

プログラミング時の Web 検索行動に関する分析

中才 恵太郎¹ 角田 雅照^{1, a)}

インターネット上には、プログラミングを行う際に有用な資料が多く存在しており、プログラミング時に検索エンジンを活用して資料を参照することにより、作業効率を高めることができる。検索エンジンを用いることにより、プログラミングのライブラリの名称を正確に記憶していない場合でも、その使い方を知ることができたり、バグが発生した場合に、その原因のヒントを知ることができる。ただし、検索行動には個人による差異があるため、Web 検索を行い、提示されたすべてのページが、必ずしもプログラミングに有用なページとはならない。本研究では、Web 検索に不慣れたプログラミング初心者に対し、検索方法の指針を与えることにより、プログラミングの効率を高めることを目的とする。そのために、プログラミングの熟練者の検索行動を分析し、そこから熟練者の検索方法の指針を得ることを試みる。実験では被験者に対し、プログラミングに関する出題を行い、それに対してプログラミングにより解答してもらった。その結果、一度使用した検索のキーワードを含むキーフレーズを再び用いることはプログラミング時の検索行動としては効率が低い可能性があることなどがわかった。

1. はじめに

インターネット上には、プログラミングを行う際に有用な資料が多く存在しており、プログラミング時に検索エンジンを活用して資料を参照することにより、作業効率を高めることができる。プログラミング言語の公式リファレンスの多くはインターネット上に公開されている。その他に、プログラミングに有用なサイトとして、プログラミングに関する Q&A サイト、プログラミング言語入門のサイト、プログラミング言語の使用方法が記載されたブログなどがあげられる。Google に代表される Web 検索エンジンで検索を行うことにより、これらの Web サイトの URL (Uniform Resource Locator) を知らなくとも、これらの Web サイトを閲覧することができる。また、プログラミングに特化した検索エンジンも公開されている[1]。

検索エンジンを用いることにより、プログラミングのライブラリの名称を正確に記憶していない場合でも、その使い方を知ることができたり、バグが発生した場合に、その原因のヒントを知ることができる。これは、検索エンジンはウェブサイト上の語句、ページの新しさやページランク [6] などを利用したアルゴリズムにより、適切な Web サイトをユーザに提示するように構築されているためであり、情報が古くて役に立たない Web ページや、重要でない Web ページはある程度検索結果から除外される。また、検索エンジンにはオートコンプリートや関連するキーワードを表示する機能があるため、もし必要な情報のための検索キーワードが曖昧だとしても、必要な情報にたどり着くことができる。

ただし、検索行動には個人による差異があるため、Web 検索を行い、提示されたすべてのページが、必ずしもプログラミングに有用なページとはならない。有用なページを参照するためには、ある程度の適切な検索キーワードを選択する能力、検索結果から必要な情報を取捨選択する能力、関連するキーワードの中から更に必要なキーワードを想起

する能力などが必要となる。これらの能力は経験や検索対象に対する背景知識の影響が大きい。

本研究では、プログラミングに関連した検索を行う際、プログラミングの初心者は、プログラミングの熟練者よりも有用な検索結果に到達する能力が高くないと仮定する。プログラミングの熟練者は、プログラミングに対する検索キーワードの想起や、検索結果から必要なページを取捨選択することを、初心者よりも数多く実行している。また、プログラミングの熟練者は、プログラミングに関する知識が多いため、検索結果から必要な情報を取捨選択することが、初心者と比べて容易に行うことができる。逆に初心者の場合、プログラミングに関する技術用語や知識が不足しているため、情報の取捨選択が容易ではないと考えられる。

Web 検索により適切な情報を得られない場合、プログラマは初めて対処する問題に関して、作業効率が低下すると考えられる。これは、初めて対処する問題であるために、これまでに習得している知識だけでは十分に対応できない可能性があるためである。逆に適切な情報を得られた場合、そのような問題でも比較的高い作業効率で解決できる可能性がある。Web ページは速報性が高いため、類似の問題に対する解決方法がすでに Web ページに公開されている可能性があり、そのページを参照することができれば、比較的容易に問題を解決できると考えられる。

