

蔵書検索のための Web 情報源を用いた件名の提案

上田 洋[†]

大阪市立大学大学院工学研究科[†]

村上 晴美[‡]

大阪市立大学大学院創造都市研究科[‡]

蔵書検索のために、ユーザの入力文字列に関連する件名を提案する手法を検討した。本手法では、Wikipedia, Amazon Web Service, Google の Web 情報源を用いて検索質問を拡張することにより、BSH4 件名標目を提案する。コンピュータ用語を入力文字列とした実験の結果、出力された件名の関連度が一定水準以上であったこと、Web 情報源の統合使用は単体使用よりも有効であったこと、OPAC のデータを用いる手法と比べて遜色のない件名が出力できたこと、がわかった。また、コンピュータ用語や流行語を入力文字列とした実験の結果、OPAC のデータを用いる手法では件名が提示されない場合に本手法では提示できることを確認した。

Suggesting Subject Headings using Web Information Sources for Subject Search in OPACs

Hiroshi Ueda[†]

Graduate School of Engineering, Osaka City University[†]

Harumi Murakami[‡]

Graduate School for Creative Cities, Osaka City University[‡]

We propose a method to suggest BSH4 subject headings by expanding query according to user's input using such Web information sources as Wikipedia, Amazon Web Service (AWS), and Google. Experimental results revealed the following. When computer terms were input, suggested subject headings were related to the input term; suggested subject headings were better when we used a mixture of Wikipedia, AWS, and Google than when we just used one of them; suggested subject headings were not inferior to those suggested by other methods that use OPAC data. We also confirmed that when computer and buzz terms were input, our method can suggest subject headings where other methods cannot.

1. はじめに

図書館における蔵書検索では、探したい蔵書の主題（テーマ）による検索を主題検索と呼ぶ。主題検索とは、著者名、件名、分類記号などの主題情報を検索するものである。この中で、件名とは、主題検索のために蔵書に付与される、あらかじめ決められた用語（統制語）である。図書館では、蔵書を目録データベース OPAC (Online Public Access Catalog) に登録する際に、図書館員が蔵書の内容を分析して、蔵書に対して 1 つあるいは複数の件名を選んで付与する（この作業は件名作業と呼ばれる）。件名作業は図書館員が人手により行うため精度が高い。また、書名が主題を表わしていない場合に、件名は有用なアクセスポイントとなる。たとえば、「インターネット時代の情報探索術」という蔵書に「情報検索」という件名が付与されている場合、ユーザが「情報検索」という件名で検索を行うとこの蔵書がヒットする。また、検索結果には情報検索に関連のない蔵書はほとんど含まれない。このように、件名を用いた検索の大きな長所は「書名に含まれていない主題で検索できる」「精度の高い検索ができる」ことである。OPAC における一般的な件名検索は、ユーザの入力文字列に基づき、件名をパターンマッチにより検索し、ヒットした件名を持つ蔵書を検索する。しかし、現状では、件名検索は一般のユーザにはあまり使われていない。それには以下のような理由がある。

一般的な OPAC では簡易検索（キーワード検索）と詳細検索がある。件名検索は詳細検索の一つの検索項目として実現されている。Google 世代のユーザは簡易検索を好み、詳細検索をあまり使わない。また、詳細検索画面に表示される件名という言葉の意味と使い方をユーザが理解できないために機能が使われないことがある。さらに、日本語の件名の多くは語彙が少なく、日常的にユーザが使う言葉と異なることが多いため、ユーザの入力した文字列でヒットしないことが多い。このため、件名検索を試みてもノーヒットとなることが多く、統制語の検索になじみのないユーザはすぐ諦めてしまう。

上記の問題を克服するためにさまざまな取り組みが行われている。最も単純な方法は、簡易検索（キーワード検索）時に、同時に件名検索も行うものである。ユーザに件名検索をしているという意識を与えずに件名検索ができるという長所があるが、キーワード検索の結果と件名検索の結果を統合するため、件名検索の長所の一つである精度を犠牲にする。

件名を探すために、図書館員や OPAC に慣れているユーザが行う経験的な方法として、まず、キーワード検索を行って検索結果を得てから、適当な蔵書をいくつか選択して、その中に含まれる件名を選択して件名検索を行うことがある。この方法を自動化することによりユーザが件名を選択するための候補を提示することができる。

