

Web 検索サービスにおける多義的なクエリ推薦手法

今井 良太[†] 戸田 浩之^{††} 関口裕一郎^{††}
望月 崇由^{†††} 鈴木 智也^{†††} 今井 桂子^{††††}

[†] 中央大学大学院理工学研究科 情報工学専攻 〒112-8551 東京都文京区春日 1-13-27

^{††} 日本電信電話株式会社 NTT サイバーソリューション研究所 〒239-0847 神奈川県横須賀市光の丘 1-1

^{†††} NTT レゾナント株式会社 〒108-0023 東京都港区芝浦 3-4-1 グランパークタワー

^{††††} 中央大学理工学部 情報工学科 〒112-8551 東京都文京区春日 1-13-27

E-mail: [†]rimai@imai-lab2.ise.chuo-u.ac.jp, ^{††}toda.hiroyuki@lab.ntt.co.jp, ^{††††}imai@ise.chuo-u.ac.jp

あらまし 今日、多くの Web 検索サービスではクエリ推薦機能が提供されている。ユーザは、推薦されたクエリから自身の意図に合うクエリを選択することで、容易に検索結果を改善させることができる。本研究では、入力クエリが複数の意味をもつ場合や、複数の意図が考えられる場合に、それぞれの意味に関連するクエリをログから抽出するクエリ推薦手法を提案する。本研究で用いる手法は、検索エンジンのログに含まれる、入力クエリと選択された URL の関係に着目することで、推薦クエリを生成するものである。この手法を用いた既存研究では、入力クエリが複数の意味や分野で用いられる場合に、より一般的な用法に推薦クエリが偏ってしまう傾向があった。提案手法では、クエリと URL から構築したグラフをクラスタリングすることで、特定の意味に偏ることなく推薦クエリを生成できる。
キーワード Web とインターネット、検索エンジン、クエリ推薦

Mutisense Query Suggestion Method for Web Search Services

Ryota IMAI[†], Hiroyuki TODA^{††}, Yuichiro SEKIGUCHI^{††}, Takayoshi MOCHIZUKI^{†††},
Tomoya SUZUKI^{†††}, and Keiko IMAI^{††††}

[†] Graduate School of Science and Engineering, Chuo University Kasuga, Bunkyo-ku, Tokyo, 112-8551 Japan

^{††} NTT Cyber Solutions Laboratories, NTT Corporation 1-1, Hikarino-oka, Yokosuka-shi, Kanagawa, 239-0847 Japan

^{†††} NTT Resonant Inc. 4-1-8F Granpark tower Shibaura 3-chome, Minato-ku, Tokyo, 108-0023 Japan

^{††††} Department of Science and Engineering, Chuo University Kasuga, Bunkyo-ku, Tokyo, 112-8551 Japan

E-mail: [†]rimai@imai-lab2.ise.chuo-u.ac.jp, ^{††}toda.hiroyuki@lab.ntt.co.jp, ^{††††}imai@ise.chuo-u.ac.jp

Abstract Today, most web search services have a function called query suggestion. Users can improve search results by selecting queries relevant to their intents from suggested queries. In this paper, we propose an improved algorithm for query suggestion. When the inputted query has multiple meanings, or is used with different intents, this algorithm can generate queries based on each meaning. The method we use generates queries focusing on relations between queries and selected URLs in search logs. In the existing research, the suggested queries tend to be occupied by a particular meaning. The algorithm we propose can generate equal queries by clustering the graph constructed from logs.

Key words Web and the Internet, Search Engine, Query Suggestion

1. はじめに

現在、ほとんどの検索エンジンでは、ユーザが入力したクエリをもとに Web ページを検索し、その結果を何らかの基準に

よって順位付けして提示している。このとき、あいまいな単語や、同じ表記で複数の意味に用いられる単語をクエリとして入力すると、ユーザが意図した意味に対応する項目が検索結果の上位に表示されるとは限らない。例えば、工具の意味で「ドラ

イバ」というクエリを入力したユーザは、ソフトウェアの「ドライバ」について言及しているページの結果も同時に受け取ることになるだろう。また、単語自体が複数の意味をもたなくとも、ユーザによって意図が異なる場合も考えられる。例えば、「DVD」というクエリの場合、あるユーザは DVD の売り上げランキングを閲覧しようとし、別のユーザはレンタル DVD のサービスを探そうとするかもしれない。このような場合、ユーザは意図した結果が現れるまで下位の結果をたどるか、単語を追加したり置き換えたりする形でクエリを変えて再検索するといった対応を求められる。ユーザの視点から考えると、このような操作をより容易に行なえるような手段が望まれる。

