマケドニア共和国代表, バスケットボールマケドニア代表		1 (2)			1 (2)	
プリンス・G 型エンジン, プリンスエンジン	1				1	
遠藤武彦, エンタケ		1			1	
インスタントラーメン発明記念館, インスタントラーメン記念館		1 (2)			1 (2)	
アットゥシ, アトゥシ		1			1	
オレゴン境界紛争, オレゴン紛争		1 (2)			1 (2)	
チャルトリスキ美術館, チャルトリスカ	1			1		
富里市農業協同組合, 富里市農協		1			1	
なう NOW スタジオ, ナウスタジオ	1			1		
エロ生☆パラダイス, エロパラ (※) エロ生が拾えなかった		1	1 (※)		1	1 (※)
辻アジア国際奨学財団, 辻国際奨学財団		1 (3)			1 (3)	
佐賀インターナショナルバルーンフェスタ, 佐賀バル	1			1		
沖縄県労働金庫, 沖縄労金		1			1	
名古屋中郵便局, 名古屋郵便局		1 (3)			1 (3)	
滋賀青年師範学校, 滋賀青師	1				1	
イングランド国教会, イングランド教会		1 (2)			1 (2)	
京丹後簡易裁判所, 京丹後簡裁		1			1	
D［di:］, Ddi	1 (6)			1 (6)		
巨文島, 巨島	1				1 (3)	
伊佐具神社, 伊佐具社	1				1 (3)	
名神大社, 名神社		1 (3)			1 (3)	
竹内畝流, 竹内流	1 (7)			1 (7)		
折田先生像, 折田像		1 (2)			1 (2)	
東近江簡易裁判所, 東近江簡裁		1			1	
ハッピーハードコア, ハピハコ (※) ハーコー、ハピコアが拾えなかった		1	2 (※)		1	2 (※)

酸化還元反応, 酸化反応		1			1	
エプスタイン・バール・ウイルス, エプスタイン・バーウイルス		1			1	
ベーシックマスター, ベーマス	1				1	
イヴェッチ・サンガロ, イヴェチ・サンガロ		1 (4)			1 (4)	
千葉信用金庫, 千葉信金		1 (2)			1 (2)	
兼高かおる世界の旅, 兼高かおる旅	1			1		
推力重量比, 推重比		1			1	
東文章・こま代, 東こま代	1				1	
合同ボランティアネットワーク, 合ボラ		1			1	
人体の不思議展, 人体展		1 (2)			1 (2)	
キツツキ亜目, キツツキ目	1				1	
累進課税, 累進税		1 (2)			1 (2)	
岡山電気軌道, 岡電		1			1	
多治見姫テレビ中継局, 多治見姫中継局		1 (2)			1 (2)	
幡豆テレビ中継局, 幡豆中継局		1 (2)			1 (2)	
教授言語, 教授語		1			1	
へそで茶をわかす, へそ茶		1			1	
テープレコーダー, テープコーダ		1 (3)			1 (3)	
七十門徒, 七十徒		1			1	
カライワシ上目, カライワシ目	1				1	
中津川簡易裁判所, 中津川簡裁		1			1	
明治薬学専門学校, 明治薬学校		1 (3)			1 (3)	
最小極大マッチング問題, 最大マッチング問題	1			1		
住友銀行, 住銀		1			1	
町田市農業協同組合, 町田市農協		1			1	
行政副主席, 行政主席	1			1		
性的快感, 性感		1 (2)			1 (2)	
日本・精神技術研究所, 日精研		1			1	
能登有料道路, 能登道路		1			1	
加古川運動公園陸上競技場, 加古川陸上競技場		1 (2)			1 (2)	

