# Web 情報資源を用いた件名と分類の提案

上田　洋(cabgr220@hcn.zaq.ne.jp)　　村上　晴美(harumi@media.osaka-cu.ac.jp)
大阪市立大学大学院創造都市研究科

概要: 我々はこれまで、利用者のキーワード入力により BSH4 件名標目と NDC9 分類項目名の検索が可能な OPAC を開発してきたが、単純なパターンマッチで検索しているため、検索結果がヒットしない場合が多くあった。本研究では、利用者のキーワード入力により関連する BSH4 件名標目（以下、件名）と NDC9 分類項目名（以下、分類）を提示する手法を検討する。利用者の入力は多様であり、新語に対応するため、Web 情報資源に着目する。本稿では、ベクトル空間モデルに基づき、利用者入力に基づき Web 情報資源（Wikipedia、Amazon Web Service、Google）を利用して作成した検索質問ベクトルと、件名や分類から作成した文書ベクトルの類似度を計算して、類似度の高い順番に出力する手法を提案する。本手法を実装したプロトタイプシステムを用いた実験にて、コンピュータ用語を対象とした場合の件名と分類の提示手法の一定の有効性を確認した。

## 1. はじめに

我々はこれまで、利用者のキーワード入力により BSH4 件名標目と NDC9 分類項目名の検索が可能な OPAC[1]を開発してきた。このシステムでは単純なパターンマッチ（部分一致）で検索しているため、検索結果がヒットしない場合が多いという問題があった。

本研究では、検索結果がヒットしない場合でも、利用者が入力したキーワードに関連する BSH4 件名標目（以下件名）と NDC9 分類項目名（以下分類）を提示する手法を検討する。利用者の入力は多様であり、新語に対応するため、Web 情報資源に着目する。

Web 情報を利用した関連語の研究では、Google を使用するものが多い（例えば[2]）が、本研究では、Web 情報として、(1)Wikipedia、(2)Amazon Web Service、(3)Google の使用を検討する。

## 2. 提案手法

本稿では、ベクトル空間モデルに基づき、Web 情報資源（Wikipedia、Amazon Web Service、Google）を利用して作成した検索質問ベクトルと、件名や分類から作成した文書ベクトルの類似度を計算して、類似度の高い順番に出力する手法を提案する。

### 2.1. Web 情報を利用した検索質問ベクトルの作成

利用者の入力する多様なキーワードに対応するために、情報源として、インターネット上のフリー辞書である Wikipedia[3]と、Amazon の書籍データを利用できる Amazon Web Service（以下 AWS）[4]、Web 検索エンジンである Google[5]を利用し、検索質問ベクトルを作成する。

以下では、Wikipedia、AWS、Google の処理手法の概要を述べる。

#### 2.1.1. Wikipedia

Wikipedia[3]とは、Web 上で自由に利用することのできる百科事典である。Wikipedia 日本語版の記事総数は、2005 年 7 月 4 日現在、約 126,618 本である。Wikipedia の記事作成、更新作業は利用者の手にゆだねられている。Wikipedia の中立性を保つ、などの基本方針に遵守すれば、誰でも記事作成や更新が可能である。しかし、誰でも自由に記事作成や更新が行えるため、悪意を持った利用者が記事を作成できるという危険性もはらんでいる。また、宗教上や政治上の問題など、人々によって見解が大きく分かれる記事では、中立性を保ちにくいという問題がある。それらの場合、記事更新を一時凍結するなどの措置がとられている[3]。

本手法では、Wikipedia のトップページにある検索フォームを利用し、利用者が入力したキーワードを用いて検索を行い、記事取得する。記事取得が行えなかった場合、Google Web APIs[6]を利用し、入力キーワードで Wikipedia のサイト内検索を行い、入力キーワードを記事のタイトルに含む結果が存在した場合、その記事を取得し、複数あった場合は、最上位の記事を取得する。入力キーワードをタイトルに含む結果が存在しなかった場合、Google の結果の最上位の記事を取得する。検索結果がない場合は、Wikipedia の記事を利用しないこととした。

Wikipedia の記事は、該当記事が存在しない場合を除き、1 ページ分の情報を取得する。取得したページは、まず該当記事の部分のみを切り出し、記事の目次の部分と、タグを全て削除する。加えて、記事の中の別の記事へとリンクが貼られている文字列と、～で囲まれた文字列を抽出し、それらを重みとして用いた。これらに該当する文字列は、記事の中でも、重要度の高い単語である場合が高いと考える（図 1 参照）。

