DBSJ Journal, Vol.7, No.3
December 2008

商用検索エンジンのヒット数に対する信頼性の検証

Reliability Verification of Search Engines' Hit Count

舟橋 卓也[1]　上田 高徳[1]
平手 勇宇[2]　山名 早人[3]

Takuya FUNAHASHI　Takanori UEDA
Yu HIRATE　Hayato YAMANA

近年，検索エンジンを本来の用途である検索以外に，翻訳支援や自然言語処理等の様々なアプリケーションに利用しようとする研究が進んでいる．これらの研究の多くは，クエリに対する検索結果のヒット数を利用している．従来，検索エンジンが返すヒット数は信頼できるという仮定のもとで用いられてきたが，ヒット数の信頼性に対する検証は行われていない．ヒット数が不正確な場合，ヒット数を利用した研究の信頼性は疑わしいものとなる．そこで本稿では，検索エンジンのヒット数に対してその信頼性の検証を試みた．検証実験では，商用検索エンジンである Google, Yahoo! JAPAN, Yahoo!，Live Search が提供する Web 検索 API を利用した．検証実験の結果，Google，Live Search，Yahoo! JAPAN では 5 割から 6 割のクエリにおいて，検索エンジンのヒット数は実際に検索結果として取得できる結果数の 100 倍以上大きな値を取ることがわかった．また，Yahoo!では検索エンジンから得られたヒット数と実際に検索結果として得られる結果数がほぼ一致していることがわかった．

In recent years, a number of application studies have been carried out in the area of both translation support and natural language processing by using search engines, while search engines themselves are originally used for searching the Web. Many of these studies use search engines' hit count. Conventional studies assume that the hit count is reliable even though none of the studies have verified the reliability of search enginesn though none of the studies have e area of both translation support and natural language processing by using search engines, while search engines themselves engines' hit count. In this experiment, we use Web search APIs provided by Google, Yahoo! JAPAN, Yahoo! and Live Search. The experiment shows that hit count, resulted from 50 to 60% of queries in Google, Live Search and Yahoo! JAPAN, is more than 100 times larger than the number of search results we can actually get. Moreover, the experiment shows that hit count is almost same as the number of search results we can actually get in Yahoo!.

[1] 学生会員 早稲田大学大学院基幹理工学研究科修士課程
{takuya, ueda}@yama.info.waseda.ac.jp
[2] 正会員 早稲田大学メディアネットワークセンター
hirate@yama.info.waseda.ac.jp
[3] 正会員 早稲田大学理工学術院，国立情報学研究所
yamana@yama.info.waseda.ac.jp

1. はじめに

これまでに，検索エンジンのヒット数を利用した研究が数多く行われている．検索エンジンのヒット数とは，検索エンジンが返す，ユーザが入力したクエリに適合するWebページ数の概算を指す．検索エンジンのヒット数は，検索エンジンがインデックス化している全Webページにおいて，クエリキーワードが出現するWebページ数の概算とみなすことができる．そのため，ヒット数を利用して，翻訳支援などの自然言語処理支援を行う研究[7][13][17]や，検索エンジンがインデックス化しているWebページ数の推定を行う研究[5]などが行われている．

これらの研究は，検索エンジンが返すヒット数が正しいという仮定の下で行われている．しかし，実際には検索エンジンのヒット数は，「いつ検索を実行したか」「検索開始オフセットをいくつにしたか」によって変化する．ここで，検索開始オフセットとは検索結果の何ページ目を表示しているかを示すものであり，検索結果として表示されているWebページのうち，最もランキングが高いWebページのランキングを指す．つまり，検索結果１ページにつき，10件の検索結果を表示する設定の場合，３ページ目の検索結果を見ているとすると，検索開始オフセットは21[4]となる．

検索の実行日時や検索開始のオフセットによって検索エンジンのヒット数が変動する理由は，検索エンジンがヒット数を算出するプロセスを明らかにしていないため断定することはできない．しかし，現在，一般に提案されている検索エンジンの構成から考えると，ヒット数が変動する理由として，検索エンジンが持つインデックス更新の仕組み，インデックスの構成，検索結果やインデックスのキャッシュが原因であると考えられる．

このように，ヒット数の変動は，検索エンジンの構成上やむを得ないものであると考えられるが，我々がヒット数を研究に用いるためにはその特徴や信頼性を知る必要がある．なぜなら，ヒット数の信頼性が低ければ，ヒット数を用いた研究の信頼性も低くなるためである．しかし，これまでに検索エンジンのヒット数の信頼性に対しての検証は行われていない．

