F-08

モノとコトに関連した web ページ検索

Web Search related to Object and Action

船井 一　　小林　優也　　岡留　剛
Hajime　Funai　　Yuya　Kobayashi　　Takeshi　Okadome

1. はじめに

Web 検索研究分野においては，ユーザが必要とする情報を取得しやすくするための検索技術が重要な課題である[11].

一つの方法として，目的と用途を絞り，検索対象とする web ページ群を限定することでその規模を抑える論文情報検索や，求人情報検索などの目的特化型 web サーチエンジンが提案されている.

目的特化型 web サーチエンジンの仕組みはページタイプ選別処理と情報収集・分類処理という二つのステップから成る[4]．ページタイプ選別処理では，個々の目的に適合したページタイプの web ページのみを検索対象として切り出し，情報抽出・分類処理では，それらを目的に応じた分類軸で検索可能にする．一つ目のステップは，web 上の膨大なページに対する作業であり，膨大な時間を必要とするので前処理として行なわれる．あらゆるジャンルのページに対応させるために，このステップを様々なジャンルに対して行なうことは負担が大きい.

既存の目的特化型 web サーチエンジンを統合することよって，それらの多大な作業を省き，なおかつ様々なジャンルに対応した新たな検索エンジンを開発する研究が行なわれている[6]．その研究では，ユーザが入力したクエリに対して適した検索エンジンを選択し，その検索エンジンを使って検索を行なう方略をとっている．この方法では，ある分野に対する目的特化型検索エンジンが存在しなければ，その分野に特化した検索を行なうことができない.

その問題に対処すべく，本研究では，ページ収集の段階でジャンルに則してページをふるいにかけるのではなく，メジャーな検索エンジンを用いて集めたページ群から，より関連性の高いページを取り出しやすくするために，ページのジャンルごとにリランキングを行ない，また，ランキングの際に用いるスコアの計算方法を変更するだけで新たなページのジャンルに対応することのできる web 検索手法の提案する.

2. 関連研究

Web 検索エンジンには様々な研究がある．その中でも本研究に強く関係するものは，ユーザのほしい情報を取り出しやすくするために検索結果を改善するというものである．以下にその中の代表的なアプローチを紹介する.

関西学院大学大学院　理工学研究科
Kwansei Gakuin University

2.1 Blog 情報による検索結果の向上

Blog の持つ情報によって，ユーザに対してより有用な検索結果を与えるように改善する研究がある．この研究は，一般的な検索エンジンの持つ以下の 2 つの問題に対する方法として考案されている．1 つの問題は，最近更新された web ページや，現在話題となっているトピックを観覧することが困難であるということが挙げられる．これは，一般的な検索エンジンで用いられているインデックスを構築することに手間と時間がかかり，その作業中に更新されたページを見ることができないことを意味する．二つ目の問題は，ユーザが見たい情報を持つページは必ずしも有名なページとは限らないということがある．リンク解析手法でランキング上位になるページは他の多くのページからリンクされているものである．リンクが多く貼られたページよりも人が直に推薦するページを重視するほうが直感的である．これらの問題に対して blog が有効であると考えられる理由は，まず blog は書き手が自発的に新たに発見した web ページに対する参照と共に文章を添えて記事にするため，blog を解析することで web 上に新たに追加されたコンテンツや最近の話題を追いやすくなるということと，その記事からリンク先のページに対する評価も推定できると思われるということにある[1].

2.2 ユーザとのインタラクションによる検索結果の向上

検索結果に人手による操作を加えることによって検索結果を向上させる手法がある．この手法は，ユーザがほしい情報のページをより簡単に発見することを目的として研究されている．同一のクエリであってもユーザやその時々によって求める情報が異なる．さらに，現在の検索エンジンでは，クエリの入力とクリックによるリンク先へのアクセスしかユーザは操作できず，ユーザの意志を反映しきれていないという問題がある．この問題に対してこの研究では，ユーザが能動的に検索結果の編集を行ない，その編集操作を通してシステムと対話を行なうことによって，検索結果をユーザの意図を反映したものに変更する．この操作によって「検索結果のこの部分が不要」や「この部分は欲しい」といったユーザの検索意図を推定することによってランキング結果をよりよいものとする[2].

