

編集操作を用いたウェブ検索結果の最適化

山本 岳洋[†] 中村 聡史^{††} 田中 克己^{††}

[†] 京都大学工学部情報学科 〒 606-8501 京都市左京区吉田本町

^{††} 京都大学大学院情報学研究科社会情報学専攻 〒 606-8501 京都市左京区吉田本町

E-mail: †{tyamamot,nakamura,tanaka}@dl.kuis.kyoto-u.ac.jp

あらまし 本研究では、ウェブ検索においてユーザが編集操作を用いることで検索結果を動的にリランキングする手法を提案する。ユーザの検索意図には様々な種類が存在し、検索エンジンが返す結果は必ずしもユーザの意図を反映したものではない。このような場合、ユーザは検索結果を下位まで1つずつチェックするか、新しいクエリで再検索を行う必要がある。本研究の目的は、ユーザとシステムが編集操作を通して対話を行うことを可能とすることによって、ユーザの検索意図を反映することである。本稿では、検索結果に対するユーザの削除操作や強調操作によって、検索結果をリランキングする手法を提案した。また、システムを実装し、システムの有用性を検証した。

キーワード 情報検索, Web とインターネット, 検索意図, インタラクション

Optimizing Web Search Results Using Editing Operations

Takehiro YAMAMOTO[†], Satoshi NAKAMURA^{††}, and Katsumi TANAKA^{††}

[†] Department of Informatics and Mathematical Science, Faculty of Engineering, Kyoto University,
Yoshida-Honmachi, Sakyo, Kyoto 606-8501 Japan

^{††} Department of Social Informatics, Graduate School of Informatics, Kyoto University Yoshida-Honmachi,
Sakyo, Kyoto 606-8501 Japan

E-mail: †{tyamamot,nakamura,tanaka}@dl.kuis.kyoto-u.ac.jp

Abstract This paper proposes a method of a dynamically reranking Web search results according to the user's editing operations while browsing the search results. Because users' Web search intentions are diverse, search engines cannot always return search results that satisfy the user's search intention adequately. Hence, the user must check search results sequentially, or re-search using a new query. Our goal is to reflect the users' search intentions by editing search results through user interaction and propagating the intention of the editing operations. In this paper, we propose a method of reranking web search results depends on users' deletion operations and emphasis operations. We describe the implementation of our system and show experimental results.

Key words information retrieval, Web and internet, search intention, interaction

1. はじめに

近年、ウェブページの検索技術の進歩はめざましく、様々なランキングアルゴリズムが考えられ、実際の検索エンジンに用いられている。ユーザは検索エンジンを利用して検索を行うことにより求める情報を探している。しかし、検索エンジンはいつもユーザの検索意図を十分に反映したものを返してくれるとは限らない。

例えば、A という製品名がクエリとして用いられている場合、あるユーザは A についての情報が知りたい、またあるユーザは A が実際に購入できるページを探しているといったように、ユーザの検索意図には様々な種類が存在する。つまり、同じク

エリを利用していてもユーザによっては検索エンジンに求める結果が異なることが多々ある。同じ人、同じクエリでも、場合によっては求めるものが異なることがあるだろう。そのため、ユーザの検索における意図をユーザの入力したクエリだけを利用して推定することは困難である。

検索エンジンがユーザの意図を推定することが困難なため、ユーザの欲しい結果は検索結果の上位にはなかなか表示されない。その場合、ユーザは自分が欲しい情報が十分に見つかるまで、何番目に登場するか分からない求めるページを探して検索結果を上位から一つ一つチェックしていく必要がある。検索結果を一つ一つ確認していく作業はユーザにとって負荷が大きいため、検索結果の上位の一部を見ただけで検索をあきらめてし

まうことも多い。求める情報が見つからない場合、新たなクエリを検索エンジンに与え、再検索を行う必要がある。しかし、新たなクエリで再検索しようと思っただとしても、どのようなクエリを入力すると求める情報に出会えるかということの思いつくことは容易なことではない。特に、自分が欲しい結果が明確にイメージできない時ほど、新たなクエリを言語化することは難しい。

このように、求める情報を含むページが検索結果ページにおいて上位に表示されておらず、下位に表示されていると予想される場合、ユーザは上位に表示されている自分とは関係ないものを下位へ、下位に表示されていると予想される自分の求めているものを上位へとリランキングしたいと考えるだろう。しかし、現状の検索エンジンではこうしたリランキング機能はサポートされておらず、検索結果のリストをユーザの意図を反映したものにすることは困難である。その理由の一つとして、検索結果閲覧ページとユーザとのインタラクションの少なさが考えられる。現状の検索エンジンでは、ユーザとの間でやり取りされる操作は、キーワードの入力や結果のブラウジング、URLのクリック等の操作しか許されていない。そのため、検索結果の「この話題はいらぬ」とあると「もっとこの話題に関する結果を上位に表示して欲しい」といったユーザの検索意図を反映することができない。

