修 士 論 文

# テキスト分類用辞書の自動学習

指導教官　佐藤理史 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

桜井 裕

1999 年 2 月 15 日

## 要 旨

テキスト自動編集システムの 1 つである Sun QA-Pack では、分野固有語辞書 (分類に利用する分野固有語とその分類カテゴリを記載した辞書) を用いて、テキストを分類する。この分類の精度は、分野固有語辞書の完全さに強く依存する。しかし、Sun QA-Pack が対象とする分野では、新たな分野固有語が頻繁に現れる。これを手作業でテキスト中から抽出し、その分類カテゴリを判断し、辞書に追加するのはかなりの労力を要する。本研究では、テキストから分野固有語を自動的に抽出し、分野固有語辞書に自動的に追加する機能の実現について検討した。

上記の機能は、分野固有語の抽出、分類カテゴリの推定、の 2 つを実現することによって達成できる。まず、テキスト中から、分野固有語と成り得る語 (分野固有語候補) を抽出する。次に、抽出した分野固有語候補の分類カテゴリを、その候補が出現する文の文脈を用いて推定する。分類カテゴリ推定では、Sun QA-Pack で用いている階層的分類の最も上位の分類カテゴリを推定することを行ない、その後、さらにより詳細な下位の分類カテゴリを推定することを行なった。この分類カテゴリ推定方法は、ニュースグループの記事以外のテキストに対しても適用可能である。ネットワーク上の色々なテキストを利用して分類カテゴリ推定を行なうこともできる。

以上の手法に対して評価実験を行なった結果、実現した手法を用いることで、そのテキスト中に存在する分野固有語の分類カテゴリをある程度推定することが可能であることが分かった。

# 目 次

# 第 1 章

# 序論

## 1.1 本研究の背景

近年、WWW(ワールドワイドウェブ) を代表とするネットワーク上の大量の電子データを個人が取り扱えるようになった。しかし、この中から必要な情報を素早く見つけ出すことは容易ではない。

これを解決する 1 つの方法として、テキストの自動編集がある [1][2]。これは、ユーザの情報探索が容易になるように、あらかじめテキスト情報を編集しておくものである。このような自動編集システムの 1 つに、Sun QA-Pack[3] がある。Sun QA-Pack は、Sun ワークステーションを対象としたニュースグループの記事を、幅広いユーザの利用を考慮に入れて、質問応答集としてパッケージ化したものである。Sun QA-Pack は、分類木を基本構造とすることで、「目次」式の検索手段を提供する。Sun QA-Pack のシステム概略は「収集」「選択」「組織化」「掲示」の 4 ステップからなる。「組織化」の 1 ステップとして、記事の分類が行なわれる。これは、ニュース記事の内容が分類木のどこに分類されるべきかを決定する処理であり、「目次」式の検索手段を提供するために不可欠な処理である。

Sun QA-Pack では、現在、ニュース記事に対して、以下のように分類を行なっている (図 1.1)。まず、記事本文から要約を生成する。次に、分野固有語辞書 (分類に利用する分野固有語とその分類カテゴリを記載した辞書) を用いて、要約中の分野固有語を抽出し、それぞれの分類カテゴリを得る。最後に、それらを集計し、最も数の多い分類カテゴリをそのニュース記事の分類カテゴリとする。

このような分類方法を取るため、Sun QA-Pack の分類精度は、既知データを用いた場合、85%であるが、未知データを用いた場合、64.5%となる。このことから、Sun QA-Pack の分類精度は、分野固有語辞書が分野固有語をどれだけ網羅しているかに強く依存する。

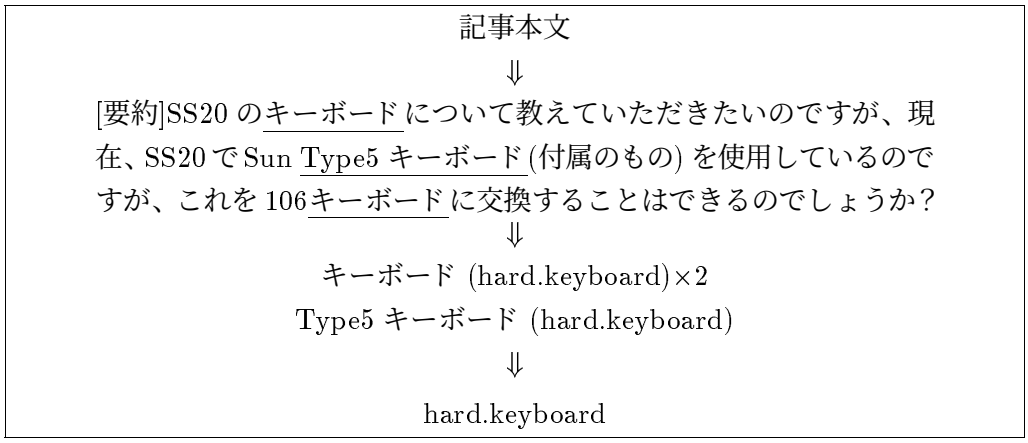


図 1.1: 現在の Sun QA-Pack の分類方法

しかし、Sun QA-Pack が対象とする分野では、新たな分野固有語が頻繁に出現する。これを手作業でテキストから抽出し、その分類カテゴリを判断し、辞書に追加するのはかなりの労力を要する。

## 1.2 本研究の目的

このような背景より、本研究では、テキストから分野固有語を自動的、あるいは半自動的に抽出し、分野固有語辞書に自動的に追加する機能の実現について検討する。これを以下の 2 つを実現することによって達成する。

1. テキスト中から分野固有語を抽出する。

2. 抽出した分野固有語の属する分類カテゴリを推定する。

## 1.3 本論文の構成

本論文の構成を、以下に示す。

第 2章では、テキスト中からどのようにして分野固有語を発見するかについて、それを実現するアルゴリズムを用いて述べる。

第 3章では、第 2章で得られた分野固有語候補の属する分類カテゴリを推定する方法を述べる。まず、この分類カテゴリ推定で中心的役割を果たす文脈を用いた推定方法につい

て述べた後、それを実現するアルゴリズムの説明をする。次に、詳細分類の方法について述べ、最後に、Web を利用した推定方法について説明する。

第 4章では、2章と 3章の方法について、それぞれ実験を行ない、検証を行なう。

第 5章では、本研究の結論、ならびに、今後の課題を述べる。

# 第 2 章

# 分野固有語の発見

本章では、電子ニュース記事群から分野固有語を発見する方法について述べる。まず、分野固有語の発見手法を説明し、次に、それを実現するアルゴリズムについて説明する。

## 2.1 分野固有語の発見

分野固有語辞書の学習のためには、まず、テキスト中から分野固有語を発見することが必要である。

Sun QA-Pack では、Sun ワークステーションを対象としたニュースグループの記事群を扱う。このニュースグループは、質問応答型ニュースグループであり、Sun ワークステーションに関する質問とそれらに対する応答が数多く掲載される。

Sun QA-Pack は、図 2.1のような分類木を基本構造とする。それぞれの節点が 1 つの概念を表し、親節点は上位概念を、子節点は下位概念を表す。それぞれの節点は、その節点が表す概念に関連した記事が分類される分類カテゴリとしての役割を持つ。

分野固有語辞書の一部を図 2.2に示す。このように分野固有語辞書には、分野固有語とその分野固有語が属する分類カテゴリを記載する。

分野固有語は分類に有効に働く語である。分野固有語に成り得る語として、(a) ある特定の概念を表す専門用語、(b) その概念 (分類カテゴリ) に属するプログラム名、システム名、製品名などの固有名詞、の 2 種類が考えられる。(a) は、数が限られているため、あらかじめ用意することが可能である。しかし、(b) は、頻繁に新たに出現する。このことより、テキスト中からプログラム名、システム名、製品名などの固有名詞を探し、これを分野固有語候補とする。

Sun QA-Pack が扱うニュースグループの記事群では、プログラム名、システム名、製

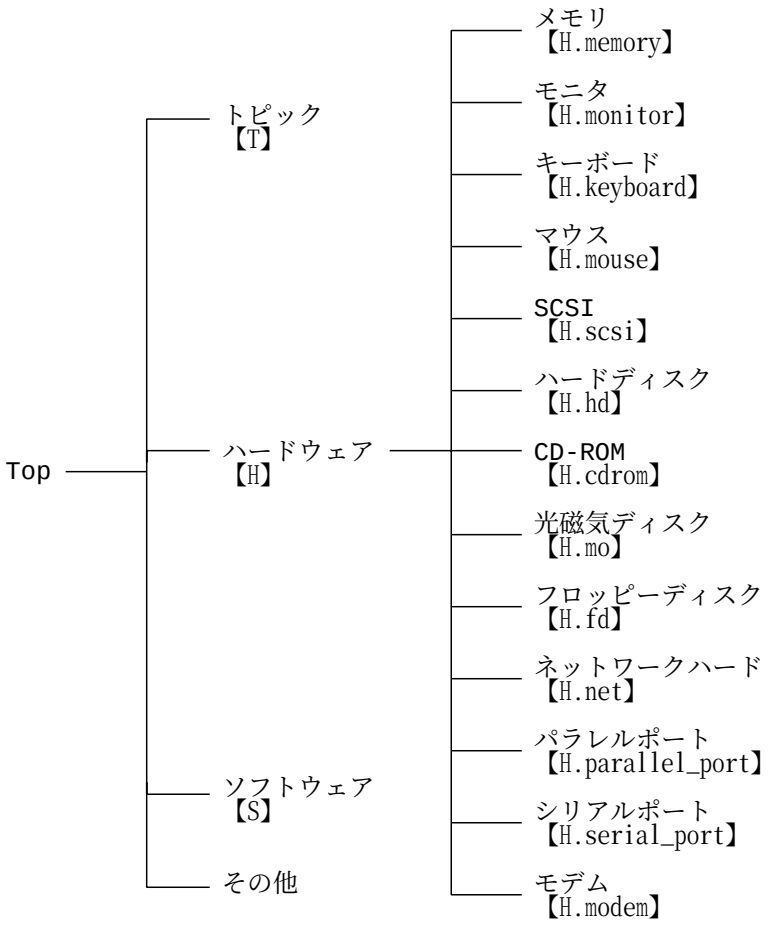


図 2.1: Sun QA-Pack の分類木 (一部)