2.3 分類された目的による検索結果の向上

ユーザが必要とする情報のうち，あるジャンルに焦点を置いて，そのジャンルに特化した検索エンジンを作成する

ことでそのジャンルに分類される情報の検索結果のランキング向上を測る手法がある．例として，ある地域の情報の検索のための研究がある[3]．この研究では，地域に関する情報がほしい場合，web ページ中に地域について示している情報が少なく関係のないページを多く取得してしまうという問題に対して web ページのリンク構造と地理情報システム(GIS)を用いて対処しており，「web ページの地域における人気度」と「web ページの地域に対する指向性」という二つについて検討している．

2.4 現在の状況から得られる情報による検索結果の向上

ユーザの状況を web 検索に反映させることで検索結果を向上させる研究も行なわれている．例えば，閲覧している文書を用いるものがある[10]．これは，PC で文書の作成や閲覧している際に必要な情報を探すことがよくあることから考えられたものである．現在，閲覧している文章からクエリのキーワードの周辺のテキストや検索結果の中にあるキーワードの周辺のテキストを用いてクエリをより適したものに修正し，検索結果をよりよいものにする．また，センサデータを扱うものもある[5]．その研究の主な目的は検索結果の向上ではないが，得られたセンサデータを基にクエリを生成し，そして検索結果に対してもセンサデータを用いてリランキングを行なっている．その他に，web ページの内容と検索ワードに対して web ページの間の相互の関係を利用する研究がある[9]．

3. 手法の概要

本手法は，まず「名詞・動詞・ジャンルごとの追加語」または「名詞・ジャンルごとの追加語」の 3 または 2 単語から成るクエリに対して既存の web search API を使用して web 検索を行なう．API で与えられる検索結果について，クエリと関連性の高い web ページをランキング上位に，関連性の低いページをランキング下位に変更するため各ページのスコアを計算し，それを基にリランキングを行なう．本手法の流れを図 1 に示す．

「ジャンルごとの追加語」とは，名詞や動詞に関連するページが膨大にある中でユーザがほしいページの内容に絞り込むための語句である．例えば，ユーザが自分で車の掃除を行ないたいと思い，web からその情報を集めようとする際，「車・掃除」だけでは，洗車の方法を示すページの他に洗車を行なっている店の情報や洗車用の商品の情報のページも多く結果として返されてしまう．そのようなページを出来るだけ減少させるためクエリを「車・掃除・方法」として検索を行なうことが少なくない．このときの「方法」が「ジャンルごとの追加語句」に当たる．

単語数を 3 または 2 とする妥当性は以下の通りである．まず，1 単語だけのクエリでは広範囲なページが取得され，所望のページではないページが多く含まれてしまう．また，3 単語以上のクエリは取得されるページが限定され過ぎ，所望のページが取得できないことが知られている[7]．これらの理由からクエリの主な内容は名詞と動詞の 2 単語とし，ジャンルに対応したページを取得するためにジャンルごとの追加語を加える．しかし，ジャンルによっては，ジャンルごとの追加語句を加えた 3 単語でもページが限定され過ぎてしまうことがあるので，そのような場合に対処するため単語数を 2 または 3 単語とした．

リランキングのために行なう各ページのスコア計算は，一つの定まった式を用いるのではなく，ジャンルごとに異なるスコア式を設定する．

一般的に，ジャンルとしては tips やニュース・グルメ・ショッピングなど多様なものがある．本研究で扱うページ内容のジャンルは，追加語が単純であることを理由に「tips・方法，やり方・ニュース・読物」の四つとした．

ジャンルごとの追加語には，「tips・方法，やり方・ニュース・読物」の順に「tips・how to・news・short story」を使用した．これらは，web 検索において実際にそれぞれのジャンルで使われているものである．

