EaRDB: Web集約質問処理のためのプラットフォーム

EaRDB: A Platform for Processing Web Aggregate Queries

大島 裕明† 小山 聡† 田中 克己†

Hiroaki OHSHIMA Satoshi OYAMA Katsumi TANAKA

本稿では，「Web集約質問」の処理を行うためのプラットフォームEaRDBを提案する．Web検索の目的がページを発見することであるのに対し，Web集約質問の目的はWeb全体から集約的な情報を発見することである．Web集約質問の処理には，(1)Web検索，(2)自然言語処理によるデータ抽出，(3)情報集約，といった機能が必要となる．我々はそのような機能を持つ環境，EaRDBを関係データベース上に作成した．関係データベースは強力な集約機能を保有しているため，関係データベース環境上にWeb検索結果を求める機能や自然言語処理の機能を仮想テーブルとして実装した．ユーザはSQLによって，アドホックなWeb集約質問処理や，そこで得られる知識とローカルデータベース上の既存のデータとの結合を行うことなどが可能となる．本稿では，EaRDBの設計と実装，それを利用したアプリケーションについて述べる．

We propose EaRDB, a platform for processing "Web aggregate queries." Whereas the purpose of a Web search is usually to find specific Web pages, the purpose of processing a Web aggregate query is to obtain aggregated information from the overall Web. To process Web aggregate queries, we need several functions such as (1) Web searching, (2) natural language processing for text extraction, and (3) aggregation of data. Because a relational database system has a robust aggregation capability, we implemented such a platform on a relational database environment, named EaRDB. We add functions for obtaining Web search results from conventional Web search engines and functions for natural language processing. Using SQL through the relational interface, users can formulate and execute several *ad hoc* SQL queries. Local databases can also be joined with Web search results through the relational database interface. We describe the design and implementation of EaRDB, and its applications.

1. はじめに

Web 検索エンジンは世界中の Web ページをクローリングし，莫大な量のインデックスを常に更新しながら，我々に Web ページの検索という非常に重要なサービスを提供している．我々が Web 検索を利用する際の目的は，ほとんどの場合，Web ページを発見することである．しかし，時には Web 検索エンジンが保有する情報のみを利用して，ある種の知識を取得することもある．例えば，ある語の定義を求める場合，ある商品の世間の評判の概要を知りたい場合，ある山の高さが知りたい場合などには，Web 検索を行い，その検索結果ページに含まれている情報を集約することによって，ある種の知識を取得する．

具体例として，「渋谷」の典型的な印象を表す語（ここでは形容詞とする）を求める場合を考える．種々の手法が考えられるが，一つの手法は，まず，「渋谷」というクエリで Web 検索を行い，検索結果のタイトルやスニペット中で，「渋谷」と共起して現れる形容詞を収集し，その出現回数を数えることである．多くの Web 検索結果を調べることによって，Web 全体ではどのような語が「渋谷」の典型的な印象を表す語であると考えられているかという集約的な知識が得られる．一般的な知識は Wikipedia[1]や Wiktionary[2]に記述されていることも多いが，世の中のあらゆる語が網羅されているわけではなく，また，評判情報などが記述されることは少ないため，Web 検索と簡単な言語パターンを用いた手動での集約的知識の取得が行われる機会は多い．しかし，このような作業はユーザにとって負担が大きいものである．

このような，Web からの集約的知識を取得する一連の処理を我々は Web 集約質問処理と呼ぶ．典型的な Web 集約質問処理は次の 3 つの段階から成る．

(1) **Web 検索**：知識取得のための言語パターンを考慮した Web 検索を行い，検索結果を取得する．
(2) **NLP によるデータ抽出**：自然言語処理によって，言語パターンに適合する語やフレーズを集約質問の答えの候補として抽出する．
(3) **情報集約**：得られた語に対して出現回数を数えたり，Web 検索の検索ページ数を用いたりして評価を行う．