特に別の記事へリンクが貼られている単語に関しては、記事の上部であるほど、記事に対しての関連する度合いが高い場合が多い。そのため、これらの単語の重み付けは、文章における単語の位

置によって変えることとし、より上部に存在するものにより高い重みを設定した。

以上の処理を行った後、形態素解析を行い、2文字以上から構成される名詞のみを抽出し、不要語処理を行ったものを検索質問ベクトルの一部とする。AWS、Google のデータに関しても同様である。

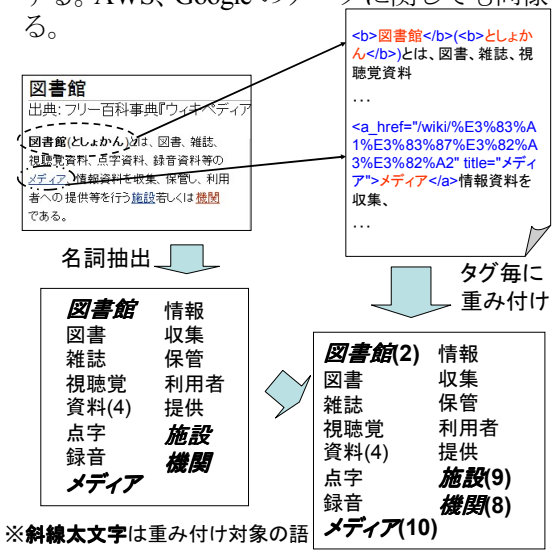


図 1 Wikipedia 重み付け例

### 2.1.2. Amazon Web Service

Amazon Web Service[4]は、Amazon に蓄積された商品に関するさまざまデータを提供する開発者向けサービスである。

本手法では、XML 形式のデータから入力キーワードで検索された書籍の<BrowseNodes>タグ内の情報を利用する。<BrowseNodes>タグ内の情報は、主に、Amazon 独自に付与した書籍に関連する語が記述されている。この情報は、件名・分類などと比較的近い語が利用されていることが多いため、提案作業を行う場合、精度を上げる一要因になるのではないかと考える。

AWS の書籍は、上位 3 冊を利用し、該当書籍の<BrowseNodes>タグ内の情報全てを利用する。この部分を XML 形式のデータから抽出し、タグを全て削除し、使用することとした。

### 2.1.3. Google

Google[5]は、Web 上では最も有名な Web 検索エンジンのひとつである。本手法では、Google Web APIs [6]を利用し、入力キーワードで検索を行い、検索された、Web ページ 5 件を利用する。検索結果の 5 件のページを取得し、タグ除去を行った上で、情報源として用いている。これらの情報については、上記 2 つの情報源の補助的な情報として使用しているため、それらを加工せず使用している。

### 2.1.4. 検索質問ベクトル作成例

本手法で、キーワード「Java」 を入力した場合の検索質問ベクトルの作成例について述べる。抽出情報は、図 2 の記述に沿い説明する。( )内の数字は、語の頻度であり、( )のない語は頻度 1 である。

まず、「Java」 を用いて、Wikipedia、AWS、Google のそれぞれに対し、検索を行う。Wikipedia は、該当記事を取得し、上述した重み付け、形態素解析を行い名詞のみを抽出する。Wikipedia のみのベクトルデータは、「Java(2)、オブジェクト(10)、指向(10)、プログラミング(9)、言語(9)」となる。AWS は、BlowsNodes タグ内のみの情報を抽出し、形態素解析、名詞を抽出する。AWS のみのベクトルデータは、「コンピュータ、インターネット、プログラミング(2)、全般」となる。Google は、検索ヒットした Web ページからタグを除去した文書群に対し、AWS と同様の処理を施す。Google のみのベクトルデータは、「Java(2)、ソフトウェア(2)、ダウンロード、無料」となる。

次に、それぞれのベクトルに対し、重み付けを行い、全てを結合し検索質問ベクトルとする。最終的に、キーワード「Java」での検索質問ベクトルは、「オブジェクト(30)、指向(30)、プログラミング(29)、言語(27)、Java(8)、ソフトウェア(2)、指向、コンピュータ、ダウンロード、無料、全般」となる。