クエリは，「方法，やり方」と「読物」は 3 単語，「tips」は 2 単語，「news」は 2 単語と 3 単語の両方を選択した．ジャンルによりクエリの単語数を変えるのは以下の理由による．すなわち，「方法，やり方」では動詞が最も重要な単語となるため名詞だけでなく動詞もクエリに含ませた．逆に，「tips」ではクエリに動詞を含んでしまうと「方法，やり方」の内容と重複してしまう可能性が大きく，また「tips」ジャンルのページはその名詞に関する豆知識を含むページなので動詞は重要でないからである．どのジャンルにおいても「名詞・動詞・ジャンル」というトピックに則したページを取得するためには，動詞を含めた 3 単語をクエリとするのが望ましい．しかし，「news」のクエリは動詞によってはそれを含むことでクエリに対応するページが大幅に減少する可能性がある．そのため「news」に対しては 2 単語と 3 単語の両方のクエリを試みた．また，web の中には様々な言語で書かれたページが存在するが，本研究では英語で書かれたページのみを対象とする．

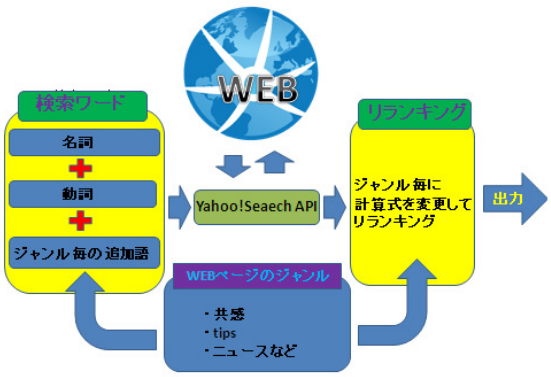

図 1 本手法の概要

4. Web ページのスコアリング

Web ページのリランキングは，(1)Web ページの取得，(2)各ページのスコアリング，(3)リランキングという手順で行なう．(3)は(2)で与えられたスコア基に降順にランキングし直すということである．(1)と(2)に関しては以下で説明する．

4.1 Web ページの取得

Web ページの取得にはメジャーな web search API を用いる．本研究では Yahoo Search API を使用する．Yahoo Search API を選択した理由としては，Google Search API では一度に入手できるページは最大 8 件，一つのクエリに対してその結果を 8 回分しか用意されていないので 1 つのクエリにつき 64 件しか取得できない. しかし, Yahoo Search API は，一度に最大 50 件入手することができ，1 つのクエリに対して 1000 件の結果を返すことができるので，本研究では，1 つのクエリに対して 100 件の結果を対象としたので Yahoo Search API を選択した．

4.2 スコアリング

4.1 で取得したクエリごとの 100 件のページに対してリランキングを行なうために各ページにスコアを付ける．スコアは HTML 内のタグや文字によって計算した．Web ページ解析器には HTML パーサや Beautiful Soup ではなく, SGML パーサを使用した. 当初は Beautiful Soup を使う予定だったが，近年，ページが解析されることを嫌う傾向があり 100 件のページ結果の中に解析不能なページが数多く存在した. HTML パーサも同様に解析不能なページが数多く存在したので，解析不能なページが最も少ない SGML パーサを選択した．英語で書かれた web ページを対象にするので動詞の変化などによって同じ意味を示すものでも異なった形で表現される．この問題を解決するために Porter Stemming Algorithm を用いて，クエリと HTML 内の単語に対して stemming を行なう[8]. 本研究で扱う HTML から検出できる単語は，英数字以外の文字で文章を分割されている文字列である．そのため“C++”や“3.14”のような単語や数値は正しく単語として分割されていないという問題が存在する．しかし，本研究で生成するクエリに含まれる単語は英単語のみなのでそれは問題にはならない.

スコアは，単語の出現頻度や単語間距離などの様々なスコア要因ごとに計算し，その要因ごとのスコアに重みを付け，そのスコアの合計をページのスコアとする．スコア式は文献[12]のもの改変した以下である．

$$\mathrm{Score} = w_1 \cdot \mathrm{ES} + w_2 \cdot \mathrm{FS} + \frac{(w_3 - \min(\mathrm{DS}, w_3))}{\frac{w_3}{w_1}} + w_4 \cdot \mathrm{SS} + w_5 \cdot \mathrm{TS} + w_6 \cdot \mathrm{LS}$$

ここで，ES, FS, DS, SS, TS, LS のそれぞれは，与えられたページに対するスコア要因ごとのスコアであり，w_1からw_6はジャンルに依存する重みである．これをジャンルごとに最適化することでジャンルごとの最適なスコア評価式とする．以下にスコア要因ごとのスコア式について述べる．