このような背景のもと，本稿では，検索エンジンのヒット数に対してその信頼性の検証を行う．信頼性の検証にあたっては，従来実施されている検索エンジンのヒット数を利用した研究が，（1）複数のクエリによって得られたヒット数の大小関係を用いたもの[7][13][17]と，（2）１つのクエリのヒット数の値を用いたもの[5][8]の大きく２つに分類できることを踏まえ，これら２種類の利用方法におけるヒット数の信頼性を検証する．なお，今回の検証では検索日時に依存するヒット数の変動については検証対象とはせず，検索開始オフセットの変化によるヒット数の変化を検証対象とする．

以下，次のような構成をとる．２節では，現在の検索エンジンの構成について述べる．３節では，本研究の関連研究について述べる．４節では，検索エンジンのヒット数に対する信頼性の検証方法について述べる．５節では，本研究で使用するデータの取得方法である．６節では，検証結果について述べ，最後に７節において本研究のまとめを述べる．

[4]実際に API に指定するオフセットとは異なる．検索エンジンによっては検索開始オフセットが 0 から始まる場合がある．本論文では表記法を統一するため，検索結果の n 番目（1～）からを出力する場合のオフセットを n として表記する．

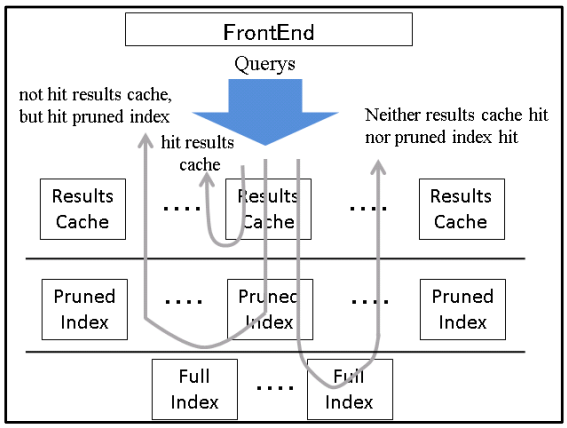

図 1. 検索システムの構成図

2. 一般的な検索エンジンの構成とそのヒット数

2.1. 一般的な検索エンジンの構成

一般的に，検索エンジンは転置インデックスを用いたブーリアン検索モデルで構築されている[12]．転置インデックスを用いることによって，クエリに合致する Web ページを短時間で検索することができる．しかし，単位時間あたりのクエリ数が増加すると，単純に転置インデックスを用いるだけでは高速な検索の提供が困難となる．そこで，検索エンジンは次のような方法を用いて検索の高速化を図っている．

・転置インデックスの分散
システムを高速化するために，検索エンジンはシステムの分散を図っている[15]．ユーザが検索エンジンにアクセスをすると，検索エンジンが内部的に複数持つサーバのうちの 1 つにアクセスをすることになる．検索エンジンが内部的に持つ複数のサーバでは，それぞれインデックスを保有する．複数のサーバを用いることにより，サーバ一台当たりの負荷を分散し，検索の高速化を図っている．

・転置インデックスの削減
現在，主な検索エンジンでは，クエリに合致する Web ページのうちランキング上位一定件数(1000 件)までしか検索結果として表示しない．そのため，検索エンジンはクローラによって収集した Web ページ全てをインデックス化する必要はなく，静的なランキングが上位となる Web ページのみインデックス化すれば十分である．転置インデックスの規模を縮小することによって，検索エンジンは高速に検索処理を行うことができる[1]．また，検索エンジンではクエリログをもとに検索結果をキャッシュ（Results Cache）することによって，検索の高速化を図っている[4]．[4]では，検索結果のキャッシュと，ページランクを用いてインデックスサイズを縮小した転置インデックス（Pruned Cache）を用いることによって，検索の高速化が図ることができたと述べられている．

これらの検索エンジンの構造を，簡略化した図を図 1 に示す．

2.2. ヒット数に関する考察とヒット数の定義

2.1 で述べたように，検索エンジンが縮小された転置インデックスを利用していることを考えると，ヒット数に対して次のような考察を行うことができる．

・ヒット数は縮小転置インデックスにインデックス化されている Web ページ，つまり静的なランキングが上位の Web ページより統計的に算出されている

つまり，検索エンジンのヒット数で示される検索結果の Web ページ全てを検索結果としては表示できないことを意味している．

また，Yahoo!が公開している Web 検索 API リファレンスには，ヒット数は検索結果を用いて算出しているため，検索開始オフセットと結果表示数（何件同時に表示するか）によって変化する可能性があることが記載されている[11]．この文章は，検索結果として表示する件数が多いほど，推定を行う際のサンプル数が多くなるため，ヒット数として正確な値が算出できることを意味している．

上記を踏まえた上で，本稿での検証に使用する 3 種類の検索エンジンのヒット数を定義する．なお，「検索を行うことによって実際に取得できた結果数」は，検索結果の「次へ」をクリックしていった時に，実際に検索結果として表示される検索結果（Web ページの URL）の総数を示す．