Azar ら [2] は、同じクエリを入力するユーザが異なる意図をもってると仮定し、各ユーザの意図に沿うように検索結果を並び替える手法を提案した。これは、あるユーザが検索結果から意図に沿う項目を選択するとき、その項目が下位にあるほど選択する際の労力が大きいと考え、すべてのユーザの労力の総和を最小化するものである。

この問題のもう一つの解決策として、クエリ推薦という機能が挙げられる。これは、検索エンジンが検索結果を表示する際、結果と同時に別のクエリを推薦するというものである。ユーザは、この推薦されたクエリを選ぶことで、検索結果を自身の意図に容易に近付けることができる。

しかし、先ほどの「ドライバ」などのクエリの場合、推薦されたクエリが必ずしもユーザの意図に沿うものであるとは限らない。ユーザの意図を検索エンジンが判断することは難しいため、複数の意味をもつクエリについては、それぞれの意味について偏りなく推薦クエリを選び出すという方法が考えられる。先の「ドライバ」の例で言えば、工具とソフトウェアについてそれぞれ推薦クエリを提示することができれば、ユーザは自身の意図に合う推薦クエリを選択し、容易に所望の結果に到達することができる。

そこで本研究では、クエリ推薦手法について示す。本研究の手法では、Mei ら [7] の手法を基にして、ログから構築したクエリと URL からなる 2 部グラフを利用して推薦クエリを生成する。この URL とは、ユーザが検索結果の中から選択した URL を指す。例えば、ある 2 つのクエリの検索結果から同じ URL が頻繁に選択されていれば、その 2 つのクエリは互いに有効な推薦クエリであると考えることができる。これをグラフ上で考えると、一方のクエリから頻度の高い URL をたどり、この URL から頻度の高い別のクエリへとたどることで、このようなクエリを見つけ出すことができる。図 1 の例では、「dvd」と「blu-ray」というクエリからは「rental.example.co.jp」という URL が選択されることが多く、この 2 つのクエリは互いに推薦クエリとして適しているといえる。しかし、Mei らの手法を日本語向け検索エンジンのログに対して適用したところ、推薦クエリが特定の意味に偏ってしまう傾向があった。そこで本研究では、この 2 部グラフをクラスタリングすることで、複数の意味への対応を実現した。実験では、2 種類のクラスタリング法を適用し、既存研究との比較を行なった。

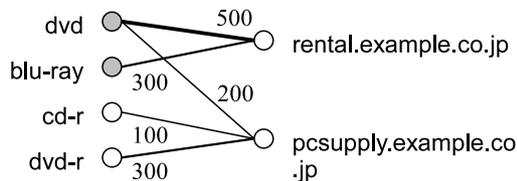


図 1 推薦クエリの考え方

2. 関連研究

クエリ推薦には多くの手法が提案されており、既存の検索エンジンにおいても検索結果と同時に表示される形で提供されていることが多い。いくつかの有名な検索エンジンでの推薦クエリを見てみると、入力クエリに単語を追加したり、クエリ中の単語を置き換えたものが推薦されていることがわかる。言い換えれば、クエリを単語の集まりとして捉え、単語同士の関係性を基にしているものが多いように見える。しかし、各エンジンがどのように推薦クエリを生成しているかは公表されておらず、具体的なアルゴリズムは不明である。

関連研究としては、推薦クエリを生成するために、検索エンジンが収集したログを基にするものが多数提案されている。これらの手法は、クエリの生成方法やログの利用方法である程度分類することができる。

推薦クエリの生成方法としては、単語の追加・置換 [9] や、クエリの置換 [4] ~ [8] が考えられる。単語の追加・置換とは、入力クエリを単語の集まりとして考え、それらの単語に共起しやすい単語を加えたり、入力クエリの一部と置き換えたものを推薦することで、検索結果を改善させるものである。例えば、「ドライバ」というクエリに対して「ドライバ 工具」のように単語を加えたり、「映画 レンタル」というクエリに対して「DVD レンタル」のように単語を置き換えたものを推薦する。Wang ら [9] は、確率的翻訳モデルと確率的文脈モデルを用いて、準同義語や共起しやすい単語を解析し、効果的にクエリを改良する手法を提案した。一方、クエリの置換とは、検索結果を改善するために、入力クエリそのものを置き換えるためのクエリを推薦するものである。単語の追加・置換と異なり、入力クエリの文字列によらず全体を置き換えることができるため、柔軟な推薦が可能と考えられる。例えば、「ドライバ」というクエリに対して「工具セット」のようなクエリを推薦することも可能である。Jones ら [6] の研究では、クエリ全体を置き換えるほか、クエリを成句で分割して一部を置き換えるといったことも行なわれている。