4.2.1 Exist 要因

クエリの単語が HTML の<body>タグ内に存在するかどうかを反映する要因である．存在した単語の合計をページのスコアとする．クエリ内のジャンル別の追加語句が 2 単語のものは 2 単語が続いて出現するならば追加語句が存在するとする．Exist 要因は以下のように定義される．

$$\mathrm{ES}(Q, D) = \sum_{i=1}^{N} \mathrm{exist}(q_i, D)$$

$$Q = [q_1, q_2, q_3]$$

ここで，exist(q, D)は，リスト D 中に単語 q が存在するならば 1 を返し，存在しなければ 0 を返す．Nはクエリ中の単語の長さになるので$N = 3$または$N = 2$とする．D は<body>タグ内のすべての単語のリストである. Qはクエリ内のすべての単語のリストである．

4.2.2 Frequency 要因

クエリの単語が HTML の<body>タグ内に存在する回数である．クエリ内の単語が出現した回数にその単語の重みを掛け合せ，その合計をスコアとする．2 単語以上のジャンル別の追加語句に関しては exist 要因と同様に 2 単語が続いて出現すると追加語句が出現したものとする．Frequency 要因は以下のように定義される．

$$\mathrm{FS}(Q, D) = \sum_{i=1}^{N} \sum_{j=1}^{M} \mathrm{equal}(q_i, d_j)$$

$$\mathrm{equal}(q, d) = \begin{cases} 1 & q = d \\ 0 & q \neq d \end{cases}$$

ここで，Q, D, Nは exist 要因のものと同様である．M はページの<body>タグ内の単語の総数である．

4.2.3 Dist 要因

この要因は，クエリ内の単語が複数ある場合にページ内のその単語の距離が近いほど 2 単語の関連性が高く，より適当なページであるという考えに基づく．クエリ内の動詞と名詞との距離によってスコアを計算する．ここで距離は，2 単語の間に存在する単語の数とする．クエリ内の動詞と名詞が<body>タグ内での単語間の距離をスコアとする．

<body>タグ内に名詞や動詞が存在しない場合，又は距離が大きすぎる場合は設定している値を単語間の距離とする．Dist 要因は以下のように定義される．

$$\mathrm{DS}(\mathrm{Q},\mathrm{D}) = \mathrm{dist}(\mathrm{q}_2,\mathrm{q}_3,\mathrm{D})$$

ここで，Q,Dは exist 要因のものと同様である．$\mathrm{dist}(\mathrm{q}_2,\mathrm{q}_3,\mathrm{D})$は単語のリスト D 中の単語$\mathrm{q}_2$と$\mathrm{q}_3$の最小距離である．

4.2.4 Sentence 要因

クエリ内に名詞と動詞が含まれる場合，1 文中にその両方の単語が含まれている文が存在するページはクエリの内容と関連性が高いという考えに基づく．例えば，「私はコーヒーを飲んでいた．子供たちは積み木でお城を作っていた。」という文章があり，クエリ内の動詞が「作る」で名詞が「コーヒー」のとき，単に単語間の距離を測ると小さく，クエリの動詞と名詞が関連していると判断されるが，上記の文章でコーヒーを作っている描写は一切ない．このような問題に対処し，2 単語により関連のあるページを取り出すことを反映させる要因である．この要因は，クエリ内の動詞と名詞が同じ 1 文に存在する割合によってスコアを計算する．ここで 1 文とは，<body>タグ内の内容を“.”, “?”, “!” 又は改行によって分割されたものである．<body>タグ内のクエリに使われている名詞と動詞が同じ 1 文に存在する文の数を動詞の出現回数で割ったものをスコアとして計算する．Sentence 要因は以下のように定義される．

$$\mathrm{SS}(\mathrm{Q},\mathrm{D},\mathrm{S}) = \frac{\sum_{\mathrm{i}=0}^{\mathrm{L}} \mathrm{sen}(\mathrm{q}_2,\mathrm{q}_3,\mathrm{s}_\mathrm{i})}{\sum_{\mathrm{j}=1}^{\mathrm{M}} \mathrm{equal}(\mathrm{q}_2,\mathrm{d}_\mathrm{j})}$$

$$\mathrm{sen}(\mathrm{q}_2,\mathrm{q}_3,\mathrm{SW}) = \sum_{\mathrm{i}=0}^{\mathrm{K}} \mathrm{exist}(\mathrm{q}_2,\mathrm{sw}_\mathrm{i}) \sum_{\mathrm{j}=0}^{\mathrm{K}} \mathrm{exist}(\mathrm{q}_3,\mathrm{sw}_\mathrm{j})$$