検索開始オフセットが 1 のときのヒット数　：h_f
検索開始オフセットが最大のときのヒット数　：h_l
検索を行うことによって実際に取得できた結果数：h_r

検索エンジンの構成を考えると，[11]で説明されている通り，h_l は h_f より正確なヒット数となることが推定できる．

3. 関連研究

本節では検索エンジンのヒット数の信頼性に関連する研究を紹介する．松尾ら[12]は「検索エンジンはクエリに対応するヒット数が取得可能な範囲（m 件以下;一般的には 1,000 件）に収まっていた場合に常に正確なヒット数を返す」という前提のもと，複数の検索結果から統計的に正確なヒット数を推定する手法を提案している．具体的には，クエリ q に対応する正確なヒット数を次のような手順で求める．

・q を含む文章に普遍的に含まれていると考えられるキーワード（以後，プローブ語と呼ぶ）を求め
・q とプローブ語を用いて AND 検索を行うことによってヒット数を取得可能な範囲（m 件以下）に落とし込み
・プローブ語を加えることによって減少したであろうヒット数を統計的に求めることにより，q に対応する正確なヒット数を求める

しかし，松尾らの手法では，「検索エンジンはクエリに対応するヒット数が取得可能な範囲（m 件以下）に収まっていた場合に常に正確なヒット数を返す」という前提に基づいている．すなわち，AND 検索を行い取得したヒット数が m 件以下となった場合，そのヒット数は正しいと仮定している．しかし，検索エンジンが最終的に上位にランキングされるであろう Web ページのみを保持していた場合，AND 検索の結果得られたヒット数は正確なものとはならない．そのため，松尾らの手法で推定した結果はすでに誤差が含まれた結果から推定を行ってしまう可能性がある．さらに，ヒット数が取得可能な範囲に収まっていた場合に常に正確なヒット数を返すとも限らない．

これに対して，本稿では，「検索結果数が取得可能な範囲内であっても信頼できるとは限らない」という前提を置き，ヒット数の検証を行う．

4. 検索エンジンのヒット数の検証手法

検索エンジンのヒット数を用いた研究は，1 節でも述べたように，（1）複数のクエリによって得られたヒット数の大小関係を用いたもの[7][13][17]と，（2）1 つのクエリのヒ

表 1. Web 検索 API の検索設定

	Google	Yahoo! JAPAN	Live Search	Yahoo!
1度に取得する結果数	10	10	10	10
類似フィルタ	off	off	off	off
アダルトフィルタ	off	off	off	off
検索対象とする言語	全言語	全言語	日本語	日本語

ット数の値を用いたもの[5][8]の大きく２つに分類できる．前者の研究では，複数クエリ間においてヒット数の大小関係が変化した場合に，後者の研究では，ヒット数の変動が発生するだけで問題となる．そこで本研究では，

・ヒット数の変動がどの程度の変動幅をもつか
・複数のクエリ間におけるヒット数の大小関係の変化がどの程度発生するか

の 2 点について検証を行う．

4.1. ヒット数の変動幅の検証

２節において，検索エンジンの構成より検索開始オフセットが大きいほどヒット数の値は正確であると考えられることについて述べた．そこで本稿では，より正確であると考えられる，検索開始オフセットが最大のときのヒット数 h_l と，多くの研究で最もよく利用されているヒット数である検索開始オフセットが 1 のときのヒット数 h_f の比較を行い，これらの間にどの程度の差が存在しているのか検証する．また，h_l と実際に取得することができた検索結果数 h_r との比較も行い，実際に取得できた結果数とヒット数の間にどの程度の差が存在しているのかについても検証する．なお，実際に取得することができた検索結果数として正確な値を用いるために，h_r を比較に用いる際には，h_r が取得可能な結果数（1000 件）未満に収まったクエリにおいてのみ用いる．

4.2. 複数のクエリ間におけるヒット数の大小関係の変化に対する検証

複数のクエリ間におけるヒット数の大小関係の変化を調べるために，検索開始オフセットが 1 のときのクエリペア q_1, q_2 に対するヒット数 h_{f1}，h_{f2} と，検索開始オフセットが最大のときの q に対するヒット数 h_{l1}，h_{l2} を比較し，ヒット数の大小関係の入れ替わりが発生しているかどうかを検証する．

5. データセット取得方法

5.1. 使用する検索エンジンと検索時の設定

本研究では，日本において広く利用されている Yahoo! JAPAN[10]，Google[3]，Live Search[6]の 3 つの検索エンジンが提供する Web 検索 API と，米国の検索エンジンである Yahoo![9]が提供する Web 検索 API を利用して実験を行う．Web 検索 API は各検索エンジンがアプリケーションを通して検索が可能なように API を設けているもので，実際に Web ブラウザから検索を行った検索結果と異なる場合がある[2]．しかし，多くの場合，検索エンジンを用いた研究は Web 検索 API を通して行われる．そこで，本研究でも Web 検索 API を通して実験を行う．