これまで，このような Web 集約質問処理を行うためにはプログラムを書くしか方法が無かった．そこで我々は，関係データベース処理環境に Web 検索や自然言語処理機能を付加することで，より容易に Web 集約質問処理が実現可能な環境，EaRDB を提案する．関係データベース上に Web 集約質問処理環境を実現することにはいくつかのメリットが存在する．

- Web 検索 API などの既存の Web サービスや自然言語処理機能を統合するより上位層の API を提供可能である．
- ローカルデータベース上のデータと Web 集約質問によって得られる知識の統合が可能となる．
- Web 集約質問処理はしばしばアドホックに行われるが，SQL はアドホックに利用する言語として優れている．

関係データベースには情報の集約を行うための機能が存在しており，Web 検索を行い，その結果に対して自然言語処理を行えるようになれば，Web 集約質問処理に必要な機能が揃う．Web 検索結果の取得や，自然言語処理機能を関係データベース上での仮想テーブルとして実現するため，我々は，Microsoft SQL Server 2005 のテーブル値関数を利用し，実装を行った．

以降，2 節で関連研究について，3 節で Web 集約質問処理について，4 節で実装について，5 節でアプリケーション例について，6 節でまとめと今後の課題について述べる．

2. 関連研究

WSQ/DSQ[3]は関係データベース環境で Web 検索を仮想テーブルとして扱うものであり，Web 検索結果のランキング，URL，

† 正会員 京都大学大学院情報学研究科社会情報学専攻
ohshima@dl.kuis.kyoto-u.ac.jp
oyama@dl.kuis.kyoto-u.ac.jp
tanaka@dl.kuis.kyoto-u.ac.jp

日付などが取得可能である．WSQ/DSQ 上では複数の Web 検索エンジンの検索結果の統合などを行うことが可能である．しかし，検索結果に含まれるタイトルやスニペットといったテキスト情報は扱われず，自然言語処理機能も無いため，検索結果のテキスト情報からのデータ抽出による知識取得を行うことは難しい．大島ら[4]は，与えられた語の同位概念を表す語を発見する手法を提案した．同位概念を表す語が「や」で接続されることに着目し，Web 検索結果から同位概念を表す語を抽出した．これは Web 集約質問処理の一例であり，EaRDB 上でも実行可能な知識取得手法の 1 つである．Cafarella ら[5]はクエリとして言語パターンが利用できる文書検索システムを作成した．言語パターンによる質問処理に特化して設計されたシステムであり，大量の文書からインデックスを作成することで，さまざまな自然言語処理アプリケーションを作成することが可能となる．EaRDB には Web 検索結果を取得するための仮想テーブルがあり，そこには実データは存在しておらず，Cafarella らのシステムとは異なる．Web 検索の結果ページ数のみを用いて語の関係性や類似度を計算する研究も存在する．類義語を発見する手法に利用するために，Turney[6]は語の共起性を，Baroni ら[7]は相互情報量を計算する手法を提案した．また，Google Similarity Distance[8]は語の類似性を計算する手法である．これらの研究は，Web 検索による検索ページ数が大規模な文書コーパスの解析の代わりに利用できることを表している．

3. Web集約質問処理

3.1 Web 集約質問処理の概要

Web 集約質問処理とは，関係データベースにおける集約質問処理と同様に，Web の情報を集約して 1 つの値を求める処理である．関係データベースのクエリ言語である SQL では，max(), min(), average(), count()などの集約関数と GROUP BY 節を利用し，データの最大，最小，平均，レコード数などを求めることができる．Web 集約質問処理では，まず，Web 検索を用いて集約処理するためのデータを収集し，そこで得られたデータの最大，最小，平均，出現数などの取得する．Web 集約質問処理は，典型的には以下の 3 つの段階から成る．

