

Wikipediaからの拡張クエリ生成によるWeb検索とその評価

Web Retrieval with Extended Queries Generated from Wikipedia and Its Evaluation

堀 憲太郎^{1*} 大石 哲也¹ 峯 恒憲² 長谷川 隆三² 藤田 博² 越村 三幸²
Kentaro Hori¹, Tetsuya Oishi¹, Tsunenori Mine², Ryuzo Hasegawa², Hiroshi Fujita², Miyuki Koshimura²

¹ 九州大学大学院システム情報科学府

¹ Graduate School of Information Science and Electrical Engineering, Kyushu University.

² 九州大学大学院システム情報科学研究院

² Faculty of Information Science and Electrical Engineering, Kyushu University.

Abstract: This paper proposes a web retrieval system with extended queries generated from the contents of Wikipedia. By using the extended queries, we aim to support user's retrieval and knowledge acquisition. To extract extended query items, we make much of hyperlinks in Wikipedia in addition to the related word extraction algorithm. We evaluated the system through experimental use of it by several examinees and the questionnaires to them. Experimental results show that our system works well for user's retrieval and knowledge acquisition.

1 はじめに

近年、インターネットの進歩により、一般家庭からでも容易に Web(World Wide Web) にアクセスすることができる環境になっている。また、目的のページを見つけるための手段として検索エンジンを利用することが普及した。

大手検索エンジンに google[2], yahoo[3], goo[4] といったものがある。これらはユーザが自分の興味ある事柄について、単一、或いは複数のキーワードを入力するだけで、膨大なデータベースから最適なページを取捨選択してくれるものである。例えば、google はページの評価に各ページのリンクに基づいた PageRank アルゴリズム [5] を用いて、ユーザの必要とするページを検索上位に提示することに成功している。

しかし、Web 上に存在するデータ量は莫大で、かつ常に増大しているため、その中からユーザの意図に沿ったページを、短時間で見つけ出せないことがしばしばある。

検索結果を絞り込む手段として、複数のキーワードを与える AND 検索がある。ユーザの目的に適したキーワードを追加すると、単一のキーワードの時に比べて検索結果を絞り込むことが可能である。しかし、ユーザは検索する際に必ずしも適切なキーワードを思いつ

くとは限らない。キーワードを追加しても、満足な結果が返ってこないこともある。何故なら、サーチエンジンは一般に入力したキーワードを含まない文書を検索しないからである。また、入力したキーワードに関する新たな知識を獲得したいとユーザが考える時もある。彼らは入力したキーワードに対してある程度の知識を持っているが、そのキーワードに対して知らない知識もある。このためユーザの知らない知識で絞込検索できないのは、新たな知識を得る際には非常に不便である。

そこで、ユーザの入力したキーワードに関連した単語(関連語)を提案するシステムがいくつか研究されている [1][9][13]。この関連語を提案するためには、ユーザが入力したキーワードに加えてさらに他の情報源が必要となる。その情報源としてユーザからのキーワードを一旦検索エンジンに入力し、そこから得られた検索結果を用いる手法がある。例えば擬似フィードバックと呼ばれる手法は、検索結果の上位 10 件をユーザの意図に沿った文書(以降適合文書と呼ぶ)、それ以下を不適合文書とし、それらの文書集合から関連語を抽出する。この手法の利点は、ユーザに負担を与えない点、そして入力したキーワードに対して何かしらの検索結果が得られる点、検索意図に縛られないという点である。しかし昨今、blog や掲示板、オンラインショップが発達しており、これらは検索意図とは関係のない情報を多量に含んでいる。検索エンジンから得られた検索結果の上位にこれらが表示されると、機械的な情