この方法の一つの成功例として米国研究図書館グループ

の総合目録データベース RedLightGreen[1]がある。RedLightGreen では、ユーザが入力した文字列で検索された結果の蔵書に含まれる件名の一覧を提示し、絞込み検索に利用できる。RedLightGreen のデータベースには複数の参加図書館の大規模の数の図書を含むが、その所在情報等を利用して、適合度順出力と件名提示を行っている。また RedLightGreen は独自開発であるために実用的な速度で実装されている。つまり、この成功のポイントは、総合目録(を利用した適合度順出力)+独自開発であるといえる。

しかし、一般的な OPAC は単館のシステムであり、検索結果を適度順出力はしていない。通常は出版年や書名のヨミの順で出力している。また、コストや人的資源の理由から、一般的な図書館で OPAC を独自開発することは困難であり、ベンダーによるパッケージソフトを導入している。RedLightGreen のように OPAC から件名をマイニングする手法を実用的な速度で実現するためにはベンダーにパッケージ改造を依頼しなければならない。また、依頼が可能となったとしても、単館システムにおける、適合度順出力と件名提示手法は明らかではない。

そこで、我々は、ベンダーから提供される OPAC 本体のシステムはそのまま、無料のインターネット上の情報源を利用して件名を提案する手法を検討する。ユーザの文字列入力に基づき、Web 情報源を利用して検索質問拡張を行い件名を出力する。この手法であれば、企業のパッケージソフトを改造する必要がなく、どの図書館においても利用可能である。また、OPAC から件名をマイニングする手法では、ユーザの入力文字列が、書名、著者名、件名に含まれない場合には何も提案できないが、Web 情報源を用いることで、入力文字列が OPAC に含まれない場合にも件名の提案ができる。本研究では、Web 情報源として、無料で利用できる (1)Wikipedia, (2)Amazon Web Service, (3)Google を使用する。

本稿は以下のように構成する。まず、2 節では件名について説明し、3 節で提案手法について述べる。4 節では提案手法を用いた実験を行い、5 節では関連研究と議論について述べる。

2. 件名とは

図書館における件名とは蔵書を検索するために蔵書に付与される、あらかじめ定められた用語のことである。代表的な件名として米国議会図書館件名標目表 (LCSH)、基本件名標目表 (BSH)、国立国会図書館件名標目表 (NDLSH) などがある。本研究では、基本件名標目表第 4 版 (BSH4) を提案する手法を検討する。BSH4 をとりあげた理由は、BSH が日本の図書館において最もよく使われる日本語の件名であり題材として適当であると考えたことと、BSH4 がシソーラスと類似した概念構造を持ちその構造が利用できるからである。

BSH4 のデータには、件名標目、参照語、説明付き参照、細目の 4 種類がある。それぞれ、7,847, 2,873, 93, 169

項目が含まれている。本研究では「件名標目」と「参照語」を提案の対象とする。件名標目は「を見よ参照あり (UF: Used For)」、「最上位標目 (TT: Top Term)」、「上位標目 (BT: Broader Term)」、「下位標目 (NT: Narrower Term)」、「関連参照 (RT: Related Term)」を持つ。

以下では、件名標目と参照語をまとめて「件名」と呼ぶ。

3. 提案手法

3.1 概要

本研究では、ユーザの入力文字列から Web 情報源 (Wikipedia, Amazon Web Service, Google) を用いて拡張した検索質問ベクトルと、上位下位関係を用いて拡張した件名ベクトルの類似度を計算して、ベクトル空間モデルに基づき類似度の高い順番に出力する手法を提案する。

検索質問ベクトル q と件名ベクトル d_i の類似度を余弦を用いて以下のように定義した。

$$\text{sim}(q, d_i) = \frac{\sum_{j=1}^t w_{qj} w_{ij}}{\sqrt{\sum_{j=1}^t w_{qj}^2} \sqrt{\sum_{j=1}^t w_{ij}^2}}$$

なお、 t は索引語の総数、 w_{qj} は検索質問 q に含まれる索引語 t_j の重み、 w_{ij} は件名 d_i に含まれる索引語 t_j の重みである。件名集合全体のテキストを形態素解析にかけ抽出した名詞 2 文字以上の語を索引語とする。

以下では、検索質問ベクトルと件名ベクトルの作成方法について述べる。

3.2 なぜ Wikipedia, Amazon Web Service, Google を使用するのか

検索質問拡張においてはデータベース中の語を追加することが一般的である。しかし本研究はデータベース中の語を使えないところから出発している。ユーザの多様な入力に対応して、精度の高い検索質問拡張を行うためには、網羅性と精度の両方を高めるような情報源の選定が必要である。そこで我々は Wikipedia, Amazon Web Service, Google の統合使用を検討することとした。