本研究では、Web 検索に不慣れたプログラミング初心者に対し、検索方法の指針を与えることにより、プログラミングの効率を高めることを目的とする。そのために、プログラミングの熟練者の検索行動を分析し、そこから熟練者の検索方法の指針を得ることを試みる。実験では被験者に対し、プログラミングに関する出題を行い、それに対してプログラミングにより解答してもらった。その際の検索行動を計測しておき、アンケート、解答の正誤、解答時間によってグルーピングを行い、結果を分析した。Google は検索のヒント[3]を公開しており、プログラム熟練者の検索行動と類似している可能性がある。ただし、これは一般的な事項に対する検索のヒントであり、高度な技術が要求されるプログラミングにおいてもそのまま適応できるかどうか

¹ 近畿大学
Kindai University, Japan
a) tsunoda@info.kindai.ac.jp

は明らかではない。

2. 実験方法

本研究では、プログラミング時の初心者と熟練者の検索行動を分析するために、プログラミングに関する問題を作成した。また、問題解答プログラム（被験者の解答を確認し、正解の場合に次の問題を表示する）を作成して実験環境を構築した。その後、問題を被験者に解答させ、アンケートを実施した（図 1）。実験後に、プログラミング時の検索行動を、アンケート、問題の正誤、解答時間に基づいて分析した。以降では、実験環境、問題解答プログラム、被験者、問題、分析項目、グルーピングのそれぞれについて詳細を述べる。

2.1 実験環境

本実験では Windows7 のノート型パソコンを使用した。マウスはノートパソコンに付属していた一般的なものを利用し、実験では、被験者にトラックパッドとマウスのどちらを利用するかを任意で選ばせた。

表 1 に実験に用いたソフトウェアを示す。検索時の行動を計測するために、キー入力やアクティブにしたウィンドウタイトルなどを記憶するきいろがぁ[4]を用い、実験時のパソコンの画面を録画するためにアマレココ[1]を用いた。プログラミングの問題を解く環境として Java 言語の統合開発環境である Eclipse[2]を用い、各ソフトの使用時間などを取得するために ManicTime[6]を利用した。ブラウザは Google Chrome に統一し、検索エンジンは Google を指定した。分析を容易にするために全被験者に同じブラウザと検索エンジンを使用させた。解答時間と問題の正答を記録するため問題解答プログラムを Java 言語で作成し、Eclipse 上で動作するようにした。

情報系学科に所属している学部生 9 人と修士学生 1 人の計 10 人を被験者とした。被験者は Java 言語についてある程度学習しており、基本的な文法などは理解している。また、被験者は Eclipse を開発環境として全員使用した経験がある。

2.2 問題

被験者全員が使える言語である Java 言語を題材とした問題を作成した。また、開発環境も被験者全員が使用したことがある Eclipse を指定した。問題は全部で 4 問とした。以下において、問題 1 から問題 4 について説明する。

- 問題 1: Java では、小数点の計算において誤差が生じ

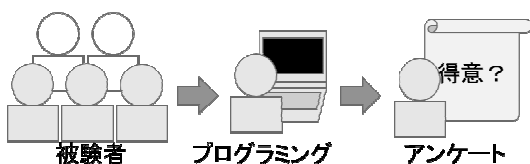


図 1 実験の概要

表 1 実験に使用したソフトウェア

ソフトウェア	用途
きいろがぁ	キー入力とウィンドウタイトルを取得するため
Eclipse	Java 言語の統合開発環境
アマレココ	パソコンの画面を録画するため
ManicTime	各ソフトの使用時間を把握するため
Google Chrome	Web ブラウザ

る場合がある。そこで、Java で正確な計算を行うためのライブラリを使い、正しい計算を行う問題を作成した[5]。適切なライブラリを探す必要がある問題である。図 2 に被験者に出題した実際の問題を示す。

- 問題 2: ArrayList を使ったデータ構造で実装されたプログラムから、連想配列を使ったデータ構造に書き換える問題である、ただし、連想配列を直接使うとは問題文には記載しておらず、どのクラスを使うのかを推測する必要がある問題である。
- 問題 3: ある URL からデータを取得した際に、文字が表示されずに文字コードが表示されているプログラムがある。これを文字が表示するように書き換える問題である。どこが問題点なのかを発見し、検索により解決方法を発見する問題である。
- 問題 4: 正しい例外のクラスを指定して、条件文により場合分けする問題であり、どのクラスが例外を出すのかを調べる必要がある。

いずれの問題も、検索を適切に利用することができれば容易に正解に到達できるように問題を作成した。被験者の実験に対する慣れや実験時間の制限を考慮し、前半の問題は難易度を低くし、後半の問題は難易度が高くなるように順序を設定した。