品名は、例えば、「Sparc Station 20」、「SunOS4.1.4」、「ソラリス 2.6」のように、カタカナ、もしくは、英数字の並びで表記されることが多い。このことより、ニュース記事から、カタカナ、英数字列を抽出し、分野固有語の候補とする。

## 2.2 分野固有語発見アルゴリズム

分野固有語発見アルゴリズムの概略を図 2.3に示す。本アルゴリズムは、入力されたニュース記事中に存在する分野固有語候補の前後にタグを挿入する。本アルゴリズムは、大きく、(1) 文選別、(2) タグ付け、(3) 一般語の削除、 の 3 ステップからなる。

```
## OS
atx*                【O】
solaris*            【O】
                    ⋮
## マシン
ipc*                【M】
ss*                 【M】
                    ⋮
## ハードウェア
cd-rom              【H.cdrom】
cd-r                【H.mo】
                    ⋮
## ソフトウェア
gcc*                【S】
su                  【S.command】
                    ⋮
```

図 2.2: 分野固有語辞書 (一部)

## 文選別

文選別では、まず、入力されたニュース記事のテキスト (本文) を文に分割し、次に、得られた文の中から、実際にタグ付けを行なう文を抜き出す。

文への分割は、基本的に、各行を連結し、文末記号 (「。」、「.」、「？」) を用いて文に分割する方法を取る。但し、空行は、段落の切れ目を表すことが多いので、その前後の行は連結しない。また、特殊行[1]も、前後の行とは連結せず、独立した文として扱う。具体的には、以下のアルゴリズムで行なう。

### 文への分割アルゴリズム

1. テキストを 1 行入力する。これを Line とする。テキストが存在しない場合は、6 へ。

[1] 行の最初の文字が「>」、「#」などの記号であるような行。コマンドやエラーメッセージである場合が多い。

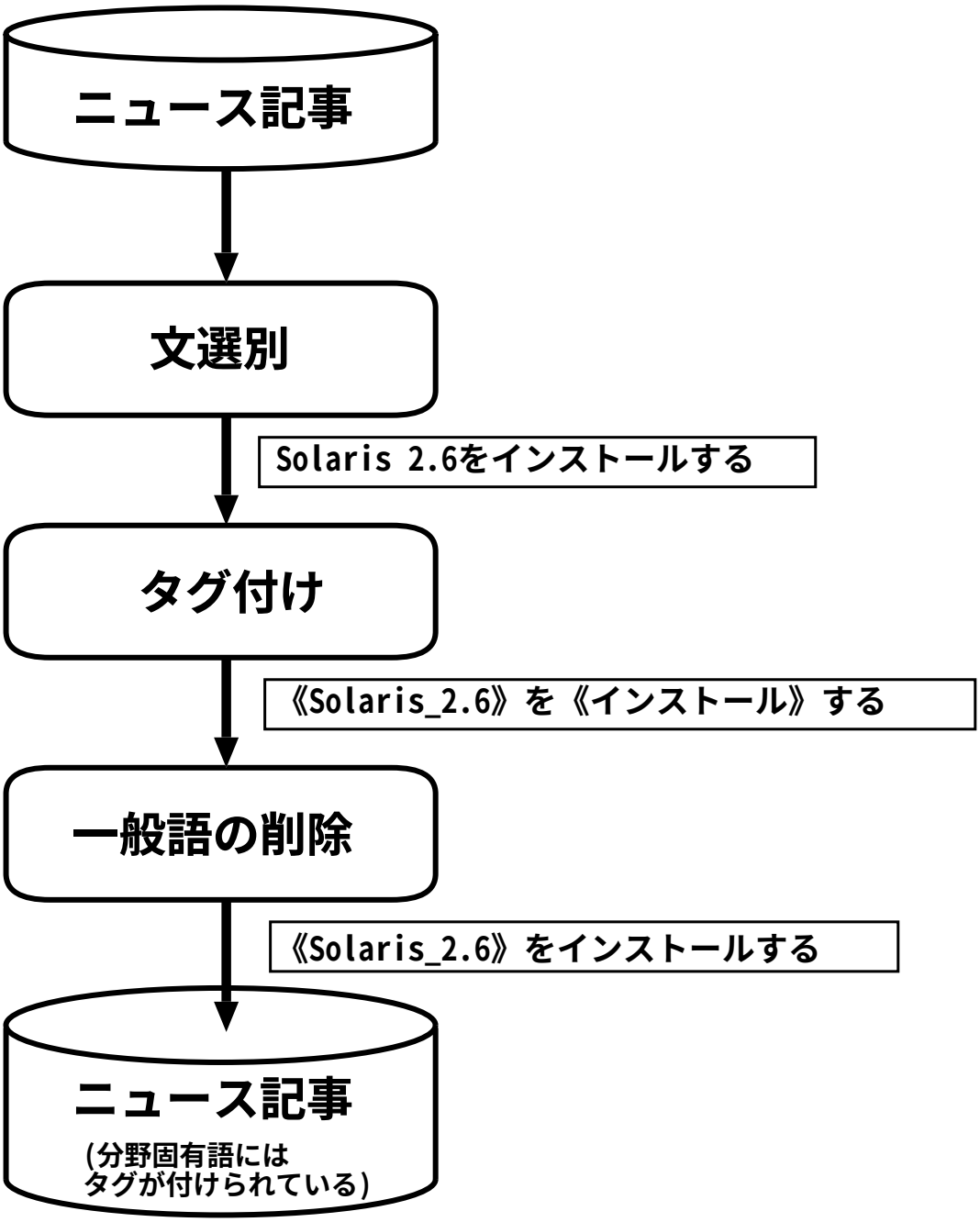


図 2.3: 分野固有語発見アルゴリズム

2. Line が空行であるか、特殊行である場合は、変数 BUF の内容を 1 文として出力する (但し、BUF が空でない場合)。Line が空でない場合は、それを 1 文として出力する。

3. それ以外の場合は、BUF の末尾に Line を連結する。

4. BUF に文末記号が存在する限り、その前方を切りだし、それを文として出力する。

5. 1 へ。

6. BUF の内容を 1 文として出力する (但し、BUF が空でない場合)。

次に、得られた文の中から、実際にタグ付けを行なう文を抽出する。文分割で得られた文の中には、明らかに以降の処理を行なう必要のない文や、タグ付けを行なうと誤りが多く発生する文が存在する。これらの文を排除する。

タグ付けを行なう必要がない文は、以下のような文である。

- カタカナ、英数字を 1 文字も含まない文
  タグ付けは、カタカナ、英数字列に対して行なうので、カタカナ、英数字が 1 文字も存在しない文は、タグ付けを試みる必要がない。

タグ付けを行なうと誤りが多く発生する文は、以下のような文である。

- コマンド、エラーメッセージ
  コマンドやエラーメッセージ (図 2.4) に含まれる英数字列は、多くの場合、分野固有語となることがない。1 文中に存在する英数字、記号の文字数が、それ以外の文字 (カタカナ、ひらがな、漢字) の 4 倍以上の場合は、コマンドやエラーメッセージを表す文と見なし、この文を排除する。

- 署名
  ニュース記事の末尾には、署名 (記事投稿者の氏名、所属、住所など) が書かれることが多い。この署名には、電子メールアドレスや URL を表す英数字列が含まれる場合があるが、これらは分野固有語とはならない。記事本文と署名の間には、多くの場合、図 2.5のように記号列だけの行が存在するので、記号列だけの文以降は署名とみなし、排除する。

- @を含んだ文
  「@」は、あいさつにおける「桜井@北陸先端大です。」や、フォロー記事での

```
% ypmatch foo.baa.co.jp hosts
Jan  9 10:56:14 machinename sendmail[294]: NOQUEUE: SYSERR(root):.......
/var/log/syslog に sendmail[8069]: UAA08061: UAA08069: return to sender:
```

図 2.4: コマンド、エラーメッセージ例

```
〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜
北陸先端科学技術大学院大学 情報科学研究科
知識工学講座 佐藤研究室 博士前期課程二年
桜井　裕
Email address:  yskr@jaist.ac.jp
URL:  http://www.jaist.ac.jp/~yskr/
〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜
```

図 2.5: 署名例

「yskr@jaist.ac.jp さんは書きました。」などのように、そのほとんどが分野固有語の存在しない文を構成するので、「@」を含む文は排除する。

文選別の実行例を図 2.6に示す。

## タグ付け

ここでは、以下の規則にしたがって、タグを挿入する。

**基本規則** カタカナ、英数字または、特殊な記号[2]が 2 文字以上連続して現れる場合、その前後にタグを挿入する。

[**例**] Gcc-1.8 をコンパイルしたいのですが、方法が分かりません。
⇒ 《Gcc-1.8》を《コンパイル》したいのですが、方法が分かりません。

[2]「.」「_」「-」「/」の 4 記号

但し、以下の場合は例外規則を優先する。

**例外規則 1** 空白をはさんで連続する英数字列は、それらを一体化してタグを付ける。

[**例**] Solaris 2.6 を Sparc Station 20 にインストールする。
⇒ 《Solaris_2.6》を《Sparc_Station_20》に《インストール》する。

**例外規則 2** 数字、記号だけの列にはタグを付けない。

[**例**] コンパイルに 24 時間かかる。⇒ 《コンパイル》に24時間かかる。

**例外規則 3** 数字列で始まる英数字列には、タグを付けない。

[**例**] 10GB のハードディスクが欲しい。
⇒ 10GBの《ハードディスク》が欲しい。

例外規則 1 は、例えば、「Sparc Station 20」のような複数の単語からなる単語列を、1 つの分野固有語としてタグ付けするためのものである。

記事中に存在する数字列の多くは、バージョン番号、箇条書きの番号、日付や時間、のいずれかである。このうち、バージョン番号は、例外規則 1 によって、直前の名称と一体化した形でタグが付けられる。残りの 2 つ (箇条書きの番号、日付や時間) は分野固有語となることはない。例外規則 2 は、これらを排除するためのものである。

