修 士 論 文

HTML 文書のカテゴリ階層への自動割り当て

指導教官　奥村学 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

片山研一

1998 年 2 月 13 日

目 次

図 目 次

表 目 次

第 1 章

はじめに

1.1　研究の背景と目的

今日，インターネットの普及により世界中の多くのコンピュータが接続されるようになった．特に　World Wide Web の発達により，膨大な文書が手軽に手にはいるようになった．しかし，そのあまりの膨大さのため目的の文書を探し出すのは困難になってきている．そこで目的文書検索の補助を行うシステムとして，全文検索システム[1]とディレクトリ検索[2]システムが使用される．全文検索システムは，WWW robot と呼ばれる自動文書取得プログラムによって文書が集められる．その文書内の単語をインデキシングし，単語をキーワードとしてその単語を含む文書を見つけだして，検索者に提示する．それに対し，ディレクトリ検索システムでは，あらかじめ人手で分野別にカテゴリを作っておき，各カテゴリに文書を人手で分類する．検索者は目的とする分野のカテゴリをさがすことによって，関連文書を探し出すことができる．

しかし，表 1.1　のようにそれぞれ長所と短所があり，全文検索システムで自動的に集めた文書を自動的に決められたカテゴリに配置されることが望まれる．

本研究では，あらかじめ人手で作成したカテゴリに多くの訓練事例を使って自動的に配置する実験を行う．実験に使用するカテゴリは，木構造をしており，上位のノード (カテゴリ) は，抽象的なラベルがふられており，下位カテゴリになるほど具体的なラベルがふられている．このような階層的な分類を行う研究は，教師無しクラスタリングが主流で

[1] goo,ODIN,altavista などがある

[2] yahoo, yahho, NTT Directory などがある

表 1.1: 全文検索とディレクトリ検索の長所と短所

	長所	短所
全文検索システム	ほとんどの部分を自動化できる ロボットプログラムにより 大量の文書を収集することが可能	目的の文書を探し出すキーワードを 考えなければならない
ディレクトリ検索 システム	カテゴリがきめられてるため, 目的が漠然とした状態で検索する ことができる	ほとんどの作業を人手で 行うためコストがかかる. 人手で分類するため 大量の文書を集められない.

あった [1]．教師付き学習は，大量にラベル付けされた事例が必要でありデータ収集が困難なためである．しかし，教師無しクラスタリングの場合，分類されたカテゴリにラベル付けが行われていないため，ディレクトリ検索システムには向かない．そこでデータ収集を既存のディレクトリシステム[3]のカテゴリを使用し，そのカテゴリに人手で配置された文書を集めることによりラベル付けされた事例を揃えた．

文書をカテゴライズする研究は多く行われてきた．しかし満足のいく精度の得ることは非常に難しい．文書を分類する手法として，大きく分けて二つの手法がある．一つは言語情報や知識を利用する方法であり，もう一つは，統計的な情報を利用した方法である．言語情報や知識を利用する方法は高い精度の分類を行える可能性があるが，大規模な知識ベースの構築・維持管理が難しく実用化が困難である．それに対し，統計的な方法は高い精度が得られないかもしれないが，手軽に実現可能である．統計手法 HTML 文書を使った分類研究に落谷の研究 [3] があるが，高い精度が得られていない．WWW 上の文書は，内容，文書サイズにばらつきが多く既存の手法を単純に利用しただけでは良い精度が得られないという理由もある．しかし，HTML 文書には，タグと呼ばれる文書整形コマンドがある．この情報を利用することにより，文書中で重要であるキーワードが分かり，文書の特徴となると考える．さらに，ハイパーテキスト構造を利用して，リンク先の文書の特徴も合わせて利用する．

本研究は，一般的な分類手法である種々の統計的な手法の中で，対象となるカテゴリに適した手法を使う．また，統計的な手法に加え，HTML 文書特有の情報を使いその有効

[3] 豊橋技術科学大学 河合研究室 近多 泰宏氏の主催する
旧 yahho (現在は wave(http://www.wave.co.jp/wave/)) を使用

性を考える．また，最上位層より完全に分類されるまでの精度を計り，実際の運用と同じモデルでどの程度利用可能であるか調べる．

1.2 論文の構成

2 章で，分類対象となるカテゴリの説明を行う．3 章では，HTML 文書の特徴について述べる．4 章では，データ取得方法とデータ加工の方法について述べる．5 章では，統計的手法によるカテゴリと文書間の類似度計算法について述べる．さらにそれを用いたカテゴリ割り当て手法と階層的分類法について述べる．また，HTML 文書の特徴を利用した分類の方法について述べる．6 章では，4,5 章で述べた手法を使って分類実験を行い，結果を示す．最後に 7 章で結論及び今後の課題について述べる．

第 2 章

カテゴリ階層

本研究では, 分類先となるカテゴリに実際にディレクトリ検索システムで使われている yahho[17] の階層 (図 A.1,A.2) を使用する[1]．これは，図 2.1 のような階層構造をしており，上の階層ほど抽象的なラベルの付いたカテゴリになっており，下の階層ほどより具体的なラベルの付いたカテゴリになっている．検索者は，階層の上の方の抽象的な概念から下の階層に降りていくことにより具体的なものへと漠然とした検索ができる．

各カテゴリには，人手であらかじめそのカテゴリに属すると思われる文書へのリンクが張られている．この情報を元にデータ収集を行う．表?? は，一番上の階層での分類で各カテゴリとその下位カテゴリにリンクされている文書数の比率である．これから分かるように文書数に非常に偏りが大きい．

この傾向は詳細分類の時も起こり，特に エンターテイメント· 書籍· 小説 · 作家 のノードの下には 50 音別のカテゴリがあり，各ノードに文書がリンクされていない (ノードのラベルである音ではじまる作家がいない) 場合もありデータスパースが生じている．教師付き学習をする際このノードには分類できない (してはいけない) ので取り除く．しかし，ノード内文書数が 1 桁のノードは多数存在する．教師付き学習においてデータ不足は深刻な問題となる．兄弟ノードがすべて葉ノードで所属文書数が少ない場合上位ノードにノードをまとめてしまうことも考えられる．しかし，兄弟間のデータ量に差があることが多い．データスパースネスの解消は難しく，データ量の少ないノードには人手でデータを集めてデータ量を増やすのが最も効果的であり，効率も高いと考える．

本研究で使用する web 階層は，内部ノードにも文書がリンクされている．内部ノード

[1] そのままの階層を用いるのではなく階層が思われるところを改変

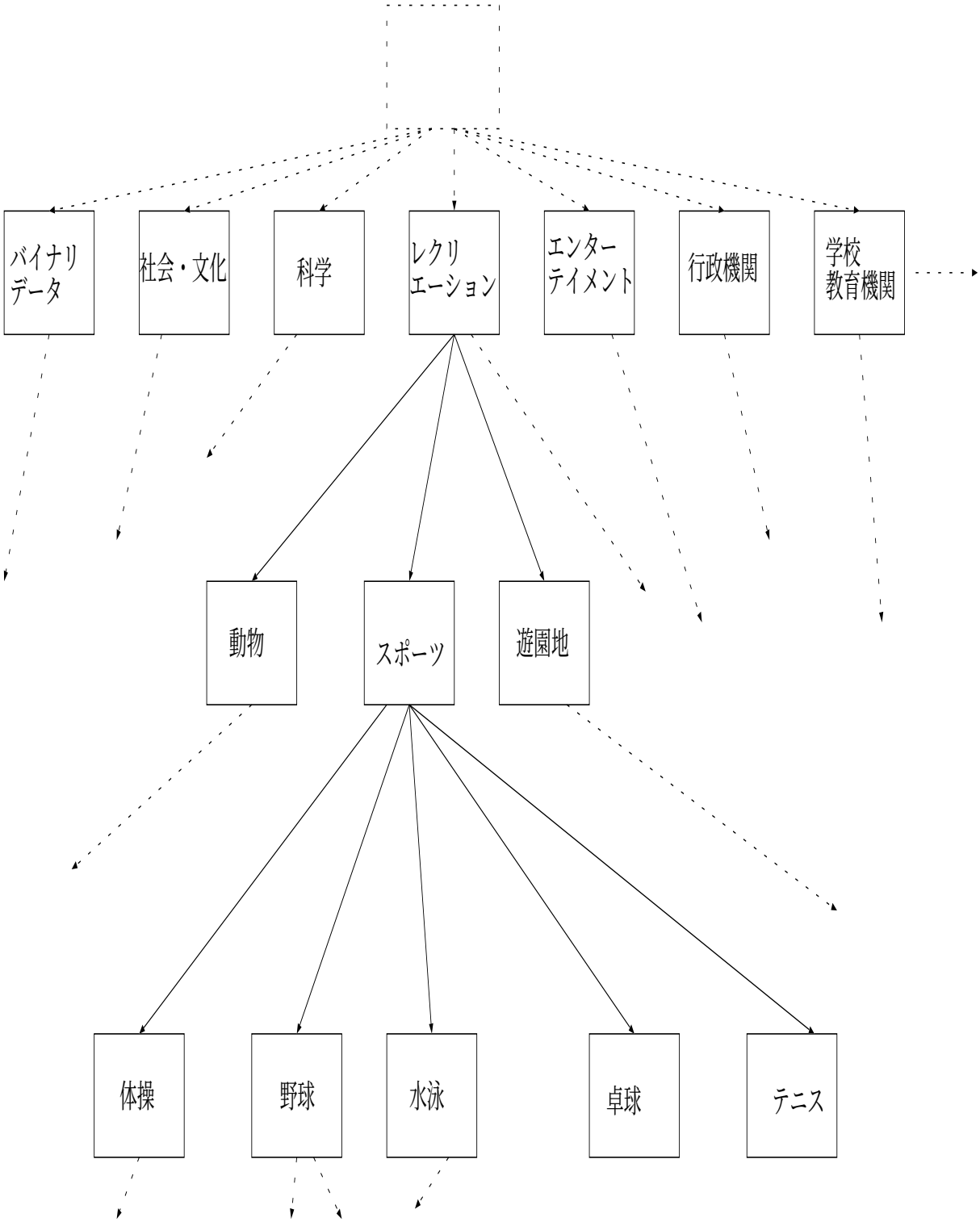

バイナリ
データ
社会・文化
科学
レクリ
エーション
エンター
テイメント
行政機関
学校
教育機関
動物
スポーツ
遊園地
体操
野球
水泳
卓球
テニス

図 2.1: web 階層

表 2.1: 大分類カテゴリとそこにある文書比率

大分類カテゴリ	比率%	大分類カテゴリ	比率%
バイナリデータ	1	イベント	3
企業	12	政治団体	4
コンピュータ	6	インターネット	14
学校教育機関	5	各種団体	12
工学	3	エンターテイメント	21
雑誌・読み物	2	レクリエーション	6
科学	5	社会・文化	5