ここで，$\mathrm{Q},\mathrm{D},\mathrm{exist}(\mathrm{q}_2,\mathrm{sw}_\mathrm{i})$は exist 要因のものと同様であり，$\mathrm{M},\mathrm{equal}(\mathrm{q}_2,\mathrm{q}_3)$は frequency 要因のものと同様である．S は<body>タグ内の文ごとの単語のリストのリストであり，L は文の総数である．SW は 1 文中に含まれる単語のリストであり K は 1 文中の単語の総数である．

4.2.5 Title 要因

Web ページのタイトルにジャンル別の追加語句が存在するかどうかを反映する要因である．ここでのタイトルとは<title>タグ内の単語のことである．この計算式は exist 要因のものと類似している．異なる点としては単語の存在を調べる位置が<body>タグ内か，<title>タグ内かという点とクエリ内の全ての単語に対して行なうか，ジャンル別の追加語句のみに対して行なうかという点のみである．Title 要因は以下のように定義される．

$$\mathrm{TS}(\mathrm{Q},\mathrm{Ti}) = \sum_{\mathrm{i}=0}^{\mathrm{R}} \mathrm{exist}(\mathrm{q}_1,\mathrm{ti}_\mathrm{i})$$

ここで，$\mathrm{Q},\mathrm{exist}(\mathrm{q}_1,\mathrm{t}_\mathrm{i})$は exist 要因のものと同様である．Ti は<title>タグ内の単語のリストであり，R はその単語の総数である．

4.2.6 List 要因

実際に「方法，やり方」に分類される内容をもつページは，その手順をわかりやすく示すために手順をリストで順に書いているものが多く見られたことから HTML 内に<li>タグが存在するかどうかを一つの要因とした．List 要因は以下のように定義される．

$$\mathrm{LS}(\mathrm{Tag}) = \mathrm{exist}(" < li > ", Tag)$$

ここで，Tag は HTML 内のタグのリストである．$\mathrm{exist}(" < li > ", \mathrm{tag}_\mathrm{i})$は exist 要因のものと類似しており，Tag の中に<li>が存在するならば 1 を返し，存在しなければ 0 を返す．

5. 重みの最適化と評価

本研究では，重み最適化のため，各ジャンルについて 12 種類のクエリで各 100 ページを収集した．このクエリにおける名詞は身近なモノから選び，また，最適化に適したデータを集めるために，クエリに関連するページが 100 件中に多く含まれるように名詞に対応した動詞を選択した．選択したクエリ一覧を付録に記載する．そのページを人手によってどの程度クエリに関連しているかを評価し，各ページにクエリとの関連度の得点を付けたものをデータとして重みの最適化を行なった．重みの最適化の手法として模擬アニーリングと遺伝アルゴリズムの 2 種類の方法を採用した．使用したコスト関数は

$$\mathrm{cost} = \sum_{\mathrm{i}=1}^{100} (101 - \mathrm{rank}_\mathrm{i} * \mathrm{PS}(\mathrm{PE}_\mathrm{i}))$$

であり，このコストが最大になる重みを求めた．ここで，rank_i はページ i の score 値によるランクを表わす．PE は人手によるページのクエリとの関連度である．これは，(1)クエリ内の単語全てに対応しているもの，(2)ジャンル毎の追加語と名詞に対応しているもの，(3)名詞のみに対応しているもの，(4)関連のないもの，(5)ページが存在しないもの，という 5 段階である．なお，動画ページは名詞にのみ対応しているものとしている．PS はその評価に対して点数を与える関数である．今回は(1)から(5)の順に[4, 2, 0, -2, -4]とし