これらの Web 検索 API を用いて検索を行う際の設定を表 1 に示す．基本的な方針として，フィルタの影響を無くすため可能な限りオフにするように設定を行った．

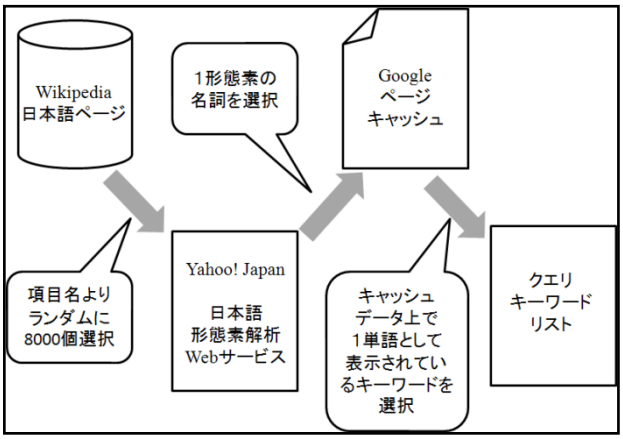

図 2. クエリキーワード選択手法

5.2. クエリキーワードの選択手法

本研究では，３節で述べたように AND 検索による不定要因をできるだけ排除するため，検索エンジンが保有する転置インデックスのキーとなっていると考えられるクエリキーワードを選択する方針をとった[5]．以下に，クエリキーワードの選出方法（図 2）を示す．

(1) 日本語版 Wikipedia の項目名よりランダムに 8,000 個のキーワード候補セットを作成する．

(2) 作成したキーワード候補セットを，Yahoo! JAPAN が提供をしている日本語形態素解析 Web サービスを用いて形態素解析を行う．形態素解析を行った結果，1 形態素かつ名詞であるキーワードのリストを作成する．

(3) (2)において作成したキーワードリストをクエリとして，Web ブラウザより Google で検索を行う．それぞれのクエリに対して得られた検索結果からキャッシュを選択し，ハイライトされているキーワード[6]が単独であるキーワードのリストを作成する．

(2)では，Yahoo! JAPAN が公開している形態素解析を利用することにより，Yahoo! JAPAN において検索インデックスのキーとして用いられているキーワードを推定している．また，(3)では，Google の形態素解析の結果を利用することにより，Google において検索インデックスのキーとして用いられているキーワードの推定している．

(1)～(3)によって，最終的に 500 個のキーワードからなるキーワードリストを作成した．実際に抽出されたキーワードの例として，「ごま油」，「クロロフィル」，「Mathematica」，「札」，「永字八法」等がある．

5.3. ヒット数の取得方法

ここでは，5.2 で述べた手法によって選出されたクエリを用いて，ヒット数を取得する方法について述べる．

[5] 実際の検索エンジンにおいては，n-gram が利用されている可能性もあり，転置インデックスのキーとなっていることを保証することはできない．しかし，検証で用いるクエリが，検索エンジン内部で AND 検索となることをできる限り避けるための処置として実施した．

[6] Google の検索結果においてキャッシュを選択すると，クエリキーワードがキャッシュされている文章中でハイライトされる．ハイライトが行われる際に，クエリキーワードが分割，色分けされて結果として表示をされる．本稿ではこの分割されている 1 要素が，Google が形態素解析を行った形態素となっていると仮定を置いている．