解答時間は指定せず無制限とした。なお、問題解答プログラムには、各問題で解答時間が 20 分以上経過している場合、問題をスキップすることができる機能を持たせた。この機能は、これ以上調べても問題を正解することは不可能だと被験者が考えた時のみ、使用を許可した。また、問題を正解した場合、スキップした場合のどちらも、現在解答している問題以外の閲覧、解答はできないように解答プログラムで制御した。これは、現在取り組んでいる問題における経過時間や検索回数などをできるだけ正確に計測するためである。

2.3 分析項目

被験者に対する実験を行った後に、きいろがぁ、解答時間を記録したデータ、Google Chrome の閲覧履歴などから、以下の 5 項目を定義し、それぞれについて集計を行った。

```
1万1000円の買い物をして2万円を支払いました。

現在のプログラムでは
正しく計算することは出来ません
結果は
お釣りは0.8999999999999999万円
となります。

0.90万円と表示するように変えてください
```

(a) 問題文

```
double answer = 2.00 - 1.10;

System.out.println("お釣りは" + answer + "万円");
```

(b) 問題

```
BigDecimal answer = new BigDecimal("2.00");

subtract(new BigDecimal("1.10"));

System.out.println("お釣りは" + answer + "万円");
```

(c) 解答例

図 2 被験者に出題した問題例

2.3.1 検索結果ページ数÷表示した Web ページ数

この項目を定義した目的は、検索結果ページから、結果に示されている各ページにアクセスせず、タイトルやスニペット (Web ページの説明文やタイトル) だけで検索キーワードを変えるかどうかなど判断をしたほうがよいのか、逆に、検索結果ページから実際に各ページへアクセスして内容を確認したほうがよいのかを分析するためである。

分母はブラウザで表示した全 Web ページ数を示し、分子は、そのうちの Google 検索の結果ページの数を示す。この項目の値が高い場合、検索結果から別のページを参照した回数が少ないことを示し、逆に値が低い場合、検索結果から別のページを参照した回数が多いことを示す。

2.3.2 キーフレーズの種類÷検索結果ページ数

キーフレーズの種類とは、検索時に入力された全ての検索キーフレーズから、重複を取り除いた数を指す。この項目の値が高い場合、被験者は検索キーフレーズを頻繁に変更していることを示し、検索結果ページは 2 から 3 ページ程度まで確認されていないことになる。逆に、値が低い場合、被験者は検索結果ページを、数ページ分確認していることになる。なお、同一のキーフレーズで検索を何度か行

ったり、ブラウザの戻るボタンを多用した場合も、この値が低くなる。

2.3.3 キーワードの種類÷キーフレーズの種類

Google が公開している検索のヒント[3]では、検索のキーワードは少ないほうがよいと指摘されている。この方針がプログラミングにおいても適切であるかどうかを確かめるため、この項目を定義した。この項目の値が高い場合、毎回複数個のキーワードを並べて検索していることを示し、逆に、値が低い場合、少ないキーワードで検索していることを示す。

2.3.4 検索結果ページ数÷問題解答時間

この項目の値が高い場合、問題に解答している時間や検索結果以外のページを閲覧するよりも、Web 検索を多く行っていることを示す。逆に、値が低い場合、検索結果を詳細に閲覧していることがわかる。

2.3.5 キーワードの種類÷キーワードの合計

この項目を定義した理由は、検索効率を高めるためには、類似したキーワードで検索したほうがよいのか、もしくは毎回キーワードを大きく変更したほうが変えたほうがよいのかを分析するためである。

キーワードの種類とは、検索におけるキーフレーズをキーワードに分けて、重複しているものを除いた数である。キーワードの合計とは、上記の重複を許して合計したものである。この項目の値が高い場合、同じキーワードを何度も別のキーフレーズ中で用いず、キーワードを新たに考案していることが多いことを示す。逆に値が低い場合、同じキーワードを各キーフレーズ中で多用していることを示す。

2.4 被験者のグルーピング

分析では、プログラミング熟練者と初心者の検索行動の差異を定量的に分析するために、熟練者と初心者でグルーピングを行い、2.3 節において定義した 5 つの項目について、差異を確かめた。グルーピングを行うために、被験者に対し、実験実施後にアンケートを行い、プログラミングが得意であるか (すなわち熟練者であるか) 不得意であるか (すなわち初心者であるか) を解答してもらった。ただし、自己申告でプログラミングを得意であると回答している場合でも、実際に得意であるかどうかは確かではない。実際にプログラミングが得意であるかどうかを判断するためには、各被験者の問題の正答率や解答時間を考慮する必要がある。

