

筑波大学 情報学群 情報メディア創成学類

卒業研究論文

クエリ依存型リンク解析手法における  
スコア近似に関する研究

佐藤 豪

指導教員 古瀬 一隆 陳 漢雄

2016年1月

## 概要

私たちが普段使用している WEB ページの検索エンジンに用いられている手法の 1 つに、リンク構造に基づいて得点を与える SALSA アルゴリズムがある。SALSA アルゴリズムはクエリ依存型のリンク解析手法であり、よりクエリに適した検索結果を得ることができる。一方で、クエリが与えられてから WEB グラフの抽出を行いリンク構造を取得し、各ページのスコア計算を行うため多くの応答時間がかかってしまう。

先行研究では、SALSA アルゴリズムの一部をクエリが与えられる前に処理することで高速化を図る手法が提案された。従来の SALSA アルゴリズムに比べて高速化することに成功したものの、ランキング精度においては著しく低い結果となり、クエリに適した検索結果が得られるというクエリ依存型リンク解析手法の長所が失われてしまった。そこで、本論文では先行研究の手法を改善し、ランキング精度を保ちつつ、高速化を行う手法を提案する。

本研究では、クエリ依存型リンク解析手法 SALSA において、クエリに適した検索結果が得られるという長所を保ちつつ、検索の高速化を行うことを目的とする。

# 目次

<b>第1章 序論</b>	<b>1</b>
<b>第2章 関連研究</b>	<b>3</b>
2.1 WEB ランキングアルゴリズムの歴史 . . . . .	3
2.2 PageRank . . . . .	4
2.3 HITS . . . . .	5
2.4 SALSA . . . . .	5
<b>第3章 提案手法</b>	<b>6</b>
3.1 クエリ依存型リンク解析手法高速化の概要 . . . . .	6
3.2 提案手法の概要 . . . . .	6
3.3 クラスタリングアルゴリズム . . . . .	6
3.4 スコアの近似 . . . . .	6
<b>第4章 実験と考察</b>	<b>7</b>
4.1 評価指標 . . . . .	7
4.2 実データでの実験 . . . . .	7
4.3 大規模データでの実験 . . . . .	7
<b>第5章 まとめ</b>	<b>8</b>
<b>第6章 形式</b>	<b>9</b>
6.1 表紙 . . . . .	9
6.2 本体 . . . . .	9
<b>謝辞</b>	<b>11</b>
<b>参考文献</b>	<b>12</b>

## 図 目 次

2.1	WEB グラフ . . . . .	3
2.2	PageRank の概念図 . . . . .	4
6.1	図の例 . . . . .	10

# 第1章 序論

世界中に 10 億件以上存在する WEB サイトの中から、必要な情報を手探りで探すことは困難である。そのような場合、検索エンジンを用いることが有効である。検索エンジンは、ユーザーが必要とする情報に関連した検索キーワードを受け取り、そのキーワードに関連する WEB ページの集合を抽出し、検索結果としてユーザーに提示する。検索結果として提示される WEB ページの集合は、検索キーワードとの関連度が高く、ユーザーにとって役に立つ WEB ページが上位に表示されることが望ましい。このように、ある検索ワードに関連する WEB ページの集合に対して、ユーザーに提示するための順位付けを行う手法を WEB ランキングアルゴリズムと呼ぶ。

WEB ランキングアルゴリズムは大きく分けて 2 種類ある。1 つ目は、WEB ページのテキストや HTML 構造を解析することで内容得点を計算し、順位付けを行う手法である、2 つ目は、WEB ページ間のリンク構造に基づいてスコア計算を行い、順位付けを行うリンク解析手法である。本論文では後者のリンク解析手法を扱うものとする。

リンク解析手法のアルゴリズムで有名なものとして PageRank アルゴリズムや HITS アルゴリズムが挙げられる。さらに近年の研究では、PageRank アルゴリズムと HITS アルゴリズムの長所を取り入れた SALSA アルゴリズムと呼ばれるリンク解析手法が、ランキング精度において他の WEB ランキングアルゴリズムより優れているという報告がされている。

PageRank アルゴリズムは、検索キーワードに依存せずスコア計算を行い、順位付けをすることができる。このようなリンク解析手法をクエリ独立型と呼ぶ。一方で HITS アルゴリズムや SALSA アルゴリズムは、検索キーワードが与えられてからスコア計算と順位付けを行う。このようなリンク解析手法をクエリ依存型と呼ぶ。したがって、これらのアルゴリズムを検索エンジンで用いる場合、検索キーワードが与えられる前にスコア計算を行う PageRank アルゴリズムは高速に応答することができるのに対し、検索キーワードが与えられてからスコア計算を行う HITS アルゴリズムや SALSA アルゴリズムは応答時に無視できないほどの時間がかかってしまう。