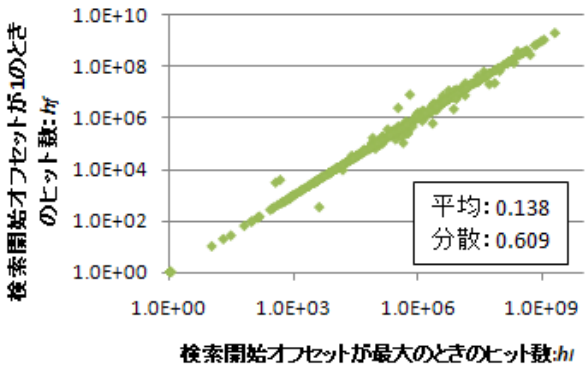

図 3. Google における h_l と h_f の相関図

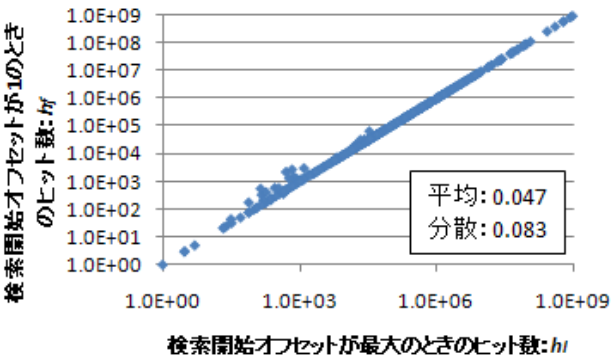

図 4. Live Search における h_l と h_f の相関図

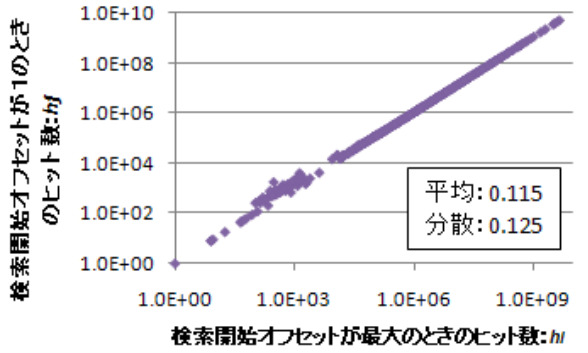

図 5. Yahoo! JAPAN における h_l と h_f の相関図

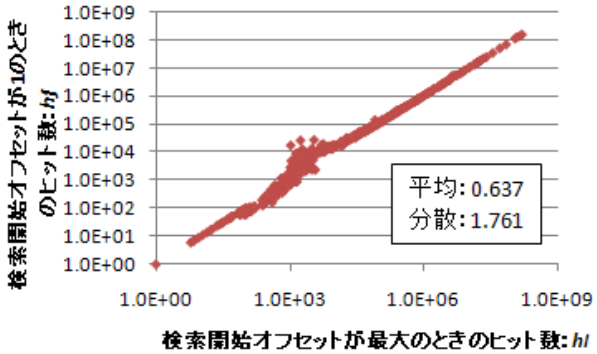

図 6. Yahoo! における h_l と h_f の相関図

本研究では，あるクエリ q を用いて検索を行った際の，検索開始オフセットが 1 の時のヒット数 h_f，検索開始オフセットが最大の時のヒット数 h_l，検索を終えた際に実際に取得することができた結果数（Web ページ数）h_r を求め比較した．具体的には，q に対し，以下の手法を適用した．

- 検索開始オフセットが 1 のときの q に対するヒット数 h_f を求める
- その後，検索開始オフセットを 10 ずつ増加させて検索結果を繰り返し取得し，それ以上検索結果が得られなくなった時点で表示されているヒット数 h_f を求める
- また，同時点で実際に検索結果として表示された Web ページ数を h_r をとして求める．

6. 検証結果

6.1. ヒット数の変動幅に対する検証結果

まず，検索開始オフセットが最大のときのヒット数 h_l と，検索開始オフセットが 1 のときのヒット数 h_f の関係を調べる．図 3～図 6 に，h_l と h_f の関係を両対数グラフにプロットした図を示す．なお，Google は検索開始オフセットが最大のときの検索エンジンが返すヒット数を実際に取得できた検索結果数に揃える調整を行っていたため，「検索開始オフセットが最大になった検索時の直前のオフセットを用いた検索によって得られるヒット数」を h_l として用いた（6.2 における検証でも採用）．具体的には，図 7 に示すように，h_r が 500 件以下の場合，h_l を h_r と同じ値を返すといった傾向がある．

次に，任意のクエリにおける h_l に対する h_f の変動率 r_d を，

$$r_d = \frac{|h_l - h_f|}{h_l}$$

と定義すると，それぞれの検索エンジンにおける r_d の平均と

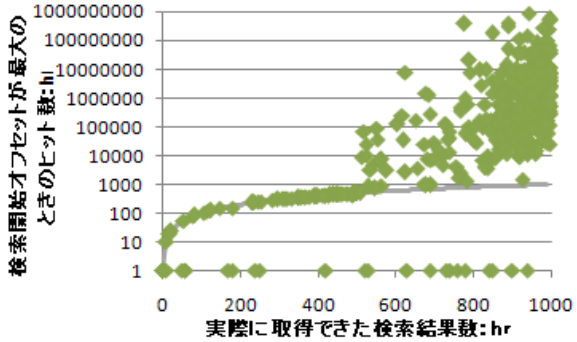

図 7. h_r に対する，検索開始オフセットが最大のときの検索エンジンが返すヒット数
（Google 要素数:410）

分散は図 3～図 6 中に示した値となる．図中の値に示される通り，h_f と h_l の値が必ずしも一致しておらず，平均で 4.7%(Live Search)～63.7%(Yahoo!)の変動があることがわかる．また，分散は Live Search や Yahoo! JAPAN では 0.047～0.125 と小さいが，Google は 0.609，Yahoo!は 1.761 と大きな値になっている．このことから，h_l を h_f の代わりに使う必要性が高いことがわかる．

h_r と h_l の関係

続いて，h_l の信頼性を検証するため，検索を行うことによって実際に取得できた結果数 h_r と，h_l の値にどれだけの差異が生じているかを調べた．h_r と h_l の関係をプロットした図を図 8～図 11 に示す．なお，図中の要素数とは h_r が取得可能な結果数（1000 件）未満に収まったクエリの数である．