にある文書の特徴として，次のことがあげられる．

1. 現ノードを代表する文書
2. 現ノードより具体的な事象について述べているが，下位ノードには，その事象が入るべきノードが存在しない文書
3. 下位ノードのほとんどの事柄について述べられた文書

図 2.1 の例でいくと，現ノードをスポーツとすると 1 は，スポーツの一般的なことについてかかれている文書．2 は，柔道，空手などについてかかれた文書，3 は，体操，野球，水泳など個々のスポーツの比較文書などがあげられる．内部ノードがこのような構成になっているため，分類する文書とノードとの類似性計算において，内部ノードに割り当てるべき文書をノードの全探索によって導きだすのは難しい．そこで，上位ノードからの下向きに文書をカテゴリに分類していくことにより，2, 3 の状態の文書を内部ノードにいれることができるであろうと考える．また，前に述べたデータスパースネスの問題によって，正解のカテゴリには割り当てることができなかったが，下向きの分類により本来割り当てるべきノードの上位ノードまでは割り当てることが可能になる．

また，この階層中に割り当てられている文書は，一つのノードだけではなく，複数のノードに所属する場合が数多くある[2]．このように，一つの事柄について書かれていても

[2]例えば，読売巨人について書かれているページは野球のカテゴリにも入るし，読売グループという企業のカテゴリにも入れられる

複数に割り当てられる場合がある．また，HTML 文書の場合一つの文書に全く異なる分野のことが書かれることがあり，複数割り当ては重要である．

第 3 章

web 上の文書の特徴

前章では，分類するカテゴリについて述べた．本章では実際に分類する文書はどのような構造になっているかについて述べる．web 上の文書は基本的に Hyper Text Markup Language と呼ばれる文書整形言語を使用して書かれ，ブラウザと呼ばれるソフトウェアによって解釈表示される．解釈の方法は，ブラウザの種類によって若干異なる．この言語の仕様は，World Wide Web Consortium から出され，短期間のうちに仕様が更新されている．最新の HTML の仕様では, アニメーション等も扱えるようになり，複雑になってきている [12].

HTML 文書の構造は基本的に，図 のようになっている．

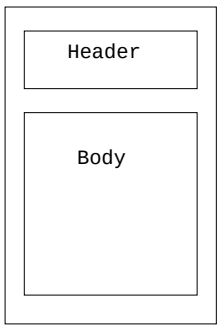


図 3.1: HTML 文書の構成

Header 部には，ページタイトル，キーワード, 著者，文字コード, Java script などの script 言語などが書かれる．Body 部に実際に表示する本文が書かれる．しかし，Header 部を書かなくても表示できるため，Header 部が書かれていない文書も少なくない．

本章では，web 上文書特有の特徴について述べる．

3.1 タグについて

HTML は，タグと呼ばれるコマンドで構成される．タグによって，文字の位置，フォントサイズ，色などの修飾や箇条書き，表，画像張り付けなどさまざまな制御が可能である．

図 3.2 は，HTML 言語で記述された文書の表示例である．この文書は，HTML で以下のように記述される．

図 3.2: HTML 文書表示例

```
<HTML><HEAD>
<TITLE>Japan Advanced Institute of Science and Technology(JAIST),
Hokuriku Home Page (In Japanese)</TITLE>
</HEAD>
<BODY BGCOLOR="#d9d9d9">
<CENTER>
<h1>北陸先端科学技術大学院大学<BR>
Japan Advanced Institute of Science and Technology</h1>
<IMG SRC="/images/JAIST-logo.mini.gif" border=0 alt="JAIST">
</CENTER>
<B>NOTE:</B> English home page is <A HREF="/">here</A>.
<HR><P><center>
<b>お知らせ</b>
<p>
<A HREF="/ kouhou/rule/index.html">
北陸先端科学技術大学院大学教員の任期に関する規則を制定</A>
<img src="/images/new2.gif">
<BR></center>
<P><hr>
<UL>
<LI> <A HREF="whats-news-jp.html">新着情報</A>
<LI> <A HREF="/gakusei/index-jp.html">学生募集案内</A>
<UL><LI> <A HREF="/gakusei/index-jp1.html">大学説明会</A> </UL>
<LI> <A HREF="/ kouhou/General_info/mokuji.html">大学総合案内</A>
<LI> <A HREF="/is/index-jp.html">情報科学研究科</A>
<LI> <A HREF="/ms/">材料科学研究科</A>
<LI> <! A HREF="/misc/jaist_info/knowlegde.html">
 <A HREF="/ks/index.html">知識科学研究科</A>
<LI> <A HREF="/ricenter/ricenter-j.html">先端科学技術研究調査センター</A>
<LI> <A HREF="/iscenter/index-jp.html">情報科学センター</A>
```

この中で，< と > で囲まれた，部分がタグになる．また，タグには有効範囲がある．P, LI, IMG などその場のみでタグの効果を出すもの，<タグ名> と </タグ名> で囲まれている間の文に効果を発揮するタグある (TITLE, CENTER, B など)．代表的なタグを表 3.1 に示す [10]．

表 3.1: 代表的なタグ

HTML	HTML の開始と終了	HEAD	ヘッダ部の開始と終了
BODY	ボディ部の開始と終了	TITLE	文書タイトル
META	文書情報の記述	FONT	フォントの変更
HR	横罫線	BR	改行
P	段落付け	CENTER	中央寄せ
BLINK	点滅	A HREF	リンク
B	太文字	I	イタリック体
EM	強調	STRONG	より強い強調
H	フォントサイズ変更	IMG	画像張り付け

この他にも多数のタグが存在していて，その数は，HTML のバージョンが上がる度に増えていっている．また，World Wide Web Consortium が策定した HTML の仕様以外にブラウザ独自のタグも存在している．

図 3.2 を見てみると，タイトルの情報，画像の上にある “北陸先端科学技術大学院大学” といった語は，この文書において重要な語である．これらの語を囲んでいるタグを見ると，それぞれ，TITLE, (CENTER, H1) というタグに囲まれている．逆に，これらのタグに囲まれている文（単語）は，重要であると考えられる．本研究では，このような知識を加え，文の特徴を抽出する．使用するタグは，解釈に依存性が少なく，将来仕様が変更されず，使用頻度の高いタグを利用する．

3.2 ハイパーテキスト

HTML(Hyper Text Markup Language) は，その名の通り，ハイパーテキスト構造をしている．A HREF タグによって文書間にリンクが張られ，世界中の文書がクモの巣状に結びついている．

リンク先の文書の内容は，リンク元の文書と関連性が高い文書が多いと考え，その情報の有効利用を考える．

図 3.3 のように文書が図にようって構成され文字情報が，得られない．この情報不足をこの文書からリンクされているページから補う．しかし，このページからは web ブラウ

図 3.3: 文字情報が得られない文書

ザであるネットスケープのことについて書かれた文書へのリンクもはられている．これは，ネットスケープで，この文書を見てくださいという意味でリンクが張られており，この文書とは内容が異なる．このように，リンク先を取捨選択し，できるだけ関連性のある文書へのリンクだけを利用しなければならない．このように一文書からとれる特徴素が少ないのも web 上の文書の特徴といえる [3]．

3.3 文書内容特徴

本節では，HTML 文書の特徴ではなく，web 上の文書の特徴について述べる．
web 上の文書について以下の項目に挙げた情報取得の難しさがあると考えられる．

1. 文調が会話調になっている文書

   インターネットの普及により誰でも気軽に WEB 上に文書がおけるようになったため多種多様な文体で文章が書かれている．一般的に公の機関が書いた文書は文体が

丁寧[1]である．それに対し，個人が書いて公開している文書には，会話文で書かれることも多い．またインターネット上のメッセージのやりとり特有の顔文字,”(笑)” など情報となりにくい語が多くある．

文書の特徴素として単語を取り出すために形態素解析を行うが，文が丁寧に書かれていないと正しく形態素解析が行えず特徴素の信頼性が落ちる．

2. 画像，プログラム，音声等の存在

   web 上の文書が一般的な文書と大きく異なる点は，画像，音声等のマルティメディアを取り扱えることと，JAVA 等の言語により文書にアクセスするだけで多種多様なことができてしまうことである．このことが，文書内の文字だけで文書の特徴を表すことを困難にする．特に図 3.3のように，分類するためのキーワードとなる文字が，画像として埋め込まれている場合に有用な情報を手に入れることができない．画像処理を行い輪郭抽出等によって文字を抽出することも考えられるが現在の技術では非現実的であるので本研究では行わない．しかし，現状では，半数以上が文字主体の文書であるので，文章を解析して分類に役立つ情報を引き出すことは有用な手段である．

3. 話題の非均一性

   web 上の文書は，大半が非公式文書であり，多種多様の文書が置かれている．決められたフォーマットも無いため，個々が色々な内容を同じ文書につめこんでいる場合があり，人手での分類でも困難を伴う場合が多々ある．

4. 意味のない情報

   web 上には，作成途中の文書，テスト用の文書，全文検索エンジンに引っかけるための意味のない語の挿入されている文書が置かれていることも多く分類する意味のない文書も存在しているが，分類すべき文書との機械的分別は難しい

このように，web 文書特有の問題があり，高い精度の分類を行うことは難しい．しかし，web 上の文書特有の HTML のタグなどの情報[2]を使えば，文書特徴を抽出する手助けになるであろうと考える．

[1] 主語述語等がはっきりした文法的に正しい文

[2] ホームページ作成ソフトの発展により簡単に単語修飾，書式変更が可能になりタグの情報による文書特徴を得ることが重要になってくるであろう

3.4 位置情報

世界中に散在している HTML 文書は，Web サーバプログラムによって管理されている．そのサーバは世界中至る所に立ち上がっており，ネットワークでつながれている．文書を見る際，どのサーバのどの位置に文書があるか位置を指定しなければならない．これは，URL と呼ばれる共通した

protocol://host.subdomain.domain/path

という形式でアクセス可能である．protocol には，プロトコル名[3] , host は，サーバホスト名 domain は，そのサーバのドメイン名，path には，文書のあるパスが書かれる [19]. domain は，日本の場合 x.y.jp という形になり y には，表 3.2にあげる文字列が入り組織種別毎に決められている [20].