・Web 検索
・NLP によるデータ抽出
・情報集約

Web 集約質問処理において集約されるデータは Web 上に存在する．まず，Web からデータを収集する必要がある．既存の Web 検索エンジンを利用することが容易な手法の 1 つである．典型的な Web 集約質問は何らかの条件を満たす語のうち，しばしば現れるものを発見するというものである．

Web 集約質問の一例として，「cat」の上位語を求めるために「such as a cat」の直前に最もよく現れる語を求めるというものがある．「such as a cat」というクエリで Web 検索を行えば，Web 検索結果に含まれるタイトルやスニペットの部分に必要な部分が存在しており，元の Web 文書をダウンロードして収集する必要はない．実際の「such as cat」での Web 検索結果のスニペットには，「for exercising a small animal such as a cat or kitten is disclosed」や，「taking in a domesticated animal such as a cat or dog」といった文が含まれており，「such as a cat」の直前の語として「a small animal」や「a domesticated animal」を得ることができる．

次に，ある条件を満たす語を発見することが必要となる．下記のような自然言語処理機能を利用して，Web 検索結果から語を抽出することで，集約されるデータが収集される．

・パターンに適合する語の抽出機能
・共起する語の抽出機能
・語の品詞情報の付加

最後にデータの集約処理が必要である．Web 検索結果から抽出された語の出現回数を数えるなどして，Web 集約質問に対する回答を作成する．

以上が，Web 集約質問処理の概要である．

3.2 関係データベースの仮想テーブルによる Web 質問処理環境

Web 集約質問処理では，「Web 検索」「NLP によるデータ抽出」「情報集約」が行われるが，我々は，関係データベース処理環境に Web 検索結果取得機能や自然言語処理機能を付加することにより，Web 集約質問処理環境を作成した．

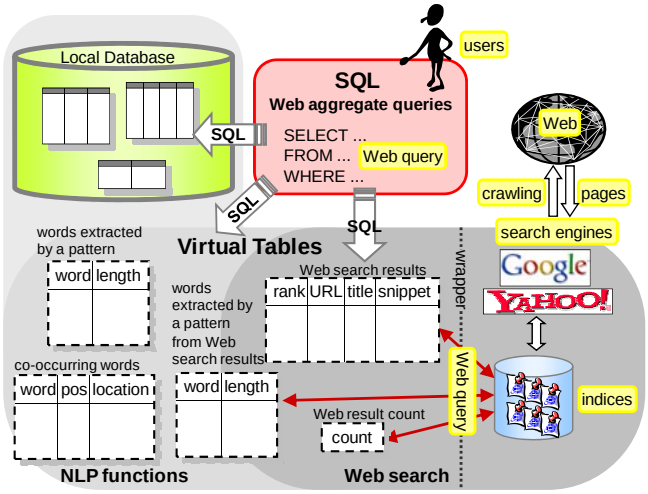

図 1 関係データベースの仮想テーブルによる Web 集約質問処理環境

Fig.1 Web aggregate query processing on a relational database environment using virtual tables

図 1 は，関係データベースにおいて，Web 検索結果取得機能や自然言語処理機能が仮想テーブルとして実現されることによる，Web 集約質問処理環境の実現の概念図である．ユーザは SQL によりそれらの機能を利用することができ，また，ローカルデータベースにも SQL でアクセスすることができる．各種 Web 検索エンジンが提供する API により，検索ランキング，URL，タイトル，スニペットといった情報が得られるが，それらの情報を関係データベース上でテーブルとして扱えるようにラッパーを作成することで，Web 検索結果の仮想テーブルを作成した．Web 検索結果の仮想テーブルは通常のテーブルとは異なり，クエリが与えられることによって異なる値が出力される．自然言語処理機能としては，文字列から指定したパターンに適合する語の抽出機能，指定した語と共起する語を取得する機能などを仮想テーブルとして実現した．

Web 検索結果のタイトルやスニペットの文字列から，ある言語パターンに適合する語を抽出することは，SQL では Web 検索の仮想テーブルと語抽出の仮想テーブルの JOIN を得ることになる．「Web 検索」と「NLP によるデータ抽出」が SQL で記述できれば，得られたデータの集約のためには，関係データベースが持つ集約機能を最大限利用することができる．主に，出現回数を数えることが多くなるため，SQL の count()

関数がよく使われることになる.