ログには様々な情報が記録されているが、基にする情報という観点から見ると、クエリログを用いる手法 [5], [6], [9] や、クリックログを用いる手法 [4], [7], [8] が挙げられる。ここで、クエリログは入力クエリのみが記録されているログを指し、クリックログは入力クエリとその検索結果中で選択された URL の組が記録されているログを指す。クエリログを用いる手法とは、クエリの出現頻度や時系列の情報を基にして、推薦クエリを生成するものである。Chien ら [5] は、クエリ間の時間的相関

関係に基づいて、意味的に類似したクエリを抽出する手法を提案した。これは、各クエリにはイベントやニュースなどによって時間的な頻度の変化が生じると仮定し、近い頻度変化をもつクエリを推薦クエリとして抽出するものである。一方、クリックログを用いる手法とは、入力クエリと、その検索結果の中から実際に閲覧された URL の情報を基にして、クエリと URL の関係から関連性の高いクエリを抽出するものである。ここで得られる URL は、過去のユーザが何らかの判断により実際に選択したものであるため、他のユーザにとっても意味のある情報が得られやすいと考えることができる。Mei ら [7] は、ログ中のクエリと URL のペアから 2 部グラフを構築し、Hitting Time という概念を導入することで、入力クエリと他のクエリ間の関連性を求める手法を提案した。

ログから構築したグラフに対してクラスタリングを行なうものとしては、Wen [10] らの研究がある。これは、人手によって管理・維持されるドキュメントを扱う検索エンジンにおいて、ログに含まれるクエリとドキュメントをクラスタ化することで、「よくある質問」(FAQ) を自動的に検出するための手法を提案するものである。この研究の目的は、サイト運営者がドキュメントを更新する際に、よく閲覧されるドキュメントを効果的に見つけられるようにすることであり、ユーザの入力クエリに対して推薦クエリを生成する本研究とは方向性は異なる。

3. 既存研究の概要

本研究では、クリックログを用いたクエリ推薦手法について検討した。この手法を用いた既存研究として、Mei らの研究 [7] が挙げられる。以下に Mei らの手法の概要を述べる。

3.1 2 部グラフの構築

検索エンジンには、それが利用されたときの様々な情報がログとして蓄積されている。ある利用者がクエリを送信し、検索結果からある URL を選択・閲覧したとき、ログには次のような情報を含むレコードが記録される。

- 日付と時刻
- 入力クエリ
- 閲覧した URL
- その URL の検索結果における順位

したがって、ログの各レコードから〈クエリ, URL〉のペアを抽出することで、送信したクエリと閲覧した URL の対応関係を見ることができる。

ここで、頂点集合 $V = V_1 \cup V_2$ 、辺集合 E からなる 2 部グラフ $G = (V, E)$ を考える。クエリと URL を頂点とすると、 V_1 をクエリの集合、 V_2 を URL の集合と考えることができる。任意のクエリ $q_i \in V_1$ と URL $u_j \in V_2$ について、ログにペア $\langle q_i, u_j \rangle$ が 1 つ以上出現するとき、 $e = (i, j) \in E$ が存在する。各辺は重み $w(i, j)$ を持ち、 $w(i, j)$ はペア $\langle q_i, u_j \rangle$ がログ中に現れる回数に等しい。

3.2 Hitting Time の導入

この 2 部グラフ上で、ランダムウォークを考える。まず、 V_1 上の頂点を始点とし、ここから辺の重みに比例する確率によって辺を選び、 V_2 上の頂点に移動する。同様にして、 V_2 上の頂