表 3.2: ドメイン第二レベル割り当て

AC	教育および学術機関
CO	企業（または営利法人）
GO	日本国政府機関
OR	その他の団体
AD	ネットワーク管理団体
GR	法人格を有しない団体
NE	ネットワークサービス
地域名	地域密着型ドメイン

この場所を見ることにより，学校教育，企業などの発行する文書であると特定することができる[4]．現在，アメリカで策定中の新ドメイン (shop, arts, rec など[5]) が新設されれば，ドメイン名によって分野が特定できる可能性がでてくる．

[3]http, ftp, gopher などがあるが，本研究では web 上データのやりとりをする取り決めである http のみを取り扱う

[4]アメリカなどの場合企業でなくても，日本の co に対応する com ドメインを取得できるのでドメイン情報に信頼性がない

[5]firm(ビジネスまたは企業), store(購買できる商品を提供するビジネス), web(WWW に関連する活動を強調する組織), arts(文化的および娯楽敵な活動を強調する組織), rec(レクリエーションまたは娯楽敵な活動を強調する組織), info(情報サービスを提供する組織), nom(個別のまたは個人の名称を希望する者)

第 4 章

文書取得と内部表現形式への変換

前章までに，今回使用するカテゴリの構成と割り当てようとする文書の web の特徴について述べた．本章では，実際にカテゴリを利用して文書取得を行い，計算機の内部表現形式へ変換して管理する方法について述べる．

4.1 ディレクトリ検索システムを利用した文書自動取得

前にも述べたように，ラベル付きの事例を集めるために，人手で作成されたカテゴリ階層中にある文書の取得を行う (図 4.1)．

この階層には既存のディレクトリ検索システムである yahho[17] を利用する．この階層には，あらかじめ人手で各ノード (カテゴリ) に該当する文書へのリンクを張ったリストが置かれている A.2．このリストをもとに URL index を作成して www robot[22] をネット上に投入して自動的に文書を取得する．前章で述べたように，リンク先文書も重要な情報源であると考えられるのでこの時同時にノードからリンクされている文書だけではなくその文書からリンクされている文書．さらにそのリンク先文書からリンクされている文書も同時に取得を行う．

しかし，文書からリンクを張るのには，以下のような動機がある．

1. 現文書の内容とと関連性があるためリンク

2. ブラウザやプラグイン[1]を必要とする文書のためそのソフトウェアダウンロードのた

[1]ブラウザの機能を拡張するプログラム

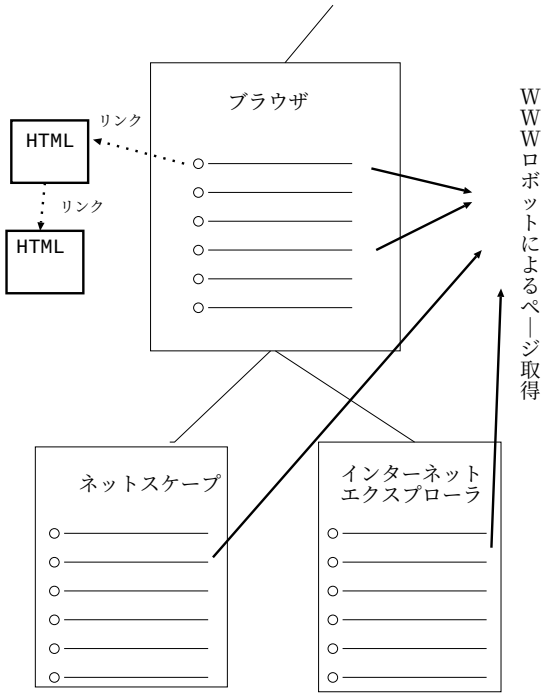


図 4.1: カテゴリ中の文書取得

めのリンク

3. 現文書があるドメインのホームページ等へのリンク (著者の所属部署を代表する文書)

4. リンク集のリンク

5. アクセスログから調べたある文書へどこからリンクが張られているかのリンクリスト (リバースリンク)

これらの中でリンク先の情報が重要であると思われるリンクは，1 と 4 である．リンク種別が 1 と 4 であるリンク先のみを取得するために, 次の規則を適用する．

- http://www.microsoft.co.jp/ , http://www.netscape.co.jp/, http://www.macromedia.co.jp/, http://www.realaudio.co.jp/ などブラウザ，プラグインと深くかかわるページへグラフィックをプッシュすることによってリンク先にたどるようなページはのぞく[2]

- jp ドメイン以外ドメインへのリンクはたどらない

[2] `<a href=''URL''><img src=''hoge''></a>` というリンクの張られ方

- ユーザディレクトリ[3]にあるページから同一ドメインのユーザディレクトリ以外へのリンクはたどらない 4.2)
- ポストスクリプト，バイナリ，プログラム等のテキスト以外へのリンクはたどらない

  gz, gif, lzh 等拡張子によって，テキストではないと判別できるリンク先はたどらない．さらに，web サーバの返す content-type の値をみて，text/html 及び text/plain を返した文書を取得する．

この方法で有用な情報が得られる関連性の高いリンクのみが選択されるわけではない．しかし，不必要であると機械的に判別できるリンクは除去すべきである．

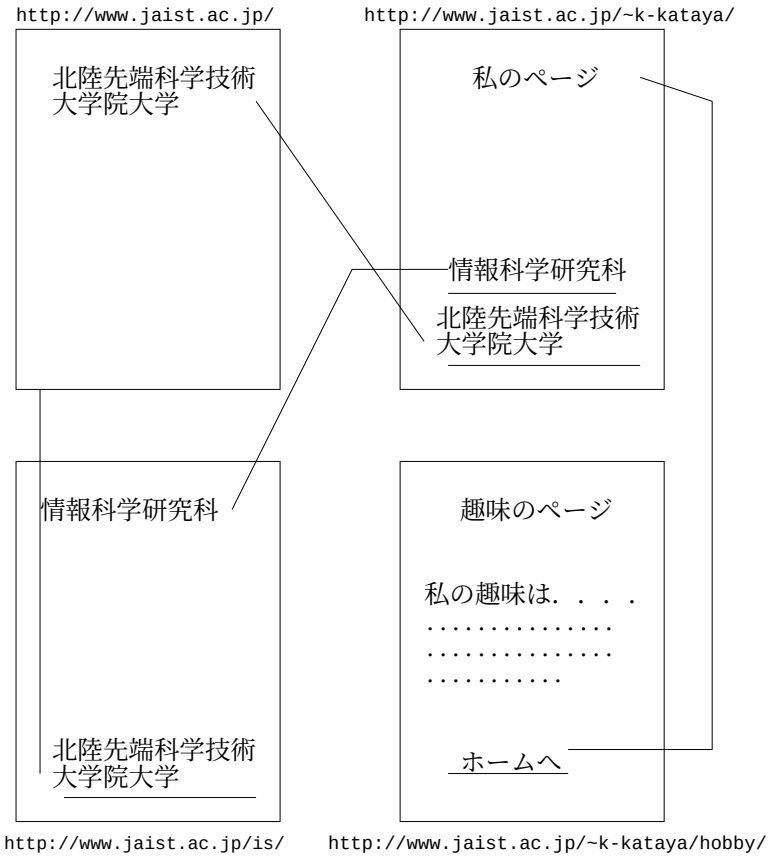


図 4.2: 同一ドメイン上位パスへのリンク

[3]URL の path に , /usr/, /home, /people が含まれている場合

4.2　文書の内部表現形式への変換

前節の方法で取得した文書をベクトルに変換し管理する方法を述べる．
基本的な内部表現形式への変換の流れは，以下の通りである．

1. 取得した文書より文とタグを分離する．
2. 文を形態素解析する
3. 分類に必要であると思われる単語だけを取り出す
4. 文書中の単語頻度を数える．

取得した文書の例を以下に示す.

例 4.1 取得文書

```
<html><head>
<title>RFC in Japanese </title>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=iso-2022-jp">
</head>
<body>
<center><h1> USENET メッセージ交換のためのスタンダード </h1></center>
このメモの位置づけ
<BR>
本稿は USENET ホスト間において、ネットワークニュースのメッセージ交換を行うための標準フォーマットを定義する。本稿はニュースプログラムバージョン B2.11 を反映しており、RFC-850 を更新、置換するものである。このメモはインターネットコミュニティが入手しやすいように、RFC として配布される。本稿はインターネット標準を規定しない。このメモの配布は無制限である。
<P>
<center>原文</center>
<h1>
<center>
A BGP/IDRP Route Server alternative to a full mesh routing
</center></h1>
Status of this Memo
<br>
This memo defines an Experimental Protocol for the Internet
community. This memo does not specify an Internet standard of any
kind. Discussion and suggestions for improvement are requested.
Distribution of this memo is unlimited.
```

4.2.1 文とタグの分離

取得した文書から普通の文タグを分離する.

例 4.2 文とタグの分離

```
RFC in Japanese(title)
USENET メッセージ交換のためのスタンダード (center,h1)
このメモの位置づけ
本稿は USENET ホスト間において、ネットワークニュースのメッセージ交換を行う

～中略～

原文 (center)
A BGP/IDRP Route Server alternative to a full mesh routing(h1,center)
Status of this Memo
This memo defines an Experimental Protocol for the Internet
```

各文の右側の括弧にかこまれている語は，各文を修飾しているタグである.

4.2.2 文の形態素解析

タグと分離した普通の文を形態素解析器にかける．代表的な日本語形態素解析器に CHASEN[4] と JUMAN[5] があるが，数度の形態素解析実験の結果，JUMAN の方が意図する形態素区切りに切ってくれる傾向がある[4]ので，日本語形態素解析には，JUMAN version 3.4 を用いた.

英語の品詞付けには，Brill Tagger[6] version 1.14 を用いた．次に形態素解析を実際に行った結果を示す.

例 4.3 日本語の形態素解析

[4] 専門用語が多く，会話形式の文書も含まれる今回のような文書には辞書単語数の多い JUMAN の方が精度が良いようである

“USENET メッセージ交換のためのスタンダード ”

は，*JUMAN3.4* にかけ，

```
USENET USENET USENET 未定義語 15 その他 1 * 0 * 0
 特殊 1 空白 6 * 0 * 0
メッセージ メッセージ メッセージ 未定義語 15 カタカナ 2 * 0 * 0
交換 こうかん 交換 名詞 6 サ変名詞 2 * 0 * 0
の の の 助詞 9 接続助詞 3 * 0 * 0
ため ため ため 名詞 6 副詞的名詞 9 * 0 * 0
の の の 助詞 9 接続助詞 3 * 0 * 0
スタンダード スタンダード スタンダード 未定義語 15 カタカナ 2 * 0 * 0
```

例 4.4 英語のタグ付け

“This memo defines an Experimental Protocol for the Internet community.”