一方、我々は日常的なアプリケーションを用いる中で多種多様な操作を行っている。例えば、文書編集の際には、不要な箇所を削除したり、重要な箇所を強調したりする。また、不要なファイルをゴミ箱へドラッグアンドドロップして捨てるという操作を行う。我々はこうしたウィンドウシステムで利用される削除や強調といった操作をウェブ検索結果閲覧時にも導入し、ユーザが能動的に検索結果の編集を行い、その編集操作を通してシステムと対話を行うことによって、検索結果をユーザの意図を反映したものに更新するシステムを提案する。

システムはユーザの削除操作により「検索結果のこの部分が不要」、ユーザの強調操作により「この部分をもっと欲しい」といったユーザの検索意図を推定する。その後、システムはその意図に合わせて結果をリランキングし、ユーザに結果を表示する。このようにすることによってユーザは手軽に検索結果を最適化できると考えられる。

以降、2章では関連研究について述べ、3章でユーザの操作と検索意図について述べる。4章で実際にキーワードに対する削除操作と強調操作を行ったときのリランキング手法について説明し、5章で実装について述べる。6章で評価実験の結果を示し、7章で考察を行う。その後ユーザが操作の対象とするキーワードの決定支援について述べ、全体のまとめを行う。

2. 関連研究

2.1 ウェブブラウジングと操作

ウェブブラウジングに様々な操作を取り入れる研究は多くある。Zero-Click [1] やズームングクロスメディア [2] はクリックを用いないブラウジング手法を提案している。Zero-Click ではリンク先の内容をマウスクリックすることなく表示することが

できる。しかし、これらの研究はリンク関係にあるコンテンツを閲覧しやすくすることに注目しているだけであり、検索においてリランキングを行うことは考えられていない。

ウェブ検索結果を編集するものに vi search [3] がある。vi search はウェブ検索結果をキーボードのみで閲覧することができる。ユーザは指定したキーワードのハイライト表示や、不要な結果の削除などを行うことができ、ウェブ検索結果ページを編集することが可能となっている。本研究もウェブ検索結果ページを編集するというアイデアを用いているが、vi search では実際にリランキング等が行われることはなく、ユーザの意図を反映することは考えられていない。

2.2 検索意図

ユーザの検索意図を扱った研究は昔から盛んに行われてきた。Broder [4] や Rose と Levinson [5] はユーザの検索意図をそれぞれ三種類に分類している。クエリからユーザの検索意図を分類する手法も多数提案されている [6] [7]。また、田らはユーザのクエリが複数キーワードの場合に、キーワード間の意味的関連を分類し、キーワードの近接性を利用してウェブ検索の精度を向上させる手法を提案している [8]。これらの研究はユーザの検索意図をクエリから推定するものであるが、我々の研究はユーザとシステムとの操作の中から検索意図を推定し、ウェブ検索結果ページに反映するものである。

ユーザが自ら検索意図を指定するものとして、Yahoo! Mindset [9] がある。Yahoo! Mindset では、各ページを Research と Shopping という二つの軸で重みづけをしている。ユーザが自分の検索意図を、Research よりであるか Shopping よりであるかをスライダーを用いて指定すると、システムは指定された重み付けに基づき検索結果をリランキングするものである。また、One to One Ranking System [10] は各ページを様々な軸で重み付けしておき、ユーザが自分の好みをグラフを用いてシステムに直接伝えることによって、ユーザの好みにあった商品をリランキングして表示することを動的に行うことができる。これらのシステムではユーザはバーやグラフを操作することでシステムに検索意図を伝えている。我々の研究は、ユーザの検索結果リストとして表示されているページに対する直接的な編集操作を通してユーザの検索意図を推定するものであり、これらとは異なる。

2.3 リランキング手法

文書をユーザのフィードバックを用いてリランキングする手法の代表的なものとして、適合フィードバック [11] がある。適合フィードバックはベクトル空間モデル [12] にもとづいて、ユーザからのフィードバックをもとに文書をリランキングする手法である。リランキング方法としてはユーザのクエリを再修正するものが良く用いられている。

また、非適合文書のみから適合フィードバックを行う手法も提案されている [13]。適合フィードバックや非適合文書を利用した適合フィードバックは我々のシステムにも適応できるものと考えられる。

3. 編集操作によるユーザの意図反映

例えば、京都の三条に関する情報を探して「三条」というクエリで検索した場合、返ってくる結果の上位は「新潟の三条市」に関する話題が大半である。従って、ユーザは検索結果を上位から一つずつ京都の三条に関するものかどうかチェックしていかなくてはならない。ここで、ユーザの検索意図は、「三条市」という京都の三条に関係の無いキーワードを含む結果を下位にし、「三条通り」といった「京都の三条」に関するキーワードを含む結果を上位にリランキングすることである。このような場合に、ユーザは編集操作を用いてシステムに働きかけ、システムはユーザの操作に基づき検索結果をリランキングしユーザに提示する。

3.1 ユーザとシステムの対話

ユーザは検索結果を編集し、システムはユーザの編集操作の種類と操作の対象に基づき検索結果をリランキングしユーザに提示する。ユーザとシステムの対話に関する一連の流れは図1の通りである。

