# 論文 / 著書情報
# Article / Book Information

| | |
|---|---|
| 論題(和文) | 機関リポジトリと外部情報源を連携した関連論文探索の実現 |
| Title(English) | Searching related papers from institution ’ s repository using external information sources |
| 著者(和文) | NGUYENMANH CUONG, 渡辺陽介, 横田治夫 |
| Authors(English) | Cuong, Yousuke WATANABE, Haruo YOKOTA |
| 出典(和文) | , , , F7-1 |
| Citation(English) | , , , F7-1 |
| 発行日 / Pub. date | 2010, 3 |

# 機関リポジトリと外部情報源を連携した関連論文探索の実現 (O)

NGUYENMANH CUONG† 渡辺 陽介†† 横田 治夫††,†††

† 東京工業大学工学部情報工学科 〒 152-8552 東京都目黒区大岡山 2-12-1
†† 東京工業大学学術国際情報センター
††† 東京工業大学院 情報理工学研究科 計算工学専攻
E-mail: †{cuong,watanabe}@de.cs.titech.ac.jp, ††yokota@cs.titech.ac.jp

**あらまし** 近年，大学などの機関リポジトリが構築され，多数の論文が蓄積されるようになってきた．蓄積された論文の有効利用のため，引用・被引用関係を利用して，関係のある論文を推薦することは重要である．しかし，機関リポジトリでは引用・被引用関係のすべての情報を収集できるわけではないため，的確に関連する論文を推薦することが困難である．本研究では，引用・被引用関係を取得するために外部情報源を使用して，機関リポジトリ内の論文の関係を発見する手法を提案する．引用情報から探索可能な論文をすべて対象とする全探索と，引用数優先の枝刈り探索と出版年優先の枝刈り探索の 3 つの手法を用いる．機関リポジトリとして東京工業大学の T2R2 を用い，外部情報源として CiNii，Google Scholar を利用するプロトタイプシステムを実装した．また実験を行い，関連論文の検索精度について評価する．

**キーワード** 関連論文検索，書誌結合，機関リポジトリ

# Searching related papers from institution's repository using external information sources

Manh CUONG NGUYEN†, Yousuke WATANABE††, and Haruo YOKOTA††,†††

† Department of Computer Science, Tokyo Institute of Technology 2-12-1 Oookayama, Meguro-ku, Tokyo, 152-8552 Japan
†† Global Scientific Information and Computing Center, Tokyo Institute of Technology
††† Graduate School of Information Science and Engineering, Tokyo Institute of Technology
E-mail: †{cuong,watanabe}@de.cs.titech.ac.jp, ††yokota@cs.titech.ac.jp

**Abstract** Since repositories of institutions such as university are being constructed, the numbers of stored electronic papers are now rapidly increasing. Searching related papers using reference and citation information is required. However, in institutions' repositories, since the reference and citation information is not enough, it is difficult to find and recommend related papers. We propose methods to find related papers from an institution repository, using reference and citation information extracted from external information sources. Brute-force method searches all the candidate papers, while cited number priority method and publish year priority method search only the high priority papers. We implemented a prototype system using T2R2 as institution's repository, and CiNii, Google Scholar as outer information sources. The experiments showed high precision in searching related papers. We also evaluated our proposed methods.

**Key words** Related papers searching, Bibliographic coupling, Institution's repository

## 1. はじめに

近年，ネットワーク技術の発達，情報インフラの普及に伴い，電子的に利用可能な研究論文の数が増大してきている．それに伴い，大学，研究所などの機関リポジトリが構築され，多数の論文が蓄積されるようになってきた．東京工業大学には T2R2 (Tokyo Tech Research Repository) [7] という大学における教育・研究活動の産物である多様な知識資源リポジトリが構築されている．このリポジトリでは学術論文だけでなく，著書，学会発表，学位論文，特許なども格納されている．文献の数は，論

文だけでも 167,985 件（2010 年 1 月時点）が格納されている．

機関リポジトリに多数の文献が蓄積されることにより，機関内で必要となる文献の情報を電子的に入手することが可能となったが，タイトルや著者を用いたキーワード検索が主体である．一方，ある機関内で，特定の研究項目に対して関連する論文を抽出したいという要求も強くある．このため，機関リポジトリ内の関連する論文を掲示する機能を提供することは重要である．しかし，キーワード検索を用いるだけでは，関連する論文をすべて探し出すことが困難である．