は，*Grill tagger* にかけ，

```
corpussize = 12, linenums = 93696, tagnums = 113819
START STATE TAGGER::LEXICON READ
START STATE TAGGER:: CORPUS READ
START STATE TAGGER:: RULEFILE READ
START STATE TAGGER:: BIGRAMS READ
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
FINAL STATE TAGGER:: READ IN OUTPUT FROM START
FINAL STATE TAGGER:: READ IN LEXICON
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
This/DT memo/NN defines/VBZ an/DT Experimental/JJ Protocol/NNP for/IN
  the/DT Internet/NNP community./JJ
```

4.2.3 単語の取捨選択

形態素解析を行った結果，文書の特徴を表すのに有用と思われる品詞

日本語の場合 未定義語，普通名詞，サ変名詞，固有名詞，地名，人名，組織名

前節の例では，USENET, メッセージ，交換, スタンダード

英語の場合 NN(名詞単数), NNP(固有名詞), NNS(名詞複数)

前節の例では，memo, Protocol, Internet

である単語のみを使う.

複数名詞の単数化

英語の名詞複数形である単語は，単数名詞にして同じ綴り (意味)[5]の単数名詞はまとめる．ここで複数名詞から単数名詞には，以下の方法によって変換する.

規則変化 EDR 電子化辞書 [7] の英単語辞書から，表 4.1の変化規則を取り出して変換する

不規則変化名詞 不規則変化対応表を作り変換する

上記二つの方法で変換できない単語は，'s がある場合それを取り除くなどヒューリスティクスに変換を行う.

ストップワードである語の除去

多頻出語で，あまり分類には役立たないと思われる単語のデータベースを作っておきその単語が出現した場合除去を行う．(例: 月, 日, 人，私, copyright, japanese, english など)

4.2.4 頻度の計測

前述の処理により選択され得られた語の文書内頻度を計測する.

例 4.5 文書内単語出現頻度

[5] ここでは，多義性は考えない

表 4.1: 名詞の語形変化

変化型	例	不変化部	単数形語尾	複数形語尾
s	boy	boy	-	s
es	box	box	-	es
y	lady	lad	y	ies
fe	wife	wi	fe	ves
f	leaf	lea	f	ves
s & es	potato	potato	-	s,es
's			-	's
s &'s	NP	NP	-	s, '

表 4.2: 頻度の計測

出現した回数	単語名
8	*USENET*
2	メッセージ
1	交換
4	スタンダード
3	*internet*
6	*Protocol*

この単語を要素とするベクトルを頻度ベクトルという．取得した全文書に関してこの頻度ベクトルを作り出す作業を行い, 文書をこのベクトルで管理する．

例 4.6 文書 D の頻度ベクトル

$$\begin{array}{rlllllll} & (USENET, & WWW, & メッセージ, & intenet, & 工業, & \ldots) \\ vector(D) = & (8, & 0, & 2, & 3, & 0, & \ldots) \end{array}$$

第 5 章

文書とカテゴリの類似性と階層的分類法

本章では，前章の頻度ベクトルを使用した，統計的分類の方法と HTML 文書特有の情報を使用した分類の方法について述べる．

5.1 特徴ベクトルの抽出

文書をカテゴリに割り当てるためには，文書の特徴とカテゴリの特徴との類似性を調べる必要がある．文書の特徴を表す特徴ベクトルを作り出すために情報検索の世界で一般的に使われている tf/idf 法を使用する．これは，他の文書にはあまり現れず，対象文書内ではよく現れる単語を重要視するという考えに基づいている．tf と idf は，それぞれ，

- $tf(d,t) =$ (文書 d にける $term\ t$ の割合)

- $idf(t) = log$(文書総数/単語 t が現れる文書数) $+ 1$

である．この tf と idf を掛け合わせた値が，文書 d における単語 t の重みとなる．$(weight(d,t) = tf(d,t) * idf(t))$

この重みを文書内のすべての単語に対して計算してベクトル化したものが，その文書の特徴を表す特徴ベクトルとなる．

$$\text{特徴ベクトル } vec(d) = (weight(d,t_1), weight(d,t_2), \ldots, weight(d,t_n))$$

文書とカテゴリとの類似性を計るために，カテゴリと割り当てる文書の特徴ベクトルを計算する．

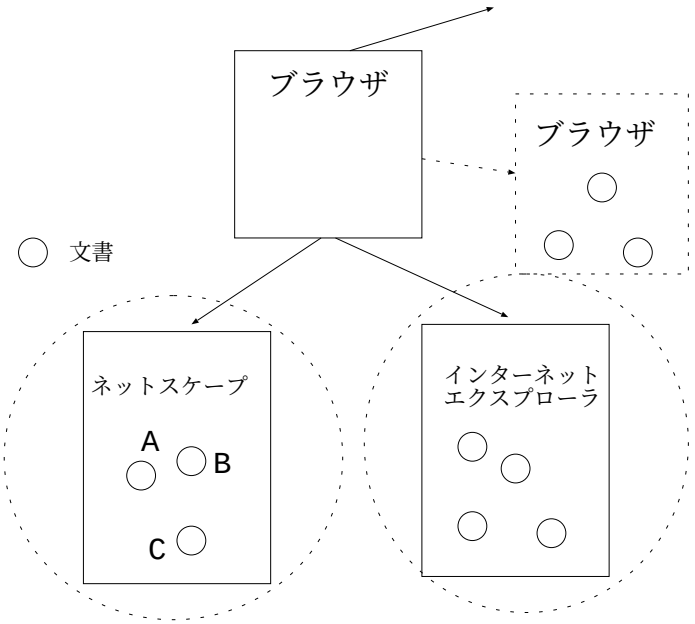
5.1.1 割り当て文書の特徴ベクトル

割り当てる文書は，1 つの文書であり，他の文書との相関関係は考えない．そこで，idf 値は 1 になり, 単語の重みは，$weight(d,t) = tf(t)$　となる.

5.1.2 カテゴリの特徴ベクトル

葉ノードの特徴ベクトル

階層中の葉ノード (カテゴリ) では，複数の文書が人手で割り当てられている．このノードの特徴を表すために本研究では，ノードを一つの文書と考え，ノード内の文書の頻度を足し合わせる.



	ネットスケープ，	インターネット，	マイクロソフト，…
文書A	8，	2，	0，
文書B	4，	1，	0，
文書C	5，	0，	0，
合計	17，	3，	0，

図 5.1: 葉ノードの特徴

図 5.1のようなノードがあった場合ネットスケープのノードの特徴ベクトルを計算する

ために．ブラウザの下位ノードすべての頻度ベクトルを作る．ネットスケープのノードに文書 A, B, C が存在するならば，図のようにネットスケープのノードの頻度ベクトルは，それぞれの頻度ベクトルを足し合わせたものになる．このベクトルを使用すれば，ネットスケープノードを　1 文書と見立てた tf/idf の計算が行える．インターネットエクスプローラのページも同様に，頻度ベクトルの加算を行う．また，ブラウザの内部ノードに割り当てられた文書もブラウザの下位ノードと考え，ネットスケープ, インターネットエクスプローラと同等に取り扱い，頻度ベクトルを計算する.

この 3 つのノードを 3 つの文書とみる．そこで 3 つのノード間で tf/idf による重みの計算を行い，ネットスケープ，インターネットエクスプローラ，ブラウザそれぞれの，特徴ベクトルを計算する．

中間ノードの特徴ベクトル

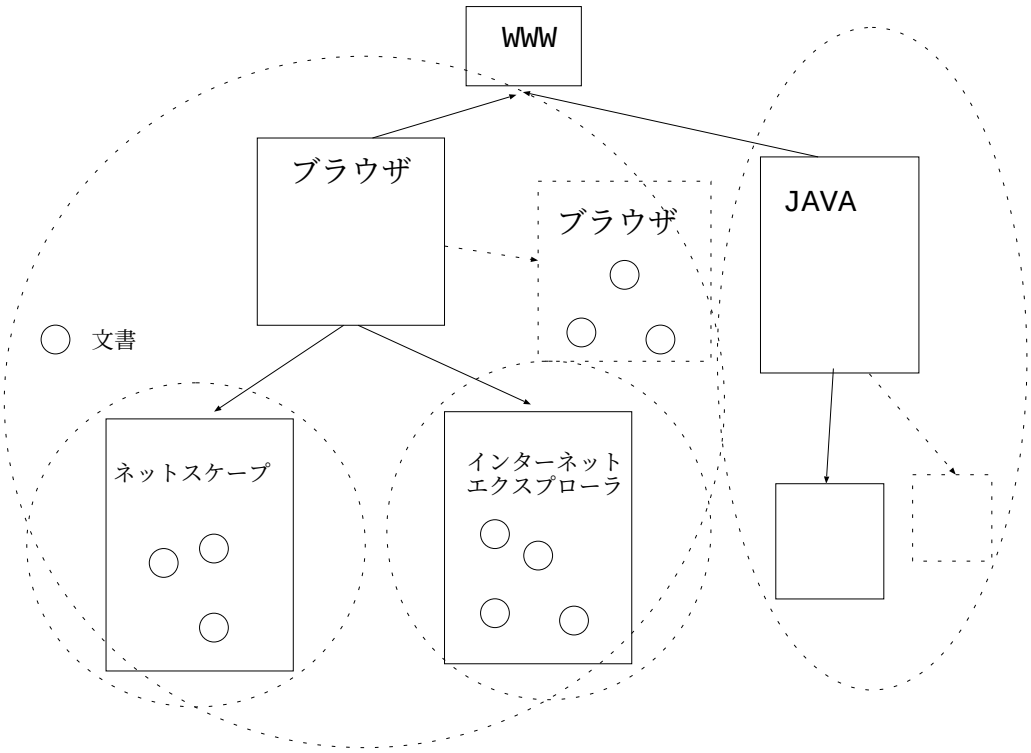


図 5.2: 中間ノードの特徴

階層中の中間ノードの特徴[1]は，下位カテゴリの文書と自ノードに割り当てられている

[1]注）中間ノード自体に割り当てられている文書の特徴ではない

文書を合わせて計算する．つまり自分自身と下位のノードをあわせて，一つの文書とみなす．図 5.2の例では，ブラウザの内部ノード，ネットスケープ，インターネットエクスプローラに割り当てられている文書の頻度ベクトルを合計して，ブラウザ中間ノードの頻度ベクトルとする．同様に，JAVA も自分自身と下位ノード中に割り当てられている文書の頻度ベクトルを足し合わせることによって JAVA 中間ノードの頻度ベクトルを作る．これにより，ブラウザ，JAVA という二つの文書間で tf/idf を計算することによって，ブラウザ，JAVA の特徴ベクトルを計算できる．中間ノードはこのように，自分自身と全下位ノード中の文書を合わせて一つの文書とみなす．