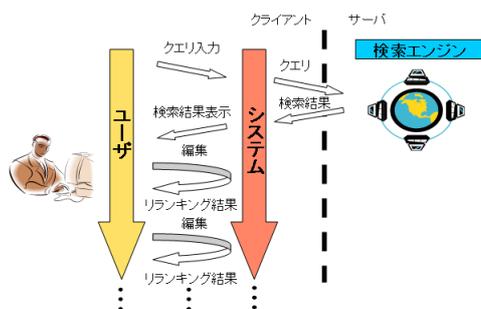


図1 システムのイメージ

- (1) ユーザはシステムにクエリを入力
- (2) ユーザはシステムから検索結果を受け取る
- (3) ユーザはシステムが提示する結果ページの様々な対象に対して各種の編集操作を行い、検索結果を編集する
- (4) システムはユーザの編集操作を検知しユーザの意図を推定する。それに基づき結果をリランキングしユーザに結果を提示する

- (5) ユーザが検索結果に満足していない場合は(3)へ

3.2 操作の種類

● 削除操作

我々は文書を編集している際に、間違った部分や不要な部分に対して削除操作を行う。ウェブ検索結果ページにおける削除操作は、自分の求める結果に明らかに不要なキーワードや、不要な検索結果を削除するといったことが考えられる。そこで、本稿では削除操作を、ユーザが自分にとって不要な情報をシステムに伝える操作として導入する。

● 強調操作

強調操作とは、あるものを周りのものよりも目立たせるために行う操作であり、削除操作と対になる動作であると考えられる。そこで本稿では、強調操作をユーザが自分にとって必要な

情報をシステムに伝える操作として導入する。

3.3 操作の対象とユーザの意図

通常の検索エンジンは図2のような検索結果ページを返す。

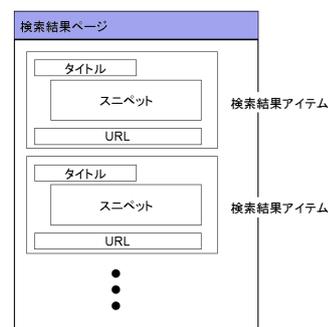


図2 検索結果ページ

● タイトル

ページのタイトルは基本的に短いキーワードで表現されることが多い。ユーザが検索結果のタイトルからあるキーワードを削除するということは、そのキーワードを含む結果を下位にしたいという意図を汲み取ることができる。また、そのキーワードに関連する話題の検索結果も下位にしたいとユーザは考えているかもしれない。例えば「田中克己」というクエリの検索結果において、ユーザがタイトルの「ピアニスト」というキーワードを削除したということは、そのユーザが「ピアニストである田中克己の検索結果はいらぬ」という意図を持っていると推定することができる。

● スニペット

スニペットとは数個の文から構成される、検索クエリに基づきページを要約した文章である。ユーザの操作の対象がスニペットとなった場合は、ユーザはスニペット内のキーワードをだけでなく、スニペットの文章を削除することもあるだろう。ユーザがスニペット内のある文章を削除した場合は、その文章に関連した話題を含む検索結果を下位にすることを望んでいると推定できる。例えば、賛成の意見を述べている検索結果を下位に表示させたい場合は、ユーザは「～に賛成する」といった文章を削除するだろう。

● URL

ユーザがURLやURLの一部を削除した際は、その選択した文字列をURLを含む検索結果を下位にしたいと推定できる。例えば、jpドメインの結果が不要である場合は、ユーザは検索結果のURLから.jpという文字列を削除すると考えられる。

● 検索結果アイテム

検索結果アイテムとは、タイトル、スニペット、URLといったページのメタデータで構成される検索結果を表す要素である。検索結果アイテムを削除するという操作は、ユーザにとって、その検索結果アイテムと似ている結果を下位にしたいという意図を汲み取ることができる。

● 複数検索結果アイテム

ユーザにとって1位の結果が必要で、2位と3位の結果が不要である場合、ユーザは2位から3位までの検索結果アイテム

を削除すると考えられる．その場合，システムは1位の結果に似た結果を上位に表示しつつ，ユーザが削除した結果と似ている結果を下位にリランキングすることでユーザの意図を反映できる．

いずれの場合も，強調操作の場合は必要なものを上位に表示したいという意図を汲み取ることができる．強調操作におけるユーザの意図は，削除操作におけるユーザの意図とちょうど反対となる．

4. リランキング手法

我々の目的は，ユーザの操作が行われたときにユーザの意図を推定し，その意図を検索結果ページに反映することである．その方法として，キーワードの削除と強調による検索結果のリランキングの方法を考案した．リランキングの流れを以下に示す．

- (1) ユーザがシステムにクエリ q を入力．
- (2) システムは検索エンジンに q を与え上位 N 件の検索結果 $R = \{r_1, r_2, \dots, r_N\}$ を取得 (r_i は検索エンジンで i 位の検索結果アイテム)
- (3) 各検索結果アイテム $r_i \in R$ について，次式に従いスコアを割り当て，スコア順にユーザに表示する

$$Score(r_i) = N - i \quad (1)$$

- (4) システムはユーザの操作を検知するたびに以下を行う
 - (a) 用意されたアルゴリズムを用いて検索結果アイテムのスコアを再計算する
 - (b) 各検索結果アイテムをスコア順に整列し，新たな検索結果ページを生成しユーザに提示する．