Google，Live Search，Yahoo! JAPAN において，h_r は h_l と

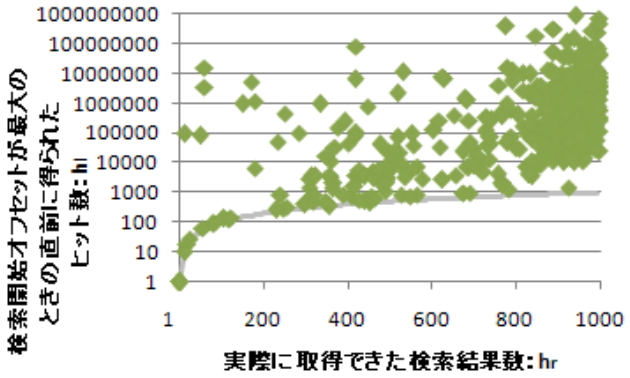

図 8. h_r に対する h_l (Google 要素数:410)

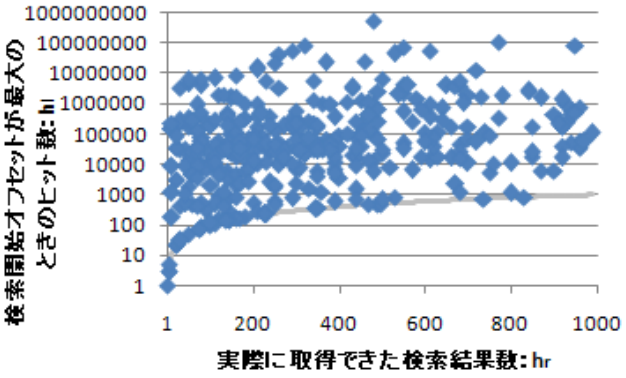

図 9. h_r に対する h_l (Live Search 要素数:374)

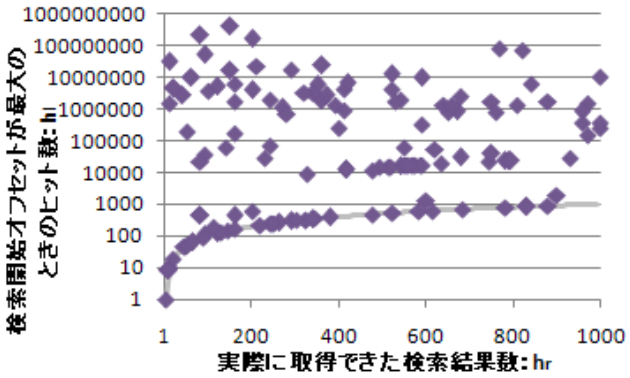

図 10. h_r に対する h_l (Yahoo! JAPAN 要素数:132)

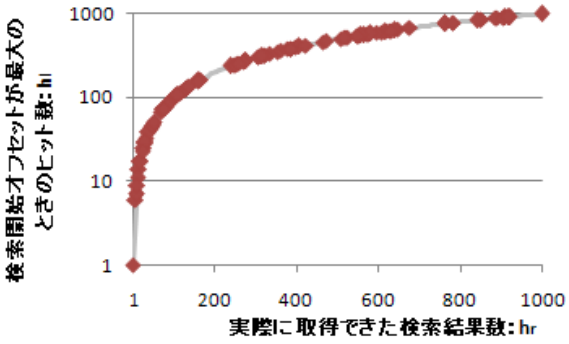

図 11. h_r に対する h_l (Yahoo! 要素数:106)

表 2. 各検索エンジン毎の r_g の分布

	Google	Live Search	Yahoo! JAPAN	Yahoo!
$0\% < r_g \leqq 10^2\%$	3.41%	7.49%	26.52%	99.06%
$10^2\% < r_g \leqq 10^3\%$	14.63%	10.43%	6.06%	0.94%
$10^3\% < r_g \leqq 10^4\%$	20.49%	23.26%	18.94%	0.00%
$10^4\% < r_g \leqq 10^5\%$	22.44%	32.35%	9.85%	0.00%
$10^5\% < r_g \leqq 10^6\%$	25.12%	17.38%	18.94%	0.00%
$10^6\% < r_g \leqq 10^7\%$	10.49%	6.95%	11.36%	0.00%
$10^7\% < r_g \leqq 10^8\%$	3.41%	1.87%	6.06%	0.00%
$10^8\% < r_g \leqq 10^9\%$	0.00%	0.27%	2.27%	0.00%
合計	100.00%	100.00%	100.00%	100.00%