関連する論文を抽出する手法として論文の引用情報を利用して，関係のある論文を抽出するアプローチが古くから提案されている．引用情報を利用し，論文間の類似度を知る代表的な手法として，書誌結合（bibliographic coupling）[1] ，共引用分析（co-citation analysis）[2] などがある．書誌結合とは共通の論文をどれだけ引用しているかで論文間の関連度を計算し，関連度の高い論文を抽出する手法である．共引用分析は多くの論文から引用される論文を関連度の高い論文として抽出する．これらの手法は，引用・被引用関係にある論文は関連する主題を扱っているということを前提にしている．しかし，機関リポジトリに蓄積される論文は，その機関の在籍メンバーに関するものだけで，引用している論文の情報や引用されている論文の情報が十分にないため，的確に関連する論文を抽出することが困難である．

本研究では，引用・被引用関係を取得するために，機関リポジトリ以外の外部情報源を使用することで，機関リポジトリに蓄積された論文の関係を発見する手法を提案し，それを実装する．ここで，外部情報源とはウェブ上にある公開論文データベースシステムを指す．例えば CiNii [8] や Google Scholar [9] などのように引用・被引用関係を公開しているものを対象とする．関連論文の探索において，関連する引用情報から探索可能な論文をすべて対象とする全探索と，引用数優先の枝刈り探索と出版年優先の枝刈り探索の 3 つの手法を用いる．全探索手法では可能なすべての引用論文を探索対象とするため，結果の再現率が高いがコストも高い．一方，引用数優先探索手法と出版年優先探索手法では論文の引用数を優先度として扱い，優先度が高いもののみに対して検索を行うことにより，処理時間を短縮できる．本研究ではプロトタイプシステムを実装し，この 3 つの手法について評価実験を行う．

以下，2. 節では提案手法の全体の処理について述べ，3. 節において作成したプロトタイプシステムについて述べる．そして，4. 節でプロトタイプシステムを利用して各提案手法に対する評価実験を行う．5. 節で関連研究について議論し，最後に 6. 節においてまとめと今後の課題について述べる．

## 2. 関連論文発見

### 2.1 全体の流れ

関連論文を検索する手順を説明する．まず，機関リポジトリ内の対象とする論文に対して，外部情報源の引用情報を基に，その対象論文が参照している引用論文の情報を取得する．次に，外部情報源から被引用情報を利用して各引用論文に対し，それぞれの論文を引用している候補論文を取得する．この候補論文の中で，対象論文と異なるものでかつ機関リポジトリにも蓄積されているものがあれば，それが対象論文と関連しているものとする．それらの関連論文に対して，書誌結合の関連度計算に基づいて対象論文との関連度を算出し，関連度が高いもののみ結果として出力する．この様にすることで，機関リポジトリ内の論文間の関連性を抽出できる．

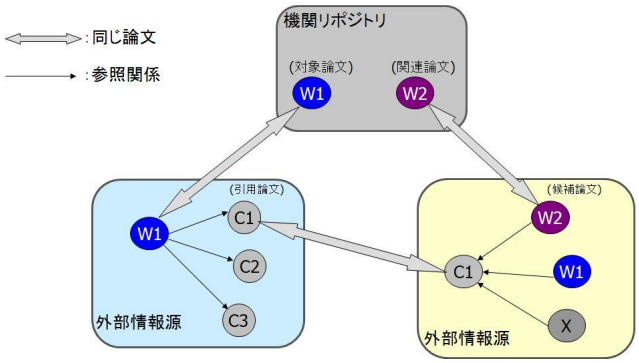


図 1　アプローチ

上記の流れを図 1 の例を用いて説明する．機関リポジトリ内の対象とした論文を W1 とする．これに対して，外部情報源を利用して W1 が引用している論文 C1，C2，C3,…を検索する．次に，C1，C2，C3,…に対して，別の外部情報源を用いてそれを引用している候補論文を検索する．例えば C1 に対して，C1 を引用している論文の W1，W2，X が検索できたとする．この場合，W2 が機関リポジトリにも入っており，かつ W1 と異なるものである．よって，W2 は W1 と関係しているとして出力される．

#### 2.1.1 全探索手法

全探索手法は取得可能な論文の引用・被引用情報をすべて解析に用いる．全探索手法の検索手順を図 2 に示す．

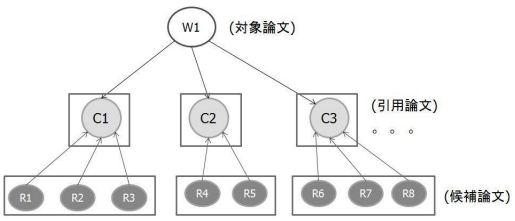


図 2　全探索手法

まず，対象となった論文 W1 に対して，外部情報源から，引用情報を利用して W1 が引用している論文（C1，C2，C3,…）の情報を取得する．次に，それらの引用論文の各々に対して，別外部情報源の被引用情報を利用してそれを引用している論文を検索する．図 2 では C1 を引用している論文の R1，R2，R3 の情報を検索し，次に C2，C3,… すべてに対して同様に検索する．この手法は単純であるが，1 つの対象論文に対して，外部情報源に問合せする回数が多くため，処理時間が長くなる．