この問題の解決策として、クエリ依存型リンク解析手法において時間のかかるスコア計算を前処理化することで、応答時間を減らすという手法が過去に提案された。これらの手法では、検索キーワードが与えられる前にあらかじめ WEB ページのクラスタリングを行い、クラスタ内での各ページのスコア計算を行う。検索キーワードが与えられた後は、そのクラスタとスコアを用いて結果を近似し、WEB ランキングを作成する。しかし、これらの手法では近似したスコアと本来の SALSA アルゴリズムによるスコアが大きく異なるため、応答時間は早くなるがランキング精度が下がる結果となった。

そこで本研究では、クエリ依存型リンク解析手法 SALSA の前処理化におけるクラスタリングとスコア近似を改善することで、従来の SALSA アルゴリズムにより近い WEB ランキングを作成することができ、かつ応答時間の少ない手法を提案する。

## 第2章 関連研究

本章では、クエリ依存型リンク解析手法 SALSA に関する技術について、特に関連のある研究について紹介する。

### 2.1 WEB ランキングアルゴリズムの歴史

巨大化した WEB の中から目的の WEB ページを見つけるには、検索エンジンを使うのが有効である。検索エンジンに与えた検索キーワード、別名クエリと関連のあるページに対して順位付けをするのが WEB ランキングアルゴリズムである。

1998 年まで WEB ランキングアルゴリズムは、WEB ページのテキストや HTML 構造といった内容得点で順位付けを行っていた。この内容得点というのは、各 WEB ページにどこにどれほどクエリの文字列が含まれているのか計算したものであった。しかし、WEB が巨大化し限りなく増大していったことや、内容得点がスパムの影響を受けやすかったことから、1998 年までには従来の内容得点は有効でないことが明らかになっていた。

そこで、内容得点だけではなく、WEB のハイパーリンク構造に対して有向グラフを作成し、その有向グラフを用いて WEB ページの人気得点を計算するリンク解析手法が登場した。これによって従来の内容得点によるランキングよりも品質が向上し、検索エンジンの利用者は劇的に増え、リンク解析手法は WEB ランキングアルゴリズムの主流となった。

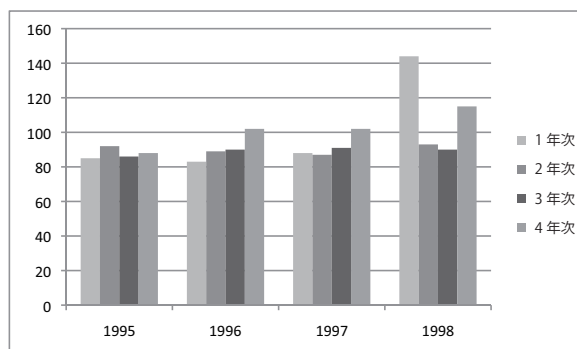


図 2.1: WEB グラフ

## 2.2 PageRank

PageRank は、当時 Stanford University に在学していた Sergey Brin と Larry Page の 2 人の計算機科学科の学生によって開発されたものであり、開発者の名前がアルゴリズムの由来となっている。これがのちの WEB 検索エンジン Google であり、このアルゴリズムの基本的な概念は「多くの良質なページからリンクされているページは、やはり良質なページである」という再帰的な関係をもとに、全てのページの重要度を判定したものである。

以下の図 2.2 は PageRank の概念図である。あるページのスコアを、そのページに存在する出リンク数で割った数が、それぞれの被リンク先のスコアに加算されるという関係になっている。

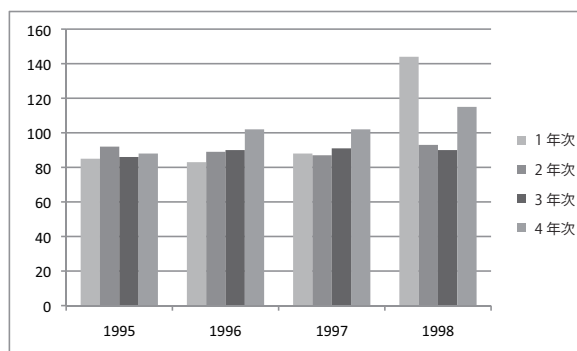


図 2.2: PageRank の概念図

PageRank は、WEB グラフにランダムウォークモデルを適応し、各ページへの遷移確率によって WEB ランキングを作成する。あるページからそのページが指している全リンク先へとランダムに遷移する確率と全ての WEB ページへとランダムに遷移する確率を足したスコアが PageRank のスコアになる。

PageRank のスコア計算は以下の式 (2.1) で定義される。

$$\pi^T = \pi^T(\alpha S + (1 - \alpha)E) \quad (2.1)$$

ここで  $\pi^T$  は PageRank ベクトルであり、 $\alpha$  は 0.85 程度のスカラー値であり、行列  $S$  はページ  $i$  からそのリンク先であるページ  $j$  への遷移確率行列である。また行列  $E$  は全ての WEB ページへのレポーターション行列である。