4.1 選択キーワード法

ユーザがあるキーワード t を削除，強調した際，そのキーワードを含む結果を下位，または上位にする手法である．システムは，タイトルもしくはスニペットにキーワード t を含むような検索結果 $r_i \in R$ について，削除操作が行われた場合は (2) 式，強調操作が行われた場合は (3) 式に従ってスコアを再計算する．

$$Score(r_i)_{new} = Score(r_i)_{last} - N \quad (2)$$

$$Score(r_i)_{new} = Score(r_i)_{last} + N \quad (3)$$

4.2 拡張キーワード法

例えば，ユーザが“田中克己”というクエリで検索し，その検索結果ページから“ピアニスト”というキーワードを削除した場合，システムはユーザの意図を「ピアニストである田中克己」についての検索結果を下位にすることだと推定することができる．従って，ユーザが“ピアニスト”というキーワードを削除した際に，システムは“ピアノ”や“コンサート”といったキーワードを含む検索結果も一緒に下位にする方がユーザの意図に即しているといえる．これを実現する手法として，単語の出現頻度変化を利用した手法を試みた．拡張キーワード法では，“田中克己”というクエリでの検索結果よりも，“田中克己 AND ピアニスト”というクエリでの検索結果の方が“ピアノ”

や“コンサート”といったピアニストに関する単語の出現回数が増えることを利用する．具体的な流れは以下のとおりである．

- (1) あらかじめクエリ q における検索結果ページ p を形態素解析し， p に出現するキーワードの集合 W を求めておく
- (2) 各キーワード $w \in W$ について検索結果ページ p 内の出現回数 $TF(w, p)$ を計算
- (3) ユーザが削除，強調したキーワード t を取得
- (4) システムは検索エンジンに“ q AND t ”というクエリを与え，上位 N 件を取得する (検索結果ページ p' とする)
- (5) $w \in W$ について p' 内の出現回数 $TF(w, p')$ を計算
- (6) $TF(w, p') - TF(w, p)$ が大きい単語を順に k 個 w_1, w_2, \dots, w_k を抽出
- (7) システムは t, w_1, w_2, \dots, w_k の各キーワードそれぞれについて，タイトルまたはスニペットにそのキーワードを含む検索結果アイテム $r_i \in R$ のスコアを，削除の場合は式 (2)，強調の場合は式 (3) に従って再計算する

この手法を用いた場合，例えば“田中克己”というクエリの検索結果ページから“ピアニスト”というキーワードを削除すると，“プロフィール”，“コンサート”，“レッスン”，“ピアノ”というキーワードを含むような検索結果も同時に下位にすることができる (検索エンジンとして Google [14] を使用し $N = 500$ ， $k = 4$ とした場合)

5. 実装

提案手法の有効性を示すため，システムのプロトタイプを実装した．システムは図 3 のようにモジュール分割されており，各モジュールが連携を行って 1 つのシステムを形成している．

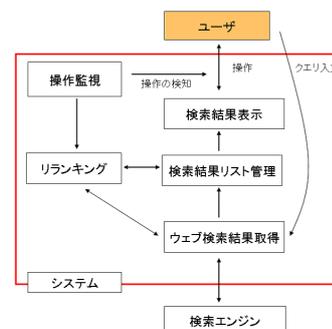


図 3 システムの設計

5.1 モジュール

5.1.1 ウェブ検索結果取得モジュール

ウェブ検索結果取得モジュールでは，ユーザからクエリが与えられるか，リランキングモジュールが検索を行おうとすると，検索エンジンにクエリを送信する．検索エンジンから検索結果ページを取得すると，取得した検索結果を形態素解析し検索結果ページに含まれるキーワードの出現頻度を保存する．

形態素解析には奈良先端科学技術大学院大学自然言語処理講座にて開発された茶筌 [15] を用いた．本研究では，検索結果ページのキーワードとして，検索結果アイテムのタイトルとスニペットに含まれる一般名詞と固有名詞を茶筌を用いて抽出

した。

5.1.2 操作監視モジュール

操作監視モジュールは、ユーザが検索結果ページに対して行う操作を監視するモジュールである。ユーザが削除や強調等の操作を行うと、監視モジュールはユーザの操作を検知し、検索結果ページの編集対象部分を取得し、対象となった文字列などをリランキングモジュールに対して通知する。

5.1.3 リランキングモジュール

リランキングモジュールでは、監視モジュールから受け取った情報を元に、検索結果のリランキングを行う。監視モジュールから通知を受け取ると、検索結果リスト管理モジュールから現在の検索結果の内容とスコアを受け取り、それをリランキングして検索結果リスト管理モジュールにリランキング後の検索結果リストを渡す。リランキングモジュールは必要があればウェブ検索結果取得モジュールから検索結果を受け取り、リランキングに利用する。