割り当てられる文書と全カテゴリの特徴ベクトルを作り出す．これにより，文書とカテゴリのベクトル間距離が計れ，類似性の計算ができる．

5.2 類似度計算

文書とカテゴリの類似性を計るための尺度として本研究では，以下の式を用いる．

1. 内積

$$inner\_product(C, D) = \sum_{i=1}^{t} c_i \cdot d_i$$

2. ダイス距離

$$Dice(C, D) = \frac{2\sum_{i=1}^{t} c_i \cdot d_i}{\sum_{i=1}^{t} c_i^2 + \sum_{i=1}^{t} d_i^2}$$

3. コサイン距離

$$cosine(C, D) = \frac{\sum_{i=1}^{t} c_i \cdot d_i}{\sqrt{\sum_{i=1}^{t} c_i^2 \cdot \sum_{i=1}^{t} d_i^2}}$$

4. ジャッカード距離

$$jaccard(C, D) = \frac{\sum_{i=1}^{t} c_i \cdot d_i}{\sum_{i=1}^{t} c_i^2 + \sum_{i=1}^{t} d_i^2 - \sum_{i=1}^{t} c_i \cdot d_i}$$

5. ユークリッド距離

$$euclid(C, D) = 1 - \sqrt{\sum_{i=1}^{t} (c_i - d_i)^2}$$

(C はカテゴリ, D は，文書 c_i は，カテゴリの特徴ベクトルの i 番目の要素, d_i は，文書の特徴のベクトルの i 番目のベクトル) 本研究では，この 5 つの類似性尺度を実装して，比較を行っている．

5.3 階層的分類

本節では，前節で計算した，文書とカテゴリ間のベクトル間類似度を使用して，階層的に文書を割り当てる方法を説明する．

2 章でも述べたように，本研究で取り扱う階層の内部ノードには，

1. 現ノードを代表する文書
2. 現ノードより具体的な事象について述べているが，下位ノードには，その事象が入るべきノードが存在しない文書
3. 下位ノードのほとんどの事柄について述べられた文書

という特徴がある．内部ノードが 1 の文書がほとんどである場合は，割り当てる文書と全てのカテゴリとの類似性をとって，類似性の高いカテゴリに割り当てればよい．しかし，2,3 の文書の場合上位ノードからトップダウンに下位カテゴリを含んだ包含関係にある上位ノードとの類似性比較を行っていかないと割り当てることができないと考える．この説明を合わせ，以降階層的分類方法について述べる．

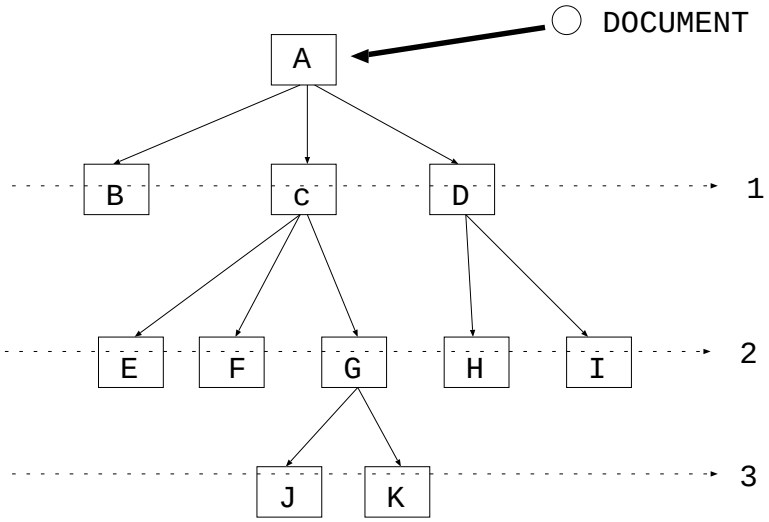


図 5.3: 階層的分類

図 5.3 のような階層を考える．DOCUMENT が今から割り当てる文書である (以下，Doc)．まず，文書 Doc は第一階層であるカテゴリ B, C, D (下位ノードを含めた文書で

計算した特徴ベクトル) との距離計算を行い，以下の表 5.1の順位で類似度がランク付けされたとする．

表 5.1: 類似度のランキング

カテゴリ	類似度
C	0.8
D	0.7
B	0.2

ここで，ある閾値 h を設定する．この値を越える類似度を持つカテゴリに文書は割り当てられる．ここでは，$h = 0.5$ とすると，カテゴリ C, D の類似度が閾値より大きいので，文書 Doc は，カテゴリ C, D に割り当てられる．次に第二階層での分類を行う．最初の分類で Doc は，カテゴリ C, D に割り当てられているので，カテゴリ C, D の下位ノードとの類似性検査を行う．ここで，D を例にとると，文書 Doc は，D 自身，H, I との類似度を計る．その類似度がそれぞれ，D: 0.8, H: 0.6, I: 0.1 とすると閾値より高い類似性を持つカテゴリは，D 自身, H ということになる．この場合，D のみに文書を割り当てる．文書配置ノードからトップノードまでの同一パス上のカテゴリに 1 つの文書が複数配置されることはないからである[2]. すべての下位カテゴリとの類似度が閾値より低い場合，前に述べた内部ノードに配置される条件 2 により内部ノードに割り当てられる．この判断は，トップダウンに分類を行わないとできない．カテゴリ C でも同様に分類を行う．下位カテゴリに割り当てられ，そのノードが内部ノードの場合下の階層のそのノードで同様の分類処理を行う．これ以上分類することができないという階層まで分類を行ってこの処理は終了する．最後に割り当てられたノードが，文書 Doc の所属カテゴリということになる．

5.4 階層を考えた割り当て戦略

前節行った分類処理は，すべて同じ閾値により割り当て先を決定していた．しかし，web 階層では，データ数に偏りがあり，上の方の階層では，多くの下位カテゴリ文書を含んで

[2]逆に D が，トップランクでなければ，配置されることはない

いるので所属単語数が多く，ノード毎に異なる閾値を使う方が良い結果が得られると考える．そこで，あらかじめ人手で分類されている文書を使い最適な閾値を設定する．この閾値を決定するために，各ノードでそのノード及び下位のノードに含まれる文書をランダムに取り出し，以下の式の値を最も高くする閾値を設定する．

$$F = \frac{(1+b^2)PR}{b^2P + R}$$

$$P(recision) = \frac{\text{正しく割り付けたカテゴリ数}}{\text{割り付けたカテゴリ数}}$$

$$R(ecall) = \frac{\text{正しく割り付けたカテゴリ数}}{\text{検査事例セットの正解カテゴリ数}}$$

この F は，Rijesbergen's F と呼ばれていて，recall と precision の 2 種の値を集約して考えることができる．

5.5 タグによる重みづけ

今までの手法は，語の出現頻度情報のみ考えてきた．本節では，HTML 文書の特徴であるタグにより単語の重要度を変化する試みを考える．

本研究では，比較的使用頻度が高く語の重要性に作用していると思われるタグを利用した (表 5.2).

タグを囲んでいる単語への影響力の大きさでグループを A, B, C の 3 つに分けた．4.2.1 によりタグが修飾している単語は分かる．単語の重要度の指針として，単語の出現頻度がある．この情報をタグによって変化させる．今回，グループ A の場合 4，B の場合 3, C の場合 2 回出現したことにして，出現頻度を上げた．出現頻度を変化させる手法により全節までの方法がそのまま使用し分類できる．

この他にも多くのタグが存在し，文書構造の決定に使われる．キーワードタグといわれるキーワード列を記述するタグや分類のためのインデックス列を記述するタグも存在するが，使用頻度が極めて低い．今後利用頻度が高まれば，分類におおきな影響を与えるようになるであろう．

表 5.2: タグのグループ分け

タグ	タグ内容	グループ	タグ	タグ内容	グループ
TITLE	タイトル	A	CENTER	中央寄せ	B
FONT_C	色づけ	B	FONT_1	font max	A
FONT_2	fsize2	B	FONT_3	font min	C
BLINK	点滅	B	NOTE	注意	C
B	ボールド体	B	I	イタリック体	B
S	抹消線	B	TT	等幅	B
U	下線	B	BIG	大文字	B
EM	強調	B	STRONG	さらなる強調	A
DT	表見出し	B	LI	項目	C

5.6 リンクの情報を利用した分類

前節では, HTML 文書の最大の特徴であるタグを利用した特徴抽出の方法を示した. 本節ではもうひとつの特徴である, ハイパーテキスト構造を利用した, テキスト間のリンク関係を利用した特徴抽出の方法について述べる.

HTML 文書は, 図 5.4 のようにリンクによって結ばれている. リンクは, ある文書から関係のある文書に張られており, リンク先の文書の情報も特徴に関連性があるといえる. リンク先文書の情報は, 図 3.3 のように対象の文書から得られる情報が少ない場合に非常に有効であると考えられる.

そこで, 本研究ではリンク先文書の特徴利用のため以下の式を用いる.

$$vec = vec\_org + w_1 * vec_1 + w_2 * vec_2 + \ldots + w_n * vec_n$$

(vec: 加えられたベクトル, vec_org:対象文書ベクトル, vec_n: リンクの深さ n の文書ベクトル, w_n: リンクの深さ n の文書ベクトルに対する重み)

対象文書からできるベクトルと，その文書からリンクされている文書からできるベクトルを加える.