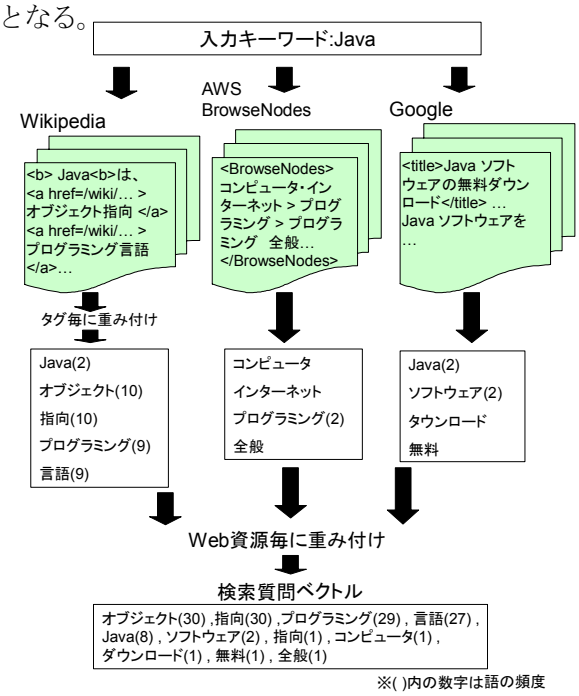


図 2 検索質問ベクトル作成の一例

## 2.2. 件名、分類からの文書ベクトルの作成

本手法では、件名と分類から文書ベクトルを作成する。件名は BSH4 件名標目、分類は NDC9 分類項目名を使用する。

件名の扱いについては、該当件名標目の下位標目と、その下位標目の下位標目を加える。該当件名標目に下位標目が存在しない場合は、該当件名標目のみとなる。下位標目以外の、上位標目、関連標目等については、今回は使用していない。例えば、件名「情報検索」の場合、「情報検索」「索引法」「パンチカード」「データベース」の 4 つの件名を用いて文書ベクトルを作成する。件名「パ

ンチカード」の場合では、「パンチカード」には、下位標目が存在しないので、「パンチカード」のみを用いて文書ベクトルを作成する(図 3 参照)。

分類については、第 3 区分以下を用い、下位 2 区分までの分類項目を該当分類項目の文書として用いた。下位区分の存在しない分類項目は、該当分類項目のみとする。分類には、分類記号が付与されているが、本手法では、分類記号は事前に除去している。例えば、分類「データ処理. 情報処理」の場合では、「データ処理. 情報処理」「データ管理」「システム分析. システム設計」「コンピュータシステムソフトウェア」「エキスパートシステム」「オペレーティングシステム[OS]」「漢字処理システム」「機械翻訳」「図形処理ソフトウェア」「コンピュータ プログラミング」「コンピュータ グラフィックス」「各種の記憶媒体」を 1 つの文書として、文書ベクトルを作成する(図 4 参照)。

なお、これらは事前に処理を行い、形態素解析を行い名詞のみ抽出している。その上で、抽出した名詞の索引ファイルと文書ベクトルファイルを作成した。検索質問ベクトルと同じ名詞を含む件名・分類を文書ベクトルファイルから検索することで処理の高速化を図っている(図 5 参照)。