*連絡先: 九州大学大学院 システム情報科学府 知能システム学専攻

〒 819-0395 福岡県福岡市西区元岡 744
E-mail: hori@ar.is.kyushu-u.ac.jp

報収集を考えた場合に、目的とする情報と同時に有用でない情報も引き出だしてしまう。

そこで、本論文では関連単語抽出の情報源として Web 上の百科事典、具体的には Wikipedia[6] を使った関連単語提案システムを提示する。Wikipedia を使うことにより、通常の検索結果では表示されてしまう不必要な情報を排除し、より未知のものに対して Web を検索した際の検索精度を向上させる。しかし、Wikipedia は Web ページ全ての情報量と比べると非常に少なく、ユーザが入力するクエリも多岐に渡るため、ユーザがクエリに使ったキーワードが辞典の中に含まれていない可能性がある。そこで、ユーザが入力するクエリを数種類に分類し、特に Web を辞書のように使用する目的でクエリを入力した際に有用となるようなシステムを構築した。ここで有用とは、システムを用いることで検索結果の精度向上となることと、システムから新たな興味と知識を獲得できることの 2 点を意味する。

2 関連研究

疑似フィードバックの他には、ユーザが直接文書に点数をつける明示的フィードバック、スクロールおよびページ拡大、ページのクリックなどの操作に関して、ユーザがいずれかの操作を行ったかどうかを調査して、ユーザの意図にあった文書を収集する暗黙的フィードバックといったものもある。

他の情報源として、ユーザ個人の特性をあらわす情報（個性情報と呼ぶ）を用いるシステムもある。個性情報は一般にユーザプロフィールと呼ばれ、例えばスケジュールであったり趣味趣向をあらかじめデータベース化しておいたり、と様々なものがある [7]。これらを用いることで、ユーザの意図にあった関連単語を提案することができる。例えばユーザの住んでいる付近の天気予報を調べるときなど、このシステムがユーザの住んでいる地域に限定してくれる。個性情報に沿った検索をしている限りは、Web という様々な情報が混濁している場から関連した単語を見出すより、ユーザプロフィールから関連語を導くほうがより有効な手段であると言える。しかし、このシステムはユーザが新しいことについて知識を得ることが困難である。知らない外国の文化や歴史、見たことがない単語について検索するときには個性情報を参照しても、そこに有用な情報は無いからである。

3 関連単語提案システムの概要

全体の流れは、図 1 のように行う。

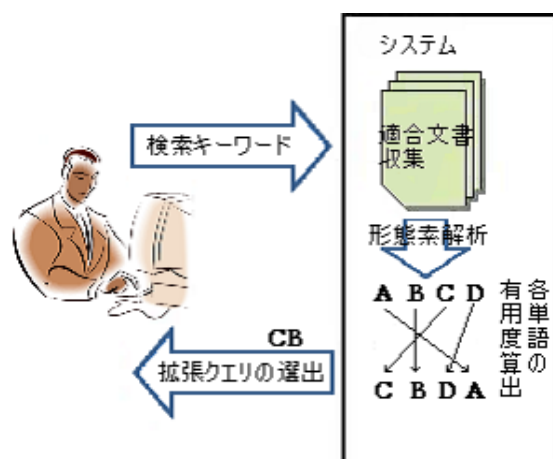


図 1: システムのイメージ

はじめに、ユーザに元となる検索キーワード（以降元クエリと呼ぶ）を入力してもらう。この元クエリに関連した単語を返すことがシステムの目的である。

次に、関連語を見つけ出すための情報源として、元クエリに関連した文書（以降適合文書と呼ぶ）の収集を行う。本研究では、Wikipedia 内の、元クエリに関連する記事から段落を抜き出し、それを適合文書として用いる。詳しくは第 4 章で述べる。

収集した適合文書を、高速形態素解析システム「MeCab」[8] を利用して形態素解析を行う。このとき MeCab を通すことで現れる単語を精練する。まず名詞のみに絞り、その上で不必要な単語、例えば「こと」「もの」といった検索に適していない単語は予めリストアップしておき、削除する。また、「卒業論文」といった連結語は「卒業」「論文」と分かれてしまうので、名詞が連続して出現した場合は連結語と判断する、というルールを作り、本システムで「卒業論文」と戻す。