たものと，[12, 4, 0, -2, -3]としたものの 2 種類を用いた．

最適化された重みの評価には，クエリごとのページ群 12 個のうち 11 個を用いて重みの最適化を行ない，残りの 1 個で生成された重みの評価をする，ということをすべての組み合わせに対して行なう leave one out 法を行なった．

表 1 は，クエリごとのページ群 12 個のうち，リランキング前のコストとリランキング後のコストを比較し，コストが増加したものの割合を表している．行は最適化の方法の違いを表しており，列はジャンルの違いを表している．Genetic と annealing はそれぞれ，最適化に遺伝アルゴリズムと模擬アニーリングを用いたものを表している．また，anearing1 と anearing2 の違いは PS の違いである．前者は[4, 2, 0, -2, -4]を使用しており，後者は[12, 4, 0,-2,-3]を使用している．Genetic に関しても同様である．

News-2 は news のジャンルでクエリに用いた単語数が 2 単語のものを表している．また，news-3 は news のジャンルでクエリに用いた単語数が 3 単語のものを表している．

表 2 は，最適化に遺伝アルゴリズムを用いて，PS に[4, 2, 0, -2, -4]を使用した際に生成された重みの平均値の表である．行は各スコア要因を表しており，列はジャンルの違いを表している．

result	annealing1	annealing2	genetic1	genetic2
how to	42%	33%	42%	33%
short story	83%	83%	83%	83%
news-3	33%	25%	42%	25%
news-2	67%	67%	75%	67%
tips	92%	83%	100%	92%

表 1 ランキング改善率

	exist	frequency	dist	sentence	title	list
how to	12	20	47	50	62	16
short story	11	24	50	74	44	0
news-3	20	0	45	17	10	0
news-2	14	31	0	0	10	2
tips	12	30	0	0	0	0

表 2 最適化された重み

6. 考察

2 単語のもの方が 3 単語のものよりも良い結果を示しているが，これは単語数が少ないことで web 上にあるクエリに関連するページの数が 3 単語のものよりも多くなり，100 ページ中にも関連したページが含まれやすくなるということが原因であると考えられる．

How to と new-3 のジャンルに関しては結果が悪化しているもののほうが多い．他のジャンルに比べ，クエリに対応するページ内容を持つページが少ないことが理由だと考えられる．Tips と news-2 は，前述の通り動詞を含んでないのでより多くのページがクエリに対応する．short story は，クエリの動作が含まれた読物なので小説のような実際には起きていないことを書いているページと日記のような実際に起きたことを書いているページの両方が対応しているページである．しかし how to は，ある動作をする方法なのでこのジャンルに対応するページに書かれていることにほとんど差が無く，ページに書かれている内容の種類が少なくなる．そして news-3 は名詞と動詞に関連するニュースそのものが少ない．なぜならニュース記事は新たな発見や珍しい出来事について書かれているので，一般的な動作はニュースになりにくいからである．この how to や news-3 のジャンルの結果を改善するためにはそれぞれのジャンルのページの特徴をより詳しく調べ，新たなスコア要因を検討する必要がある．

次に最適化を行なった重みに注目すると，short story のジャンルは他のジャンルに比べて sentence の重みが大きい結果となっている．これは物語の中で描かれている描写がクエリと同じであるとページがクエリに対応しているとしているので，単に動詞と名詞の距離が近いということよりも 1 文中にクエリ内の動詞と名詞があるものがよりクエリに対応しているページであると考えられる．

また how to のジャンルが他のジャンルに比べて list の重みが大きい結果となっている．これはやはり how to に関するページは list 形式で説明しているものが多いことが理由であると考えられる．

News-3 のジャンルの frequency の重みが 0 である理由と

しては，ニュースのトピックにクエリ内の名詞や動詞が入っていたとしても，ニュース記事内ではその出来事が起こった経緯などのより詳しい説明になるのでトピックに使われた単語はそれほど多く出現しないことが考えられる．また，他のすべてのジャンルが frequency にある程度の重みを与えているので，frequency に大きな重みを与えると，他のジャンルの内容を持つページが上位にリランキングされてしまうという理由も考えられる．