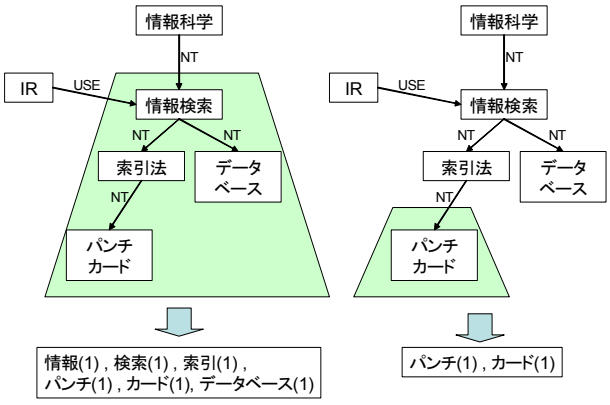


図 3 BSH4 件名標目「情報検索」の文書範囲と BSH4 件名標目「パンチカード」の文書範囲と文書ベクトル

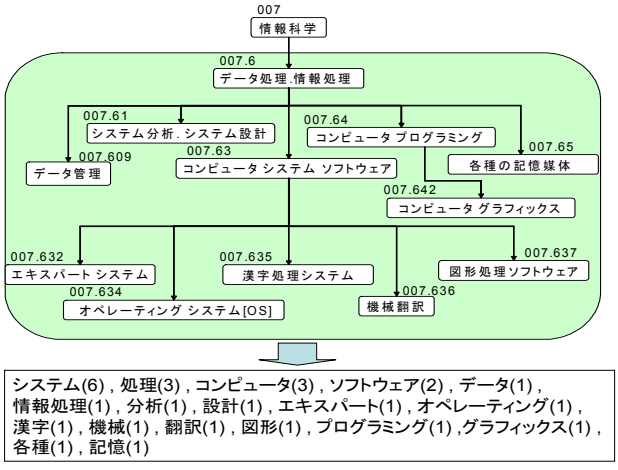


図 4 NDC9 分類項目名「データ処理･情報処理」の文書範囲と文書ベクトル

## 2.3. 処理手順

本手法では、件名・分類の文書ベクトルと、Web 情報資源で構成される検索質問ベクトルの間の類似度を算出し、数値の高い各上位 10 件を関連語として提示する(図 5 参照)。

文書ベクトル $d_x$ と、検索質問ベクトル $d_y$ の間の類似度 $sim(d_x, d_y)$ を

$$sim(d_x, d_y) = \frac{\sum_{i=1}^{T} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{T} x_i^{\,2} \times \sum_{i=1}^{T} y_i^{\,2}}}$$

を定義した[7]。なお、$T$ は索引語の総数を、$x_i, y_i$ は $d_x$, $d_y$ に出現する名詞の頻度を表す。索引語の重みは、$d_x$, $d_y$ に出現する名詞の頻度を 2.2 節のように加工したものである。

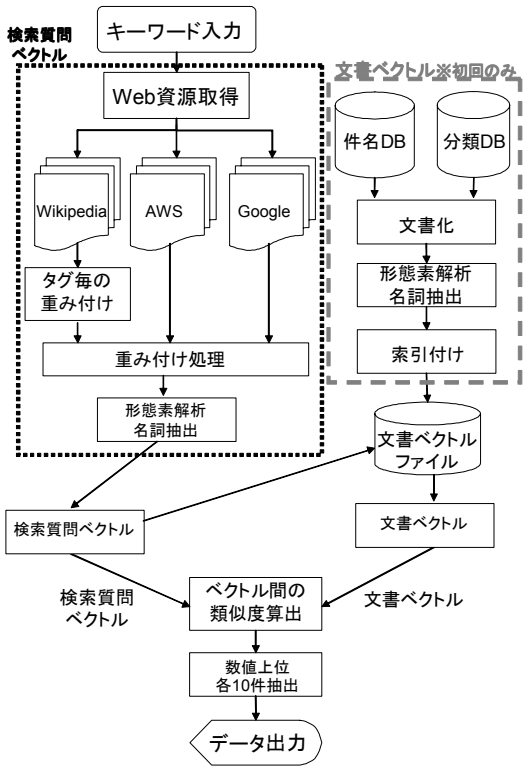


図 5 処理手順

## 2.4. 実行例

入力キーワード「Java」を用いて、本手法を実装したシステムでの実行例について述べる。

まず、システムのトップ画面の入力フィールドに「Java」を入力し、検索ボタンを押す。システムは、キーワード「Java」を用いて、上述した手順を実行したのち、ベクトル間の類似度の数値が高い件名・分類各 10 件を表示する。

キーワード「Java」での 2005 年 7 月 10 日現在の件名の実行結果は、「コンピュータ　プログラミング」「プログラミング(コンピュータ)」「インターネット」「コンピュータ　グラフィックス」「コンピュータ　アート」「コンピュータ音楽」「コンピュータ犯罪」「パーソナル　コンピュータ」「コンピュータ　ネットワーク」「漢字処理

（コンピュータ）」である。同じく分類の実行結果は、「007.64|コンピュータ　プログラミング」「007.642|コンピュータ　グラフィックス」「376.158|言語」「800|言語」「801.02|言語学史」「021.49|コンピュータによる編集」「368.66|コンピュータ犯罪」「375.199|コンピュータ教育」「021.4|編集．編纂」「374.79|教具．教育機器．コンピュータ」である。なお、本手法は日々更新され続ける Web 情報資源を用いているため、常に同じ結果が表示されるとは限らない。