ユーザが利用する言語が SQL であるため，ローカルデータベースのデータをこれらの処理に組み込むことが容易に可能である．例えば，ローカルデータベースに保有する文書が格納されているとする．Web 集約質問処理によって「フェラーリ」のライバルを表す語を発見する手法が実現できたとすると，ローカルデータベースの検索において，「フェラーリ」のライバルについて書かれた文書を検索する，といったことが可能になる.

4. EaRDBの実装

4.1 Microsoft SQL Server 2005 における SQL CLR と APPLY 演算子

我々は，Microsoft SQL Server 2005[9]を利用し，Web 集約質問処理環境 EaRDB を構築した．SQL Server 2005 には SQL CLR という，SQL 上での関数やストアドプロシージャをプログラミング言語で記述できる機能がある．ユーザが作成可能な関数として，文字列や数値などの単一のスカラ値を返すスカラ値関数，結果としてテーブルを返すテーブル値関数，データの集合が与えられたときにそれらを計算した結果を単一のスカラ値として返す集約関数という 3 つのタイプがある.

我々は，SQL CLR の機能を利用して，Web 検索結果を取得する機能や自然言語処理機能をテーブル値関数やスカラ値関数として実装した.

APPLY 演算子は SQL Server 2005 において，テーブルと，その列のいくつかの列の値を引数とするテーブル値関数を仮想的に結合(JOIN)させるための演算子である．通常は FROM 節の後にテーブル名が記述され，続けて APPLY 演算子とテーブル値関数が記述される.

4.2 Web 検索結果取得と自然言語処理のためのテーブル値関数

テーブル値関数は SQL Server 2005 上で仮想テーブルを実現する手段であり，テーブルとしてのデータは保持せず，関数が呼び出されたときに，引数に応じたテーブルの内容を返す．ここでは，我々が実装した，Web 検索結果取得や自然言語処理を行うテーブル値関数について述べる.

WebSearch(query, num)は，Web 検索を行った結果を返すテーブル値関数である．引数 query が Web 検索で用いられるクエリ文字列であり，num が検索結果として取得する最大数である．結果テーブルの各行は，Web 検索結果の各ページの順位，URL，タイトル，スニペットを保持する．設定に応じて Yahoo!ウェブ検索や Live 検索を利用することが可能である.

WordExtract(value, extractPattern)は，ある言語パターンに適合する 1 語を与えられた文字列から抽出するテーブル値関数である．引数 value が対象となる文章であり，extractPattern が抽出するルールを表す正規表現である．extractPattern は内部に<term>という文字列を含む必要があり，value の文字列中でその位置に存在する語が抽出される．結果テーブルの各行は，抽出された語と，その語がいくつの単語・形態素から成るかを表す数値からなる.

CooccurringWords(value, targetWord)は与えられた文章中で対象の語と共起する語を取得するテーブル値関数である．引数 value が文章であり，targetWord が対象の語である．結果の各行は，共起している語，その語の品詞，対象の語からの位置を保持する．品詞情報付加ツールとしては日本語では MeCab[10]を，英語では SStagger[11]を利用した．品詞情報が付加されているため，ある語に共起する名詞のみを取得するといったことが可能である.

5. EaRDBの利用例

本節では，EaRDB 上におけるアプリケーションとして，「渋谷」の典型的な印象を表す語を取得する例と，より一般的に，ある語の印象を表す語を取得する関数を作成する例を示す.