Query	URL
q_1	u_1
q_1	u_2
q_2	u_2
q_2	u_3
q_3	u_1
q_3	u_2

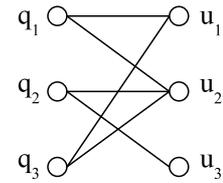


図 2 ログに含まれるペアとそれに対応する 2 部グラフ

点に接続する辺を確率によって選び、 V_1 に移動する。クエリ q_i と URL u_j の間の辺を選択する際の遷移確率は、次の式で定義される。

$$p_{ij} = \frac{w(i, j)}{d_i}, d_i = \sum_{j \in V_2} w(i, j)$$

また、 V_1 上の 2 頂点間の遷移確率、すなわちクエリ q_i から URL u_k を経由してクエリ q_j に移動する確率は、次のように表される。

$$p_{ij} = \sum_{k \in V_2} \frac{w(i, k)}{d_i} \frac{w(k, j)}{d_k}$$

ここから、グラフ G 上の Hitting Time [7] を考える。 A を V の部分集合とし、 X_t を離散時間 t におけるランダムウォークの位置とする。ランダムウォークが部分集合 A に到達する最短の時間を T^A とすると、 $T^A = \min\{t : X_t \in A, t \geq 0\}$ と表せる。 T^A は確率変数であるので、この変数の期待値を h_i^A で表す。ここで、 i は $t = 0$ におけるランダムウォークの位置である。すなわち、 $h_i^A = E[T^A | X_0 = i]$ と表せる。これを式変形すると、最終的に次の式が得られる。

$$h_i^A = \begin{cases} 0 & (i \in A) \\ \sum_{j \notin A} p_{ij} h_j^A + 1 & (i \notin A) \end{cases}$$

3.3 アルゴリズム

前述の Hitting Time を、ログから構築した 2 部グラフ $G = (V, E)$ に適用する。 $V = V_1 \cup V_2$ であり、 V_1 はクエリに、 V_2 は URL にそれぞれ対応する。クエリ q_s がいくつかの URL にアクセスするのに用いられるとき、それらの URL にアクセスするために別のクエリ q_i が用いられていれば、 q_i は q_s に対する推薦クエリとして適していると考えられる。

アルゴリズムの方針は次のとおりである。まず、入力クエリを q_s とする。 $A = \{q_s\}$ として、Hitting Time h_i^A をすべてのクエリについて求める。最後に、 h_i^A に基づいて k 個のクエリを選出し、これらを推薦クエリとして出力する。

しかし、ログから構築した 2 部グラフは非常に大きく、すべてのクエリを扱うと計算時間が膨大になってしまう。したがって、元のグラフに対してある深さまで深さ優先探索を行ない、ここで探索された部分グラフを対象に Hitting Time を求める。よって、アルゴリズムは次のようになる。

(1) グラフ G に対して入力クエリ q_s を根とする DFS を実行し、探索された部分グラフを新たに G とする。

(2) 遷移確率 p_{ij} を用いて、 V_1 上の頂点 i, j 間のランダムウォークを構成する。

(3) q_s 以外のクエリについて、Hitting Time を求める。 $h_i(0) = 0$ として、時間 $t = 0, \dots, m$ について、 $h_i(t)$ を次のようにして求める。

$$h_i(t+1) = \sum_{j \neq s} p_{ij} h_j(t) + 1$$

(4) h_i^* を最終的な $h_i(t)$ の値とし、 h_i^* の小さい順に k 個のクエリを選び出し、推薦クエリとする。

4. 提案手法

前述の既存手法を実際のログに対して適用すると、表 1 のように、類似したクエリが複数出現することがわかる。これらのクエリが小さい Hitting Time をもっていると、昇順で k 個のクエリを選び出した際に、特定の意味に関するクエリが多くを占めてしまうことになる。提案手法では、グラフをクラスタリングすることで、複数の意味に対応した推薦クエリの選択を実現する。具体的には、2 つのクラスタリング法を提案し、評価実験にて結果を比較する。

4.1 クラスタリング法 1

この手法では、グラフ上の各 URL について、その URL に接続するクエリの集合をクラスタとし、その中から最も Hitting Time が小さいクエリを選出する。これは、同一の URL が選択されたクエリは類似した意味で用いられるという仮定をもとにした手法である。基本的な考え方は次のとおりである。

(1) ある URL u_j について、 u_j に接続しているクエリの集合をクラスタ C_j とし、 C_j から最も Hitting Time が小さいクエリを選ぶ。

(2) 手順 (1) をすべての URL について行ない、各クラスタから選出されたクエリを出力する。

ここで、任意の 2 つのクラスタは 1 つ以上のクエリを共有する可能性があることに注意する。すなわち、共有されるクエリが非常に小さい Hitting Time をもっている場合、複数のクラスタからそのクエリが選出される場合がある。