以上のような形態素解析の後、精練した単語について次は関連語としての有用度を算出する。この計算手法についても様々な研究が行われている。例えば、RSV(Robertson Selection Value)[9] と呼ばれるアルゴリズムがある。RSV は適合文書か不適合文書のどちらかに偏って出現している単語を重要とするものである。しかし、提案手法は Wikipedia から適合文章を抜き出すという構造上、妥当な不適合文章の設定が難しい。そこで今回使用したのは我々の研究室で提唱している関連単語抽出アルゴリズム [1] である。これにより、どの単語が関連語として有用か、単語間の距離を基に算出する。更に、Wikipedia の内部リンクを利用した各単語の補正値を計算する。詳しくは第 5 章で述べる。

以上より、Web を辞書として検索する際に有用となる、関連語のリストを提示するシステムを構築する。

4 提案手法

4.1 適合文書

このシステムでは、適合文書に Wikipedia[6] を使用する。その理由については主に3点が挙げられる。

- Wikipedia が Web 上で最大の百科事典であり、日本語だと現在約 55 万語載っている。(2009 年 1 月現在)
- 充実した内容が書かれている [11][12]。
- 誰でも更新ができるというその特性から、多数のユーザが頻繁に更新し常に最新の内容が維持される。

4.2 抽出方法

適合文章の収集方法は、元クエリを構成するキーワード数によって異なる。

元クエリが1つのキーワードからなる場合

元クエリは A であったとする。Wikipedia 内に A についての記事があるかどうか検索を行う。検索結果の上位 1 ページを抽出し、そのページ内で A が出現している段落を検索する。抽出された段落を適合文書として用いる。

元クエリが2つのキーワードからなる場合

元クエリは「A B」であったとする。Wikipedia 内を「A」「B」「A B」の3つのクエリでそれぞれ検索を行い、各上位 1 ページを収集する。収集したそれぞれを「記事 A」「記事 B」「記事 AB」とし、記事 A は B の出現している段落を、記事 B は A の出現している段落を、記事 AB は A と B が同時に出現している段落を抽出する。

例えば、「卵焼き 味付け」というクエリで検索したときに、まず Wikipedia で「卵焼き」を検索する。「卵焼き」のページがヒットするので、次は「味付け」という単語で「卵焼き」ページ内を探索する。見つかったら「味付け」がある段落を適合文章として抽出する。次に「味付け」で検索し、「味付け海苔」というページがヒットする。ところが、単語「卵焼き」が存在する段落がないので、抽出は行わない。最後に、「卵焼き 味付け」で検索を行い、「オムレツ」のページがヒットする。このページ内で「卵焼き」と「味付け」の両方の単語が存在する段落を抽出する。以上により抽出された段落を適合文書として、形態素解析に移る。図 2 は、卵

焼きページ内から味付けという単語のある文章を探し出し、その段落を抜き出す様子である。

卵焼き

出典: フリー百科事典『ウィキペディア (Wikipedia)』

卵焼き、玉子焼き(たまごやき)は、卵を溶きほぐしたものをいれ、こけしなどで焼く。ここでは溶いて焼く狭義の「卵焼き」について記述する。ヨウなどの卵を使用する場合もある。

黄身、白身を共に混ぜ(黄身は割りほぐして白身とよく混ぜ合っ素加工したフライパンでは油をひく必要はない)。家庭でよく延ばし、表面だけ軽く焼いた状態で巻いていったものである。

味付けには、多くの場合、塩、胡椒を加えるが、寿司や懷石料理地方では、おかずとして食べる場合にも砂糖を加える家庭がある。このほかめんつゆやナンプラーを混ぜることもある。卵に加えて焼く、ウナギを中に入れて巻く(う巻き)などのバリエーションがある。

なお、細片状になる「炒り卵」については普通「卵焼き」というレツの相違点は、焦げ目の有無のほか、オムレツには牛乳やシなどを混ぜることがあるのが卵焼きとの大きな違いである。