5.1.4 検索結果リスト管理モジュール

検索結果リスト管理モジュールは検索結果アイテムの内容とスコアを保持するモジュールである。検索結果リスト管理モジュールは、保持する検索結果がリランキングモジュールによってリランキングされたり、ユーザが新しくクエリを入れ直して検索結果が変更されると、新たな検索結果ページを作成しなおし、検索結果表示モジュールを通してユーザに提示する。

5.2 実行例

実装にはプログラミング言語である C# を用いた。また、検索結果の取得及び形態素解析等のプログラムには C# のライブラリである SlothLib を用いた。

システムの実行例を図 4, 5 に示す。ユーザは検索エンジンの種類や検索条件数を指定して検索を行い、検索結果を受け取る。その後、ユーザは検索結果ページ内のキーワードを選択し、画面上部にある削除ボタンや強調ボタンを押すことによってキーワードの削除・強調を行い、検索結果をリランキングすることができる。



図 4 実行例：キーワード削除前



図 5 実行例：キーワード削除後

6. 実験

6.1 実験内容

本稿で提案する手法の有用性を明らかにするため、4 章で提

案したリランキング手法を用い、キーワードの削除、強調操作を行っていくことでどのように検索結果が変化していくかを検証するため実験を行った。まず、実験にあたり 15 個のクエリと、そのクエリに対する正解セットを用意した。なお検索エンジンには Google を用い、 $N = 500$, $k = 4$ を用いた。

6.1.1 キーワードの削除

ユーザはシステムにクエリを与え検索結果を受け取る。ユーザはシステムから返ってきた結果を上位から順に適合結果であるかどうかチェックしていき、不適合結果を見つけると、その結果の中から 1 つキーワードを削除する。ユーザはシステムがリランキングした結果を受け取り、再び上位から順に結果をチェックしていき、不適合な結果を見つけるとその中から 1 つキーワードを削除するという操作を繰り返す。ユーザは 1 つのクエリについて削除操作を 3 回繰り返し、それぞれの操作終了ごとにリランキングされた検索結果の適合率及び、誤って適合文書を下位にリランキングした検索結果数を調べた。これを選択キーワード法と拡張キーワード法とで比較を行った。

6.1.2 キーワードの強調

ユーザはシステムにクエリを与え検索結果を受け取る。ユーザはシステムから返ってきた結果を上位から順に見ていき、自分の求める結果に関係ありそうなキーワードを強調する。キーワードの削除と同様にユーザは 1 つのクエリについて強調操作を 3 回繰り返し、それぞれの操作終了ごとにリランキングされた検索結果の適合率を調べた。これを選択キーワード法と拡張キーワード法とで比較を行った。

なお、両手法ともリランキングされた結果の上位 20 件が全て適合結果となった場合はそこで操作を終了した。

6.2 結果

図 6 は削除操作によって、図 7 は強調操作によってリランキングした結果の、上位 $K(K = 1, 2, \dots, 20)$ 件までの検索結果の平均適合率の比較結果である。表 1 に削除操作によるリランキングによって上位 20 件中の適合文書を誤って下位にしてしまった文書の数と、総削除数に対する割合を示した。総削除数とは、3 回の削除操作によって上位 20 件中の結果を下位にリランキングした検索結果アイテムの総数である。また、誤り数とはその中で正解セットに含まれていた検索結果アイテムの数の 15 クエリでの総数を表し、誤り率は総削除数に対する誤り数の割合である。

7. 考察

7.1 キーワードの削除

図 6 から分かるように、両手法ともキーワードの削除操作を行うごとに上位 20 件中の適合結果の数が上昇していくのが分かる。また、上位 20 件中の適合率の変化という点で見れば、1 回目の削除操作が最も効果が大きく、2 回目、3 回目と操作を