そこで、自己申告による得意不得意に基づいたグルーピングに加えて、2 つの観点からグルーピングを行い、それらに基づいてプログラミング熟練者の検索行動に関する分析を行う。ひとつのグルーピング方法は、正解者と不正解者に基づいた方法である。問題を正解している場合、被験者が適切な検索行動、すなわち熟練者と類似した検索行動を行ったために、正解することができたのみならず、これに従いグルーピングした。もう一つのグルーピング方法は、

問題を正解し、かつその問題に対する解答時間が上位5位以内の被験者とそれ以外の被験者をグルーピングした。正解した場合でも解答時間が長い場合、その被験者の検索は不適切であるとみなし、これに従いグルーピングを行った。

3. 実験結果

3.1 問題の正誤とアンケート結果

各問題の正答の結果とアンケート結果を表2から表4に示す。被験者10人をアルファベットのAからJで示す。各被験者における、各問題の正誤を表2に示す。○が正解を表し、×が不正解を表す。B, D, Hの3人の被験者が全問正解し、Fの被験者は全問不正解であった。問題2は正解者が8人であるのに対し、問題3, 4は正解者が6人であった。よって、問題2は比較的簡単な問題であり、問題3, 4は難しい問題であるといえる。問題1については正解者が7人であり標準的な難易度の問題である考えられる。このことから、出題順序は比較的適切であった（前半の問題は難易度が低く、後半の問題は難易度が高い）といえる。

表3は各問題に対する、各被験者の解答時間を示す。数値の単位は分であり、各被験者が各問題に取り組んだ時間を表す。被験者によっては実験時間が非常に長くなり、例えばJの被験者の合計解答時間は約210分となった。そのため、トイレ休憩などの休憩を取ることを許可した。このときの休憩時間を解答時間から除くため、2分30秒間パソコンの操作が加えられなかった場合、解答時間からその時間を減じる処置を行った。

正解者は不正解者の解答時間より解答時間が短い傾向がある。これは当然の傾向ではあるが、各問題の開始から20分以上経過しないと問題をスキップすることができないため、不正解者の解答時間が長くなる。逆に、問題を読んで即座に解答を推定できる被験者が存在する場合、正解者の解答時間が短くなる。例外として、被験者Bのこの問題2と3に対する解答時間は、正解している者のなかでは長い。実験中に被験者Bから2回質問があり、問題をもう一度読むように答えると、被験者Bはすぐに正答することができた。すなわち、被験者Bは当初問題の意図を正しく理解していなかったが、その後問題の意図を理解し、解答することができた。ただし、正しく理解するまでに多くの時間を消費したため、解答時間が長くなったと考えられる。

各被験者に対し、解凍後にアンケート形式により、プログラミングの経験年数とプログラミングが得意かどうかを質問した。その結果を表4に示す。表の「年数」はプログラミングの経験年数を示し、「得意」はプログラミングが得意なのかどうかを示している。プログラミングが得意であると答えた被験者のうち、全問正解者は3人中2人であり、残りの1人も4問中3問正解していた。また、解答時間も被験者Iの問題3, 4以外は比較的短い。よって、アンケートの自己申告によるものであるが、回答は比較的信頼性が高いと考えられる。

3.2 各項目に対する分析結果

2.3節で定義した項目を、プログラミングを得意と回答

表2 各被験者における各問題の正誤

	A	B	C	D	E	F	G	H	I	J
問題1	○	○	×	○	×	×	○	○	○	○
問題2	○	○	○	○	○	×	×	○	○	○
問題3	×	○	×	○	○	×	○	○	○	×
問題4	×	○	○	○	○	×	○	○	×	×

表3 各被験者における各問題の解答時間

	A	B	C	D	E	F	G	H	I	J
問題1	19.13	28.08	41.12	10.70	41.57	30.73	32.22	6.42	13.85	30.70
問題2	26.72	46.43	25.97	4.58	18.83	29.60	62.37	5.18	18.85	18.53
問題3	26.18	67.18	20.82	12.22	8.27	31.00	23.73	10.55	30.85	96.68
問題4	29.48	18.78	20.60	2.55	3.32	21.97	12.55	14.33	59.68	64.63