図 2: 実際の抽出イメージ

5 単語の関連度計算

5.1 関連単語抽出アルゴリズムの概要

このアルゴリズムは、あるキーワード群 K とそれに関するテキスト T に現れる単語から、 K に関連すると思われる単語を抽出、出力するものである。単語の関連度の算出には、単語間の距離に着目し、それを元に評価を行う。

このアルゴリズムの根幹となっている考え方が、単語間の距離である。具体的には、文章中に出現する単語の順番に注目し、単語 A について A の付近に出現している単語ほど A に関連性があるという考え方である。

5.1.1 定義

アルゴリズムの説明の前に、以下の記号を定義する。

- K … 関連単語を抽出するための基となるキーワード群
- T … キーワード群 K に関するテキスト
- $k_i (i = 1 \dots m)$ … キーワード群 K で出現する m 個の単語 (出現順に $k_1, k_2, k_3, \dots, k_m$)
- $s_h (h = 1 \dots n)$ … テキスト T で出現する n 個の文章 (出現順に $s_1, s_2, s_3, \dots, s_n$)

- $F_{k_i}(s_g)(g = 1 \dots n) \dots s_g$ がキーワード k_i を含むとき $F_{k_i}(s_g) = 1$, 含まないとき $F_{k_i}(s_g) = 0$
- $t_j(j = 1 \dots o) \dots$ テキスト T で出現する o 個の単語 (出現順に $t_1, t_2, t_3, \dots, t_o$)

但し, t_j はテキスト T を MeCab で形態素解析し, その結果得られた名詞のみ抽出し, それらを出現順に並べたものである. また, 形態素解析した結果, 同一単語が複数出現した場合もそれらは別のものとみなす.

5.2 テキスト内の単語の評価

テキスト T 内で出現する単語の評価は以下のように行う.

1. $k_i(i = 1 \dots m)$ を基準に $s_h(j = 1 \dots n)$ の基礎評価値 (BasicValue) $BV_{k_i}(s_h)$ を計算
2. 文章の単語への分解, 平滑化
3. 単語の出現頻度による最終評価値 $V(t_j)$ を計算

5.2.1 $BV(s_h)$ の算出

本アルゴリズムはキーワード群 K を基に, K とテキスト T で出現する単語間の距離を中心に単語の評価を行うことを目的としており, はじめにその基本となる評価値を求める.

文章 s_h のキーワード k_i に関する評価値 $BV_{k_i}(s_h)$ を次のように定める.

$$BV_{k_i}(s_h) = \sum_{g=1}^n (n - |g - h|) F_{k_i}(s_g) \quad (1)$$

ここで, $|g - h|$ は k_i が出現する文章と s_h との距離である. すなわち, 文章 s_q に k_p が出現するときの s_q の $BV_{k_p}(s_q)$ は, $|q - q| = 0$ で, $F_{k_p}(s_q) = 1$ なので, $BV_{k_p}(s_q) = n$ となる. また, その一つ隣の文章 s_{q+1} と s_{q-1} は $|q - (q + 1)| = 1, |q - (q - 1)| = 1$ で, $BV_{k_p}(s_{q+1}) = BV_{k_p}(s_{q-1}) = n - 1$ となる. k_i が出現する文章と s_h の距離が近いほど $BV_{k_i}(s_h)$ は大きな値をとることに注意されたい.

以上の計算をすべての $k_i(i = 1 \dots m)$ で行い, それぞれの s_h において $BV_{k_i}(s_h)$ の和を $BV(s_h)$ とする. すなわち $BV(s_h)$ は次式により求める.

$$BV(s_h) = \sum_{i=1}^m BV_{k_i}(s_h) \quad (2)$$

表1に $BV(s_h)$ を求める例を示す. テキスト T にはキーワード A が2度出現している. 例より, キーワード群 K に含まれている単語を含む文章中心に, それに近い文章ほど高評価を得ていることがわかる.