数字列で始まる英数字列は、多くの場合、「300MHz」や「1GB」など、数量を表す表現である。これらも分野固有語となることはない。例外規則 3 は、これらを排除するためのものである。

実際のタグ付けは以下の方法で行なう。

1. 連続するカタカナ、英数字列を一体化させるために、カタカナ、英数字列間の空白を「_」に変換する。(例外規則 1 に対応)
2. 連続するカタカナ、英数字、特殊な記号の前後にタグを挿入する。
3. 開始タグの直後が数字ならば、そのタグ (開始タグと終了タグ) を削除する。(例外規則 2 および 3 に対応)

タグ付けの実行例を図 2.7に示す。

| |
|---|
| 3Com, Apple, CANON, Cemax, Cemex, Compaq, EPSON, Fujitsu, HP, IBM, Intel, IO-DATA, MELCO, NEC, OKI, Quantum, S3, Seagate, SGI, Sony, STB, Sun |

表 2.1: 企業リスト

| |
|---|
| アドバイス、*アドレス、アプリケーション、インストール、エラー*、*エラー、オプション、カード、*カラー、*キー、ケーブル、ゲット、コマンド、コンパイル*、サポート、システム、ソース、ソフト*、ダメ、チェック、データ、テスト、デフォルト、トライ、ネットワーク、バージョン、*バイト、バイナリー、バグ、パッチ、*ビット、ヒント、ファイル*、フォロー*、フリー、プログラム*、プロセス、*ヘルツ、ボード、ホスト、マシン、マニュアル* 、メール、*メッセージ、ユーザー* 、リンク |

表 2.2: 一般語リスト

## 一般語の削除

上記の方法では、カタカナ、英数字列にはすべてタグを付けるため、カタカナ、英数字列で記述される一般的に使われる語 (一般語) にもタグが付けられてしまう。ここでは、一般語を削除することによって、分野固有語候補だけを残す。

削除すべき一般語は、以下の 3 種類に分けられる。

1. 長過ぎる語
2. 「テスト」「エラー」のようなカタカナ表記される一般的に使われる語
3. 企業名 (「アップル」など)

1 は、文削除の際に、削除されなかったエラーメッセージなどにタグがついたものである。50 字以上の場合は、長過ぎる語として、そのタグを削除する。

2 と 3 に対しては、それぞれリストを用意し、そのリストに掲載されている場合は、一般語と見なし、そのタグを削除する。

企業名リストと一般語リストをそれぞれ表 2.1と表 2.2に示す。

一般語の削除は開始タグ直後、もしくは、終了タグ直前にも適用する。例えば、「《ls コマンド》」とタグの付いた語に対して、一般語リストに「コマンド」が登録されていれば、これを削除し「《ls》コマンド」とする。さらに、一般語は「*」をワイルドカードとして、

一般語の先頭、もしくは、末尾に付けることで、一般語リスト増加を防ぐことができる。例えば、「エラー*」により、「《エラーメッセージ》」のタグも削除可能である。但し、「*エラー*」のように前後ともに付けることはできない。

一般語削除の実行例を図 2.8に示す。

From: Takayuki Kato ¡kato.takayuki@sm1.bch.ntt.co.jp¿
Newsgroups: fj.questions.unix,fj.sys.sun
Subject: HP OpenView on SunOS4.1.4
Date: Mon, 06 Jan 1997 20:18:09 +0900
Organization: Nichido Kasai ,Tokyo ,Japan

加藤といいます。
HP OpenView をインストールしたことのある方に質問です。

現在 HP OpenView3.3 を SunOS4.1.4 にインストールしています。
無事、インストールは終了したのですが、ネットワーク状況をグラフィカルに表示させたときに、/etc/hosts に書いてあるホスト以外はすべて IP でしか表示されません。

SunOS は NIS を使用し DNS を引くように設定してあります。
ping はホスト名だけで動作させることができますし、

% ypmatch foo.baa.co.jp hosts

などで NIS 経由で DNS が引けていることも確認しています。

どこに問題があるのでしょうか？御教授下さい。
よろしくお願いします。
————————————
kato@nichido.co.jp
加藤隆行

⇓ 文分割

| # | sentence |
|---|---|
| 1 | 加藤といいます。 |
| 2 | HP OpenView をインストールしたことのある方に質問です。 |
| 3 | 現在 HP OpenView3.3 を SunOS4.1.4 にインストールしています。 |
| 4 | 無事、インストールは終了したのですが、ネットワーク状況をグラフィカルに<br>表示させたときに、/etc/hosts に書いてあるホスト以外はすべて IP でしか表示されません。 |
| 5 | SunOS は NIS を使用し DNS を引くように設定してあります。 |
| 6 | ping はホスト名だけで動作させることができますし、 |
| 7 | % ypmatch foo.baa.co.jp hosts |
| 8 | などで NIS 経由で DNS が引けていることも確認しています。 |
| 9 | どこに問題があるのでしょうか？御教授下さい。 |
| 10 | よろしくお願いします。 |
| 11 | ———————————— |
| 12 | kato@nichido.co.jp |
| 13 | 加藤隆行 |

⇓ 文削除

| # | sentence |
|---|---|
| 2 | HP OpenView をインストールしたことのある方に質問です。 |
| 3 | 現在 HP OpenView3.3 を SunOS4.1.4 にインストールしています。 |
| 4 | 無事、インストールは終了したのですが、ネットワーク状況をグラフィカルに<br>表示させたときに、/etc/hosts に書いてあるホスト以外はすべて IP でしか表示されません。 |
| 5 | SunOS は NIS を使用し DNS を引くように設定してあります。 |
| 6 | ping はホスト名だけで動作させることができますし、 |
| 8 | などで NIS 経由で DNS が引けていることも確認しています。 |

図 2.6: 文選別の実行例

| # | sentence |
|---|---|
| 2 | HP OpenView をインストールしたことのある方に質問です。 |
| 3 | 現在 HP OpenView3.3 を SunOS4.1.4 にインストールしています。 |
| 4 | 無事、インストールは終了したのですが、ネットワーク状況をグラフィカルに<br>表示させたときに、/etc/hosts に書いてあるホスト以外はすべて IP でしか表示されません。 |
| 5 | SunOS は NIS を使用し DNS を引くように設定してあります。 |
| 6 | ping はホスト名だけで動作させることができますし、 |
| 8 | などで NIS 経由で DNS が引けていることも確認しています。 |

⇓ タグ付け

| # | sentence |
|---|---|
| 2 | 《HP_OpenView》を《インストール》したことのある方に質問です。 |
| 3 | 現在《HP_OpenView3.3》を《SunOS4.1.4》に《インストール》しています。 |
| 4 | 無事、《インストール》は終了したのですが、《ネットワーク》状況を《グラフィカル》に<br>表示させたときに、《/etc/hosts》に書いてある《ホスト》以外はすべて《IP》でしか表示されません。 |
| 5 | 《SunOS》は《NIS》を使用し《DNS》を引くように設定してあります。 |
| 6 | 《ping》は《ホスト》名だけで動作させることができますし、 |
| 8 | などで《NIS》経由で《DNS》が引けていることも確認しています。 |

図 2.7: タグ付けの実行例

| # | sentence |
|---|---|
| 2 | 《HP_OpenView》を《インストール》したことのある方に質問です。 |
| 3 | 現在《HP_OpenView3.3》を《SunOS4.1.4》に《インストール》しています。 |
| 4 | 無事、《インストール》は終了したのですが、《ネットワーク》状況を《グラフィカル》に<br>表示させたときに、《/etc/hosts》に書いてある《ホスト》以外はすべて《IP》でしか表示されません。 |
| 5 | 《SunOS》は《NIS》を使用し《DNS》を引くように設定してあります。 |
| 6 | 《ping》は《ホスト》名だけで動作させることができますし、 |
| 8 | などで《NIS》経由で《DNS》が引けていることも確認しています。 |

⇓ 一般語の削除

| # | sentence |
|---|---|
| 2 | HP《OpenView》をインストールしたことのある方に質問です。 |
| 3 | 現在 HP《OpenView3.3》を Sun《OS4.1.4》にインストールしています。 |
| 4 | 無事、インストールは終了したのですが、ネットワーク状況を《グラフィカル》に<br>表示させたときに、《/etc/hosts》に書いてあるホスト以外はすべて《IP》でしか表示されません。 |
| 5 | Sun《OS》は《NIS》を使用し《DNS》を引くように設定してあります。 |
| 6 | 《ping》はホスト名だけで動作させることができますし、 |
| 8 | などで《NIS》経由で《DNS》が引けていることも確認しています。 |

図 2.8: 一般語削除の実行例

# 第 3 章

# 分類カテゴリの推定

本章では、前章で得られた分野固有語候補の分類カテゴリを推定する方法について述べる。まず、分類カテゴリ推定で中心的役割を果たす文脈からの推定方法について説明し、次に、分類カテゴリ推定アルゴリズムについて説明する。3 番目に、分野固有語候補の分類カテゴリをさらに詳細に分類する詳細分類について説明し、最後に、Web 上のテキストを利用する方法について説明する。

## 3.1 文脈からの推定

文脈からの推定では、分類カテゴリが未知の分野固有語候補に対し、テキストの文脈を用いて分類カテゴリを推定する。この方法には、大きく分けて以下の 2 つの方法がある。

- 動詞とその格要素との関係を用いた方法
- 特定語とその前後の分野固有語との関係を用いた方法

テキスト中から発見した分野固有語候補は、プログラム名、システム名、製品名などと思われるカタカナ、英数字列である。Sun QA-Pack が対象とする Sun ワークステーションのニュースグループでは、これらのプログラム名、システム名、製品名などは、OS 名、マシン名、ハードウェア名、ソフトウェア名のいずれかであることが多い。まず、分野固有語候補を OS、マシン、ハードウェア、ソフトウェアのいずれかに分類することを考える。この 4 つの分類カテゴリを大分類カテゴリと呼び、以下では、大分類カテゴリ OS、マシン、ハードウェア、ソフトウェアそれぞれを【O】、【M】、【H】、【S】と表記する。