Wikipedia[2]とは、Web 上で自由に利用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。Wikipedia¹の情報はブリタニカ²と同じくらい正確である、という調査結果[3]があるように、Wikipedia の情報は信頼性があると考えられる。このように、最新の情報に対応しながら、なおかつ、高い信頼性も保つ Wikipedia を用いることで、入力文字列に関連する件名を提案できると考える。

Amazon Web Service (以下 AWS) [4]は、Amazon に

¹ 調査は英語版で行われている。

² 市販されている百科事典の一種。

蓄積された商品に関するさまざまな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文字列で検索された書籍の<BrowseNodes>タグ内の情報を主に利用する。<BrowseNodes>タグ内には、該当書籍の分類に当たる語が複数記載されている。これらの語は、Amazon において書籍に付与される主題情報（統制語）に相当すると考える。また、AWS の情報は、全て構造化された情報であり、必要な情報のみを確実に抽出できるというメリットがある。

Google[5]は、Web 上で最もよく使われる検索エンジンである。Google からは網羅的な情報を得ることが期待できる。

Wikipedia を主要な情報源として、AWS と Google を加味することで、網羅性と精度の両方を高めることができると考えた。

以下では、Wikipedia, AWS, Google の情報の処理について述べる。

3.3 Wikipedia

本研究では、ユーザの入力に応じて最適な記事を 1 件取得することを目指す。単純に入力文字列で日本語版の Wikipedia の検索を行うと、複数の異なる記事が出力されることがある。たとえば、「Java」と入力すると、検索結果の 1 位は「ジャバ島」であり 2 位は「Java 言語」である。このような場合に最適な記事を取得するために、Google を利用してユーザの情報要求を推測する。すなわち、Google で上位に来る記事は最近よく参照されるもので、多数のユーザの情報要求に近いだろうというものである。

本手法では、日本語版の Wikipedia のトップページにある検索フォームを利用し、ユーザが入力した文字列を用いて検索を行う。検索結果が 1 件のときには該当の記事を取得する。検索結果がない、または 2 件以上の場合には、Google Web APIs[6]を利用し、入力文字列で日本語版の Wikipedia のサイト内検索を行う。検索結果が 1 件以上の場合には、最上位の記事を取得する。

以下に取得した記事の処理方法を述べる。まず、要素の内容と位置に基づき重要句を抽出する。具体的には要素と、別の記事へのリンクである<a>要素の中身を抽出する。このとき、<a>要素に関しては上部に出現するほど重要であると推測して重み付けを行う。次に、記事から目次やタグ等の不要な部分を削除し、先ほどの重要句とあわせて、形態素解析をかけ、2 文字以上の名詞を抽出した。

以下では、AWS と Google の処理について述べる。これらは補助的な情報源であり、簡素な処理を行う。

3.4 Amazon Web Service

ユーザの入力文字列で Amazon の和書検索を行い、上位 3 件の書籍の<BrowseNodes>タグ内の情報と書籍タイトルを、タグを除去して利用する。Wikipedia と同様に、形態素解析をかけて名詞 2 文字以上の語を抽出した。

3.5 Google

Google でキーワード検索を行い、上位 5 件の Web ペー

ジを取得する。処理時間を短縮するために取得文字数を各ページ 1000 文字に制限している。形態素解析以降の処理は同様である。

3.6 重み付け

索引語 t に対して、以下のように重み付けを行った。

$$W(t) = 3Wi(t) + A(t) + G(t)$$

ただし、 $Wi(t)$ は Wikipedia から抽出した t の頻度、 $A(t)$ は Amazon から抽出した t の頻度、 $G(t)$ は Google から抽出した t の頻度である。