### 2.1.2　引用数優先探索手法

全探索手法のように，すべての引用論文に対して検索を行うとコストが高い．そこで論文の重要度を考慮して，優先度が高いもののみに対して検索を行う手法を提案する．ここでは論文の引用数を優先度として扱うことを考える．つまり，引用されている論文の数が高いものは優先度が高い．検索した引用論文に優先度を付け，高い優先度を持つ引用論文総合数中の z 割だけに対して検索を行う．

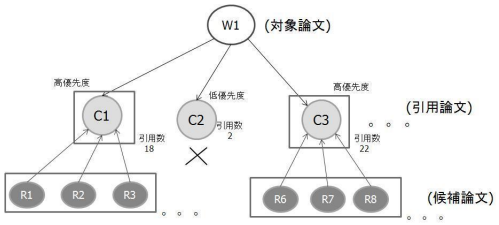


図 3　引用数優先探索手法

引用数優先探索手法の検索手順を図 3 に示す．まず，対象となった論文 W1 に対して，外部情報源から，引用情報を利用して W1 が引用している論文（C1，C2，C3,…）の情報を取得する．次に，検索結果の引用論文に対して，優先度を付ける．優先度が高いもののみに対して，外部情報源から被引用情報を利用してその論文を引用している論文を検索する．ここでは優先度が高い論文 C1 と C3 に対して検索を行う．優先度が低い論文 C2 には検索を行わない．このように，関連する可能性が高いもののみを検索し，検索論文の数を制限し，探索コストを削減する．

### 2.1.3　出版年優先探索手法

出版年が対象論文と同じ年代の論文は対象論文と関係が強いと考える．この手法では論文の出版年を考慮して，出版年の新しい順を優先度とした優先探索手法である．検索した引用論文に優先度を付け，高い優先度から引用論文総合数中の z 割だけに対して検索を行う．出版年優先探索手法の検索手順を図**??**に示す．この手法の検索手順は 2.1.2 節で述べた引用優先探索の手順と同様である．

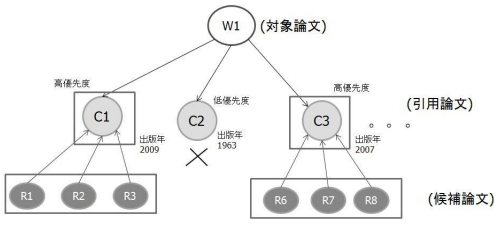


図 4　出版年優先探索手法

## 2.2　論文間関連度評価

2.1.1 節の提案手法を用いて関連論文を取得した後，それらの論文と対象論文の関連度を計算し，関連度が閾値 t 以上のものだけを出力する．論文間の関連度を評価するには書誌結合 [4] を使う．2 つの論文が同じ文献を引用しているとき，関連度を 1 点追加する．2 つの論文に対して，同じ参照文献の数が多いとき関連度が高い．

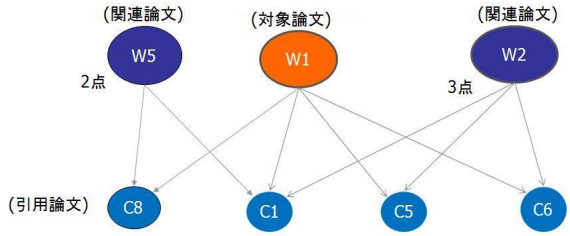


図 5　論文間関連度評価

例えば，図 5 では対象論文が W1 で，W1 と W2 が両方とも C1，C5，C6 を引用し，W1 と W5 は C8 と C1 を引用している．このとき，W1 と W2 の関連度が 3 点，W1 と W5 の関連度が 2 点となる．

## 2.3　提案システムの構成

提案手法を実現するシステム全体の構成を図 6 に示す．対象論文のタイトルやキーワードを入力とし，関連論文群を出力とする．以下の 5 つのモジュールからシステムを構成する．