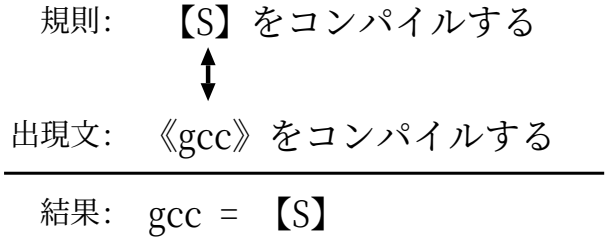


図 3.1: 動詞とその格要素との関係を用いた推定方法

### 3.1.1 動詞とその格要素との関係

動詞とその格要素との関係を用いた方法では、特定の動詞が、特定の分類カテゴリに属する分野固有語を、その格要素に取ることを利用する。

例えば、「コンパイルする」という動詞に着目した場合、この動詞は「【S】をコンパイルする」のように使われる。このことを利用し、テキスト中に、「X をコンパイルする」という表現が見つかった場合、「X」の分類カテゴリは【S】であると推定する (図 3.1)。

動詞とその格要素との関係を用いた分類カテゴリ推定規則の一覧を表 3.1に示す。

#### 信頼度の設定

動詞とその格要素との関係を用いた分類カテゴリ推定規則は、100%信頼できるものではない。そのため、これらの推定規則にそれぞれ信頼度を設定する。信頼度は高い順に $a$、$b$、$c$、$d$ の 4 段階に分ける。

表 3.1に、動詞とその格要素との関係を用いた分類カテゴリ推定規則の信頼度を示す。例えば、「X をコンパイルする」という表現がテキスト中に見つかった場合、「X」の分類カテゴリが【S】であるという推定の信頼度は $b$ となる。

### 3.1.2 特定語とその前後の分野固有語との関係

特定語とその前後の分野固有語との関係を用いた方法では、特定の語がその前後に特定の分類カテゴリに属する分野固有語を取ることを利用する。この方法には、大きく分けて以下の 2 つの方法がある。

- カテゴリ明示表現を用いた推定方法
- 並列関係を用いた推定方法

| 動詞 | 格要素 | 信頼度 |
|---|---|---|
| コンパイル | 【S】ヲ | $b$ |
| | 【S】ノ | $c$ |
| | 【S】デ | $c$ |
| | 【O】デ【S】ヲ | $c$ |
| インストール | 【S】ヲ | $c$ |
| | 【S】ノ | $b$ |
| | 【O】デ【S】ヲ | $d$ |
| | 【M】デ【S】ヲ | $d$ |
| | 【M】ニ【O】ヲ | $c$ |
| | 【O】ニ【S】ヲ | $c$ |
| 起動 | 【S】ヲ | $b$ |
| | 【S】ノ | $b$ |
| | 【S】デ | $d$ |
| | 【S】カラ | $c$ |
| | 【S】デ【S】ヲ | $c$ |
| 接続 | 【H】ヲ | $c$ |
| | 【H】ニ | $b$ |
| | 【O】ノ | $b$ |
| | 【H】ニ【H】ヲ | $c$ |
| 使う | 【S】ヲ | $c$ |
| | 【O】デ | $d$ |
| | 【M】デ | $d$ |

| 動詞 | 格要素 | 信頼度 |
|---|---|---|
| 表示 | 【S】デ | $c$ |
| | 【S】ガ | $d$ |
| | 【S】ニ | $d$ |
| 出力 | 【H】デ | $d$ |
| | 【S】ノ | $c$ |
| | 【H】ニ | $d$ |
| 使用 | 【S】ヲ | $c$ |
| | 【S】デ【S】ヲ | $c$ |
| 設定 | 【S】ノ | $d$ |
| | 【S】ヲ | $d$ |
| 実行 | 【S】ヲ | $c$ |
| | 【S】ノ | $c$ |
| 認識 | 【H】ヲ | $b$ |
| | 【H】ノ | $b$ |
| 動作 | 【S】ノ | $c$ |
| 終了 | 【S】ガ | $d$ |
| 搭載 | 【H】ヲ | $d$ |
| 運用 | 【S】ヲ | $c$ |
| 入れる | 【S】ヲ | $d$ |
| 入力 | 【S】デ | $d$ |
| 作る | 【S】ヲ | $c$ |

表 3.1: 動詞とその格要素との関係を用いた推定規則一覧

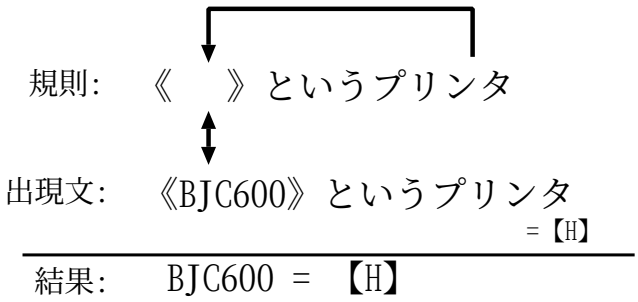


図 3.2: カテゴリ明示表現を用いた推定方法

**カテゴリ明示表現を用いた推定方法**

日本語には、あるもののカテゴリを明示的に示す表現として、「X というY」という表現がある。これは、X のカテゴリはY であることを意味する。例えば、「BJC600 というプリンタ」という表現は、「BJC600」は「プリンタ」というカテゴリに属するものであることを意味している。

この事実を利用し、「Y」の分類カテゴリがさらに分かっている場合、「X」は「Y」と同じ分類カテゴリに属すると推定する (図 3.2)。

**並列関係を用いた推定方法**

並列関係を構成する語に着目した場合、この語の前後に来る分野固有語は、同じ分類カテゴリに属することが多い。このことを利用して、前後どちらか片方の分野固有語分類カテゴリが既知の場合、もう片方の分野固有語の分類カテゴリを推定する。

例えば、「や」という並列関係を構成する語に着目しよう。テキスト中では「X やY」のように使われ、分野固有語「X」、「Y」の分類カテゴリは同じであることが多い。この事実を利用し、「X」の分類カテゴリが【O】のとき、「Y」の分類カテゴリも【O】であると推定する (図 3.3)。

**推定規則一覧**

特定語とその前後に来る分野固有語との関係を用いた分類カテゴリ推定規則の一覧を表 3.2に示す。表中の【$\alpha$】は既に分類カテゴリが分かっている分野固有語の分類カテゴリを示し、【$\beta$】は分野固有語候補の分類カテゴリを示す。【$\alpha$】、【$\beta$】は【O】、【S】、【M】、【H】のいずれかである。

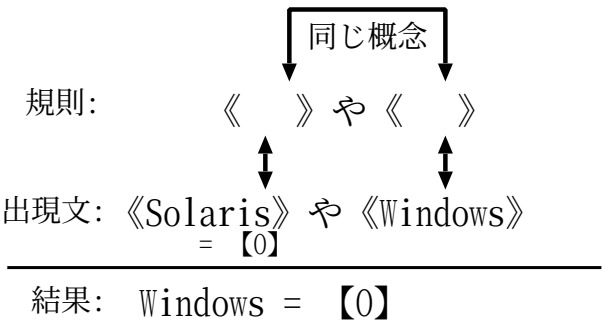


図 3.3: 並列関係を用いた推定方法

| 規則 | 信頼度 |
|---|---|
| 【$\beta$】(の\|という)【$\alpha$】 | $a$ |
| 【S】(の\|という)(ソース\|バイナリ\|コマンド\|ソフト) | $a$ |
| 【S】(の\|という)パッチ | $b$ |
| 【H】(の\|という)(ボード\|カード) | $a$ |
| 【$\alpha$】(や\|もしくは\|および\|及び)【$\beta$】 | $d$ |
| 【$\beta$】(や\|もしくは\|および\|及び)【$\alpha$】 | $d$ |

表 3.2: 特定語とその前後の分野固有語との関係を用いた推定規則一覧

### 信頼度の設定

特定語とその前後の分野固有語との関係を用いた推定規則も、100%信頼できるものではない。これらの規則に対しても信頼度を設定する。この信頼度は、「動詞とその格要素の関係」の信頼度と同じ 4 つの信頼度、$a$、$b$、$c$、$d$ を用いる。

表 3.2に、特定語とその前後の分野固有語との関係を用いた分類カテゴリ推定規則の信頼度を示す。

例えば、「X というY」という表現がテキスト中に見つかったとしよう。「Y」の分類カテゴリが【S】である場合、「X」の分類カテゴリが【S】であるという推定の信頼度は $a$ となる。

## 3.2 分類カテゴリ推定アルゴリズム

分類カテゴリ推定アルゴリズムの概略を図 3.4に示す。本アルゴリズムは、分野固有語候補にタグの付いたニュース記事を入力として、分野固有語候補の分類カテゴリを推定する。本アルゴリズムは、大きく、(1) 一般化、(2) カテゴリ埋め込み、(3) 文脈推定、(4) 得点による判定、の 4 ステップからなる。

### 一般化

抽出した分野固有語候補は、プログラム名、システム名、製品名などであるため、末尾にバージョン番号の付く語が多い。バージョン違いのプログラム名、システム名、製品名などを同一視するために、分野固有語候補に対して、一般化を行なう。

一般化は以下の規則に従って行なう。

**一般化規則** 数字、バージョン記号[1]が 2 文字以上連続して現れる場合、それ以降を全てバージョン番号であると判断し、「*」に変換する。

[例] ASCII-pTeX_2.99_j1.7 ⇒ ASCII-pTeX*

但し、分野固有語候補の末尾では、数字、バージョン記号が 1 文字だけでもバージョン番号と判断し、「*」に変換する。

[例] SS5 ⇒ SS*

### カテゴリ埋め込み

カテゴリ埋め込みでは、テキスト中に出現する分野固有語候補それぞれに対して、その分野固有語候補が既に分野固有語辞書に記載されていれば、その分類カテゴリを分野固有語候補のタグの中に埋め込み、この分野固有語候補を既存の分野固有語とする。分野固有語辞書に記載されていない分野固有語候補はそのまま分野固有語候補として出力する。