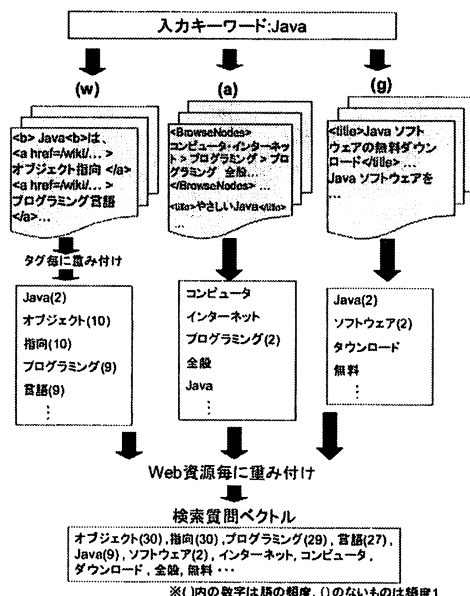


図1 検索質問ベクトル作成の一例

3.7 検索質問ベクトル作成例

文字列「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。図1の記述に沿って説明する。()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、検索と処理を行う。Wikipedia の記事が図1の(w)とすると、Wikipedia から作成したベクトルデータは、「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。同じく、AWS から得られた情報が、図1の(a)とすると、AWS から作成したベクトルデータは、「コンピュータ、インターネット、プログラミング(2)、全般、Java...」となる。Google から得られた情報が、図1の(g)とすると、Google から作成したベクトルデータは、「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。図1の(w), (a), (g)か

ら作成される最終的な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), インターネット, コンピュータ, ダウンロード, 無料, 全般...」となる³。

3.8 件名ベクトルの作成

下位標目と、その下位標目を加えることにより、件名を拡張する。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ない。

たとえば、件名「情報検索」の場合、「情報検索」「データベース」「索引法」「パンチカード」「データベース」の4つの件名を用いて件名ベクトルを作成する。図2の例では、件名「情報検索」の件名ベクトルについては「情報、検索、索引、パンチ、カード、データベース」が重み1となる。

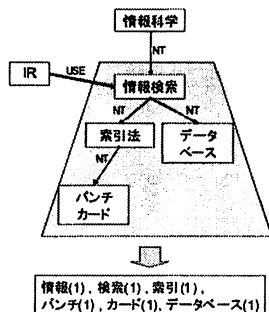


図2 BSH4 件名標目「情報検索」の件名ベクトル作成例

3.9 実行例

入力文字列「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。システムは文字列「Java」を用いて件名10件を表示する。

「Java」での2005年7月10日現在の実行結果は、「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。なお、本手法は日々更新され続けるWeb情報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。

4. 実験

本手法の有効性を確認するために4つの実験を行った。

本研究の特徴は、Web情報源を利用することにより、OPACに含まれない語に対応した件名の提案ができる点にあると考える。OPACに含まれない語の典型例は新語である。実験1-3では、比較的新しい語として、コンピュータ用語を、実験4では非常に新しい語として流行語を用いた。

実験1-3のコンピュータ用語には、IT用語のオンライン辞典サイトであるe-words[7]のアクセスランキングである注目用語ランキング100[8]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）を用いた。

4.1 実験1

学部学生を対象として、本手法の有効性を調べた。

4.1.1 方法

被験者は大阪市立大学学部学生41名である。

コンピュータ用語101語からパターンマッチで検索される2語を除く、全99語を被験者に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）させた。次に、システムの出力である件名10語を提示して、各語が入力文字列とどの程度関連しているかを3段階（3:関連がある, 2:どちらともいえない, 1:関連がない）で評定（関連度と呼ぶ）させた。

実験は2005年7月14日に質問紙調査を行った。

4.1.2 結果と考察

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ため、1人以上の被験者が既知度が3と答えたデータを分析の対象とした。その結果分析対象の用語は31語となった。

既知度が3以上のデータに関しては、最上位語の関連度が最も高く、出力位置が下がるにつれて関連度も下がっている（平均:2.34, 図3参照）。また、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、上位1件（最上位語）で21/38(55%), 3件で56/114(49%), 10件で150/370(41%), であった。

以上の結果より、被験者が既知っているコンピュータ用語を対象とした場合の件名の提示手法の有効性が示唆された。

また、この実験により、被験者が知らない言葉については関連語の評価を得ることが難しいことがわかった。そこで、以下では、被験者の知っている言葉を対象として実験を行うこととした。

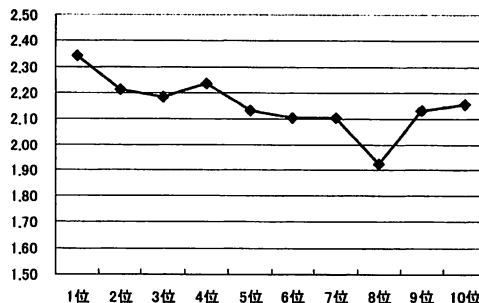


図3 既知度3以上の関連度

³ 実際にはここから索引語以外の語を省いている。

4.2 実験2

検索質問拡張の情報源として、Wikipedia、AWS、Googleを組み合わせてその有効性を確認するために、実験を行った。

4.2.1 方法

被験者は情報学を専門とする大阪市立大学大学院生1名である。

本手法と単一の情報源を比較した。(a) Wikipediaのみ、(b) AWSのみ、(c) Googleのみ、と(d) 本手法の4条件である。ただし、(a) Wikipediaのみについては、記事取得過程は本手法と同じであるが、重み付け処理は行っていない。また、Googleのみについては、本手法での文字数制限を掛せずにページ全てのテキストを取得した。