一般的に，対象文書の内容とそのリンク先の文書内容は，リンクを深くたどるほどかけ離れてくる．そこで，w_n の重みをかけ，リンク先の文書の特徴が，対象文書に与える影

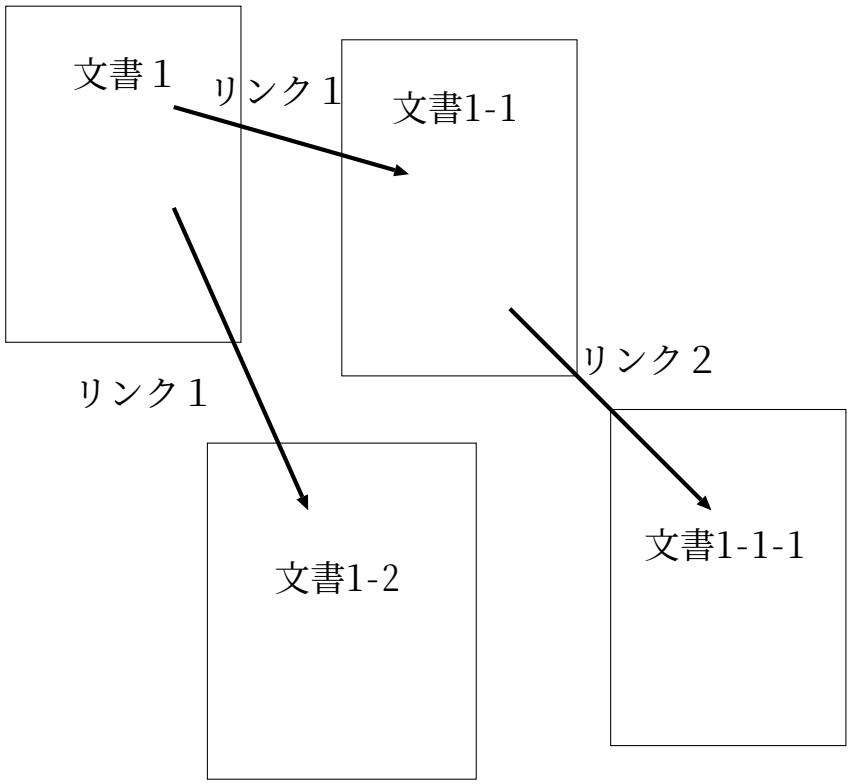


図 5.4: 文書間のリンク

響を制御する.

5.7 URL の情報を利用した分類

いままでの手法は，すべて統計的手法に基づいた分類を行ってきた．本節では統計的分類の手助けになるような知識を利用する方法について述べる．

3.4 節で述べたように，文書の位置情報である URL のドメイン情報は，表 5.3 のように種別ごと区分されており，分類情報の役に立つと思われる．

表 5.3: ドメイン第二レベル割り当て

AC	教育および学術機関
CO	企業（または営利法人）
GO	日本国政府機関
OR	その他の団体
AD	ネットワーク管理団体
GR	法人格を有しない団体
NE	ネットワークサービス
地域名	地域密着型ドメイン

図 5.5 のような階層で，URL が，http://www.hogehoge.ne.jp/ というページを分類を試みる．統計処理による類似度によって，企業というカテゴリのみ割り当てられたとすると，URL の情報からしてあきらかにおかしい．企業だけでなくプロバイダのノードへも入れるべきであり，プロバイダと企業のノードから下位ノードへ統計的分類を行う．

現在，統計的処理の補助的な役割しか果たせていない．ドメインとカテゴリとの関係は表以外にもあり，URL 以外にも文書とカテゴリを結びつける規則性はある．これらの知識をデータベース化を行い，分類に役立てることが望まれる．

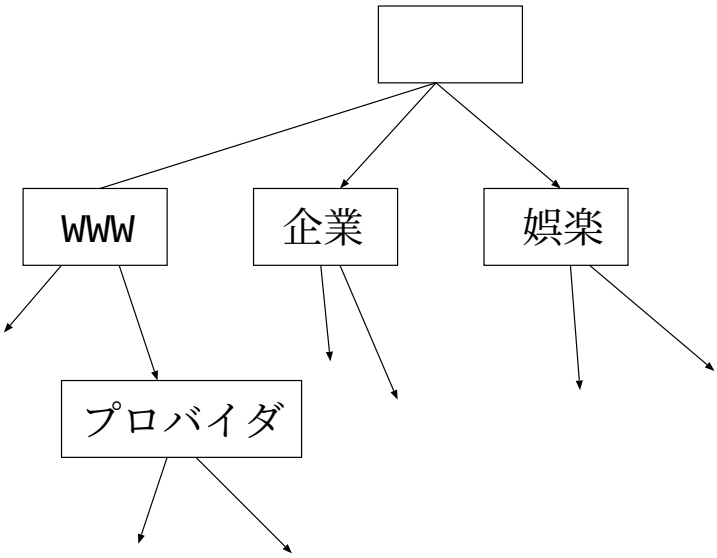


図 5.5: 階層例

第 6 章

分類実験

前章までの手法を使用し，自動分類実験を行った．

6.1 実験データ

yahho 等[1]のデータを用い所属カテゴリと URL の対のインデックスリストを作り，それをもとにして WWW robot による自動文書取得を行った．取得した HTML 文書の文書数と所属カテゴリの対応は，表 6.1である．

合計 22429 文書 (リンク先を含めると 約 86 万文書)

リンク先も含め 2 つに分ける：

訓練文書 (20429), 検査文書 (2000)

6.2 類似尺度の比較

節 5.2 で述べた 5 つの距離尺度を用いて分類実験を行う．

図 6.1,6.2,6.3, 6.4 は，それぞれ，階層 1, 2, 3 までと最後まで分類した階層 max の recall precision グラフである．

階層上位のところでの分類は，内積による距離尺度が一番良い結果が得られた．上位ノードでは，下位カテゴリの文書の特徴が加えられており，異なり語が多くなる．これと

[1]yahho のみに割り当てられているデータのみの収集では，network unreachable, page not found , proxy down 等によりデータ不足が生じる．そこで，その他の全文検索システムや，ディレクトリ検索システムを利用してデータ不足を補った．

表 6.1: 実験データ

大分類カテゴリ	所属文書数	大分類カテゴリ	所属文書数
バイナリデータ	223	イベント	773
企業	3135	政治団体	845
コンピュータ	1367	インターネット	2675
学校教育機関	1237	各種団体	2312
工学	834	娯楽	3967
雑誌・読み物	463	レクリエーション	2326
科学	1002	社会・文化	1170