アカネテンリュウ, アカネリウ	1			1		
中尾憲太郎, ナカケン		1			1	
俺の屍を越えてゆけ, オレシカ (※) 俺屍が拾えなかった		1	1 (※)		1	1 (※)
オーストリアドイツ語, オーストリア語	1			1		
楊心流, 楊流	1			1		
アナハイム・エレクトロニクス社, アナハイム社	1				1 (3)	
延岡簡易裁判所, 延岡簡裁		1			1	
国際興業観光バス, 国際観光バス		1 (3)			1 (3)	
玉里島津家, 玉里家		1			1	
可部デジタル中継局, 可部中継局	1				1	
秋保電気鉄道, 秋保電鉄		1			1	
前田利家, 前田家	1			1		
XML Metadata Interchange, XMI		1			1	
黄海南道, 黄海道		1 (3)			1 (3)	
奥田順子, オクジュン		1 (2)			1 (2)	
西日本銀行, 西銀		1			1	
ビーロボカブタック, ビカブタック	1			1		
池田友政, 池田政	1			1		
青少年保護育成条例, 青少年保護条例		1			1	
青少年保護育成条例, 青少年条例		1 (3)			1 (3)	
中国プロバスケットボールリーグ, 中国バスケットボールリーグ		1 (3)			1 (3)	
君の居た昨日、僕の見る明日, キミボク		1			1	
青森中央郵便局, 青森郵便局		1 (3)			1 (3)	
Motorola ROKR E1, MotoROKR		1			1	
遠藤哲夫, エンテツ		1			1	
Universal Serial Bus, USB		1			1	
合計	31	70	6	16	85	6

C. 本課題研究における固有表現タグと関根の拡張固有表現階層の対応

固有表現抽出において、粒度の細かい関根の拡張固有表現階層のタグを粒度の粗い新たな固有表現タグにまとめて用いた。下記の設定ファイルは 1 行が「1

つの新しいタグ=複数の関根の拡張固有表現階層のタグ」という形式で書かれており、各行の右辺のタグを左辺の１つのタグで置き換えることを意味している。CRF の学習データを作成するプログラムではこの設定ファイルを読み込み、関根の固有表現タグ付きコーパスにおける固有表現タグを新しいタグに置き換えて出力するようにした。

また、#で始まる行はコメント行である。コメント行は、(a)「1 つの新しいタグ=複数の関根の拡張固有表現階層のタグ」というフォーマットのものと、(b)「関根の拡張固有表現階層のタグ」というフォーマットの行（(a)と区別するために行末に[*]を付記した）があり、これらの関根のタグは CRF の学習データを作成するプログラムにより削除される。新しいタグはビジネス利用の観点から定義したものであるが、(b)のフォーマットの行はビジネス利用はないと判断した関根のタグであり、(a)のフォーマットの行は予備実験の結果から 4 章の評価実験では未使用とした関根のタグである。

ORGANIZATION=International_Organization,Organization_Other,Political_Organization_Other,Pro_Sports_Organization,Show_Organization,Sports_Organization_Other,Company,Company_Group,Nationality,Ethnic_Group_Other,Government,Political_Party,Cabinet,Military,Family,Sports_League,Corporation_Other
FACILITY=Facility_Other,Facility_Part,Archaeological_Place_Other,Tumulus,GOE_Other,Public_Institution,School,Research_Institute,Market,Bridge,Road,Museum,Tunnel,Railroad,Amusement_Park,Park,Water_Route,Car_Stop,Theater,Worship_Place,Canal,Airport,Port,Sports_Facility,Line_Other,Station
PERSON=Person
TITLE=Title_Other,Position_Vocation
LOCATION=Country,County,Province,City,GPE_Other,Spa,Continental_Region,Domestic_Region,Geological_Region_Other,Region_Other,Mountain,Island,Lake,River,Sea,Bay,Latitude_Longtitude,Location_Other
PRODUCT=Drug,Clothing,Material,Art_Other,Movie,Music,Picture,Show,Stock,Weapon,Character,ID_Number,Product_Other,Service,Compound,Mineral,Element,Newspaper,Magazine,Book,Printing_Other,Broadcast_Program
EVENT=Award,Event_Other,Occasion_Other,Religious_Festival,Conference