### 文脈推定

文脈推定は、3.1節で説明した文脈推定方法を用いて、分野固有語候補の分類カテゴリを推定する。

[1]バージョン番号を表す際に用いられる記号: 「-」「+」「_」「.」

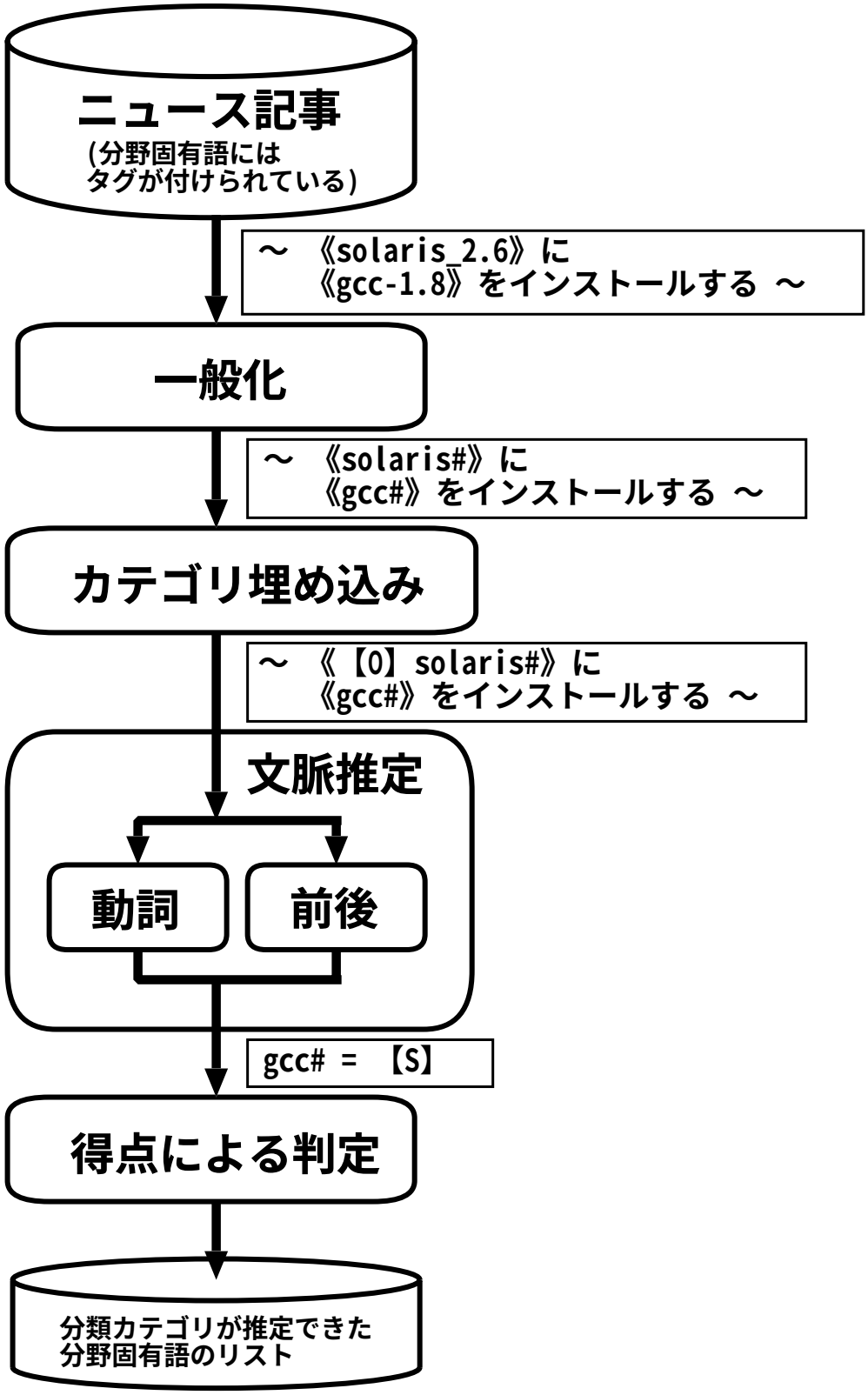


図 3.4: 文脈による推定アルゴリズム

| sendmail ＝ 【S】 | 信頼度 |
|---|---|
| sendmail のインストール | $b$ |
| sendmail をインストールする | $c$ |
| sendmail をコンパイルする | $b$ |
| sendmail というソフトウェア | $a$ |
| 集計 (信頼度) | $a(1)b(2)c(1)d(0)$ |
| 集計 (得点) | $1 \times 9 + 2 \times 6 + 1 \times 4 + 0 \times 2 = 25$ |

表 3.3: 信頼度の集計と得点への変換

## 得点による判定

得点による判定では、文脈推定規則が持つ信頼度を用いて、推定結果から、正しい推定結果のみを抽出する。得点による判定は大きく分けて、(1) 信頼度の集計、(2) 閾値による抽出、の 2 つのステップからなる。

### 信頼度の集計

ここでは、以下の 2 つを行なう。

1. 分野固有語候補とその分類カテゴリごとに信頼度を集計する。

2. 信頼度を得点に変換する。

まず、分野固有語候補の推定された分類カテゴリそれぞれに対して、適用された規則の信頼度を集計する。次に、この信頼度を次の閾値設定のために、得点に変換する。「$a = 9$」「$b = 6$」「$c = 4$」「$d = 2$」とすることで、集計された信頼度を 1 つの得点として表す。具体例を表 3.3に示す。

### 閾値による抽出

これまでのステップで、図 3.5のように、分野固有語候補と分類カテゴリの組それぞれに対して、得点が得られる。この得点を用いて、以下の方法で分野固有語候補の分類カテゴリを決定する。

**(1)** 閾値を設定し、それ以上の得点を持つ分野固有語候補と分類カテゴリの組を抽出する。

| 分野固有語候補 | 分類カテゴリ | 得点 | |
|---|---|---|---|
| telnet | 【S】 | 35 | |
| ps* | 【S】 | 32 | |
| lp* | 【H】 | 30 | |
| mount | 【S】 | 30 | |
| nic | 【H】 | 10 | |
| lp* | 【S】 | 7 | |
| dat | 【H】 | 6 | ⇑ |
| st* | 【H】 | 5 | |
| unix | 【H】 | 5 | |
| プロバイダ | 【H】 | 3 | |
| | ⋮ | | |

⇓

【S】「telnet」「ps*」「mount」
【H】「lp*」「nic」「dat」

図 3.5: 閾値による抽出

**(2)** 1 つの分野固有語候補に対し、2 つ以上の分類カテゴリが推定された場合は、得点高い方を抽出する。

閾値を 6 とした場合の具体例を、図 3.5に示す。

## 3.3 詳細分類

ここでは、今まで分類してきた大分類カテゴリを Sun QA-Pack に使われる分野固有語辞書への学習を考えて、さらに深い階層の分類カテゴリ (詳細分類カテゴリ) の推定を行なう。詳細分類は以下の方針で行なう。

現在より 1 つ深い階層の分類カテゴリを推定する。【H】は、詳細な分類カテゴリ「ハードディスク」「CD-ROM」などに分割する。【S】は、「ソフトウェア」「コマンド」の 2 つに分割する。表記方法は、それぞれ、【H.hd】【H.cdrom】【S.soft】【S.command】のように、「.」を用いて大分類カテゴリの後に詳細分類カテゴリを表記する。但し、詳細分類カテゴリが推定できないものは大分類カテゴリで出力する。

| BJC* | |
|---|---|
| 分類カテゴリ | 得点 |
| 【H】 | 12 |
| 【H.printer】 | 5 |
| 【H.printer】 | 17 |

図 3.6: 詳細分類

詳細度分類をするために、分類カテゴリ推定アルゴリズムを以下のように変更する。

**変更点 1** 信頼度の集計を大分類カテゴリではなく、詳細分類カテゴリで行なう。

**変更点 2** 詳細に推定された分類カテゴリを優先する。

変更点 1 は、例えば、今まで【S.soft】と【S.command】は、信頼度集計の際に、同じ推定結果、大分類カテゴリ【S】と判断し、集計していた。これを詳細分類カテゴリで集計することで、別の推定結果と判断する。

変更点 2 は、得点集計時に、詳細分類カテゴリが推定された結果が存在した場合、大分類カテゴリの推定結果の得点を、それに加えることで詳細分類カテゴリを優先させる (図 3.6)。

## 3.4 Web を用いた推定

3.2節と 3.3節で説明した分類カテゴリ推定方法が対象とするテキストは、ニュース記事だけに限定されるわけではない。より多くのテキストに対して適用することで、より多くの推定を行なうことができると期待できる。Web 上には多くのテキストが存在する。Web 上からテキストを取得し、このテキストに対して、分類カテゴリ推定方法を適用することで、より多くの推定を行ない、推定結果の精度を向上させることを考える。

Web を用いた分類カテゴリ推定アルゴリズムの概略を図 3.7に示す。Web を用いた推定方法は、大きく、(1)Web からのページ取得、(2) 文分割、(3) 分類カテゴリ推定、の 3 つのステップからなる。