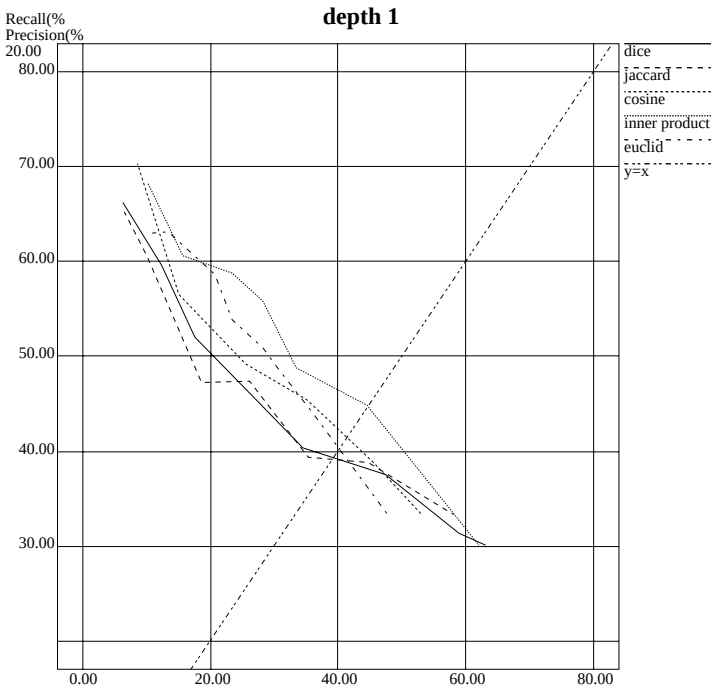


図 6.1: 階層 1

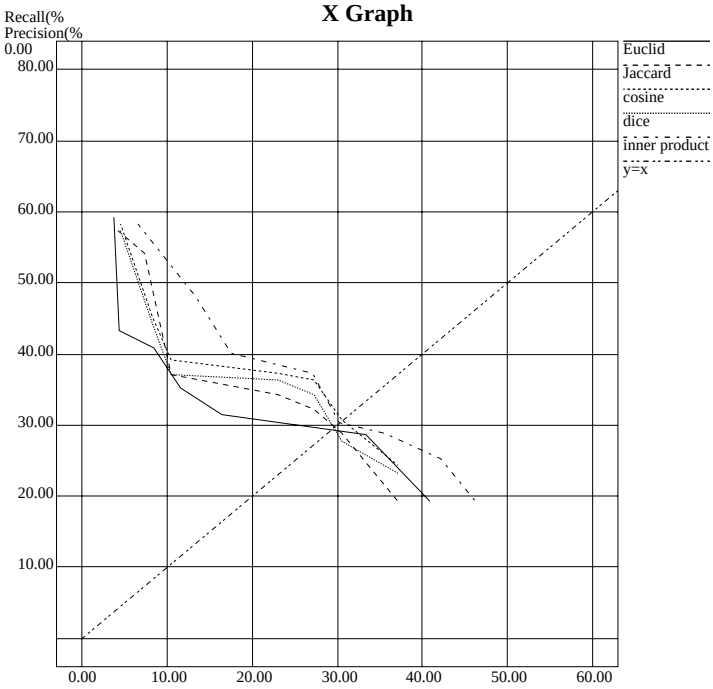


図 6.2: 階層 2

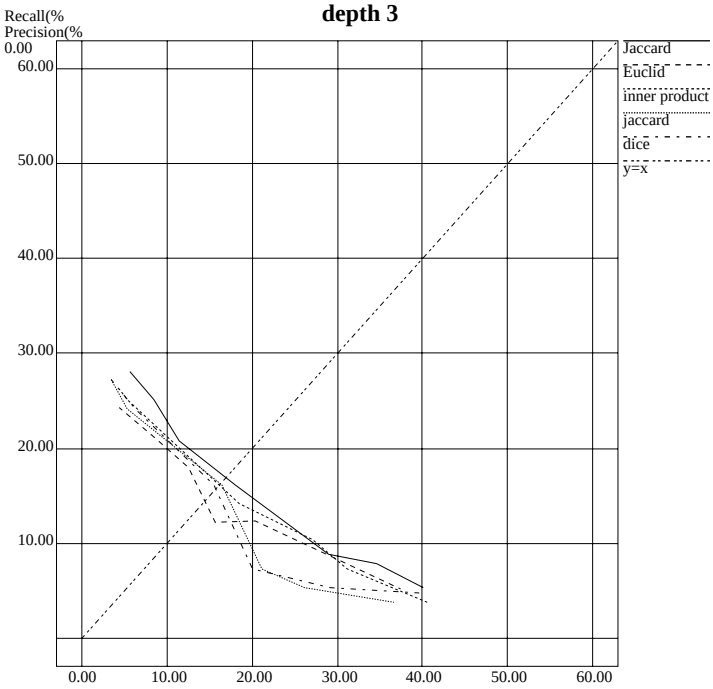


図 6.3: 階層 3

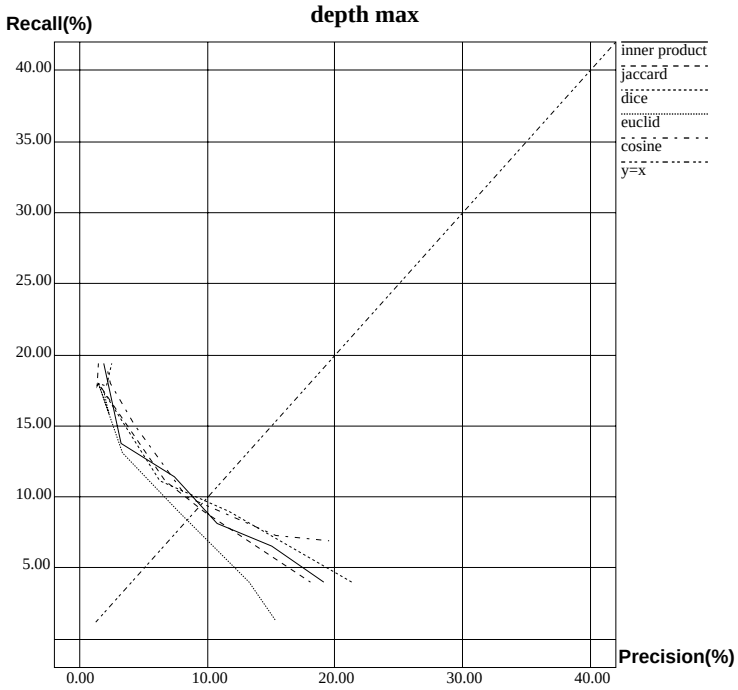


図 6.4: 階層 max

比較して，分類する文書の単語数は少ない．

単語	A	B	C	D	E	F	...)
分類する文書の特徴ベクトル (	0.3,	0.0,	0.8,	0.0,	0.0,	0.0,. . .)	
カテゴリの特徴ベクトル (	0.4,	0.2,	0.6,	0.3,	0.1,	0.4,. . .)	

コサイン，ジャッカード，ダイス距離では，単語数が多くなるほどカテゴリにしか無い単語の部分が雑音となってうまくいかないと考えられる．

階層の下の方になるとコサイン距離の方が良くなっているが，非常に僅差である．そこで，今後の実験では，内積による類似度計算を用いる．

6.3 リンク情報とタグの利用

節 5.5 で述べた手法を使いタグによる重み付けを行う．さらに，リンク先の文書の特徴も加える．以下の 3 つの実験を行い，どの手法が有効であるか調べる．

1. タグによる重みづけ

2. リンク先文書の情報抽出

3. 1,2 を合わせる

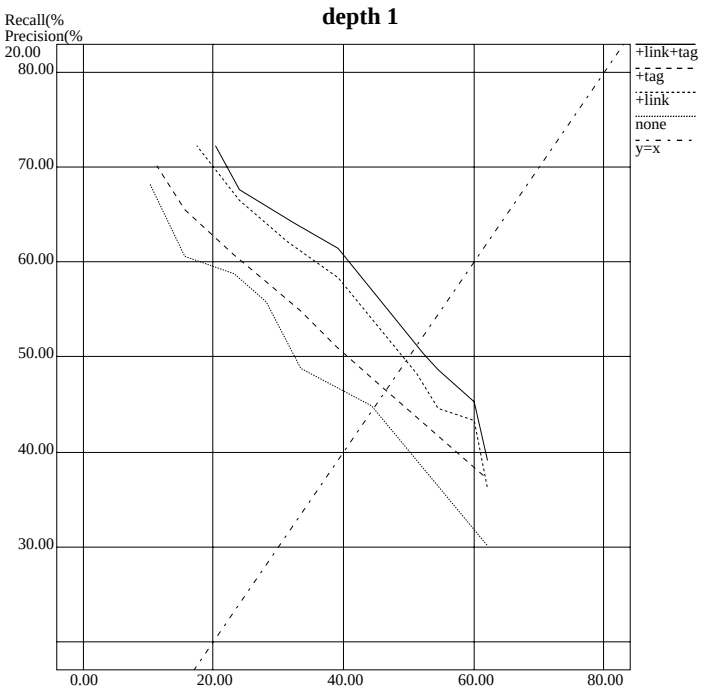


図 6.5: リンク:階層 1(link1: リンク先文書の重み, link2: リンク先文書からリンクされた文書の重み

リンク先の文書の特徴を利用した場合，リンク先の文書からリンクされた文書の情報を利用すると良い結果が得られなかった．リンクが深くなるほど元々の文書からの内容の関連性が薄れていき，余分な情報を取り出していく傾向があるといえる．

また，タグによる重みづけよりリンク先の文書の情報を利用した分類の方が効果があるといえる．さらに，リンク先の文書についてもタグによる重み付けを行った特徴抽出を行うと結果が良くなり，提案手法が有効であることが示された．

6.4 閾値の変動

節 5.4 節の方法により各ノードでの分類の際の閾値を計算する．その閾値を使っての各距離での分類結果を表に記す.

実験の結果, 各階層での閾値の変動による効果が現れている．最初の実験の recall/precision と比べトップノードでの分類以外はすべて精度の向上が計れた．しかし，閾値の設定のた

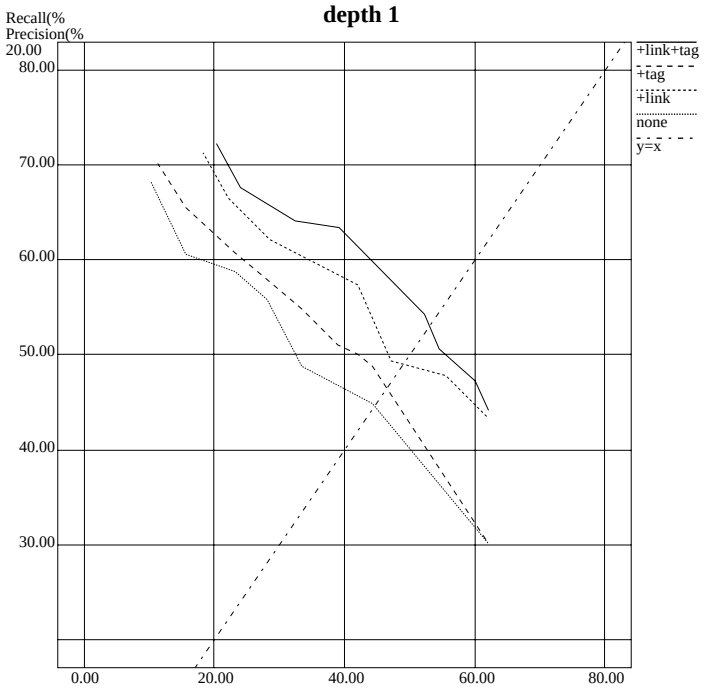


図 6.6: リンク:階層 3

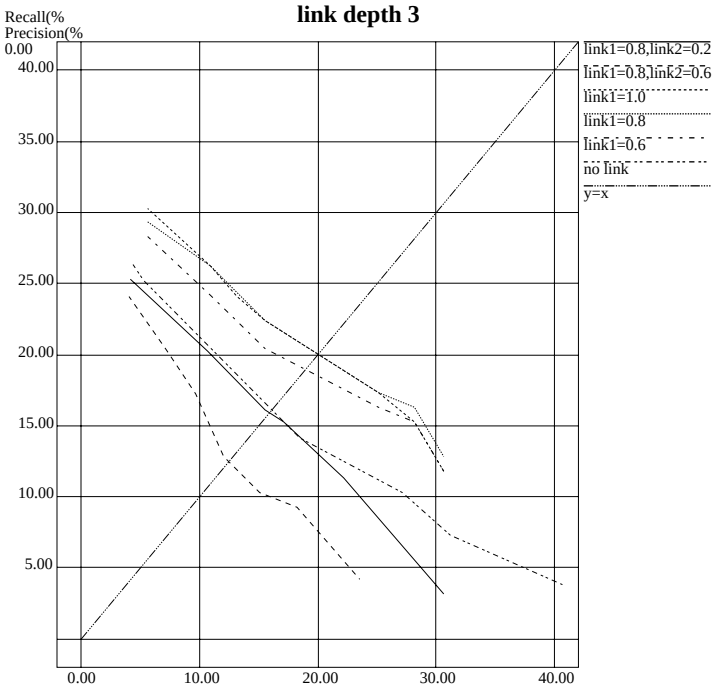


図 6.7: リンク+タグ:階層 1

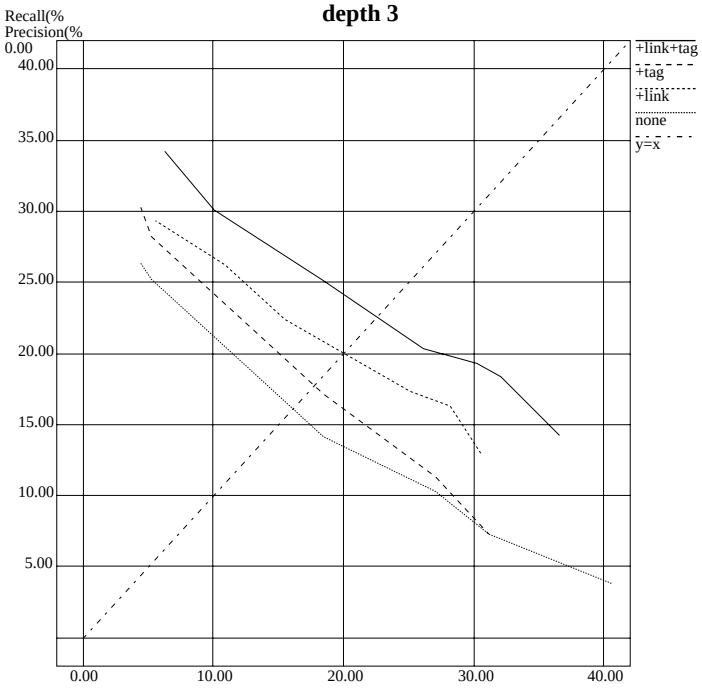


図 6.8: リンク+タグ階層 3

表 6.2: 閾値を訓練事例で設定した場合の結果 (recall/precision)

	depth1	depth 2	depth 3	depth max
内積	46.1/40.4	37.2/34.3	27.1/28.2	17.1/15.4
コサイン	44.2/32.1	35.8/31.2	29.4/24.3	15.6/16.4
ダイス	42.1/33.2	36.1/32.1	26.1/20.1	13.8/10.6
ジャッカード	41.3/42.1	33.1/29.4	23.4/22.6	14.2/13.1
ユークリッド	42.0/38.2	31.1/32.6	16.6/22.1	15.1/10.8

めに訓練事例を使用しているため訓練事例と検査事例のカテゴリ分布が一致しないといい結果が得られないと考える．また，事例不足が生じているノードが多数あるので，事例不足解消を果たせばより良い結果が得られると考える．

6.5 分類する文書の特徴ベクトルの要素に注目した距離尺度を用いた分類

前章で, 分類する文章の特徴ベクトルの特徴ある要素を使用した, 距離尺度の計算について述べた. ここでは, その有効性を実証するための実験を行なう.