,Natural_Disaster,Natural_Phenomenon_Other,Earthquake,Incident_Other,Game,War,Sport,Offense
PLAN=Plan,Academic,Theory,Style,Doctrine_Method_Other,Culture
DATETIME=Date,Period_Day,Period_Month,Period_Time,Period_Week,Period_Year,Periodx_Other,Era,Day_Of_Week,Time
#RULE=Rule_Other,Treaty,Law
#RELIGION=Religion,God
#FOOD=Food_Other,Dish
#VEHICLE=Aircraft,Car,Train,Ship,Spaceship,Vehicle_Other
#LIVING=Bird,Fish,Flora,Flora_Part,Animal_Part,Amphibia,Fungus,Mammal,Repitile,Name_Other,Living_Thing_Other,Living_Thing_Part_Other,Mollusc_Arthropod,Insect
#DISEASE=Animal_Disease
#AGE=Age,School_Age
#MONEY=Money,Currency
#COUNT=N_Organization,N_Animal,N_Country,N_Event,N_Facility,N_Flora,N_Location_Other,N_Natural_Object_Other,N_Person,N_Product,Countx_Other,Calorie,Intensity,Measurement_Other,Numex_Other,Percent,Point,Seismic_Intensity,Seismic_Magnitude,Space,Speed,Temperature,Volume,Weight,Multiplication,Frequency,Ordinal_Number,Physical_Extent,Class,Rank
#LANGUAGE=National_Language,Language_Other
#ACCESS=URL,Phone_Number,Address_Other,Postal_Address
#Color_Other[*]
#Natural_Object_Other[*]
#Nature_Color[*]
#Unit_Other[*]
#Astral_Body_Other[*]
#Constellation[*]
#Planet[*]
#Star[*]

D. CRF++のテンプレート

固有表現抽出モデルの学習の際に利用した CRF++のテンプレートファイルを示す。

```
# Unigram
U01:%x[-2,0]
U02:%x[-1,0]
U03:%x[0,0]
U04:%x[1,0]
U05:%x[2,0]

U11:%x[-2,1]
U12:%x[-1,1]
U13:%x[0,1]
U14:%x[1,1]
U15:%x[2,1]

U21:%x[-2,2]
U22:%x[-1,2]
U23:%x[0,2]
U24:%x[1,2]
U25:%x[2,2]

# Bigram
B
```

E. 固有表現タグごとの品詞の出現頻度

関根の固有表現タグ付きコーパス全体における、タグ付けされた固有表現の品詞の出現頻度（タグごとに上位 5 位まで）を示す。

DATETIME

固有表現の品詞	出現頻度
名詞-数詞	43137
名詞-普通名詞-助数詞可能	19932

接尾辞-名詞的-助数詞	6000
名詞-普通名詞-副詞可能	4921
名詞-普通名詞-一般	2502

EVENT

固有表現の品詞	出現頻度
名詞-普通名詞-一般	16361
名詞-普通名詞-サ変可能	4116
名詞-固有名詞-地名-一般	2705
接尾辞-名詞的-一般	2212
接頭辞	1950

FACILITY

固有表現の品詞	出現頻度
名詞-普通名詞-一般	10235
名詞-固有名詞-地名-一般	5099
接尾辞-名詞的-一般	3206
名詞-普通名詞-サ変可能	1023
名詞-固有名詞-一般	1022

LOCATION

固有表現の品詞	出現頻度
名詞-固有名詞-地名-一般	19940
名詞-普通名詞-一般	13545
名詞-固有名詞-地名-国	10974
接尾辞-名詞的-一般	903
名詞-普通名詞-助数詞可能	336

ORGANIZATION

固有表現の品詞	出現頻度
名詞-普通名詞-一般	25935
接尾辞-名詞的-一般	7760
名詞-固有名詞-一般	6067
名詞-普通名詞-サ変可能	5090
名詞-固有名詞-地名-一般	4465

PERSON

固有表現の品詞	出現頻度
名詞-固有名詞-人名-姓	19702
名詞-固有名詞-人名-名	14216
名詞-固有名詞-人名-一般	5326
名詞-普通名詞-一般	3686
補助記号-一般	1785

PLAN

固有表現の品詞	出現頻度
名詞-普通名詞-一般	3882
名詞-普通名詞-サ変可能	1424
接尾辞-名詞的-一般	721
名詞-数詞	304
接頭辞	218

PRODUCT

固有表現の品詞	出現頻度
名詞-普通名詞-一般	11623
名詞-普通名詞-サ変可能	2074
名詞-数詞	1624
助詞-格助詞	1457
接尾辞-名詞的-一般	1188

TITLE

固有表現の品詞	出現頻度
名詞-普通名詞-一般	25240
接尾辞-名詞的-一般	13908
名詞-普通名詞-サ変可能	5396
接頭辞	887
名詞-普通名詞-助数詞可能	695