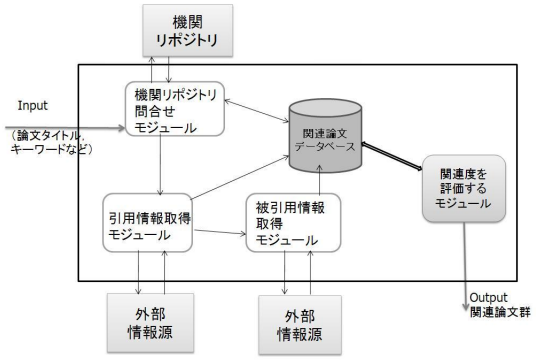


図 6　システムの構成

● 機関リポジトリ問合せモジュール：機関リポジトリへ問合わせし，タイトル，著者名，論文誌名などの論文属性を取得するモジュールである．

● 引用情報取得モジュール：外部情報源から対象論文が引用している論文の情報を取得するモジュールである．このモジュールでは全探索・優先探索手法の切り替えを行う．

● 被引用情報取得モジュール：指定された論文を引用している論文の情報を外部情報源から取得するモジュールである．

● 関連論文データベース：取得した論文のデータを格納するデータベース．

● 論文関連度評価モジュール：検索で取得した論文のネットワークから，対象論文との関連度を計算し，関連度が高いものを出力する．

## 3. プロトタイプシステムの実装

提案手法の有効性検証のために，プロトタイプシステムを実装した．プロトタイプシステムからアクセスする機関リポジトリとして東京工業大学の T2R2 [7] を対象とした．T2R2 は東京工業大学における教育・研究活動の産物である多様な知識資源リポジトリである．外部情報源は CiNii [8] と Google Scholar [9] を用いた．CiNii（NII 論文情報ナビゲータ）は，学協会刊行物・大学研究紀要・国立国会図書館の雑誌記事索引データベースなど，学術論文情報を検索の対象とする論文データベースである．国内の機関リポジトリに格納された論文の引用情報を一番多く含むと判断する．しかし，被引用情報はまだ十分とは言えない．このため，被引用情報を多く含む Google Scholar を被引用情報の取得に使う．Google Scholar は世界中の学術資料を検索できる論文データベースである．

### 3.1 機関リポジトリへの問合せ

T2R2 ではウェブサービスが提供され，文献・研究者などについての検索が可能となっている．論文情報取得には XML-RPC メッセージを利用した．T2R2 に XML-RPC リクエストを送信し，レスポンスを受信する．レスポンスメッセージを解析し，論文のタイトル，年，著者名，T2R2 における ID などを取得できる．

### 3.2 引用情報の取得

対象論文の引用情報を取得するために CiNii を利用した．CiNii に問合せするとき，対象論文のタイトルを指定し，http リクエストを送信すると，検索結果の HTML ページが返される．これを解析することにより，その論文の引用論文情報（タイトル，著者名，年，雑誌名，CiNii におけるリンクなど）を取得できる．

### 3.3 被引用情報の取得

Google Scholar を利用して，論文の被引用情報を取得する．論文のタイトルを指定し，Google Scholar に検索リクエストを送信する．返された HTML ページを解析することにより，その論文に引用している論文の情報（タイトル，著者名，雑誌名，Google Scholar における引用している論文リストのリンクなど）を取得できる． Google Scholar に連続して検索リクエストを送ると通信制限を受けるため，ここでは 1 つのリクエストに対して 3 秒ずつ待ってから送信する．

### 3.4 情報源間の論文特定

候補論文を検索できたとき，その論文が T2R2 に入っているかどうかを確認するために，外部情報源にある論文と T2R2 内の論文が同一かどうかを判断する必要がある．ここで，論文のタイトルと著者名が同じであれば，同じ論文であるとする．

また，対象論文との関連度を算出するために，Google Scholar からすべての被引用論文情報を取得したとき，被引用論文の中で同じ論文を特定する必要である．この時は，Google Scholar における論文の ID が一致すれば，同じ論文であるとする．

## 4. 実　　験

### 4.1 実験の目的

作成したプロトタイプシステムで全探索手法と優先探索手法を使った検索について実験を行った．実験の目的は提案手法の探索精度と効率の比較である．実験方法については T2R2 にある 4 件の論文を実験の対象とし，各論文のタイトルで検索して返された結果を解析した．

全探索手法に関する実験では、すべての引用論文数に対して探索を行う。引用数，そして出版年を優先度とした優先探索手法に関する実験では、枝刈を行うことにより，候補論文の総合数を減少する．優先度の高い物から探索を行い，閾値 z で枝刈を切る．ここで，$z = 1/2$ と $z = 2/3$ として，実験を行った．

3 つの手法を評価するには検索結果の precision，recall と F 値，検索時間，そして外部情報源 (Google Scholar) にアクセスする回数を尺度として比較した．

### 4.2 実 験 結 果

全探索手法を使った実験の結果を表 1 に示す．引用数を優先度として，閾値を 1/2 と 2/3 とした優先探索手法に対する実験結果を表 2 と表 3 に示す．出版年を優先度として，閾値を 1/2 と 2/3 とした優先探索手法に対する実験結果を表 4 と表 5 に示す．