6.6 知識の利用

節 5.6　節の方法により統計的方法に加えて，知識を導入する．

以下のルールを知識として利用する．

http://www.hogehoge.??.jp/??/

URL のパス部が１階層のときだけ以下の分類の開始点を以下のノードから行う[2]

- ne.jp ⟶ プロバイダ
- ac.jp ⟶ 学校教育機関
- co.jp ⟶ 企業
- go.jp ⟶ 政治団体

URL ルール適用条件に当てはまる検査文書が　186 文書あった．そのうち，類似度による分類によって以下の表の結果になった．

	分類された	失敗
類似度による分類	124	62

この失敗した文書 62 に，上記のルールを適用することにより, co.jp の場合，類似度にるって分類されたノードとは別に，企業のノードから分類を開始する．企業以下のノード

[2]第２階層以下だとユーザディレクトリになる可能性がありルールが適用できなくなるため

の分類は類似度による方法で行うため下位ノードでの正解は保証できないが，少なくとも企業のある第 1 階層での分類精度の向上にはつながる．

第 7 章

まとめ

本研究では，文書の特徴を抽出するために HTML 文書特有の情報であるタグを利用した, 出現頻度のみによる情報抽出よりもタグの情報を利用した情報抽出の方法の方が, recall, precision 共に精度の向上がみられた．このタグを統計的な手法に合わせて使うことにより，タグが特徴抽出を行うためのに有効な材料になることが分かった．研究対象の階層の下位のノードのどこにも割り当てを行わないという状態を許すために閾値を用いた割り当てを行った．しかし，文書数の偏りなどにより，ノード毎によって類似度の高さが異なる．この解決のため各カテゴリ毎に異なる閾値を用いることによって全カテゴリで同じ閾値を使うよりも良い結果が得られた．

これまでは，統計処理のみによる分類であったが，知識の利用により，精度の向上につなげることを示すため手始めとして URL のドメイン情報を利用した簡単な分類ルールを作った．このルールを用いることで，統計処理による分類の失敗をカバーすることができた．HTML 文書の分類は，他の分野の文書よりも分類のための特徴抽出に難しさがあるが，HTML 特有の情報を使うことにより結果の向上につながることが示せた．

今後の課題として，各文書毎，特徴素としてとれる単語数が異なり，特徴のあまりとれない文書に関してはリンク先の文書の重みを高くするといった文書毎の重み付けを試みたい．また，現在利用している知識が非常に少ない．web 上の文書の特殊性を調査し，知識を増やすことによって統計的方法に頼りすぎない分類を試みたい．

謝辞

本研究を進めるにあたり, 島津明教授ならびに奥村学助教授には数多くの御教示を頂きました. また, 本研究に関して多大な助言をしていただいた本田岳夫氏と望月源氏に心から感謝いたします. Thanaruk Threeramunkong 先生には, 終始あたたかいご助言を頂きました. そして, 島津・奥村研究室の皆様がたには研究に関する貴重な支援をして頂きましたことを心から感謝致します.

最後に，本研究で使用した階層である yahho の管理者である，豊橋技術科学大学の近多 泰宏氏には快く使用許可を頂き心から感謝いたします.

付録 A

yahho ディレクトリ検索システム

図 A.1: yahho の階層トップ

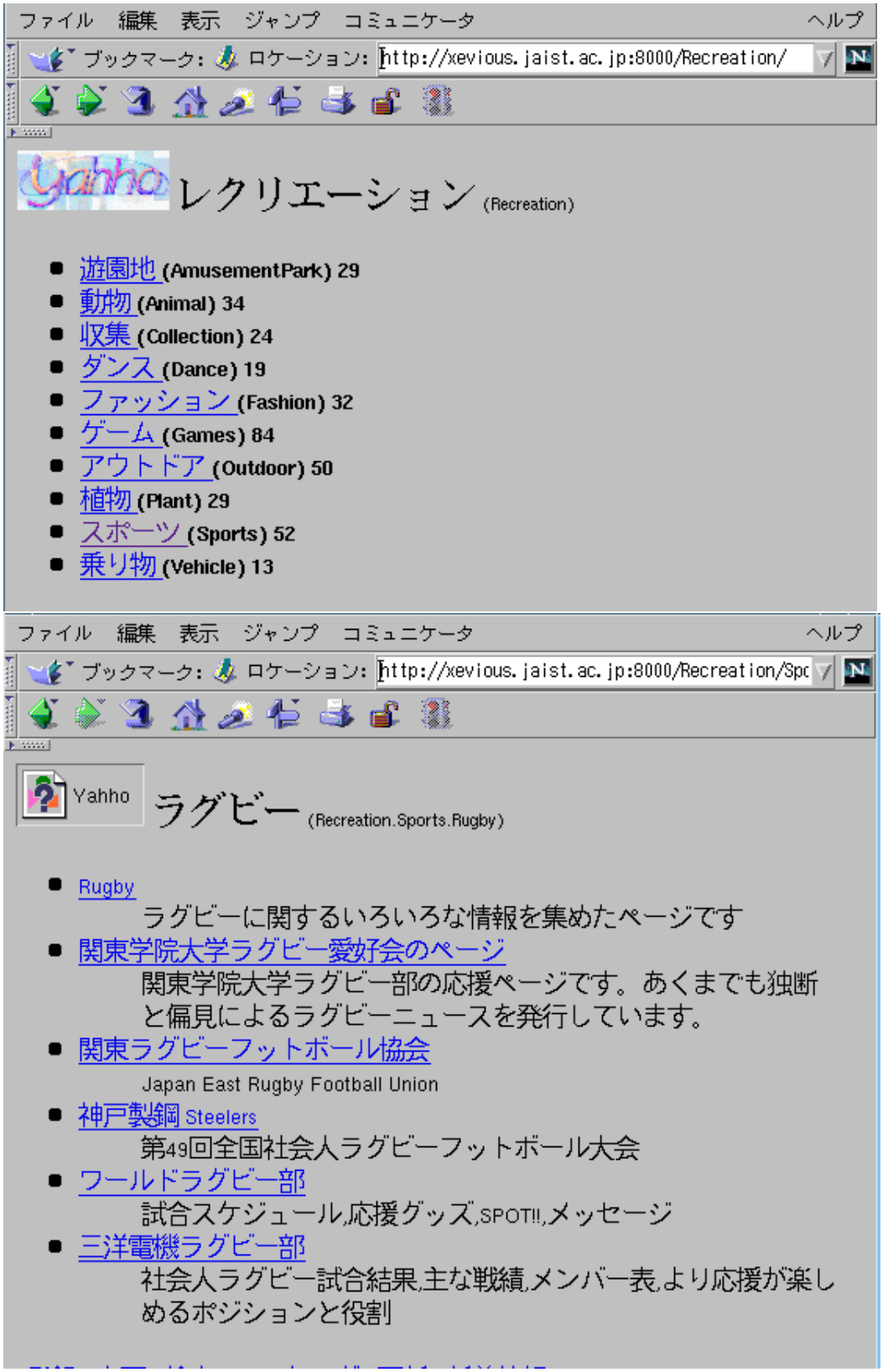


図 A.2: yahho の中間ノードと葉

参考文献

[1] Ray Liere, Prasad Tadepalli, The Use of Active Learning in Text Categorization, AAAI Machine Learning in Information Access, 1996

[2] Maarek Y. S., Shaul B., Automatically Organizing Bookmarks per Contents, Fifth International World Wide Web Conference, 1996

[3] 落谷亮, WWW ページの分類におけるテキストの特徴分析手法, 情報処理研究技報, NL118-14, 1997

[4] 松本裕治, 北内啓, 山下達雄, 今一修, 今村友明 日本語形態素解析システム「茶筌」version 1.5 使用説明書NAIST Technical Report 1997

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

[6] Eric Brill, Some advances in rule-based part of speech tagging Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), Seattle, Wa., 1994.

[7] EDR 電子化辞書仕様説明書 (第２版), (株) 日本電子化辞書研究所, 1995

[8] Beatrice Santorini, Part-of-SpeechTagging Guidelines for the Penn Treebank Project, 1991.

[9] 吉村信, 家永百合子, 鎧聡, インターネットホームページデザインエクステンション, SHOEISHA, 1996

[10] アンク, 最新 HTML タグ辞典, SHOEISHA, 1996

[11] Gerard Salton, Automatic Text Processing, Addison-Wesley Publishing Company, 1988

[12] World Wide Web consotium, <http://www.w3.org/TR/REC-html40/>

[13] “goo”, <http://www.goo.ne.jp/>

[14] “ODIN - Open Documentary Information Navigator”, <http://kichijiro.c.u-tokyo.ac.jp/>

[15] “Excite Search”, <http://www.excite.co.jp/>

[16] “Yahoo!”, <http://www.yahoo.co.jp/>

[17] “yahho”, <http://www.wave.co.jp/wave/>

[18] “NTT DIRECTORY”, <http://navi.ntt.co.jp/>

[19] URI, <http://www.w3.org/Addressing/Addressing.html>

[20] ドメイン割り当て, <ftp://ftp.nic.ad.jp/jpnic/domain/domain-name-all.txt>

[21] New domain, <http://www.fine.ad.jp/fine/c-seminar/Gtld/outlinee.htm>

[22] www robot, <http://info.webcrawler.com/mak/projects/robots/faq.html>