Tips のジャンルの title の重みが 0 である理由としては，tips のジャンルに属するものは豆知識について書かれているページと定義しており，豆知識はメイントピックの補足として書かれていることが多く，ページのメイントピックとして扱われることが少ないことが考えられる．

2 単語でクエリを生成している tips と news-2 はどちらも dist と sentence の重みが 0 になっている．これは dist と sentence のどちらもが, 動詞と他の単語との関係に基づいて計算されるスコア要因であるので，動詞を含まない上記の 2 ジャンルについては意味を持たないということを表している．

7. まとめと今後の課題

Web 検索 API から得られた小規模な web ページ群に対して，各ページにスコアを付け，それを基にリランキングを行ない，クエリに関連するページを上位にするランキング手法を提案した．また，最適化された重みの評価を通じて提案手法の有効性についての評価を行なった．

今後の課題としては考察で述べた改善点を考慮し，ページのスコア計算に新たな指標を追加することと SVN やニューラルネットワークなどを用いてクエリに関連のあるページとないページを分類することを学習することで精度の改善を目指す．

参考文献

[1] 竹原 幹人, 中島 伸介, 角谷 和俊, 田中 克己(2004). Web 情報検索のための Blog 情報に基づくトラスト値の算出方式, DBSJ Letters, Vol.3, No.1

[2] 山本 岳洋, 中村 聡史, 田中 克己(2007). 編集操作を用いたウェブ検索結果の最適化, 電子情報通信学会第 18 回データ工学ワークショップ第 5 回 DBSJ 年次大会, L4-7

[3] 井上 陽介, 李 龍, 高倉 弘喜, 上林 弥彦(2002). 地域情報検索のためのリンク構造分析によるウェブページと地域の関係抽出 Data Engineering Workshop

[4] 福島 俊一(2003). Web サーチエンジンの基本技術と最新動向（下）基本技術, 情報管理 Vol. 46, No, 7 p.436-445

[5] Makekawa, T., Y. Yanagisawa, Y. Sakurai, Y. Kishino, K. Kamei, and T. Okadome (2009). Web Search for Daily Living. SIGIR'09, pp. 19-23

[6] 廣川 佐千男 専門検索サイトの動的統合による次世代検索システムの研究開発, http://www.ipa.go.jp/NBP/13nendo/reports/softseed/nextsrch/nextsrch.pdf

[7] Henziger, M., B. W. Chang, B. Milch, and S. Brin (2003). Query-free news search. WWW2003, pp. 1-10

[8] Porter, M.F (1980). An algorithm for suffix stripping. Program, 4, pp. 130-137.

[9] 荒谷 寛和, 藤田 茂, 菅原 研次(2004). ウェブページ間類似度に基づく推薦リンクを用いたウェブ検索システムの設計. 電子情報通信学会技術研究報告, AI 人工知能と知識処理, pp. 7-12

[10] 河重 貴洋, 小山 聡, 大島 裕有, 田中 克己(2006). 質問修正と再ランキングを用いた文脈依存 Web 検索. 電子情報通信学会, 第 17 回データ工学ワークショップ , 3C-i14

[11] Glover, E J., G W. Flake, S. Lawrence, W P. Birmingham, A. Kruger, C. L Giles, and D. Pennock(2001). Improving Category Specific Web Search by Learning Query Modifications. SAINT, pp. 23-31

[12] Lawrence, S. and C. L. Giles(1998). Inquirus, the NECI meta search engine. WWW-7, pp.95-105

付録

本研究で使用したジャンルごとの 12 種のクエリ．

how to	short story	news-3	news-2	tips
clean window	brush teeth	brush teeth	bath	bath
cut paper	drink tea	cut hair	book	bike
drink tea	make tea	drink tea	car	car
maintain car	open door	drink water	chair	clothes
make chair	open window	drive car	dish	food
make juice	play piano	eat dinner	food	garbage
make tea	pour tea	eat food	garbage	letter
raise plant	read book	play piano	piano	piano
read book	sit chair	read book	scissors	plant
sew clothes	wash dish	ride bike	tea	tea
sharpen scissors	wear clothes	take pic	teeth	teeth
wash dish	write letter	throw garbage	water	water