表 1 適合文書削除率の比較

	総削除数	誤り数	誤り率
選択キーワード法	127	2	1.57%
拡張キーワード法	191	20	10.47%

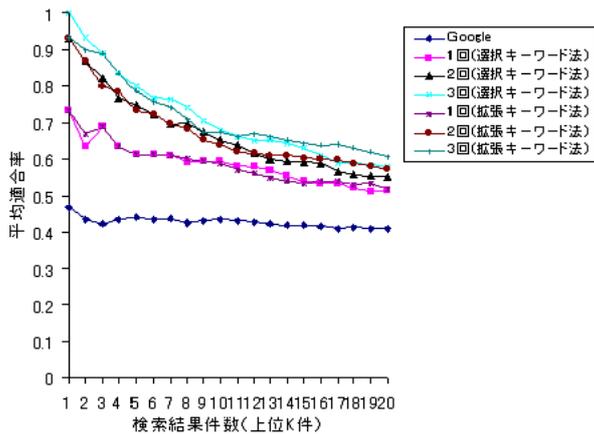


図 6 両手法による適合率の推移の比較結果（削除）

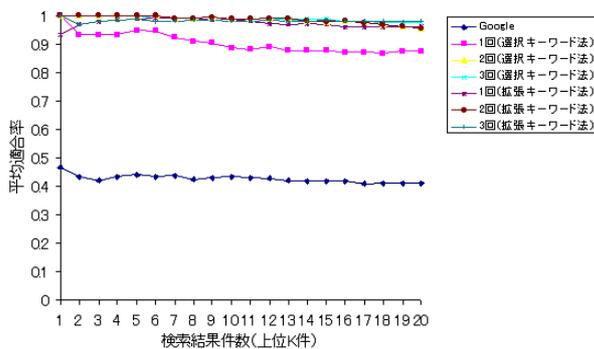


図 7 両手法による適合率の推移の比較結果（強調）

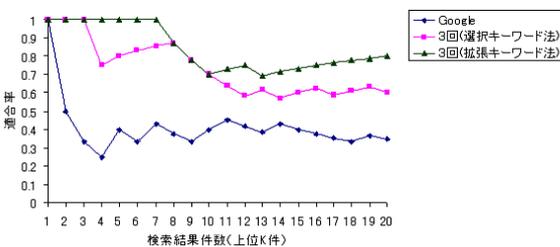


図 8 田中克己（京大教授）

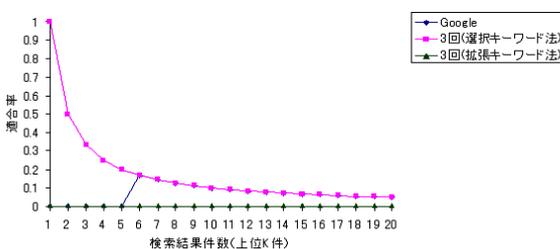


図 9 東西線（京都）

行うごとに適合率上昇の効果は薄れていく。

一方、拡張キーワード法は、選択キーワード法に比べ大きな差はなかった。特に、ユーザは検索結果の上位 10 件程度しか見ないとするならば、両手法の差はほとんどないといえる。両手法の結果に差があまり生じなかった理由として、表 1 から分かる通り、拡張キーワード法ではより多くの結果を下位にでき

るものの、誤って適合結果を下位にしてしまうことも多いことが挙げられる。あるクエリでは拡張キーワード法を用いた方が効果があったが（図 8）、別のクエリでは拡張キーワード法を用いることで適合率が下がってしまうということが起こった（図 9）。例えば、図 9 では拡張キーワード法を使用した場合、上位 20 件に含まれる適合文書が無くなってしまっている。これは、「京都の東西線」を求めて“東西線”というクエリで検索を行い、「札幌」というキーワードを削除した場合、拡張キーワード法では“地下鉄”というキーワードを含む結果も下位にしてしまうことに起因している。京都の東西線は地下鉄であるため、“地下鉄”を含む結果を下位にすると、上位 20 件から「京都の東西線」に関する結果がなくなってしまう。単語の出現頻度を用いた手法の精度を向上させるためには、このような“複数の話題に共通する語”や“様々な検索結果に含まれる一般的な語”を取得してしまう問題に対処する必要がある。

ここで、ユーザがキーワードを削除した検索結果よりも上位にある検索結果はユーザにとって必要な結果であることが多い。従って、拡張キーワード法で取得したキーワードが削除操作を行った検索結果よりも上位の検索結果に含まれる場合は、そのキーワードはランキングに使用しないという手法を用いることで、ランキングの精度も向上し、上記のような問題が発生しにくくなると期待できる。

7.2 キーワードの強調

キーワードの強調は削除操作と比べて非常に効果が高かった。特に 1 回目の操作の効果が大きく、今回の実験では 2/3 以上のクエリが、1 回目の強調操作で上位 20 件全てが適合文書となった。

また、1 回目の操作では拡張キーワード法の方が良い結果となった。キーワードの削除では拡張キーワード法で“地下鉄”のような複数の話題に関係する一般的な語を取得してしまうことが適合率を低下させる原因となった。しかし、強調では欲しい話題とは関係ない一般的な語を取得しても、ユーザが指定したキーワードや、他に取得した語が欲しい話題に関係のある語であれば適合率の低下が起こらないため、拡張キーワード法の方が良い結果となったのではないと思われる。

7.3 削除操作と強調操作

強調操作はその性質上あらかじめ欲しい結果が具体的に言語化できていないと用いることができない操作であると考えられる。なぜなら、ある単語が自分にとって必要なものであるかという判断は、ある単語が自分にとって不要なものであるかという判断より困難であるからである。例えば、「蹴りたい背中」という書籍を読んだ人の感想を探していて、“蹴りたい背中”というクエリで検索した場合、こういった単語を強調すると自分の求める結果が上位にランキングされるのかが非常に分かりづかった。一方、削除操作の場合、“ダウンロード”や“価格”といったキーワードが含まれる検索結果は、感想を記述してあるページであるとは考えにくいいため、こうした不要なキーワードを削除することで正解となるページを上位に集めることができた。今回の実験ではあらかじめ正解セットが比較的用意しやすいクエリを選んだので、強調すべきキーワードも比較的見つけやす