大きな差が存在している．一方，図 11 をみると，Yahoo!において h_r と h_l がほぼ等しい．また，どの検索エンジンでも，検索エンジンが返すヒット数 h_l は h_r を下回ることがほとんどない．

ここで，実際に取得できた検索結果数 h_r に対する検索開始オフセットが最大のときのヒット数 h_l の増加率 r_g を次のように定義する．

$$r_g = \frac{h_l}{h_r}$$

r_g を各検索エンジンについて求め，その r_g の値を $0\% < r_g \leqq 10^2\%$0，$10^2\% < r_g \leqq 10^3\%$，…，$10^8\% < r_g \leqq 10^9\%$の範囲に分類する．そして，クエリ全体の何%がその範囲の r_g をもつのか計算した結果を表 2 に示す．表 2 から，h_r に対して h_l が 10,000%より大きくなっているのは，Google では全体の 61.46%，Live Search では全体の 58.82%，Yahoo! JAPAN では全体の 48.48%，Yahoo!では 0%となっていることがわかる．

この結果より，Google，Live Search，Yahoo! JAPAN では 5 割から 6 割のクエリにおいて，検索開始オフセットが最大のときのヒット数は，検索エンジンから実際に得られた結果数の 100 倍以上大きな値を取ることがわかる．また，Yahoo!では検索開始オフセットが最大のときのヒット数と検索エンジンから実際に得られた結果数がほぼ等しいことがわかる．

検索結果数の信頼性の観点からすると，現段階では，h_l と h_r の内，どちらがより正しいかを判断することができない．しかし，「h_r が実際の検索結果数として採用すべき正しい検索結果数である」と仮定すると，検索開始オフセットが最大のときのヒット数 h_l は，Yahoo!を除き 5〜6 割のクエリにおいて実際の検索結果数の 100 倍以上の値になっており，信頼性に乏しい値であると考えられる．なお，h_l と h_r の内，どちらがより正しいかという点と，Yahoo!のみが 99.06%の確率で h_r=h_l となる理由の解明については，今後の課題である．

6.2. 複数のクエリ間におけるヒット数の大小関係の入れ替わりに対する検証

複数のクエリ間においてヒット数の大小関係の入れ替わりがどのように発生しているのか検証を行った．まず，4.2 で述べた検証手法を，検索に使用した 500 のクエリキーワードの組み合わせによって作成したキーワードペアに対して適用し，h_f と h_l の間の大小関係の入れ替わりがどの程度発生しているのかについて調査を行った．結果を表 3 に示す．表 3 における 2 つのクエリ間におけるヒット数の大小関係の入れ替わり発生率は次の式より求められる．

$$\text{入れ替わり発生率} = \frac{\text{大小関係の入れ替わりの発生数}}{\text{クエリの組み合わせ数}}$$

表 3 では，Yahoo! JAPAN のみクエリの数が 498 件となっているが，これは Yahoo! JAPAN において検索時の不具合に

表 3. 2つのクエリ間における h_f, h_l の大小関係の入れ替わり発生率

	クエリ数	組み合わせ数	発生数	発生率
Google	500	124,750	1,732	1.4%
Live Search	500	124,750	357	0.3%
Yahoo! JAPAN	498	123,753	297	0.2%
Yahoo!	500	124,750	3,361	2.7%

よって2件のクエリに対して正常に検索結果が取得できなかったためである.

表 3 より，Google，Yahoo!では大小関係の入れ替わり発生率が，それぞれ 1.4%, 2.7%とどちらも 1%を超えるのに対し，Live Search，Yahoo! JAPAN では 0.3%，0.2%と大小関係の入れ替わり発生率が低い．Live Search，Yahoo! JAPAN においてこのように低い入れ替わり発生率が生じる理由は，これらの検索エンジンでは検索開始オフセットにあまり影響を受けずに，ヒット数を算出しているためだと考えられる．

次に，表 4 は，実際に検索結果として得られる検索結果数 h_f と h_r の大小関係の入れ替わり発生率を示したものである．表 4 に示すように Yahoo!を除いて 30～40%前後の入れ替わりが発生している．なお，ここでの h_r も，h_r が取得可能な結果数（1000 件）未満に収まったクエリのみの結果を用いた．

どの検索エンジンを用いた場合にも，それぞれの検索エンジンにおいて入れ替わり発生率は，h_f と h_l の間では，高々 3%である．しかし，h_f と h_r の間での入れ替わり発生率は，Yahoo!を除き 30～40%となる．また，6.1 でも述べたように，現段階では，h_l と h_r の内，どちらがより正しいかを判断することができない．このため，ヒット数の大小関係を利用する場合は，検証対象とした検索エンジンの中では，Yahoo!を用いるのが最もよいと判断できる．