各表において，結果の関連論文数というのは対象論文に対して，出力される論文数である．対象論文と関係しているものを T2R2 に入っているかどうかを問わず，全部出力した．この中で，実際に関連しているかどうかは目で見て判断した．ここで，関連度の閾値を t=2 とした．論文の第一著者が同一であれば，同著者とし，第一著者が異なる場合は異著者に分類する．候補論文総合数は対象論文の引用論文の各々に対して，それを引用している候補論文の総合数（例えば図 1 では R1，R2，R3,... の総合数）である．時間 (s) は対象論文に対して，引用・被引用論文情報の可能なものをすべて取得し，関連度を算出して関連論文群を出力するまでの実行時間である．

論文番号 3 において，CiNii から取得した引用論文数は実際の引用論文数より少ない．これは CiNii に 3 つの引用論文の情報がないため，取得できなかった．

表 1　全探索手法を使った実験結果

| 論文番号 | 結果の関連論文数 | 関連あり | 正解率 (%) | T2R2 内あり | 同著者 | 異著者 | 実際の引用論文数 | CiNii から取った論文数 | 時間 (s) | 候補論文総合数 | Google Scholar アクセス回数 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 14 | 14 | 100 | 10 | 9 | 5 | 35 | 35 | 148 | 293 | 70 |
| 2 | 7 | 6 | 85.70 | 0 | 0 | 7 | 18 | 18 | 78 | 90 | 36 |
| 3 | 6 | 5 | 83.30 | 3 | 2 | 4 | 24 | 21 | 104 | 138 | 42 |
| 4 | 3 | 2 | 66.60 | 1 | 1 | 2 | 7 | 7 | 34 | 55 | 14 |
| 合計 | 30 | 27 | 90 | 14 | 12 | 18 | 84 | 81 | 364 | 576 | 162 |

表 2　引用数を優先度とした優先探索実験結果 (z = 1/2)

| 論文番号 | 結果の関連論文数 | 関連あり | 正解率 (%) | T2R2内あり | 同著者 | 異著者 | 実際の引用論文数 | CiNiiから取った論文数 | 時間 (s) | 候補論文総合数 | Google Scholarアクセス回数 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 7 | 7 | 100 | 7 | 6 | 1 | 35 | 35 | 88 | 156 | 51 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 18 | 35 | 28 | 24 |
| 3 | 1 | 1 | 100 | 1 | 1 | 0 | 24 | 21 | 51 | 63 | 27 |
| 4 | 1 | 1 | 100 | 1 | 1 | 0 | 7 | 7 | 21 | 16 | 3 |
| 合計 | 9 | 9 | 75 | 9 | 8 | 1 | 84 | 81 | 195 | 263 | 105 |

表 3　引用数を優先度とした優先探索実験結果 (z = 2/3)

| 論文番号 | 結果の関連論文数 | 関連あり | 正解率 (%) | T2R2内あり | 同著者 | 異著者 | 実際の引用論文数 | CiNiiから取った論文数 | 時間 (s) | 候補論文総合数 | Google Scholarアクセス回数 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 12 | 12 | 100 | 9 | 9 | 3 | 35 | 35 | 111 | 213 | 56 |
| 2 | 0 | 0 | 0.00 | 0 | 0 | 0 | 18 | 18 | 49 | 35 | 28 |
| 3 | 2 | 2 | 100 | 2 | 1 | 1 | 24 | 21 | 60 | 94 | 34 |
| 4 | 1 | 1 | 100 | 1 | 1 | 0 | 7 | 7 | 25 | 25 | 4 |
| 合計 | 15 | 15 | 75 | 12 | 11 | 4 | 84 | 81 | 245 | 367 | 122 |

表 4　出版年を優先度とした優先探索実験結果 (z = 1/2)

| 論文番号 | 結果の関連論文数 | 関連あり | 正解率 (%) | T2R2内あり | 同著者 | 異著者 | 実際の引用論文数 | CiNiiから取った論文数 | 時間 (s) | 候補論文総合数 | Google Scholarアクセス回数 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 8 | 8 | 100 | 7 | 6 | 2 | 35 | 35 | 72 | 120 | 34 |
| 2 | 6 | 6 | 100 | 0 | 0 | 6 | 18 | 18 | 33 | 56 | 18 |
| 3 | 2 | 2 | 100 | 1 | 1 | 1 | 24 | 21 | 38 | 68 | 20 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 7 | 15 | 5 | 6 |
| 合計 | 16 | 16 | 75 | 8 | 7 | 9 | 84 | 81 | 158 | 249 | 78 |