かったが、実際の検索では強調すべきキーワードが見つからない可能性は十分に考えられる。実際の利用ではユーザは削除操作と強調操作を組み合わせながら、徐々に検索結果を最適化していくことになるだろう。

キーワードには多くの検索結果に含まれるようなキーワードと、他の検索結果にほとんど出現しないキーワードがある。多くの検索結果に含まれるキーワードを削除した場合は、検索結果の多くがリランキングされるが、ほとんど出現しないキーワードを削除した場合は、下位にリランキングされる結果は少ないため検索結果ページはあまり変動しない。また、キーワードの強調についても同様の問題があり、様々な検索結果に含まれるキーワードを強調してもあまり検索結果ページは変動しない。一方、あまり他の検索結果に含まれないような単語を指定すると普段は見えないような下位の検索結果を上位に表示してやることができる。ここで、下位にしか存在しないキーワードに対して操作を適用するには、実際に結果を順に見ていき、そのキーワードをユーザが発見する必要があり、削除や強調操作によるリランキングの利点が損なわれてしまう。従って、キーワードに対して削除や強調の操作を行った際に、どのように検索結果が変わるのかをあらかじめユーザに提示してやることができれば、ユーザは削除操作や強調操作を効率的に行うことができるようになると思われる。

8. タグクラウドを用いたキーワード決定支援

タグクラウドとは様々なタグをひとかたまりの領域に並べて表示し、利用頻度が高いものや、人気があるタグほど大きく描画するという表示方法である。タグクラウドを用いたものとして代表的なものには flickr [16] があり、flickr ではユーザが画像につけた多種多様なタグを、タグクラウドを用いて一覧表示している。タグクラウドの利点としては、頻出単語が大きく表示されて一目で確認できるうえ、全体が見渡せるという点が挙げられる。

そこで我々は、このタグクラウドというテキストベースの表現手法をウェブ検索結果ページにも適用し、ユーザが削除や強調するキーワードの決定の支援を行う。

8.1 タグクラウドの生成方法

一般的なタグクラウドでは、表示されるのはユーザが付加したタグの集合であるが、ここではウェブ検索結果ページに出現するキーワードの集合をタグクラウドとしてウェブ検索結果ページに表示する。また、どのように検索結果の内容が分布しているのかということを表示するために、検索結果を上位から順に複数のグループに分割して、それぞれについてタグクラウドを表示するという手法をとった。タグクラウド生成の流れは下記の通りである。

(1) 検索結果ページ p 内のスコア順に整列された検索結果アイテムの集合 $R = \{r_1, r_2, \dots, r_N\}$ を上位から 5 分割する (5 分割された検索結果アイテムの集合を $R' = \{R_1, R_2, R_3, R_4, R_5\}$ とする)

(2) $R_i \in R'$ について以下を行いタグクラウドを生成する

(a) R_i 内の検索結果アイテムのタイトルとスニペットを

形態素解析し、キーワードの集合 T_i とキーワード $t \in T_i$ の出現回数を計算する

(b) 出現回数の多いキーワード上位 k 個を求め、そのキーワードを R_i で初めて登場する順番にタグクラウドに追加する

(c) キーワードの出現回数があらかじめ用意した閾値 $\theta_i (i = 1, 2, 3, \dots)$ を超えるごとにタグクラウドで表示するキーワードの大きさを変更する

8.2 実行例

実際のシステムの実行例を図 10 に示す。システムの左側には通常の検索結果のリストを表示し、検索結果ページの右側にはタグクラウドを表示している。図では 500 件の検索結果を取得して表示しており、その隣に 1 位 ~ 100 位, 101 位 ~ 200 位, ..., 401 位 ~ 500 位のタグクラウドを表示している。各タグクラウドではキーワードの出現回数が上位 30 件のものを 4 種類の大きさで表示している。ユーザはタグクラウドで表示されているキーワードも編集操作の対象として利用することができるため、タグクラウド上のキーワードを削除、強調することで検索結果のリランキングを行うことができる。システムはユーザの操作に応じて、4 章のリランキング手法を用いて検索結果をリランキングし、リランキング結果及び新しいタグクラウドを生成してユーザに提示する (図 11)。



図 10 タグクラウドの実装例



図 11 キーワードの強調後のタグクラウドと検索結果

8.3 利用で得られた知見

タグクラウドを表示することによって、検索結果ページにどのようなキーワードが含まれているか俯瞰することができ、検索結果にどのような話題が含まれているのかを素早く見渡すことができるようになった。また、大きく表示されているキーワードは、そのキーワードを削除したときに多くの結果が下位になること、小さく表示されているキーワードは、強調すれば検索結果ページにまばらに含まれるページをまとめて上位に表示できることが視覚的に分かるようになっており、削除や強調を行うことによってどのように検索結果が変化するのが分かりやすく表示できている。

タグクラウドを使用することによって、ユーザは検索結果の中から自分に必要なキーワード、不必要なキーワードを探し出すことが、従来のタイトルやスニペットからキーワードを選択するよりも容易となっていることが多かった。

現在は検索結果を5分割してタグクラウドを表示しているが、検索結果の分割方法やキーワードの表示順序、キーワードの色づけなど、タグクラウドのユーザへの提示手法については今後研究していく予定である。