表4 各被験者に対するアンケート結果

	A	B	C	D	E	F	G	H	I	J
年数	3	3	3	5	3	3	3	4	3	1
得意	NO	NO	NO	YES	NO	NO	NO	YES	YES	NO

したかどうか、解答時間の長短、問題の正誤によりグルーピングをした結果を示す。表 5 は各項目の平均値を示し、表 6 は各項目の中央値を示す。

3.2.1 検索結果ページ数÷表示した Web ページ数

検索結果ページ数÷表示した Web ページ数の平均値を見ると、得意、解答時間短、正答グループのいずれの場合も、不得意、解答時間長、誤答グループよりも値が低かった。中央値においても同様の結果となった。正答と誤答のグループでは、中央値の差は 0.1 しかなかったが、得意と不得意、解答時間短と解答時間長のグループ間では、平均値で 0.1 以上、中央値で 0.08 以上の差があった。このことから、プログラミング熟練者の場合、この値が低くなる可能性があると考えられる。

この項目の値は、検索結果のページからそれ以外のページを多く参照した時や、検索結果以外のページから他のページにアクセスした場合にも低くなる。すなわち、スニペットだけ、もしくは Web ページのタイトルを参照しただけで検索結果が違っていると判断し、すぐに別の検索キーワードを試すことは適切ではない可能性がある。一見調べようとしている内容と無関係に思える検索結果の場合でも、念のためそれらの Web ページにアクセスし、内容を確認したほうが、プログラミングの作業効率が高まる可能性がある。

3.2.2 キーフレーズの種類÷検索結果ページ数

キーフレーズの種類÷検索結果ページ数の平均値は、得意、解答時間短、正答のいずれのグループも、不得意、解答時間長、誤答のグループよりも低かった。中央値でも同様の結果となった。得意と不得意のグループの平均値の差

と、正解者と不得意のグループの差は 0.1 以上あるのに対し、解答時間短と解答時間長のグループの平均の差は 0.06 とあまり大きくなかった。中央値に着目すると、得意と不得意グループ、解答時間短と解答時間長グループ、正答と誤答グループの差はそれぞれ、0.12, 0.11, 0.13 となった。よって、プログラミングの熟練者の場合、この項目の値が比較的高くなる可能性がある。

この値が高い場合、検索する際に検索キーワードを頻繁に変更している、また、検索結果ページの 2 ページ目や 3 ページ目はあまり確認していないことを示している。このことから、検索結果の 1 ページ目を精読し、その後別のキーワードを考えて検索するほうが、検索結果の 2 ページ目以降を確認するよりもプログラミングの作業効率が高い可能性がある。

3.2.3 キーワードの種類÷キーフレーズの種類

キーワードの種類÷キーフレーズの種類は、得意、解答時間短、正答のいずれのグループも、不得意、解答時間長、誤答のグループよりも値が低かった。中央値でも同様の結果となった。

しかし、得意と不得意のグループにおいて、平均の差は 1.09 であったのに対し、解答時間短と解答時間長のグループの差は 0.4 であり、あまり大きくなかった。さらに、正答と誤答の差も 0.12 であり、あまり大きくなかった。中央値に関しても、得意と不得意のグループ間の差が大きく、その他のグループ間では差が大きくなかった。正答している場合でも熟練者であるとは限らず、その他のグループ間ではこの項目の値の差が小さかったことから、今回の実験

表 5 各問題における各項目の平均値

	全体	得意	不得意	解答時間 短	解答時間 長	正答	誤答
検索結果ページ数÷表示した Web ページ数	0.53	0.44	0.57	0.48	0.58	0.51	0.57
キーフレーズの種類÷検索結果ページ数	0.48	0.58	0.44	0.51	0.45	0.51	0.41
キーワードの種類÷キーフレーズの種類	2.44	1.80	2.71	2.24	2.64	2.40	2.52
検索結果ページ数÷問題解答時間	0.73	0.48	0.84	0.55	0.91	0.69	0.82
キーワードの種類÷キーワードの合計	0.59	0.78	0.52	0.72	0.47	0.63	0.50

表 6 各問題における各項目の中央値

	全体	得意	不得意	解答時間 短	解答時間 長	正答	誤答
検索結果ページ数÷表示した Web ページ数	0.56	0.47	0.59	0.50	0.58	0.55	0.56
キーフレーズの種類÷検索結果ページ数	0.45	0.52	0.40	0.50	0.39	0.50	0.37
キーワードの種類÷キーフレーズの種類	2.38	1.83	2.77	2.31	2.44	2.33	2.38
検索結果ページ数÷問題解答時間	0.61	0.46	0.69	0.48	0.73	0.59	0.69
キーワードの種類÷キーワードの合計	0.53	0.75	0.46	0.67	0.43	0.58	0.44