表 5　出版年を優先度とした優先探索実験結果 (z = 2/3)

| 論文番号 | 結果の関連論文数 | 関連あり | 正解率 (%) | T2R2内あり | 同著者 | 異著者 | 実際の引用論文数 | CiNiiから取った論文数 | 時間 (s) | 候補論文総合数 | Google Scholarアクセス回数 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 14 | 14 | 100 | 10 | 9 | 5 | 35 | 35 | 92 | 175 | 43 |
| 2 | 6 | 6 | 100 | 0 | 0 | 6 | 18 | 18 | 45 | 62 | 22 |
| 3 | 6 | 5 | 83.30 | 3 | 2 | 4 | 24 | 21 | 54 | 97 | 32 |
| 4 | 1 | 1 | 100 | 1 | 1 | 0 | 7 | 7 | 18 | 17 | 8 |
| 合計 | 27 | 26 | 96 | 14 | 12 | 15 | 84 | 81 | 209 | 351 | 108 |

### 4.3　3 種類の手法の比較

#### 4.3.1　検索精度

各提案手法を使った実験結果の precision と recall を図 7，と図 8 に示す．全探索手法については，4 回の実験で関連論文として合計 30 件の論文が出力された．この中で，実際に対象論文と関係するものが 27 件であり，関係しないものが 3 件であった．これにより，この手法では対象とした論文に対して，関連している文献を見つけることができた．また，対象論文の参考文献リストに入っていない関連論文も抽出することができた．関連文献検索結果の平均 precision が 0.9 となっている．引用数を優先度とした検索において，z=1/2 と z=2/3 の場合共に precision が 1.0 となっている．出版年を優先度とした検索において，z=1/2 の場合は precision が 1.0 となり，z=2/3 の場合は precision が 0.96 となっている．

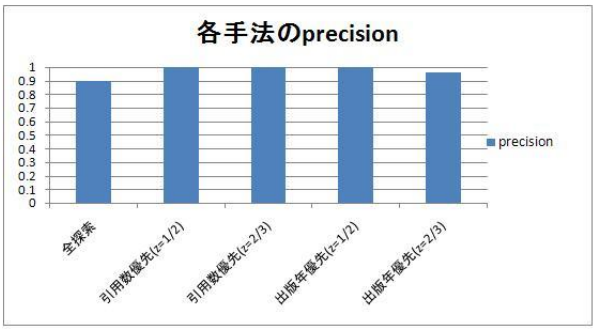


図 7　各手法における検索結果の precision

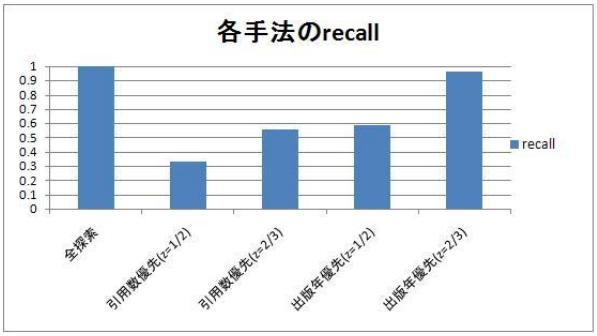


図 8　各手法における検索結果の recall

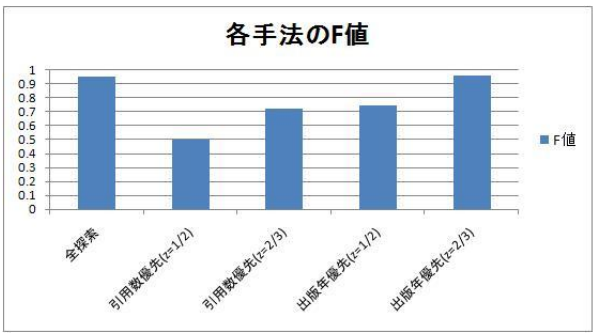


図 9　各手法における検索結果の F 値

各手法の recall の計算仕方については，全探索手法を使った実験の結果を比較基準としている．全探索法が出力した関連論文の 27 件に対して，引用数優先手法と出版年優先手法を使った実験においてどれだけ関連論文を正確に出力しているかで recall を計算する．引用数を優先度とした検索において，z=1/2

の場合は recall が 0.33 となり，z=2/3 の場合は recall が 0.55 となっている．これは，多くの論文から引用されているものは必ずしも対象論文と密な関係を持つとは限らないためと考えられる．出版年を優先度とした検索において，z=1/2 の場合は recall が 0.59 となり，z=2/3 の場合は recall が 0.96 となっている．z=2/3 の場合，枝刈を行っても recall が全探索手法とほぼ同じとなっている．これは，機関リポジトリにおいては同じグループが比較的近い時期に関連する論文を発表しているためと考えられる．