9. まとめと今後の課題

本稿では、ウェブページの検索を行う際に、ユーザに様々な操作を許可することでユーザの検索意図を推定し、結果に反映する手法を提案した。また、実際にユーザがキーワードの削除、強調操作を行った際のリランキング手法を提案し、評価を通してその有効性を検証し、削除操作と強調操作には利用のしやすさとリランキングの効果に差があることを示した。さらに、ユーザが効率的に削除・強調操作を行うためにタグクラウドを用いた手法を実装した。

今後はキーワードの削除や強調操作だけでなく、ページや文章に対する削除や強調操作や、ドラッグアンドドロップ等の操作を用いた意図推定や反映についても取り組んでいきたい。また、ユーザの検索意図には結果をリランキングして欲しいという意図以外にも様々なものが考えられる。一般に検索結果のスニペットはそのページを上手く要約したものであるとは限らないので、もっとそのページの別の箇所のスニペットを見たいということがあるだろう。そういった場合に、スニペットを削除することによってその検索結果のスニペットを別の箇所のスニペットに修正するといったことが考えられる。また、あるものを様々な側面で比較したいということがあるが、そういった比較検索にもユーザの編集操作を導入し、検索結果を編集することで簡単に比較を行うことが出来るシステムも実装する予定である。

また、今回我々が提示した手法はウェブ検索結果閲覧だけでなく、他のウェブブラウジング時においても適用できると考えられる。例えば掲示板において、特定の名前の書き込みや、スパムメッセージだけを削除することにより可読性を向上させたり、特定の話題に関係する書き込みのみ表示したりすることなどが考えられる。このような、ウェブページをユーザの好きなように修正できるシステムについても取り組んで行く予定で

ある。

謝辞 本研究の一部は、文部科学省 21 世紀 COE 拠点形成プログラム「知識社会基盤構築のための情報学拠点形成」(リーダー：田中克己，平成 14～18 年度)，文部科学省科学研究費補助金特定領域研究「情報爆発時代に向けた新しい IT 基盤技術の研究」，計画研究「情報爆発時代に対応するコンテンツ融合と操作環境融合に関する研究」(研究代表者：田中克己，A01-00-02，課題番号 18049041)，文部科学省科学研究費補助金特定領域研究「情報爆発時代に向けた新しい IT 基盤技術の研究」，計画研究「情報爆発に対応する新 IT 基盤研究支援プラットフォームの構築」(研究代表者：安達淳，Y00-01，課題番号：18049073)，および、文部科学省科学研究費補助金若手研究 (B)No.180700129 によるものです。ここに記して謝意を表すものとします。

文 献

- [1] T. Nanno, S. Saito and M. Okumura: “Zero-Click: a system to support Web browsing”, The 11th International World Wide Web Conference (2002).
- [2] T. Araki, H. Miyamori, M. Minakuchi, A. Kato, Z. Stejic, Y. Ogawa and K. Tanaka: “Zooming Cross-Media: A Zooming Description Language Coding LOD Control and Media Transition”, Proc. of the 16th International Conference on Database and Expert Systems Applications(DEXA2005), LNCS3588, pp. 260–269 (2005).
- [3] vi search.
<http://www.visearch.com/>.
- [4] A. Broder: “A taxonomy of web search”, ACM SIGIR Forum, **36**, 2, pp. 3–10 (2002).
- [5] D. Rose and D. Levinson: “Understanding user goals in web search”, Proceedings of the 13th international conference on World Wide Web, pp. 13–19 (2004).
- [6] I. Kang and G. Kim: “Query type classification for web document retrieval”, Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp. 64–71 (2006).
- [7] U. Lee, Z. Liu and J. Cho: “Automatic identification of user goals in Web search”, Proceedings of the 14th international conference on World Wide Web, pp. 391–400 (2005).
- [8] 田馳, 手塚太郎, 小山聡, 田島敬史, 田中克己: “質問キーワードの意味的関連と近接性に着目したウェブ検索の精度改善”, 電子情報通信学会第 17 回データ工学ワークショップ (DEWS2006) (2006).
- [9] Yahoo! Mindset.
<http://mindset.research.yahoo.com/>.
- [10] 121r(One to One Ranking System)
<http://www.kbmj.com/service/products/121r.html>.
- [11] G. Salton: “The SMART Retrieval System Experiments in Automatic Document Processing”, pp. 312–323 (1971).
- [12] G. Salton and M. McGill: “Introduction to Modern Information Retrieval”, McGraw-Hill, Inc. New York, NY, USA (1986).
- [13] T. Onoda, H. Murata and S. Yamada: “Non-Relevance Feedback Document Retrieval Based on One Class SVM and SVDD”, 2006 IEEE World Congress on Computational Intelligence, pp. 2191–2198 (2006).
- [14] Google.
<http://www.google.com>.
- [15] 形態素解析システム茶釜.
<http://chasen.naist.jp/hiki/ChaSen/>.
- [16] flickr.
<http://www.flickr.com/>.