表 1: 距離による文章の評価例

キーワード K	A	B			
テキスト T	AFB	ED	AFC	FE	DE
$BV_A(s_h)$	8	8	8	6	4
$BV_B(s_h)$	5	4	3	2	1
$BV(s_h)$	13	12	11	8	5

5.2.2 $BV(s_h)$ の平滑化

前節で求められた $BV(s_h)$ は, 文章 s_h のテキスト T で出現する位置を考えると不公平である. テキスト T の端に出現する文章 $s_h(h = 1, n)$ については $1 \leq BV_{k_i}(s_h) \leq n$ であるのに対し, テキスト T の中央に出現する文章 $s_{n/2}$ については $n/2 \leq BV_{k_i}(s_{n/2}) \leq n$ である. よって $BV(s_h)$ のとり得る値の期待値は文章 s_h の出現位置 h によって異なり, このままでは T の中央に出現する文章は必然的に与えられる評価値が大きくなってしまい, 公平な評価はできない.

この問題を解決するために, 求めた $BV(s_h)$ を文章 s_h の出現位置 h での評価値の期待値を用いて平滑化を行う. 文章 s_h の出現位置 h での評価値の期待値 $EBV(h)$ は以下の式で求められる.

$$EBV(h) = \frac{1}{2n} n(n + 2h - 1) - 2h(h - 1) \quad (3)$$

ここで, n はテキスト T 内で出現する文章ののべ総数である.

本アルゴリズムでは, 平滑化後の評価値を $EBV(s_h)$ とし, 以下の式で計算する.

$$EBV(s_h) = \frac{BV(s_h)}{EBV(h)} \quad (4)$$

表2に $EBV(h)$ を計算し, $EBV(s_h)$ を求めるまでの例を示す. $EBV(h)$ は出現位置 h での評価値の期待値であるので, 中心付近 ($h = n/2$ 付近) の値が大きくなっている. 中心をピークとして左右対称になっている. このように $BV(s_h)$ では出現位置により不公平な評価を得ていたものが平滑化されていることがわかる.

表 2: 出現位置 h での期待値 $EBV(h)$ とそれを用いた評価値 $EBV(s_h)$ の例

キーワード K	A	B			
テキスト T	AFB	ED	AFC	FE	DE
$BV(s_h)$	13	12	11	8	5
$EBV(h)$	3	3.6	3.8	3.6	3
$EBV(s_h)$	4.33	3.33	2.89	2.22	1.67

5.2.3 $V_T(t_j)$ の算出

本アルゴリズムは単語間の距離を最重要視して評価値計算を行っているが、それに加えて TF/IDF 法 [10] などを用いられるテキスト内での単語の出現頻度 TF(TermFrequency) の概念も考慮する。つまり、テキスト T 内で複数回出現した単語はある程度重要視すべきである、ということである。

まず、文章から単語に分解する。そのとき、テキスト T 内で複数回出現した単語についてはその単語の $EBV(t_j)$ の平均値 $AveEBV(t_j)$ を計算する。例えば文章 s_a と文章 $s_b(a \ b)$ の 2 つに単語 t_c が 1 回ずつ現れたとすると、 $AveEBV(t_c) = (EBV(s_a) + EBV(s_b))/2$ となる。無論、 T 内で出現が 1 回のみ単語 t_j (t_j は s_h 内の単語とする) に関しては $AveEBV(t_j) = EBV(s_h)$ である。

さらに単語 t_j の tf 値 (T 内での出現回数) を用いて以下の式で表される重み $W_T(t_j)$ を計算する。

$$W_T(t_j) = 1 + \frac{tf(t_j)}{n} \log tf(t_j) \quad (5)$$

ここで、 $tf(t_j)$ はテキスト T 内での単語 t_j の出現回数である。

そして得られた $AveEBV(t_j)$ と $W_T(t_j)$ を用いて以下の式でテキスト T で出現する単語 t_j の評価値 $V_T(t_j)$ を計算する。