実験1では、語についての知識不足のため、評価が行えない場合が多く見られた。そのため、実験1で用いたコンピュータ用語31語に対して用語の定義を読ませた。その後、上記4条件で上位10件の件名の関連度(実験1と同じ)を質問紙で評定させた。

実験は2005年12月7日に行った。

4.2.2 結果と考察

1語についてAWS側の検索結果がなく(b)で件名が提示されなかった。その1語の評価については、1.関連していない、と評価されたこととした。

手法別に各順位の平均を算出し、一要因の分散分析を行ったところ、各手法間に有意差が見られた($F(3, 36) = 12.88, p < .01$)。FisherのLSD法を用いて下位検定を行ったところ、本手法とWikipediaのみ、本手法とGoogleのみ、の間に有意差が見られた($p < .01$)。

上位1件については本手法の評価が最も良かった(本手法-2.35, Wikipediaのみ-1.90, AWSのみ-2.19, Googleのみ-1.90, 全て平均, 図4参照)。

また、実験1と同様に、3.関連しているを適合と判断し、適合度を計算したところ、上位1件で58%(最上位語)、3件で55%、10件で43%、であり、本手法が最も良かった(表1参照)。

これらの結果、コンピュータ用語を対象とした場合に、検索質問拡張の情報源としてWikipedia、AWS、Googleの組み合わせが有効であったと考える。

Wikipedia、AWS、Googleの各同一情報源を用いた場合、拡張される語の偏りがあると考える。この偏りが適合率を下げる要因ではないかと推測する。

Wikipediaに関しては、詳細な記事や簡単な記事等、記事によって文書の長さが多様であり、抽出される語の量も多様となる。詳細な記事では、記事の語の意味とは直接関連のない記述、例えば、その語に関する歴史やその語にまつわるニュース、が含まれることも多く、語の内容に反映される。

Googleに関しては、必ずしも入力文字列に関連するページが検索されるわけではないため、入力文字列に関連の少ない語が検索質問に含まれてしまう。

本手法では、複数の検索サイトを用いているため、上記のような語の偏りによる精度の低下をある程度防ぐことができたと考える。

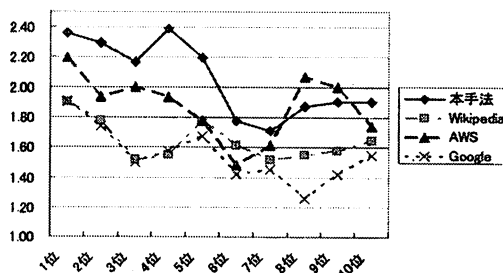


図4 本手法と単一情報源の関連度

表1 本手法と単一情報源の適合度

	上位1件	上位3件	10件
本手法	58%	55%	43%
Wikipedia	39%	25%	24%
AWS	58%	47%	38%
Google	35%	27%	19%

4.3 実験3

本手法と、OPACを用いた件名マイニングを比較するための実験を行った。

4.3.1 方法

情報学を専門とする大阪市立大学大学院生6名を被験者とした。

まず、実験1で用いた31語を用いて大阪市立大学OPACで検索を行い、検索された上位10件の蔵書の書誌情報の中からBSH4の件名を抽出した。検索の結果、蔵書が存在しないものや、件名が複数抽出できないものがあった。そのため、それらを除く17語(表2参照)を用いることとした。また、蔵書10件から抽出した件名は、3から9個とかなり幅があった。そのため、出現頻度順に上位3語を抽出した。この手法をOPACを用いた件名マイニング(以下、OPACマイニングと呼ぶ)とする。

質問紙を用いて、OPACマイニング、本手法で提示された上位3語に対して関連度を評価させた。関連度については、実験1や2で用いたものと同じである。

実験は2006年1月18日と4月19日、24日に行った。

4.3.2 結果と考察

各順位毎の関連度の結果を図5に示す。本手法、OPACとも同程度の評価であった(本手法: 1位-2.14, 2位-2.10, 3位-2.06, OPAC: 1位-2.14, 2位-2.29, 3位-1.90(全て平均))。