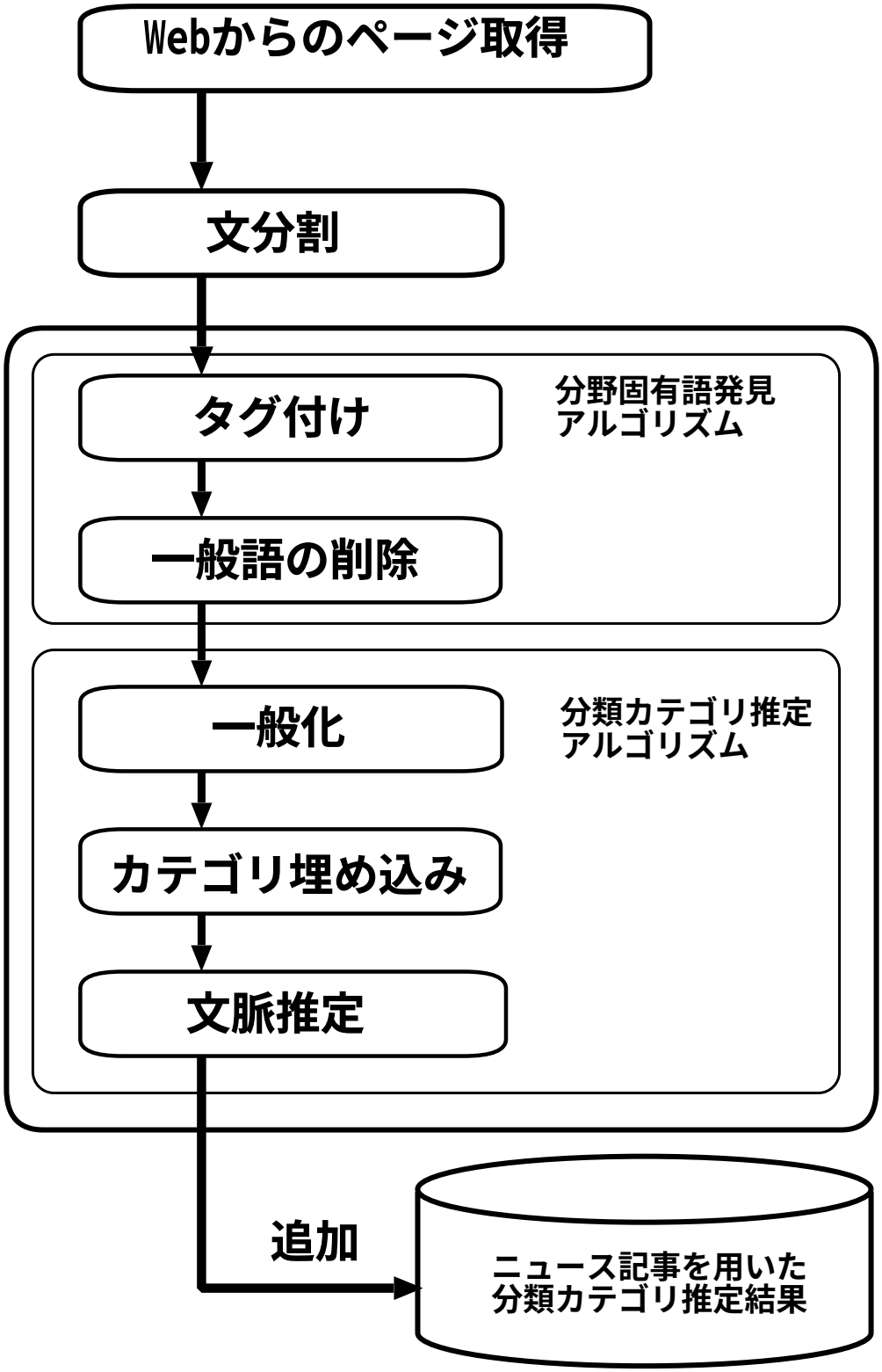


図 3.7: Web を用いた推定方法アルゴリズム

### Web からのページ取得

ここでは、2章で発見した分野固有語候補が記載されたページを Web 空間から取得する。その方法を以下に示す。

1. 2章で発見した分野固有語候補を検索キーワードとする。
2. 検索エンジン[2]にキーワードを入力し、検索された URL のページを収集する。

### 文分割

ここでは、Web 空間から取得したページを次の分類カテゴリ推定のために文単位に分割し、キーワードが存在する文を抜き出す。取得したページの文分割方法は、HTML 形式とそれ以外を分けて考える。文分割方法を以下に示す。

1. 取得したページを 1 ページ入力する。
2. ページが HTML 形式の場合、3 へ。それ以外の場合は、5 へ。
3. BODY 内をとりだす。
4. <br>や<p>など、タグの情報を用いて文ごとに分割し、5 へ。
5. 分野固有語発見アルゴリズムの文選別ステップを行ない、6 へ。
6. キーワードが存在しない文は削除する。

### 分類カテゴリ推定

ここでは、キーワード (分野固有語候補) の分類カテゴリを推定し、ニュース記事を用いて行なった推定結果に加える。

その方法を以下に示す。

1. 前ステップで文単位に分割された文に、分野固有語発見アルゴリズムのタグ付け、一般語の削除を行なう。
2. 分類カテゴリ推定アルゴリズムの一般化、カテゴリ埋め込み、文脈推定を行なう。
3. 得られたキーワードの分類カテゴリをニュース記事により得られた結果に加える。

[2]goo(http://www.goo.ne.jp/)

# 第 4 章

# 実験と検討

本章では、2章で述べた「分野固有語発見方法」、3章で述べた「分類カテゴリ推定方法」についての実験とその結果について述べる。

## 4.1 実験対象

以下の実験では、対象テキストとして、ニュースグループ「fj.sys.sun」のニュース記事のうち、1997 年 1 年間のニュース記事 (2761 記事) を用いた。

## 4.2 分野固有語の発見

ニュースグループ「fj.sys.sun」の 1997 年のニュース記事 (2761 記事) に対して、分野固有語発見を行なった。

### 文選別

各記事に対して、文選別を行った。文分割、および、文削除を行なった結果を表 4.1に示す。2761 記事の全文数が 37496 文であったのに対し、文選別では、12705 文 (33.9%) が選ばれた。削除された文のうち、その半数以上を「署名文」が占めるのは、記号列だけの行以降を署名文と判断して、削除しているためと思われる。

| 文タイプ | | 文数 (%) |
|---|---|---|
| 文分割後の文 | | 37496 |
| 文削除後の文 | | 12705 (33.9) |
| 削除された文 | | 24791 (66.1) |
| | コマンド、エラーメッセージ | 4066 (10.8) |
| | 署名 | 12206 (32.6) |
| | あいさつ部分 | 3086 (8.2) |
| | カナ、英数字を 1 文字も含まない文 | 5433 (14.5) |

表 4.1: 文選別結果

| | 述べ | 異なり |
|---|---|---|
| タグ付けによってタグの付いた語 | 33695 | 9156 |
| 削除された語 | 7517 | 1124 |
| 分野固有語候補 | 28531 | 8501 |

表 4.2: タグ付け、一般語削除結果

### タグ付け、一般語の削除

文選別で得られた文にタグ付けを行ない、一般語を削除した結果を表 4.2に示す。タグ付けにより、タグの付いた語が異なりで 9156 語得られ、一般語削除により、分野固有語候補が異なりで 8501 語得られた。削除数が、(タグ付け語のタグの付いた語の数) -(分野固有語候補数) でないのは、一般語削除が、開始タグ直後、もしくは、終了タグ直前の語に適用する際、「《ls コマンド》」は「《ls》コマンド」のように、タグを削除せず一般語をタグの外に出すためである。

## 4.3 分類カテゴリの推定

前節で得られた分野固有語候補に対し、大分類カテゴリの推定を行なった。既存の分野固有語辞書として、現在の Sun QA-Pack の辞書を用いた。

| | 述べ | 異なり (%) |
|---|---|---|
| 一般化後の分野固有語候補数 | 28531 | 7429 |
| 既存の分野固有語 | 8535 | 916 (12.3) |
| 分野固有語候補 | 19996 | 6513 (87.7) |

表 4.3: 一般化、タグの埋め込み結果

| 登場回数 | 10 回以上 | 5 回以上 | 2 回以上 | 1 回以上 |
|---|---|---|---|---|
| 分野固有語候補 | 375 | 897 | 2702 | 6513 |

表 4.4: 分野固有語候補とテキスト中の登場回数

## 一般化、タグの埋め込み

分野固有語候補に対して、一般化を行ない、分野固有語辞書を用いて、分野固有語候補を既存の分野固有語と分野固有語候補に分けた (表 4.3)。分野固有語候補と全対象テキスト中の登場回数の関係を表 4.4に示す。分野固有語発見で得られた分野固有語候補、異なりで 8501 語に対して、一般化により、異なりで 7429 語得られ、タグの埋め込みにより、分類カテゴリを推定すべき分野固有語候補が、異なりで 6513 語 (87.7%) 得られた。その内、全テキスト中に 10 回以上登場した分野固有語候補は、異なりで 375 語であった。

## 文脈推定

得られた分野固有語候補に対して、文脈推定を行なった。「動詞とその格要素との関係」、「特定語とその前後の分野固有語との関係」それぞれの適用回数を表 4.5に示す。

## 得点による判定

文脈推定結果に対して、得点による判定を行なった (表 4.6)。得点 10 以下は信頼度と得点の関係が「$a = 9$」「$b = 6$」「$c = 4$」「$d = 2$」なので、表の得点以外取ることはない。

信頼度と得点の関係より、得点 10 以上の語は、少なくとも規則が 2 回以上適用されたことになる。閾値を 10 に取った場合、111 語 (【H】22 語、【S】85 語、【O】3 語、【M】1 語) の分野固有語候補に対して、分類カテゴリが推定できた。その一例を表 4.7に示す。

| 信頼度 | 動詞 | 前後 | 計 (信頼度別) |
|---|---|---|---|
| $a$ | | 239 | 239 |
| $b$ | 145 | 6 | 151 |
| $c$ | 662 | | 662 |
| $d$ | 332 | 167 | 499 |
| 計 (規則別) | 1139 | 412 | 1551 |

表 4.5: 文脈推定結果 (適用回数)

| 得点 | 10 以上 | 9 | 8 | 6 | 4 | 2 |
|---|---|---|---|---|---|---|
| 【H】 118 | 22 | 51 | 0 | 19 | 8 | 18 |
| 【S】 628 | 85 | 53 | 44 | 28 | 250 | 168 |
| 【O】 44 | 3 | 4 | 2 | 0 | 7 | 28 |
| 【M】 5 | 1 | 3 | 0 | 0 | 1 | 0 |
| 計 795 | 111 | 111 | 46 | 47 | 266 | 214 |