ここで行列  $S$  は、ページ  $i$  からそのリンク先ページ  $j$  への遷移確率を示すので以下のよう

$$S(i, j) = \begin{cases} 1/out(i) & (out(i) \text{ はノード } i \text{ の出リンク数}) \\ 0 & (\text{ノード } i \text{ からノード } j \text{ へのリンクがないとき}) \end{cases} \quad (2.2)$$



また、行列  $E$  はページ  $i$  から全てのページへのランダムな遷移を示すので、以下 (2.3) のように定義することができる。

$$E_{(i,j)} = 1 \quad (2.3)$$

PageRank の特徴の 1 つとして、クエリ独立型のリンク解析手法であるという点が挙げられる。これは、クエリに関係なく、最初から全ての WEB ページに対する WEB ランキングが決定しているということである。クエリが与えられてからは、既に作成されている WEB ランキングに適切なフィルタリングを施し、検索結果として WEB ランキングを表示するだけである。このため PageRank は、HITS や SALSA のようなクエリ依存型のリンク解析手法に比べ、応答時間が短い。

## 2.3 HITS

## 2.4 SALSA

## 第3章 提案手法

てすです

3.1 クエリ依存型リンク解析手法高速化の概要

3.2 提案手法の概要

3.3 クラスタリングアルゴリズム

3.4 スコアの近似

## 第4章 実験と考察

てすです

### 4.1 評価指標

### 4.2 実データでの実験

### 4.3 大規模データでの実験

## 第5章 まとめ

てすです

## 第6章 形式

ここでは、論文の表紙および本体の記述方法について述べる。

### 6.1 表紙

表紙は、`\maketitle` によって作成するため、以下の項目に相当する文字列をそれぞれ記述する。

題目: 題目は `\title` に記述する。行替えを行う場合は `\` を入力する。ただし、題目の最後に `\` を入力するとコンパイルが通らなくなるので注意する。なお、4 行以上の題目の場合、表紙ページがあふれるためスタイルファイル “`mast-jp-sjis.sty`” を変更する必要がある。

著者名: 著者名は `\author` に記述する。

指導教員名: 指導教員は `\advisor` に記述する。

年月: 年月は `\yearandmonth` に記述する。年月は提出時のものを記述すること。

### 6.2 本体

本体は 1 段組で記述する。

図表には番号と説明 (caption) を付け、文章中で参照する。表 6.1 と図 6.1 はそれぞれ表と図の例である。表の説明は上に、図の説明は下にかくことが多い。図の挿入に用いるパッケージについては使用環境に合わせて自由に選択してほしい。

表 6.1: 表の例

年 度	1 年次	2 年次	3 年次	4 年次
1995	85	92	86	88
1996	83	89	90	102
1997	88	87	91	112
1998	144	93	90	115

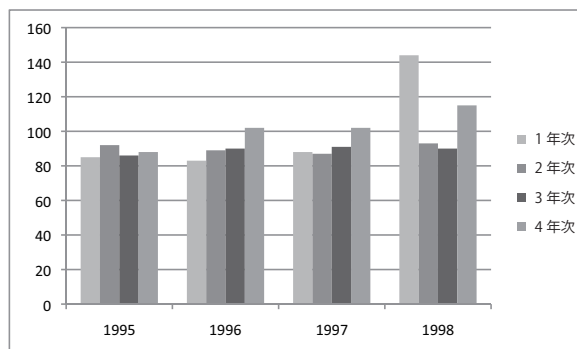


図 6.1: 図の例

詳しくは参考書 [1, 2]などを参照のこと。また、奥村晴彦氏の「 $\text{\TeX}$ Wiki」<http://oku.edu.mie-u.ac.jp/~okumura/texwiki/>は、日本語の $\text{\TeX}$ に関する情報が充実している。また、具体的な論文としての文献参照例として(本の例)[3]、(雑誌論文の例)[4]、(予稿集の例)[5]を挙げておく。

# 謝辭

感謝

## 参考文献

- [1] 奥村晴彦, LaTeX2e 美文書作成入門 改訂第 5 版, 技術評論社, 2010 年.
- [2] 吉永徹美, LaTeX2 ε 辞典, 翔泳社, 2009 年.
- [3] Colin Ware, Information Visualization — Perception for Design, Second Edition, Morgan Kaufmann Publishers, 486 p., 2004.
- [4] Miriah Meyer and Tamara Munzner, MizBee: A Multiscale Synteny Browser, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 15, No. 6, pp. 897–904, 2009.
- [5] Emerson Murphy-Hill and Andrew P. Plack, An Interactive Ambient Visualization for Code Smells, in *Proceedings of the 2010 International Symposium on Software Visualization (SOFT-VIS '10)*, pp. 5–14, 2010.