「3. 関連している」と判定されたものを適合とみなし、全体の適合度(全データ中3であったデータの割合)を集計したところ、本手法、OPACとも同程度の評価であった(本手法: 64/153 (42%) OPAC: 65/153 (42%))。

表2 使用用語と取得件名数

用語	検索蔵書数	取得件名数
サーバ	596	8
Windows	356	6
DVD+RW	286	4
Java	188	6
OS	145	4
プロトコル	120	8
クッキー	92	9
WWW	58	6
Linux	47	3
ISDN	38	5
アーカイブ	34	4
HTML	31	4
ブログ	21	7
ドメイン	13	8
CGI	11	7
JPEG	7	3
HTTP	4	3

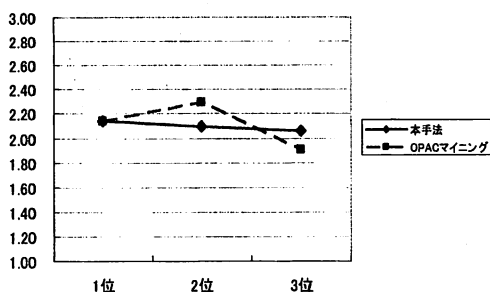


図5 実験3の本手法とOPACマイニングの比較

上記は、コンピュータ用語を対象とした場合、OPACマイニングと本手法の評価が同程度であることを示している。OPACマイニングは、一般的にうまくいく手法として考えられているため、本手法の有効性を表す結果であると考えられる。

OPACマイニングでは、全体的に硬い語が目についた。例えば「Java」では、本手法では「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」が提案されるのに対し、OPACマイニングでは、「電子計算機 -- プログラミング」「データ通信」「通信網」であった。OPACにおいては件名の付与作業において、追加することはあっても過去につけたものの削除は通常行わないため、古い蔵書には古い(当時は新しい)件名が付与されている。硬くや古い言葉が多いことが、評価の低かった一因ではないかと考える。また、大阪市立大学OPACは適合度順出力ではないことも原因の一つであると考えられる。

この結果は、提案された件名が入力文字列に関連があるかどうかに関する本手法の有用性を示しているだけで、本手法がOPACマイニングと比べて件名検索を支援するのに同等の性能を持つことを主張するものではない。この点を調べることは今後の課題である。

4.4 実験4

ある語に関する書籍は出版されているがOPACにまだ蔵書として登録されていない場合や、一時的な流行語でありその内容に関する書籍が出版されないような場合に、OPACを用いて件名を探すことはできない。このような場合、すなわちOPACマイニングができない場合に本手法が有用であると考えられる。

そこで、流行語を入力文字列として用いた場合に、OPACでどの程度検索されるか、本手法で件名を提案できるか調べた。

4.4.1 方法

流行語として、自由国民社の「現代用語の基礎知識」選2005ユーキャン新語・流行語大賞のノミネート語60語(以下、流行語)を入力文字列の候補とした。「〇〇タン」のように任意の文字列を入れるものについては修正、削除を行い、最終的には59語となった。実際の例は、「フォーー！」「愛・地球博」「のまネコ」であった。

上記の59語を用いて、大阪市立大学OPACでキーワード検索を行い、検索件数を調査した。本手法に関しては、件名が提案されるかどうかを調査した。

実験は2005年11月28・29日に第一著者が行った。

4.4.2 結果と考察

OPACでは、39語(65%)で検索結果が0件であった(図6)。検索件数が1件以上10件未満のものが16語(27%)あった。検索結果のあったものでも、入力文字列として用いた流行語と関係のない蔵書が多かった。本手法では、流行語59語全てにつき件名10件を提案できた。

以上より、非常に新しい流行語を入力文字列として用いた場合にOPACでは件名を探せないことを確認した。このような場合本手法は有用であると考えられる。また、新語が一時的な流行語や局所的な語であり、その語をテーマとした書籍が出版されないような場合にも、本手法が有用であると考えられる。

さらに、たとえば「orz(失意を表すアスキー文字)」が入力文字列である場合に、OPACでは「ORZ」という文字列をタイトルに含むドイツ語の文献がヒットしたが、本手法では「現代用語」「時事用語」などの件名を出力した。このように、本手法は、語の意味を知る手がかりとして使用できる。これは間接的に蔵書検索を支援する。