各提案手法を使った実験結果の F 値 (*F-measure*) を図 9 に示す．F 値については式 1 を用いて計算する．

$$F{-}measure = \frac{2 * precision * recall}{precision + recall} \quad (1)$$

全探索手法の F 値が 0.947 となっている．出版年優先探索手法における z=2/3 の場合では F 値が一番高く，0.96 となっている．引用数優先探索手法では F 値が一番低い．これは，引用数優先探索手法において precision が高いが recall が低いためである．

#### 4.3.2 検索時間

各提案手法を使った実験の検索時間を図 10 に示す．全探索手法において検索時間が一番長くなっている．引用数優先探索手法と出版年優先探索手法では検索時間を短縮することができた．

引用数優先探索手法の z=1/2 の場合の検索時間がほぼ出版年優先探索手法における z=2/3 の場合の検索時間と同じである．これは，引用数優先探索手法の実現において，枝刈を行う前にすべての引用論文に対して 1 回 Google Scholar にアクセスし引用数を取得していることに対して，出版年優先探索手法の実現では枝刈に必要な出版年の情報は CiNii からまとめて 1 回で取得可能なためである．Google Scholar の検索制限を回避するため，アクセスするには 1 つのリクエストに対して 3 秒ずつ待ってから送信することにより，アクセス数の増加に伴い検索時間が長くなっている．

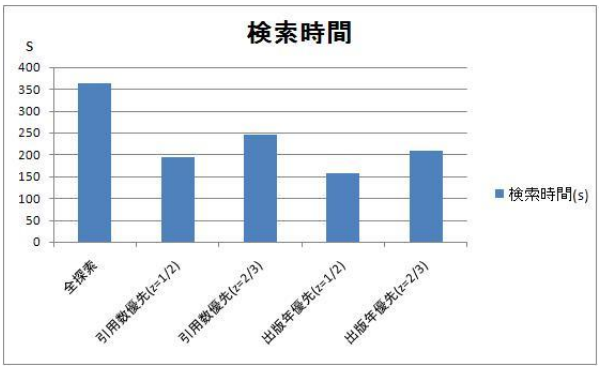


図 10　各手法に対する検索時間

#### 4.3.3 Google Scholar にアクセスする回数

Google Scholar にアクセスするとき，一定時間内のアクセス数が制限されているので，1 回のアクセスから次回まで 3 秒待ちとしている．これにより，検索時間が長くなる．このため，時間だけでなく，外部情報源（Google Scholar）にアクセスする回数を尺度として 3 つの手法を比較する．各提案手法を使った実験の検索時間を図 11 に示す．

図 11 により，全探索手法ではアクセス数が一番多い．被引用数優先手法を使った検索では Google Scholar にアクセスする回数が減ったが，まだ全探索手法のアクセス回数と近い．それに対して，出版年優先手法を使った検索では，Google Scholar にアクセスする回数が全探索手法におけるアクセス回数の 1/3 以下となった．

同様に枝刈を行っても，引用数優先手法は出版年優先探索手法と比べて Google Scholar にアクセスする回数が多い．これは，引用数優先手法の実現において，各引用論文に対して 2 回 Google Scholar にアクセスする必要があるためである．候補論文を検索する前に，すべての引用論文に対して 1 回 Google Scholar にアクセスしてその論文の ID および被引用情報のリンクを取得する．その後，枝刈における検索対象となった引用論文に対してまたもう 1 回 Google Scholar にアクセスしてリンク先の被引用情報を取得する．

出版年優先手法を使った検索では，出版年の情報は CiNii から取得できるので，出版年情報に関しては Google Scholar にアクセスする必要がないため，枝刈による探索範囲の限定の効果がより現れている．

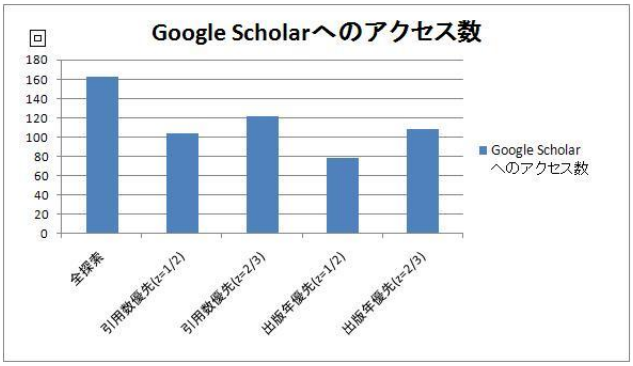


図 11　Google Scholar へのアクセス数

### 4.4 実験のまとめ

3 つの手法を使った関連論文検索の実験結果により，全探索手法では検索結果の recall が一番高いことが明確になった．しかし，この手法では検索時間も長く，外部情報源にアクセスする回数も一番多い．