4.2 クラスタリング法 2

もう 1 つの手法は、グラフ上で密度を考え、密度が高い部分をクラスタとするものである。ある 1 つの意味について考えると、その意味に関連するクエリや URL はそれぞれ複数存在することが多い。そして、それらのクエリと URL は互いに密に接続していると考えられることができる。したがって、グラフ上で密度が高い部分を 1 つのクラスタとして考え、各クラスタから Hitting Time が小さいクエリを選出することができれば、推薦クエリが特定の意味に偏ることを防ぐことができる。

この手法では、高橋ら [11] および Aizawa [1] のクラスタリング手法を基にした。また、クラスタリング法 1 と同様に、この手法でも各クラスタが同じクエリを共有する場合を考慮に入れるものとする。

この手法の基本的な考え方は次の通りである。

(1) グラフ G からクエリ q_i をもとに初期クラスタ C_i を作成する。

(2) C_i から頂点を 1 つずつ除去することを考える。このとき、ある頂点を除去することでクラスタの評価値が増加するならば、その頂点をクラスタから削除する。

(3) すべての頂点を評価した後、生成されたクラスタ C'_i が有用なものならば、出力するクラスタの 1 つとして採用する。

(4) すべてのクエリについて (1) から (3) の手続きを行ない、採用された各クラスタから最も Hitting Time が小さいクエリを選び、出力する。

手順 (1) の初期クラスタは、 q_i と接続している URL、およびそれらの URL と接続しているクエリからなる。すなわち、 q_i を根とし、深さの最大値を 2 とする深さ優先探索によって得られる部分グラフと考えることができる。

手順 (2) では、頂点を評価する順序によって生成されるクラスタに差異が生じることに注意する。ここでは [1] に倣い、各頂点の情報量の昇順で評価する。

頂点やクラスタの評価には [1] の相互情報量を用いる。すなわち、クエリと URL の情報は式 (1) (2) によって求められ、生成されたクラスタは式 (3) の値が正のとき、有用であると判定される。

$$\delta I(q_i, S_u) = \sum_{u_j \in S_u} P(q_i, u_j) \log \frac{P(q_i, u_j)}{P(q_i)P(u_j)} \quad (1)$$

$$\delta I(S_q, u_j) = \sum_{q_i \in S_q} P(q_i, u_j) \log \frac{P(q_i, u_j)}{P(q_i)P(u_j)} \quad (2)$$

$$\begin{aligned} \delta I(S_q, S_u) &= P(S_q, S_u) \log \frac{P(S_q, S_u)}{P(S_q)P(S_u)} \\ &\quad - \sum_{q_i \in S_q} \sum_{u_j \in S_u} P(q_i, u_j) \log \frac{P(q_i, u_j)}{P(q_i)P(u_j)} \quad (3) \end{aligned}$$

5. 評価実験

5.1 評価条件

実際の検索エンジンのログを用いて、既存研究の結果と 2 つのクラスタリング法を適用した結果を比較する評価実験を行った。使用したログは日本語向け検索エンジンの 2008 年 10 月から 12 月までの 3 か月分である。

5.2 評価結果および考察

表 1, 2, 3 は、入力クエリ「dvd」に対して各手法で Hitting Time の上位のクエリを出力したものである。既存手法では、上位 10 個中 5 個がレンタルサービスに関連したものである。6 位と 7 位のように表記ゆれとして統合できそうなものもあるが、一方で「dvd レンタル」と「レンタルビデオ」のように、ほとんど同じ意味であるにも関わらず、単なる表記ゆれとして処理するのは難しいものも含まれている。

クラスタリング法 1 では、クエリの総数は 9 個になり、既存手法では 11 位以下のために含まれなかった「ぼすれん」が 9 位に入っている。しかし、類似したクエリも多く残っており、クラスタリングが十分に行なわれていないと見ることもできる。

クラスタリング法 2 では、クラスタリング法 1 に比べてク

表 1 既存手法 (上位 10 個)

クエリ	HT	頻度
dvd-r dl	1.384	11
dvd 容量	2.241	30
dvd-r	5.950	136
dvd レンタル	7.080	12
レンタル dvd	7.108	64
dvd ランキング	7.287	72
dvd ランキング	7.316	86
レンタルビデオ	7.904	65
dvd レンタル	8.699	204
tsutaya レンタル	8.923	62