$$V_T(t_j) = AveEBV(t_j) * W_T(t_j) \quad (6)$$

こうして求められた $V_T(t_j)$ を、本アルゴリズムでのテキスト T 内で出現する単語 t_j の評価値とする。表 3 に $V_T(t_j)$ 算出までの例を示す。テキスト T 内では単語 C が 2 回出現しているので $tf(C)$ は 2 となっており、その他の単語については tf 値は 1 となっている。重み $W_T(t_j)$ は tf 値を基に計算しているため、 tf 値が 1 の単語については 1、2 回出現している単語 C について 1 より大きな値をとっている。さらに、単語 C に関しては $EBV(C)$ の平均も計算する。そして最終的な単語の評価値 $V_T(t_j)$ が求められ、本アルゴリズムでは、テキスト T 内の単語は、評価値の大きい F, A, B, E, D 、 C の順でキーワード群 K へ関連度が高いとみなす。

表 3: $V_T(t_j)$ 算出までの各値の例

キーワード K	A	B				
テキスト T	AFB	ED	AFC	FE	DE	
$EBV(s_h)$	4.33	1.67	2.11	2.22	2.67	
単語	A	B	C	D	E	F
$AveEBV(t_j)$	3.61	4.33	2.11	1.67	2.50	2.56
$tf(t_j)$	2	1	1	2	3	3
$W_T(t_j)$	1.28	1	1	1.28	1.66	1.66
$V_T(t_j)$	4.62	4.33	2.11	3.20	4.00	5.23

5.3 Wikipedia の内部リンクによる補正

5.3.1 概要

更に、Wikipedia の記事内にある、他記事へのハイパーリンク (内部リンク) に着目する。この内部リンクは、記事の作者が任意に設定できるものであり、内部リンクが設定されている単語は、記事に関連が深く、着目してほしい重要語であると考えられる。ただし、内部リンクは任意に設定できる点から、元クエリには関連性のない単語である可能性もある。そこで、元クエリとの関連度を計算する。

5.3.2 計算方法

- $LW_f (f = 1 \dots p)$ … 内部リンクが指定されている単語
- $RWS(LW_f)$ … 関連単語抽出アルゴリズムで算出された LW_f の評価値
- $LW_{f(k_i)}$ … キーワード k_i が、 LW_f の内部リンクが指す記事内に出現する回数

以上の 3 つから、最終的な評価値を計算する。式は以下の通り。

- $WikiEX(LW_f) = \log(\sum_{i=1}^m LW_{f(k_i)}) + 1$
- $WordScore(LW_f) = RWS(LW_f) * WikiEX(LW_f)$

実験を行ったところ、 $LW_{f(k_i)}$ の値が 0 から 100 以上までに渡ったため、極端な評価値差をなくすよう、対数を取った。この $WordScore(LW_f)$ を、最終的な評価値とする。なお、内部リンクのない単語については、関連単語抽出アルゴリズムの評価値を最終的な評価値とする。

6 実験

6.1 クエリのカテゴリ

本実験では、ユーザに入力してもらったクエリに対し、Andrei らの調査 [14] を基に分類を行い、特定のクエリにのみ本手法を適用する。分類の仕方は、

1. ナビゲーションクエリ
2. トランザクションクエリ
3. インフォメーションクエリ

の3つである。以下それぞれについて説明する。

ナビゲーションクエリとは、目的のページがただ一つだけと分かっているクエリである。例えば、ユーザが Google のページへ行きたいと思っているときは、「Google」という検索クエリを入力する。これらのクエリに対してはクエリの拡張を行う必要性がない。なぜなら、既存の検索エンジンにこれらのクエリを入れると、十分な結果が返ってくるからである。

トランザクショナルクエリとは、ユーザが何か行動を起こすクエリのことである。例えば買い物や、ファイルのダウンロード、地図を探すといったものが該当する。これらに対しては、2章で取り上げたような、ユーザプロファイルからのクエリ作成が有効である。