表 4.6: 得点による判定結果

() 内の数字はその分野固有語候補の得点を示す。

## 検討

閾値を 10 とすることで、111 語 ( 【H】22 語、【S】85 語、【O】3 語、【M】1 語) が得られた。この推定結果の評価を表 4.8に示す。それぞれの項目に対する推定結果例を以下に示す。以下で示す例において、分野固有語候補の後に付加した分類カテゴリは、実験により誤って推定された分類カテゴリである。

- 正しく分類カテゴリが推定された候補: [例] sendmail*【S】、ide【S】
- 正解か判断が付けにくい候補: [例] フォーマット【S】、UNIX【S】
- 誤って分類カテゴリ推定された候補
  これはさらに以下のように 7 種類に分割される。

【H】 pc*(47), サイズ (36), s-bus(27), スロット*(24), ide(24), ローカル, (18) ナナオ (18), トラフィック (18), スーパーフロッピーフォーマット (18), vertex*(18), news(18), dvi_file(18), cg*(18), atapi(18), lan(16), pcmcia(15), plextor(13), プロバイダ (12), サードパーティcrt(12), rshd(12), lp*(11), a*(11)

【S】 sendmail*(97), openwindows*(89), root(72), gs*(62), telnet(53), unix(42), du*(38), オリジナル (36), wnn*(31), ppp*(31), xfree*(30), admintool(30), nt*(29), free(27), fmoformat(27), inn*(26), gnu_malloc(26), db*(26), パッケージ (24), ufsdump(21), スクリプト (20), telnetd(20), mh*(20), f*(20), xdvi*(18), netscape*(18), netmaj(18), mo-mount(18), cpio(18), cde*(18), ···

【O】 win*(22), パソコン (11), macintosh(10)　【M】 ローエンド (18)

表 4.7: 得点による判定結果 (一例)

- 一般語: [例] サイズ【H】、フォント【S】、オリジナル【S】
- 企業名: [例] ナナオ【H】、plextor【H】
- 【H】に分類されるべき候補: [例] Ethernet_card【S】
- 【S】に分類されるべき候補: [例] rshd【H】
- 【M】に分類されるべき候補: [例] NEWS【H】
- 【O】に分類されるべき候補: [例] NT*【S】
- 候補ではない語: [例] a*【H】、dvi_file【H】

大分類カテゴリが推定された得点 10 以上の分野固有語候補 111 語のうち、正しく推定された語は 70 語、63%である。この結果より、提案した方法は、ある程度有効に働くことが確認できた。人間が行なう分野固有語辞書への新しい分野固有語の追加作業を支援する用途には十分利用できると考えられる。

しかし、分野固有語辞書への追加を考慮に入れた場合、更に精度を上げる必要がある。表 4.8の結果に含まれた誤りは、(1) 辞書、リストの不備、(2) 分類カテゴリ推定の失敗、の 2 通りが考えられる。それぞれの対策を以下で検討する。

(1) は、表 4.8の「正解か判断が付けにくい候補」、「一般語」、「企業名」の語が誤って推定された原因と考えられる。これらの語は、分野固有語ではないが、候補として残った語である。

| | 【H】 | 【S】 | 【O】 | 【M】 | 計 (%) |
|---|---|---|---|---|---|
| 正しく分類カテゴリが推定された候補 | 10 | 59 | 1 | 0 | 70 (63) |
| 正解か判断が付けにくい候補 | 3 | 7 | 1 | 1 | 12 (11) |
| 誤って分類カテゴリ推定された候補 | 9 | 19 | 1 | 0 | 29 (26) |
| 一般語 | 3 | 13 | 1 | 0 | 17 (15) |
| 企業名 | 2 | 0 | 0 | 0 | 2 (2) |
| 【H】に分類されるべき候補 | - | 2 | 0 | 0 | 2 (2) |
| 【S】に分類されるべき候補 | 1 | - | 0 | 0 | 1 (1) |
| 【M】に分類されるべき候補 | 1 | 0 | - | 0 | 1 (1) |
| 【O】に分類されるべき候補 | 0 | 1 | 0 | - | 1 (1) |
| 候補ではない語 | 2 | 3 | 0 | 0 | 5 (5) |
| 合計 | 22 | 85 | 3 | 1 | 111 |

表 4.8: 推定結果 (大分類)

- 「正解か判断が付けにくい候補」は、表記される文によって、大分類カテゴリが文脈によって変化することが原因で正しい分類カテゴリを確定できない。例えば、「UNIX」という分野固有語候補は、文によって、【O】、【S】両方の分類カテゴリを取り得る。対象テキスト中に含まれる文の種類によって、推定される分類カテゴリが変わってくるため、一般語として削除するか、あらかじめ分野固有語辞書に登録しておく必要がある。

- 「一般語」は、一般語リストが不十分であるために一般語削除で削除されずに残ったものである。

- 「企業名」は、表 2.1のように、企業名リストに登録してある企業名が英字表記のみだったために、カタカナ表記されている企業名は削除されず、候補として残ったものである。

(2) は、表 4.8の「間違った大分類カテゴリが推定された候補」に対する原因と考えられる。これらの語は、誤って大分類カテゴリを推定したものである。現在の信頼度設定で推定し、信頼度を集計すると、閾値以上の得点を持った誤りが出現するので、誤って推定を行なった規則の信頼度の設定を下げることで、誤りを減らすことができる。

| | 【H】 | 【S】 | 計　(%) |
|---|---|---|---|
| 正しく詳細分類カテゴリが推定された候補 | 6 | 5 | 11　(16) |
| 詳細分類カテゴリが推定できなかった候補 | 3 | 54 | 57　(83) |
| 詳細分類カテゴリ誤って推定された候補 | 1 | 0 | 1　(1) |

表 4.9: 推定結果の分布 (詳細分類)

### 4.3.1　詳細分類

詳細分類カテゴリ推定を行なった。詳細分類カテゴリ推定は、文脈推定規則適用までは大分類カテゴリ推定と同じである。その後、得点集計時に詳細分類カテゴリで集計する。

大分類カテゴリの推定で誤ったものは、詳細分類でも誤るため、表 4.8より、「正しく大分類カテゴリが推定された候補」の内、詳細分類を行なう【H】もしくは【S】と分類推定された候補 69 語 (【H】10 語、【S】59 語) に対して、詳細分類カテゴリ推定を行なった結果を表 4.9示す。

表 4.8の結果の内、正しく大分類カテゴリが推定された分野固有語候補 69 語に対して、詳細分類カテゴリが正しく推定された分野固有語候補は 11 語、16%である。誤った詳細カテゴリ推定がほとんどないことから、大分類カテゴリが正しく推定されれば、この手法を用いることで詳細分類カテゴリ推定が可能であるが、現在の方法だけでは、詳細分類カテゴリを十分得ることはできないことが分かった。

「正しく詳細分類カテゴリが推定された候補」の数が非常に少ないのは、詳細分類カテゴリ推定が可能な文脈推定規則は、「特定語とその前後の分野固有語との関係」を用いた文脈推定規則だけだからである。「動詞とその格要素との関係」を用いた文脈推定規則では、動詞によって、格要素が特定の大分類カテゴリを持つことはあっても、特定の詳細分類カテゴリを持つことはないからである。例えば、「接続する」という動詞は、その格要素に【H】を取ることは分かっても、それが、【H.printer】か、【H.hd】かは決定できない。

### 4.3.2　Web を用いた推定

#### Web からのページ取得

検索エンジンへのキーワードとして、表 4.4の登場回数 10 回以上の分野固有語候補 375 語を用いる。キーワードと検索された Web ページ数の関係を表 4.10に示す。38 キーワードは、1 ページも得られなかった。

| 検索ページ数 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| キーワード数 (375) | 0 | 192 | 93 | 23 | 17 | 4 | 2 | 1 | 0 | 5 | 38 |

表 4.10: キーワードと検索された Web ページ数

| | 【H】 | 【S】 | 【O】 | 【M】 | 計 (%) |
|---|---|---|---|---|---|
| 正しく分類カテゴリが推定された候補 | 0 | 10 | 0 | 0 | 10 (59) |
| 正解か判断が付けにくい候補 | 1 | 1 | 0 | 0 | 2 (12) |
| 誤って分類カテゴリ推定された候補 | 0 | 5 | 0 | 0 | 5 (29) |
| 一般語 | 0 | 1 | 0 | 0 | 1 (6) |
| 企業名 | 0 | 0 | 0 | 0 | 0 (0) |
| 【H】に分類されるべき候補 | - | 3 | 0 | 0 | 3 (18) |
| 【S】に分類されるべき候補 | 0 | - | 0 | 0 | 0 (0) |
| 【M】に分類されるべき候補 | 0 | 1 | - | 0 | 1 (6) |
| 【O】に分類されるべき候補 | 0 | 0 | 0 | - | 0 (0) |
| 候補ではない語 | 0 | 0 | 0 | 0 | 0 (0) |
| 合計 | 1 | 16 | 0 | 0 | 17 |

表 4.11: 推定結果の分布 (Web)

### 文分割

全 2771 ページの内、HTML 文書は 2756 ページ、それ以外が 15 ページであった。文分割を行なった結果、10474 文が得られた。

### 分類カテゴリ推定

文分割した Web テキストの文を用いて、4.3節で行なったのと同じ文脈推定を行なった。同じ閾値 (10) を設定することで、表 4.8の結果と比較して、新たに 17 語 (【H】1 語、【S】16 語) が得られた (表 4.11)。表中の記号は表 4.8と同じものとする。

### 検討

表 4.11より、Web から取得したテキストに対しても、同じ分類カテゴリ推定規則を適用して分類カテゴリを推定することが可能なことが分かった。

しかし、分野固有語候補を検索キーワードとして、検索結果上位 10 件の URL に存在する Web ページ (2771 ページ) は、fj.sys.sun の 97 年 1 年間のニュース記事数 2761 記事よりも多いにも関わらず、分類カテゴリ推定結果があまり得られない。これには (1) 関係のないページの収集、(2)HTML 文書の記述方式、の 2 点が原因と考えられる。