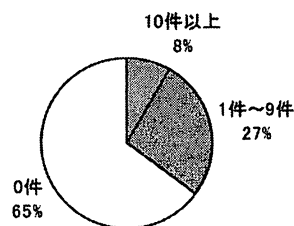


図6 実験4の流行語59語に対するOPACのキーワード検索件数

5. 関連研究と議論

5.1 関連研究

入力文字列から関連語を提示する OPAC には、先にあげた RedLightGreen のほかに、Webcat Plus[9]が存在する。Webcat Plus は、入力文字列で検索結果が上位の本の書名に含まれる語を抽出して統計的に語を提示する。RedLightGreen では統制語を提示し、Webcat Plus では自然語を提示するという違いがあるが、どちらも OPAC マイニングと考えられ、本手法とは異なる。

Web 情報源を用いて関連語を提示する手法は数多く存在する。例えば、専門用語の関連語や下位語を提示する研究[10]、検索語に関連する語を提示する研究[11]、関連用語の自動収集の研究[12]、ある人名に関連する語を提示する研究[13]、などがある。これらは、Web 情報源から抽出した自然語を関連語として用いている。本手法では、Web 情報源から抽出した自然語を提示するのではなく、件名という統制語を提示するための検索質問拡張の情報源として用いている。

件名を提示する研究や実用システムは、医療文献データベースにおいて数多くある。典型的なシステムは、MEDLINE⁴において MeSH (Medical Subject Headings)⁵を提示するものである(例:[14][15][16])。これらは二種類に大別される。一つは本研究と同じように入力文字列に対して件名を提示するものもある。もう一つは、抄録を与えると関連する件名を提示するもので、索引作成者向けの機能である。どちらも基本的な考え方は OPAC マイニングと同じであり、データベースに含まれる抄録情報を利用している。文献の抄録は、概要や著者キーワードなど、OPAC と比べて文献の内容を表わす情報が豊富に含まれており、提示の精度は高い。本研究は、利用できる情報が乏しい現状の OPAC において、Web 情報源を利用することにより、関連語提示が可能となることを示している。

一般的な検索質問拡張は、シソーラスや既存のデータベース内に存在する情報を用いるものが多い[17][18][19][20]。本研究では、内部的な情報ではなく、Web 情報源という外部的な情報を用いて検索質問拡張を行う点が特徴である。

5.2 議論

本研究では、主題検索のために、ユーザの入力文字列に基づき件名を提案する手法を検討した。

関連研究は、OPAC マイニングによるものがほとんどである。OPAC においては、図書館員が各蔵書の内容を分析して件名を付与するため、コンピュータが件名を自動付与するよりも精度が高い。適合度順出力が可能な OPAC においては、検索結果の上位に一般的に有用な蔵書が出現する

ため、その中に出現する件名も、入力文字列に対して関連する可能性が高い。

しかし、ユーザの入力文字列が OPAC の中に存在しない場合や、その文字列を含む情報が非常に少ない場合には、件名を表示できないことがある。これらの場合は、入力文字列が新語、流行語、局地的な語(例:特定の掲示板で使われるような語)などの場合におこると考える。以下ではこれらをまとめて新語と呼ぶ。入力文字列が新語の場合には、その語を書名に含む蔵書が一定量必要である。しかし、新語が生まれて OPAC に登録されるまでに数ヶ月単位のタイムラグがある。すなわち、ある語が生まれて、その語を書名として持つ書籍が出版されるためには、少なくとも1か月以上はかかる。書籍が図書館において選書、購入、蔵書として OPAC に登録されるまでにさらに1か月以上かかる。また、図書館の規模によって蔵書数はまちまちである。蔵書数が少ない場合、情報量の不足のため、適当な件名が提示されない可能性も高まるであろう。

Web においては、新語についても素早く定義や説明について記述されたページが作成される。また、多くの人々が Web ページの作成・更新を行うため、情報量が不足するという心配は非常に少ない。

我々の手法は、特に、ユーザの入力文字列が新語のように、OPAC での出現にタイムラグがある場合に、従来手法と比べて効果を発揮すると思われる。また、蔵書数の少ない図書館において有効であろう。さらに、1節で述べたように、本手法では、ベンダーのパッケージソフトを改造する必要がなく、どの図書館においても無料で利用できることが長所である。

実験3においては、OPAC マイニングと比べて本手法における、提示された語と入力文字列の関連度が遜色なかった。これは、本手法が、入力文字列の語の意味や関連語を調べるために役立つ可能性を示しておりキーワード検索を支援できるだろう。