引用数優先手法では検索時間を短縮する事ができたが，recall と F 値が低下した．また，外部情報源にアクセスする回数もまだ多い．

出版年優先手法では，検索結果の recall と F 値が高く，検索時間を短縮する事ができ，外部情報源にアクセスする回数も削減できた．z=2/3 の場合は全探索手法の検索時間の約半分で，ほぼ全探索手法の recall と F 値と同じである．

## 5. 関連研究

参照関係を利用して論文間の関係と研究の発展経緯を発見する手法としてリサーチマイニング手法 [3] が提案されている．リ

サーチマイニングではアプリオリアルゴリズムを使用して論文間の関係を抽出する．このため，まず論文データベース中のすべての論文を入手し，コストのかかるマイニング手法を適用する必要がある．これは，1 つの研究室内の論文程度の規模であれば実用的な時間で解析できるが，機関リポジトリ内の全論文のように多数の論文を対象とするにはコストが高い．本研究では，すべての論文の引用関係を先に入手するのではなく，外部情報源から逐次関係情報を取得しながら，探索するアプローチを取る．将来的には，外部情報源を用いてリサーチマイニングを行い，機関リポジトリ内の論文の発展経緯を発見することも考えたい．

書誌結合を改良した研究として，難波らによって引用の仕方を考慮した研究もなされている [6]．この手法では，被引用論文の引用の理由を考慮し，引用構造を用いて論文間の類似度を測ることを行っている．論文間の関連度を評価する方法としてこの手法を利用することも考えられるが，本研究では書誌結合を利用している．

## 6. まとめと今後の課題

本研究では，引用・被引用関係を取得するために外部情報源を使用して，機関リポジトリ内の論文の関係を発見する全探索と引用数優先探索，出版年優先探索の 3 つの手法を提案した．T2R2 と CiNii，Google Scholar を利用してプロトタイプシステムを実装し，この 3 つの手法を実現した．評価実験を行い，3 つの手法を比較した．対象とした論文に対して，関連している文献を見つけることができた．また，対象論文の参考文献リストに入っていない関連論文も抽出することができた．全探索手法では precision，recall と F 値が高い関連論文検索ができたが，外部情報源にアクセスする回数が多く検索時間が長い．出版年優先探索手法では precision，recall と F 値が高くかつ少ない外部情報源へのアクセスする回数で実行時間が短い検索ができた．

今後の課題として論文タイトルの類似度を優先度として枝刈の検索手法を実現する予定である．また，CiNii と Google Scholar の他に別の外部情報源を利用して，3 つの手法の性能を確認することも考えられる．さらに，外部情報源から取得した引用・被引用情報を基にして，リサーチマイニングを適用し，研究の発展経緯を発見することも今後の課題である．

## 謝　　辞

本研究の一部は文部科学省科学研究費補助金特定領域研究 (#21013017) の助成により行われた．

## 文　　献

[1] M. Kressler，“Bibliographic Coupling between Scientific Papers”，American Documentation, Vol. 14 No. 1 pp. 10-25, 1963.
[2] H. Small, “Co-citation in the Scientific Literature: A New Measure of the Relationship between Two Documents”, Journal of the American Society for Information Science, Vol. 24 pp. 265-269, 1973.
[3] 吉田誠, 小林隆志, 横田治夫, 公開されている論文 db からのマクロ情報抽出に対するリサーチマイニング手法と他手法の比較, 情報処理学会論文誌データベース, Vol. 45, No. SIG 7(TOD 22), pp. 24-32, 2004.
[4] Bing Liu, “Web Data Mining”, Springer, 2008.
[5] 豊田正史, Www における関連コミュニティ群の発見, 情処学会研究会報告, データベースシステム研究報告 No.122. 情報処理学会, 2000.
[6] 難波英嗣, 神門典子, 奥村学, 論文間の参照情報を考慮した関連論文の組織化, 情報処理学会, Vol. 42, No. 11, pp. 2640-2649, 2001.
[7] T2R2 (Tokyo Tech Research Repository) http://t2r2.star.titech.ac.jp/.
[8] CiNii (NII 論文情報ナビゲータ) http://ci.nii.ac.jp/.
[9] Google Scholar http://scholar.google.com/.
[10] CiteSeer http://citeseerx.ist.psu.edu/.
[11] DBLP http://www.informatik.uni-trier.de/~ley/db/
[12] Redstone Xml-Rpc Library http://xmlrpc.sourceforge.net/.
[13] ACM Portal http://portal.acm.org/.