インフォメーションクエリとは、ユーザが知識を得ようと検索するときのクエリである。ユーザ自身がわからない単語や物事があったとき、それに近い単語をクエリとして入れて、その検索結果から新たな知識を獲得するといった行動に使われる。これらのクエリに対して有効に働くことを想定しているのが本システムである。よって、本実験はインフォメーションクエリに対して行う。

6.2 評価方法

本システムの目的は、WWW を辞書のように利用する時、すなわちクエリ自体の意味、または関連語について調べる時に、ユーザの有用な手助けとなるように働くことを目的としている。よって、被験者にアンケートを取り、実際に役に立つと感じるかどうかを調査する。具体的には、検索結果の精度向上と、新たな知識獲得という2つの観点から、生成された関連語のリストを閲覧し評価してもらう。システムの比較対象として、[goo],[Web5],[RWS],[EXRWS]の4つを用意した。[goo]は、NTT レゾナントが運営する検索エンジンである。

[Web5]は、元クエリによる検索結果上位5件を適合文書として取得し、形態素解析の結果を関連単語抽出アルゴリズムで評価値計算したものである。

[RWS]は、元クエリを用いて Wikipedia から適合文書を取得し、形態素解析の結果を関連単語抽出アルゴリズムのみで評価値計算したものである。

[EXRWS]は、本研究の提案手法であり、[RWS]に Wikipedia の内部リンクを考慮した補正を追加したものである。

[goo]は、通常の実験エンジンと同様に、元クエリの検索結果を上位10件提示する。[Web5][RWS][EXRWS]は、元クエリを利用して算出された上位10件の関連語をリストにしてユーザに提示する。ユーザはアンケートの質問に沿っていると判断した関連語を自ら選択し、

「元クエリ+関連語」を新たなクエリとして goo に投げ、その検索結果を評価してもらう。事前にクエリを20個用意し、被験者5人に、各人が未知であると判断したクエリを5つ選択してもらい、計25個のクエリについてアンケートを取った。クエリのリストは表7の通りである。

6.3 結果と考察

アンケートの質問は2種類で、それぞれ5段階評価で行った。

- 質問1：元クエリの検索結果をより良いものにすると思われる関連語を選択し、評価してください。

- 比較対象 [goo][Web5][RWS][EXRWS]

1. 不満:絞り込めるような関連語がなかった
2. やや不満:欲しい情報が得られなかった
3. 普通:わずかだが情報は得られた
4. やや満足:各ページの情報は断片的だが、総合すれば得られた
5. 満足:どこを見ても文句なく情報が得られた

ここで [goo] は、元クエリでどの程度よい結果が出ているかという指標であり、関連語の選択はない。結果は以下ようになった。

表 4: アンケート1結果

手法	評価値平均	分散
goo	4.48	0.33
Web5	3.92	1.28
RWS	3.8	1.52
EXRWS	3.96	1.16

結果は表4のようになった。実験手法の中では [EXRWS] の評価値の平均が最も高いが、関連語を追加しない元クエリのままの [goo] が一番良いという結果となった。元々よい結果が出ているクエリに余分な単語を追加し、結果が悪くなる例となってしまった。原因としては、検索に慣れている実験者が、容易に目的が想像できるクエリを用意したため、goo にも良い結果が表示されてしまったことが挙げられる。そこで、被験者3人に、それぞれ goo では良い結果が出なかった3つのクエリを渡し、追加実験を行った。表5から、[goo] より [EXRWS] が上回る結果となった。今後の実験では、非常に曖昧なクエリを被験者に渡し、クエリからその目的を想像

表 5: アンケート 1 追加結果

手法	評価値平均	分散
goo	3.89	0.10
Web5	4.56	0.47
RWS	4.1	0.54
EXRWS	4.67	0.22

してもらった上で、関連語リストに目的を満たすような語が載っているかどうかを調べる予定である。

- 質問 2 : 興味を引くような関連語があればそれを選択し、評価してください。
 - 比較対象 [Web5][RWS][EXRWS]
1. 不満:そもそも興味のある関連語がなかった
 2. やや不満:期待したような知識は得られなかった
 3. 普通:新しい知識は得られたが、興味が沸く内容ではなかった
 4. やや満足:新しい知識が得られ、興味は沸いたが、詳しく知るためには検索をしない必要があった
 5. 満足:新しい知識が得られ、興味が沸いた。その興味も検索結果を見ることで満足できた。