本研究の課題として以下の点があげられる。まず、提示手法の改善が必要である。たとえば、Wikipedia のみに対し要素を用いた重み付けを行っており、他の情報源に対しては行っていない。件名の拡張に関しては、Web 情報源を用いなかった。これらのアルゴリズムの変更、実験を行う必要がある。また、本手法で提案された件名を用いた件名検索の有効性は確かめていない。今回は件名として概念構造を持つ BSH4 を用いたために件名ベクトルの拡張に上位語や下位語が利用できたが、概念構造を持たない件名についても本手法が有効かどうか今後の課題である。

6. おわりに

本研究では、ユーザの入力文字列に応じて Web 情報源を利用して関連する BSH4 件名標目を提示する手法を提案した。

本手法の評価実験では、コンピュータ用語を入力文字列とした場合に、既知語に対して一定の有効性を示したこと、

⁴ アメリカ国立医学図書館が提供するオンライン医学文献検索サービスの名称。医学系雑誌などの検索が可能。

⁵ 医学用語のための件名標目。

Wikipedia, AWS, Google の統合使用は単体使用よりも結果がよかったこと, OPAC マイニングと比べて件名の入力文字列に対する関連度が高いことがわかった。また, 流行語を入力文字列とした場合, OPAC マイニングでは件名が提示されない場合でも, 本手法では提示できることがわかった。

今後は, 5 節で指摘した課題の改善を行うとともに, コンピュータ用語以外での有効性, 主題検索として用いたときの有効性の確認を行いたい。

参考文献

- [1] RedLightGreen
<http://www.redlightgreen.com/>
- [2] Wikipedia
<http://ja.wikipedia.org/>
- [3] Jim Giles, Internet encyclopaedias go head to head, Nature Vol 438 No 15 December 2005, pp.900-901, 2005.12.
- [4] Amazon Web Service
<http://www.amazon.co.jp/exec/obidos/subst/associates/join/websservices.html>
- [5] Google
<http://www.google.com/>
- [6] Google Web APIs
<http://www.google.com/apis/>
- [7] IT 用語辞典 e-Words
<http://e-words.jp/>
- [8] 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>
- [9] Webcat Plus,
<http://webcatplus.nii.ac.jp/>
- [10] 芳鐘冬樹, 野澤孝之, 辻慶太, 影浦峽, 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.
- [11] 大塚真吾, 豊田正史, 喜連川優, 大域ウェブアクセスログを用いた関連語の発見法に関する一考察, 情報処理学会論文誌データベース(TOD), Vol.46 No. SIG 8(TOD 26), pp.82-92, 2005.6.
- [12] Satoshi Sato and Yasuhiro Sasaki, Automatic Collection of Related Terms from the Web, ACL-03 Companion Volume to the Proceedings of the Conference, pp. 121-124, 2003.7.
- [13] 松平正樹, 上田俊夫, 瀬上正睦, 大沼宏行, 森田幸伯, 文書からのキーワード抽出と関連情報の収集, 人工知能学会, 第 5 回セマンティックウェブとオントロジー研究会, 2004.
- [14] Aronson AR, Bodenreider O, Chang HF, Humphrey SM, Mork JG, Nelson SJ, Rindfleisch TC, and Wilbur WJ. The NLM Indexing Initiative. Proc AMIA Symp 2000;:17-21.
- [15] Kuang-hua Chen, Chien-tin Wu, Automatically Controlled-Vocabulary Indexing for Text Retrieval, Proceedings of Research on Computational Linguistics Conference XII, Hsinchu, 1999, pp. 171-185.
- [16] Kwangcheol Shin, Sang-Yong Han, Alexander F. Gelbukh: Balancing Manual and Automatic Indexing for Retrieval of Paper Abstracts. TSD 2004: 203-210
- [17] 栗山和子, シソーラスを用いた検索式拡張の評価, 情報処理学会研究会報告 98-FI-52, pp. 1-8, 1998.
- [18] 佐々木稔, 新納浩幸, 潜在的な文脈関連度を用いた検索質問拡張, 情報処理学会研究報告, Vol.2002-NL-151, pp.65-72, 2002
- [19] 好田勲, 拓植覚, 獅々堀正幹, 北研二, Non-negative Matrix Factorization を用いた情報検索モデルの次元圧縮および検索質問拡張, 自然言語処理, Vol.54, No.1, pp.17-22, 2003.
- [20] 金谷敦志, 梅村恭司, 相関係数を用いた実証的重みの分析と検索質問拡張, 情報処理学会研究報告 Vol.2003-FI-73, pp.17-24, 2003.