(1) は、分野固有語候補をそのまま検索キーワードとしたことが原因として考えられる。分野固有語候補を検索キーワードとしたことによって、分野固有語候補が存在する HTML 文書を取得することができるが、分野固有語候補の文字数が短い場合、特に製品名などの場合、必要のないページが検索されてしまう。例えば、「ls」という分野固有語候補をキーワードにした場合、「ls」は存在しないが、「else」という語が存在する文書も検索する。これに対しては、検索を 10 結果までを得ていたが、これを増やす。もしくは、全然関係のない文書を排除するために、検索キーワードを工夫する。また、キーワードを工夫することにより、検索結果を得られなかったキーワードにも対応させる。

(2) は、HTML 文書では製品紹介のようなページの場合 TABLE タグ等で視覚的に見せようと、図 4.1のように整理して記述されることがある。このような文書は、タグの中身が文となっていないため、現在のように、タグ情報により文分割した場合、単語列だけの文ができ、文脈による推定方法が使えない。こういう文書には、HTML 文書特有の規則を用意する。

## 4.4 検討

本研究では、テキスト中から分野固有語を発見し、その語が属する分類カテゴリを推定する方法を示した。

大分類カテゴリが推定された 111 語の内、正しく推定された語は、70 語、63%である。この大分類カテゴリ推定の結果を増やすためには、(1) 分類推定の精度を上げる、(2) テキストを増やす、の 2 通りが考えられる。

(1) の分類精度を上げるためには、削除語リストの強化、得点集計方法の見直し、が考えられる。削除語リストの強化は、今回の実験により全テキスト中に多く登場した語を登録することにより、実現できる。得点集計方法の見直しは、推定を誤った規則の信頼度設定を修正する、もしくは、信頼度と得点の関係を修正する、閾値の設定を修正する、の 3 通りの方法により、実現できる。

(2) のテキストを増やすには、Web から取得するテキストを増やすことにより、実現できる。取得するテキストを増やすには、検索エンジンから得る分野固有語候補が存在するテキストの URL の数を増やすせばよい。この時、分野固有語候補と関係のないテキストを取得しないように、検索エンジンへのキーワード設定を工夫する必要がある。また、取得したテキストから分野固有語候補を含む文を効率良く抽出する工夫や、文脈推定規則とは違った推定方法を考える必要がある。

詳細分類カテゴリ推定に関しては、現手法では、大分類カテゴリが正しく推定されれば、詳細分類カテゴリを誤って推定することはほとんどない。詳細分類推定結果を増やすには、詳細分類カテゴリ推定が可能な規則を増やすことにより、実現できる。

## 4.5 関連研究

電子化されたテキストから重要な情報を抽出する技術は近年米国を中心に盛んに研究されている。米国には、英語新聞記事からの固有名詞抽出として、ARPA(Advanced Research Projects Agency) が支援する MUC(Message Understanding Conference)[4], Tipster[5] などの情報抽出プロジェクトがある。

本研究では、テキスト中から発見すべき分野固有語候補が、プログラム名、システム名、製品名などであり、それらがカタカナ、英数字の並びで表記されることから、テキスト中から分野固有語を発見する方法として、カタカナ、英数字列を抽出する方法を取った。

従来からの日本語テキストから特定の情報 (専門用語など) を抽出する方法として、テキスト中での単語の出現頻度の高いものを専門用語として抽出する手法 [6] や、専門用語と共起する語を探し、その語から共起している専門用語を抽出する手法 [7][8] などがある。

***EPSON***

| プリンター名 | トナー名 | 標準定価 | 販売価格 |
|---|---|---|---|
| LP1000 | LP1000ETC | \21,000 | \18,800 |
| LP1500 | LP1500ETC | \28,500 | \25,650 |
| LP1600 | LPA4ETC1 | \19,500 | \17,500 |
| LP1700,700W,1700Sトナー | LPA4ETC2 | \12,000 | \10,000 |
| LP1700,700W,1700Sドラム | LPA4KUT2 | \10,000 | \9,000 |
| LP2000ﾄﾅｰ | LP2000ETC | \16,000 | \13,600 |
| LP2000ﾄﾞﾗﾑ | LP2000KUT | \29,000 | \26,000 |
| LP800ﾄﾅｰ | LPA4ETC3 | \10,500 | \9,450 |
| LP800ﾄﾞﾗﾑ | LPA4KUT3 | \10,000 | \9,000 |
| LP1800ﾄﾅｰ | LPA4ETC4 | \21,000 | \18,900 |
| LP1800ﾄﾞﾗﾑ | LPA4KUT3 | \10,000 | \9,000 |
| LP3000 | LP3000ETC | \29,000 | \26,100 |
| LP7000,G(LP7000ETC) | キャノンEP-T | \27,000 | \17,000 |
| LP8000,S,SE,LP9000 | LP8000ETC | \39,000 | \27,000 |
| LP8500,LP8000SX | LPA3ETC1 | \42,000 | \25,000 |
| LP8200,8200PS2,8300 | LPA3ETC2 | \30,000 | \22,000 |
| LP9100 | LPA91PSETC | \42,000 | \37,800 |
| LP9200,9200PS2,9200S | LPA3ETC3 | \40,000 | \35,000 |
| LP9600 | LPA3ETC5 | \50,000 | |
| LP8400,LP8300S | LPA3ETC4 | \30,000 | \27,000 |
| | LPCA3ETC1K | \14,000 | \12,600 |
| | LPCA3ETC1Y | \16,000 | \14,400 |
| | LPCA3ETC1M | \16,000 | \14,400 |
| LP8000C | LPCA3ETC1C | \16,000 | \14,400 |
| | LPCA3KUT1 | \25,000 | |
| | LPCA3TOR1 | \6,000 | |
| | LPCA3HTB1 | \3,000 | |

ご注文方法　メインメニューへもどる

図 4.1: TABLE タグで整理された例 (キーワード:lp)

# 第 5 章

# 結論

## 5.1 本研究の結論

本研究では、Sun QA-Pack の分野固有語辞書の学習において、テキスト中に存在する分野固有語の分類カテゴリを推定する方法を示した。その手順として、まずテキスト中から分野固有語を自動的に発見することを行ない、次に、発見された分野固有語の分類カテゴリの推定を、分野固有語が登場する文脈を用いることで行なった。

これにより、テキストから自動的に分野固有語を抽出し、その分類カテゴリを推定する方法を実現した。さらに、詳細な分類カテゴリを推定する方法も実現し、分類精度を上げるために Web からテキストを取得して分類カテゴリを推定する方法も実現した。

大分類カテゴリ推定された分野固有語候補の精度は 63%であり、本研究で実現した方法により、大分類カテゴリを推定することがある程度可能であると分かった。人間が行なう分野固有語辞書への新しい分野固有語の追加作業を支援する用途には、十分に利用できると考えられる。詳細分類カテゴリ推定は、本研究で実現した方法により、大分類カテゴリさえ推定されれば、誤った推定はしないことがわかった。また、Web から取得したテキストに対しても、この分類カテゴリ推定方法は適用することが可能であることが分かった。

## 5.2 今後の課題

本研究で実現した方法により、テキストから分野固有語を発見し、その分類カテゴリを推定することが可能である。しかし、分野固有語辞書の学習を考えた場合、更に分類カテゴリの推定精度を上げる必要がある。今後、(1) リストの強化、(2) 推定規則の強化、信頼度の設定変更、(3)Web を用いた分類カテゴリ推定の強化、の 3 つを行なうことで、推

定精度を向上させる。

**リストの強化**　テキスト中に多く登場する分野固有語候補内の一般語や企業名を調べて追加する。これにより、推定結果内の分野固有語でない一般語を減らす。

**推定規則の強化、信頼度の設定変更**　詳細分類カテゴリ推定のために、推定規則を増やし、それぞれの推定規則に対する信頼度を設定し直す。

**Web を用いた分類カテゴリ推定の強化**　Web から取得したテキストに対する分類カテゴリ推定の強化のために、以下の 3 つを行なう。

- キーワード検索時に、キーワードとして分野固有語候補をそのまま入力せず、工夫することで検索結果内の分野固有語候補に関係のないページを減らし、また、検索結果の少ないキーワードにも対応させる。
- HTML 文書の視覚的な構造を用いた分類カテゴリ推定規則を用意することで、取得したページを有効に利用する。
- 企業名リストに記載した企業は Sun ワークステーションの製品を製造する企業なので、その企業の製品情報ページを活用する方法を考える。

# 謝辞

本研究を進めるに当たり、終始熱心な御指導を賜りました佐藤理史助教授に心から感謝致します。

さらに、貴重な御意見、討論をしていただいた知識工学講座の皆様に感謝致します。

最後に、多くの方々の御援助によって本研究を行なうことができましたことを厚く御礼申し上げます。

# 参考文献


[1] 長尾真, 黒橋禎夫, 佐藤理史, 池原悟, 中野洋: “言語情報処理”. 岩波書店,1998.

[2] Hayes,P.J.&Weinstain,S.P.: “Construe-TIS: A system for content-based indexing of a database of news stories”. In Rappaport,A.&Smith,R.(eds.),*Innovative Applications of Artificial Intelligence 2*,pp.49-64,AAAI Press/MIT Press,1991.

[3] 佐藤円, 佐藤理史: “ネットニュース記事群の自動パッケージ化”. 情報処理学会論文誌,Vol.38,No.6,pp.1225-1234,1997.

[4] Defense Advanced Research Projects Agency: “Proceedings of the sixth Message Understanding Conference(MUC-6)”. Morgan Kaufman Publishers,1995.

[5] Defense Advanced Research Projects Agency: “Tipster Text Program(Phase II)”. Morgan Kaufman Publishers,1996.

[6] 中川 裕志, 森 辰則, 松崎 知美: “日本語マニュアル文における名詞間の連接情報を用いたハイパーテキスト化のための索引語の抽出”. 情報処理学会研究報告,NL116-10,pp.65-72,1996.

[7] 中山 拓也, 松本裕治: “シソーラスへの未登録語の自動登録”. 情報処理学会研究報告,NL120-16,pp.103-108,1997.

[8] 久光 徹, 丹羽 芳樹: “辞書と共起情報を用いた新聞記事からの人名獲得”. 情報処理学会研究報告,NL118-1,pp.1-6,1997.