結果からは、この項目とプログラミングの作業効率とに関連があるとはいえなかった。

この項目の値が低い場合、少ないキーワードで検索していることを示すが、Google の検索のヒントにおいても、検索キーワード数は少ないほうがよい[3]と指摘されている。プログラミングにおける検索の場合も、キーワード数を少なくすることにより効率が高まる可能性があるが、今回の実験結果からは検索キーワード数は少ないほうがよいと結論付けることはできなかった。

3.2.4 検索結果ページ数÷問題解答時間

検索結果ページ数÷問題解答時間の平均値は、得意、解答時間短、正答グループのいずれも、不得意、解答時間長、誤答グループより値が小さかった。中央値も同様の結果となった。正答と誤答グループの平均値と中央値の差異は0.1ほどで他のグループより差が少ない。ただし、得意と不得意グループの差と、解答時間短と解答時間長グループの差は0.36であることから、プログラミングの熟練者はこの項目の値が低い傾向にあると考えられる。

この項目の値が小さい場合、問題を解くために検索を多数繰り返すことをしていないことを示す。すなわち、熟練者は、検索結果や検索結果のリンクをたどった先のページを熟読している、または、検索キーフレーズを熟慮している可能性があり、そうすることによりプログラミングの作業効率が高まる可能性がある。

3.2.5 キーワードの種類÷キーワードの合計

キーワードの種類÷キーワードの合計の平均値は、得意、解答時間短、正答グループのいずれも、不得意、解答時間長、誤答グループよりも値が大きかった。中央値も同様の結果となった。なお、正答と誤答グループ間の差は、他のグループよりも小さかったが、得意と解答時間短のグループは不得意、解答時間長のグループよりもこの値が大きいことから、プログラミング熟練者はこの値が高い傾向にある可能性がある。

この項目の値が高い場合、同じ検索のキーワードを多用していないことを示している。すなわち、一度使用した検索のキーワードを含むキーフレーズを再び用いることはプログラミング時の検索行動としては効率が低い可能性がある。プログラミング時には、毎回新たなキーワードを用いることを考慮して検索を行うことにより、効率が高い検索ができる可能性がある。

4. おわりに

本研究では、プログラミング熟練者と初心者の検索行動の差異を明らかにするために、熟練者と初心者の検索行動を定量的に計測する実験を行った。その際、実験結果やアンケート結果に基づき、プログラミング熟練者と初心者をグルーピングし、5つの項目に関してグループ間の差異を分析した。その結果、以下の傾向が見られた。

- スニペットだけ、もしくは Web ページのタイトルを参照しただけで検索結果が違くと判断し、すぐに別の検索キーワードを試すことは適切ではない可能性がある。
- 検索結果の1ページ目を精読し、その後別のキーワードを考えて検索するほうが、検索結果の2ページ目以降を確認するよりもプログラミングの作業効率が高い可能性がある。
- 熟練者は、検索結果や検索結果のリンクをたどった先のページを熟読している、または、検索キーフレーズを熟慮している可能性があり、それによりプログラミングの作業効率が高まる可能性がある。
- 一度使用した検索のキーワードを含むキーフレーズを再び用いることはプログラミング時の検索行動としては効率が低い可能性がある。

今後の課題は、被験者に出題する内容をさらに検討することと、分析に用いる項目を新たに定義し、熟練者と初心者の間で、それらの項目の値が異なるかどうかを分析し、最適な検索行動についてさらに検討することである。

謝辞 本研究の一部は、文部科学省科学研究補助費（挑戦的萌芽:課題番号 26540029, 基盤 C:課題 番号 25330090）による助成を受けた。

参考文献

- [1] アマレココ, <http://www.amarectv.com/>
- [2] Eclipse, <http://www.eclipse.org/>
- [3] Google: 検索のヒント - 検索サービス, <https://www.google.com/intl/ja/insidesearch/tipstricks/basics.html>
- [4] きいろがあ, sword, <http://keylog.web.fc2.com/keyfree/keyfree.html>
- [5] ジョシュア・ブロック, ニール・ガフター: 罫, 落とし穴, コーナーケース Java PUZZLERS, ピアソン・エデュケーション(2005).
- [6] ManicTime, <http://www.amarectv.com/>
- [7] Page, L, and Brin, S.: The anatomy of a large-scale hypertextual web search engine, *Proc. World-Wide Web Conference* (1998).
- [8] Runnable, <http://runnable.com/>