## 3. 実験

本手法の有効性を確認するために、プロトタイプシステムを作成し、実験を行った。

### 3.1. 方法

大阪市立大学学部学生 41 名を対象として 2005 年 7 月 14 日に質問紙調査を行った。IT 用語のオンライン辞典サイトである e-words[8]のアクセスランキングである注目用語ランキング 100[9]の 2005 年 7 月 9 日のランキング（同位が存在するため計 101 語）を用い、件名・分類どちらもパターンマッチで検索される 2 語を除く[1]、計 99 語を被調査者に 5 語ずつわりあてた。

まず、その語をどの程度知っているか 5 段階（5.かなりよく知っている 4.よく知っている 3.どちらともいえない 2.あまりよく知らない 1.全くよく知らない）で評定（既知度と呼ぶ）させ、次に、システムの出力である件名、分類各 10 語を提示して、その語が入力キーワードとどの程度関連しているかを 3 段階（3.関連している、2.ちらともいえない、1.関連していない）で評定（関連度と呼ぶ）させた。

### 3.2. 結果

被験者が言葉の意味がわからず、評価を行えない場合が多く見られた。それらの回答は集計から省き、被験者が回答した範囲で集計を行った。

既知度が 3 以上であった語について集計したところ、(a) 最上位語の評定が最も高い（平均件名:2.34、分類: 2.31、図 6 参照）, (b) 関連度 3 のものを適合とみなし、適合率を判定したところ、件名の上位 1 件（最上位語）で 55%、3 件で 49%、10 件で 41%、分類の上位 1 件(最上位語)で 51%、3 件で 46%、10 件で 40%であった（表 1 参照）。以上の結果より、コンピュータ用語を対象とした場合の件名と分類の提示手法の一定の有効性を確認した。

[1] 本システムは、パターンマッチで検索結果が出力されるキーワードが入力された場合、パターンマッチでの検索を優先的に行う。

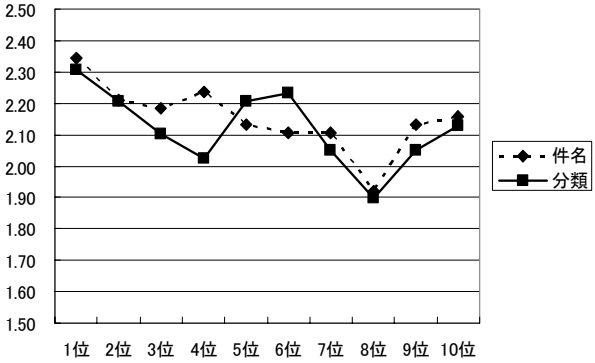


図 6 既知度 3 以上評価グラフ

表 1 実験 1 適合率

| | 上位1件 | 上位3件 | 10件 |
|---|---|---|---|
| 件名 | 55% | 49% | 41% |
| 分類 | 51% | 46% | 41% |

## 4. おわりに

本研究では、利用者が入力したキーワードに関連する件名と分類を提示する手法を検討し、新語に対応するため、Web 情報資源に着目した。

本手法を実装したプロトタイプシステムを用いた実験で、コンピュータ用語を対象とした場合の件名と分類の提示手法の一定の有効性を確認した。

今後は、検索質問ベクトル・文書ベクトル双方に対し、Wikipedia 以外にも重み付けを行うことにより、適合度の向上を目指すと同時に、コンピュータ用語以外での有効性の検証を行いたい。

## 参考文献

[1] 上田 洋, 村上 晴美, 携帯 OPAC の高度化 - 主題検索, 配置画像表示, 内容表示機能の試作 -, 2005 年度日本図書館情報学会春季研究集会発表要綱, pp.67-70, 専修大学, 2005.5.28.

[2] 芳鐘 冬樹, 野澤 孝之, 辻 慶太, 影浦 峡, ウェブからの関連語・下位語の収集手法の検討と検索システムへの応用, 第 52 回日本図書館情報学会研究大会発表要綱, pp.113-116, 関西大学, 2004.11.6-7.

[3] Wikipedia
http://ja.wikipedia.org/

[4] Amazon Web Service
http://www.amazon.co.jp/exec/obidos/subst/associates/join/webservices.html

[5] Google
http://www.google.com/

[6] Google Web APIs
http://www.google.com/apis/

[7] 徳永健伸, 情報検索と言語処理, 東京大学出版, p31, 1999.

[8] IT 用語辞典 e-Words
http://e-words.jp/

[9] IT 用語辞典 e-Words 注目用語ランキング 100
http://e-words.jp/p/s-ranking.html