表 2 クラスタリング法 1

クエリ	HT	頻度
dvd-r dl	1.384	11
dvd 容量	2.241	30
dvd-r	5.950	136
dvd レンタル	7.080	12
レンタル dvd	7.108	64
レンタルビデオ	7.904	65
dvd レンタル	8.699	204
tsutaya レンタル	8.923	62
ぼすれん	9.260	1222

表 3 クラスタリング法 2

クエリ	HT	頻度
dvd 容量	2.241	30
dvd-r	5.950	136
レンタル dvd	7.108	64
ツタヤオンライン	9.991	254

エリ数は減っているが、既存手法では含まれなかった「ツタヤオンライン」が新たに含まれている。また、既存手法のクエリと比べてみると、減少した分のクエリは「dvd レンタル」「レンタルビデオ」のように他のクエリと類似したものがほとんどであることがわかる。

2つのクラスタリング法を比べてみると、類似したクエリを除去し、下位にあったクエリがより上位に上がっていることから、この例ではクラスタリング法 2の方がクラスタリングが上手く働いているといえる。

他のいくつかの実験結果によると、クラスタリング法 1は、企業の代表的なサイトを探すような navigational なクエリ [3] に対して良い結果を出しやすい傾向があると考えられる。また、類似したクエリが別々のクラスタに分かれてしまう場合があったが、URL に基づくクラスタリングでは 1つのクラスタがあまり大きくなり、類似したクエリが 1つのクラスタにまとまりにくいことが原因と考えられる。一方、クラスタリング法 2では、クラスタリング法 1に比べて計算量は増えるものの、閲覧先が複数のサイトに分散するような informational なクエリ [3] に対しても良い結果が出やすい傾向があると考えられる。本研究の手法は、入力クエリがあらかじめログに存在していることが前提となっており、ユーザの要求より前に各クエリに対して推薦クエリを用意しておくことができる。したがって、計算量はそれほど大きな問題ではなく、より良い結果を出しやすいクラスタリング法 2の方が実用的であると思われる。

6. 結 論

本研究では [7] の手法をもとにログから 2部グラフを構築

し、さらに [1], [11] の手法によってグラフをクラスタリングすることで、特定の意味に偏らないクエリ推薦を行なう手法を構築した。実際のログを用いた評価実験では、既存手法と提案手法を比較したとき、同じ意味に対応するクエリが減少し、代わりに他のクエリが推薦クエリとして現れていることを確認した。

評価手法については、本研究では主観による評価が中心となっているが、推薦クエリの有効性について議論するためには、定量評価に向けた評価手法の検討が必要となる。ログの期間についても、より長期間のログを対象とすることで、ログ自体の内容の偏りを抑え、よりの確な推薦クエリを得られるようになるだろう。

文 献

- [1] Akiko Aizawa. A Method of Cluster-Based Indexing of Textual Data. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, pp. 1-7, 2002.
- [2] Yossi Azar, Iftah Gamzuz, Xiaoxin Yin. Multiple intents re-ranking. In *Proceedings of the 41st ACM symposium on Theory of computing*, pp. 669-678, 2009.
- [3] Andrei Broder. A taxonomy of web search. In *ACM SIGIR Forum, Volume 36, Issue 2*, pp. 3-10, 2002.
- [4] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, Hang Li. Context-aware query suggestion by mining click-through and session data. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 875-883, 2008.
- [5] Steve Chien, Nicole Immorlica. Semantic similarity between search engine queries using temporal correlation. In *Proceedings of the 14th international conference on World Wide Web*, pp. 2-11, 2005.
- [6] Rosie Jones, Benjamin Rey, Omid Madani, Wiley Greiner. Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web*, pp. 387-396, 2006.
- [7] Qiaozhu Mei, Dengyong Zhou, Kenneth Church. Query suggestion using hitting time. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 469-478, 2008.
- [8] Silviu Cucerzan, Ryan W. White. Query suggestion based on user landing pages. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 875-876, 2007.
- [9] Xuanhui Wang, ChengXiang Zhai. Mining term association patterns from search logs for effective query reformulation. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 335-344, 2009.
- [10] Ji-Rong Wen, Jian-Yun Nie, Hong-Jiang Zhang. Clustering user queries of a search engine. In *Proceedings of the 10th international conference on World Wide Web*, pp. 162-168, 2001.
- [11] 高橋 秀幸, 竹内 孔一. 多義性を考慮した同時共起クラスタリングによる同志の類語抽出. 電子情報通信学会技術研究報告, NLC, 言語理解とコミュニケーション, pp. 37-42, 2009.