表 6: アンケート 2 結果

手法	評価値平均	分散
Web5	2.24	1.78
RWS	2.52	1.61
EXRWS	3.16	1.33

結果は表 6 のようになった。提案手法である [EXRWS] が最も良い結果となった。しかし評価値平均は 3.16 と、やや低い点数であった。新しい知識や興味を得る手段としては活用できるが、その興味を満足させるためには、現在のクエリ生成法では不十分であるので、新たに考える必要がある。

7 おわりに

適合文書の収集先を Wikipedia にしても、Web 全体から適合文書を収集した時と劣らない結果が返ってくることが分かった。また、ユーザに他の知識をクエリとして推薦することも可能であることが分かった。

表 7: 実験クエリ一覧

準備したクエリ	被験者選択回数
ロールちゃん	3
縄跳び 跳び方	1
麻雀 点数計算	3
奈良時代	0
おみくじ 結ぶ	0
ほぼ日手帳	2
赤兎馬	1
麻雀 ルール	0
ゴルフ 打ち方	0
劉備 桃園	3
切手 価値	0
釘宮 日野	2
ミニマル・ミュージック	2
ホステージ	3
スタン・ハンセン	0
リボルビング払い	1
iPS 細胞	2
大富豪 地方ルール	1
NP 完全問題	0
インサイダー取引	1

今後も、検索結果の精度向上と新たなクエリ推薦の 2 点について研究を行う。前者はまず、考察で述べた再実験に加え、インターネットが苦手な層に意見を聞き、どのようなクエリを入力したときに結果が定まらず苦労するのか、について調査を行った上で、それらのクエリに本手法が有効に働くのかどうかを実験する。後者は、ユーザがどのようなクエリに興味を持ち、またどのようなクエリで検索を行うと、その興味を満足させることができるのかを検討する。

謝辞

本研究は科研費 (20240003) の助成を受けたものである。

参考文献

- [1] 大石 哲也, 倉元 俊介, 峯 恒憲, 長谷川 隆三, 藤田 博, 越村 三幸: "関連単語抽出アルゴリズムを用いた Web 検索クエリの生成" 電子情報通信学会 情報・システムソサイエティ和文論文誌データ工学特集号 掲載予定 Vol.J92-D, No.3, Mar. 2009.
- [2] google, <http://www.google.co.jp/>
- [3] yahoo, <http://www.yahoo.co.jp/>
- [4] goo, <http://www.goo.ne.jp/>

- [5] Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd: “ The PageRank Citation Ranking: Bringing Order to the Web ”, 1998,
- [6] Wikipedia: <http://ja.wikipedia.org/>
- [7] 土方嘉徳: ”情報推薦・情報フィルタリングのためのユーザプロファイリング技術”, 人工知能学会論文誌 19 卷 3 号 a,2004
- [8] 高速形態素解析システム「MeCab」:
<http://mecab.sourceforge.net/>
- [9] S.E.Robertson and K.S.Jones.: “ Relevance weighting of search terms. ” , Journal of the American Society for Information Science,27:129-146,1976
- [10] 石川博: “ 次世代データベースとデータマイニング ”, Web とデータマイニング, P182-183 , 2005 , CQ出版社
- [11] Web of the Year 2007: <http://woy2007.sbcr.jp/>
- [12] Chesney, Thomas: ”An empirical examination of Wikipedia’s credibility.”, First Monday. 2006. 11(11).
- [13] 真野博子, 伊東秀夫, 小川泰嗣:“ 文書検索におけるランキング検索技術 ”Ricoh Texhcnical Report No.29,December,2003
- [14] Andrei Broder: ”A taxonomy of web search”, SIGIR Forum,Fall 2002,Vol.36,No.2