7. おわりに

本論文では検索結果のヒット数に対する信頼性の検証を行った．具体的には，検索を行った際に一番はじめに表示がされるヒット数を h_f，一番最後の検索結果を閲覧しているときに表示されるヒット数を h_l，実際に取得することができたヒット数を h_r として，その関係性より検索エンジンのヒット数の検証を行った．500 個のクエリを用いた検証の結果，h_f と h_l の間には，平均で 4.7%(Live Search)～63.7%(Yahoo!)の変動があることがわかった．また，Google, Live Search, Yahoo! JAPAN では 5 割から 6 割のクエリにおいて，検索エンジンから得られたヒット数は h_r の 100 倍以上大きな値を取ることがわかった．さらに，Yahoo!では検索エンジンから得られたヒット数と h_r が近い値を持つため，Yahoo!は，複数のクエリ間においてヒット数の大小関係を判定する用途に最も適していることがわかった．

今回の実験は，2008 年 11 月 10 日から 11 月 18 日にかけて収集をしたデータを対象に実験を行った．本論文で記載した傾向は，今後検索エンジンのシステムの仕様変更などによって変化をする可能性がある．

今後は h_f, h_l, h_r の関係から正しい検索結果数を予測する手法について検討を行う予定である．

[文献]

[1] A. Ntoulas and J. Cho: “Pruning policies for two-tiered inverted index with correctness guarantee”, Proc. of SIGIR'07, pp.191-198 (2007.7)

表 4. 2つのクエリ間における h_f, h_r の大小関係の入れ替わり発生率

	クエリ数	組み合わせ数	発生数	発生率
Google	410	83,845	23,015	27.4%
Live Search	374	69,751	27,124	38.9%
Yahoo! JAPAN	132	8,646	3,596	41.6%
Yahoo!	106	5,565	321	5.8%

[2] F. McCown and M. L. Nelson: “Agreeing to disagree: Search engines and their public interfaces”, Proc. of the 2007 Conf. on Digital Libraries, pp.309-318 (2007.6)
[3] Google: http://www.google.com
[4] G. Skobeltsyn, F. P. Junqueira, V. Plachouras and R. Baeza-Yates: “ResIn: A Combination of Result Caching and Index Pruning for High-performance Web Search Engines”, Proc. of SIGIR'08, pp.131-138 (2008.7)
[5] K. Bharat and A. Border: “A technique for measuring the relative size and overlap of public Web search engines”, Proc. of the 7th Int’l Conf. on WWW, pp.379-388 (1998.4)
[6] Live Search: http://www.live.com
[7] M. Lapata and F. Keller: “Web-based models for natural language processing”, ACM Trans. on Speech and Language Processing, Vol.2, No.1, pp.1-31 (2005.2)
[8] R. L. Cilibrasi and P. M. B. Vitanyi: “The Google similarity distance”, IEEE Trans. on Knowledge and Data Engineering, Vol.9, No.3, pp.370-383 (2007.3)
[9] Yahoo!: http://yahoo.com/
[10] Yahoo! JAPAN: http://www.yahoo.co.jp/
[11] Yahoo! Search BOSS API Guide: http://developer.yahoo.com/search/boss/boss_guide/
[12] Y. Matsuo, H. Tomobe and T. Nishimura: “Robust Estimation of Google Count for Social Network Extraction”, Proc. of 22nd Conf. on Artificial Intelligence (2007.7)
[13] 大鹿広憲，佐藤学，安藤進，山名早人：“Google を活用した英作文支援システムの構築”，DEWS2005，4B-i8 (2005.3)
[14] 徳永健伸：“情報検索と言語処理”，pp.39-41，東京大学出版会 (1999)
[15] 西田圭介: “Google を支える技術”，技術評論社(2008.3)
[16] 日本語形態素解析 Web サービス: http://developer.yahoo.co.jp/jlp/MAService/V1/parse.html
[17] 平野孝佳，平手勇宇，山名早人：“検索エンジンを用いた英文冠詞誤りの検出”，日本データベース学会 Letters, Vol.6，No.3，pp.1-4 (2006.9)

舟橋 卓也 Takuya FUNAHASHI
早稲田大学大学院基幹理工学研究科修士課程在学中．DBSJ 学生会員.

上田 高徳 Takanori UEDA
早稲田大学大学院基幹理工学研究科修士課程在学中．ACM，IEEE，IPSJ，IEICE，DBSJ 学生会員.

平手 勇宇 Yu HIRATE
2008 早稲田大学大学院理工学研究科博士後期課程修了．博士(工学)．2006 年より同大学メディアネットワークセンター助手．ACM，IPSJ，DBSJ 各会員.

山名 早人 Hayato YAMANA
1993 早稲田大学大学院理工学研究科博士後期課程修了．博士(工学)．1993-2000 電子技術総合研究所．2000 早稲田大学理工学部助教授．2005 同大学理工学術院教授，国立情報学研究所客員教授．IEEE，ACM，IEICE，IPSJ，DBSJ 各会員.