5.1 「渋谷」の典型的な印象を表す語の取得

ここでは，「渋谷の典型的な印象」を求める手法を考える．一手法として，「渋谷」というキーワードで Web 検索を行い，その検索結果内で「渋谷」から前後 10 語以内に現れる形容詞の出現回数を求める．EaRDB の関数 WebSearch()と CooccurringWords()を用い，集約関数として COUNT()を用いる．以下が上記を実現する SQL である.

```
SELECT cw.word, COUNT(cw.word) c
FROM WebSearch('Shibuya', 100) ws
CROSS APPLY
    CooccurringWords(ws.description, 'Shibuya') cw
WHERE cw.pos = 'JJ'
AND cw.location >= -10
AND cw.location <= 10
GROUP BY cw.word
ORDER BY c DESC;
```

まず，WebSearch()関数により，「Shibuya」というクエリで Web 検索結果を取得する．各検索結果のスニペットは description という列に格納され，そこから「Shibuya」と共起する語を求める．WHERE 節では，pos 列が「JJ」であること，すなわち品詞が形容詞であることと，語の出現位置が「Shibuya」から 10 語以内であることが条件指定されている．そして，抽出された語の出現回数を数えている．この結果の一例は以下のようになる.

word	c
trendy	4
special	3
new	3
Japanese	2
famous	2
fashionable	2

5.2 典型的な印象を表す語を取得する関数

Microsoft SQL Server 2005 ではユーザが SQL で簡単に関数を作成することが可能である．ここでは，ある語が与えられたときに，その語の典型的な印象を表す語を取得する関数を作成する SQL を以下に示す.

```
CREATE FUNCTION Impressions (@query varchar(100))
RETURNS @Results TABLE (word nvarchar(max), c int)
AS
BEGIN
INSERT @Results
SELECT cw.word, COUNT(cw.word)
FROM WebSearch(@query, 100) ws
CROSS APPLY
    CooccurringWords(ws.description, @query) cw
WHERE cw.pos = 'JJ'
AND cw.location >= -10
AND cw.location <= 10
GROUP BY cw.word;
RETURN
END;
```

作成されたこの関数は，下記のような SQL で利用することができる．

```
SELECT * FROM Impressions('Kyoto')
ORDER BY c DESC;
```

ここでは，「京都」の印象語を求めており，この結果の一例は以下のようになる．

word	c
ancient	3
important	3
Japanese	2
traditional	2
histric	2
beautiful	2

以下の例では，この関数を用いて「京都」と「奈良」の共通の印象語を求めている．

```
SELECT kyoto.word, kyoto.c + nara.c v
FROM Impressions('Kyoto') kyoto,
Impressions('Nara') nara
WHERE kyoto.word = nara.word
ORDER BY v DESC;
```

結果例は以下のようになる．

word	v
Japanese	8
ancient	6
old	3
public	2
up-to-date	2
next	2

6. まとめと今後の課題

本稿では，Web 集約質問処理を行うための環境 EaRDB について述べた．EaRDB は関係データベース環境上に実現されたシステムであり，Web 検索結果の取得や自然言語処理機能を仮想テーブルとして利用することが可能である．典型的な Web 集約質問の処理は，「Web 検索」「NLP によるデータ抽出」「情報集約」の 3 つの段階から成る．Microsoft SQL Server 2005 上にテーブル値関数やスカラ値関数として Web 検索結果取得や自然言語処理の機能を実装した．実装したシステムでのアプリケーション例として，Web からの知識取得の事例を紹介しした．今後は，Web2.0 系の各種サービスや，クラスタリング機能など，さまざまな機能を EaRDB 上に追加する予定である．

[謝辞]

本研究の一部は，文部科学省グローバル COE 拠点形成プログラム「知識循環社会のための情報学教育研究拠点」(研究代表者: 田中克己，平成 19～23 年度)，文部科学省科学研究費補助金特定領域研究「情報爆発時代に向けた新しい IT 基盤技術の研究」における計画研究「情報爆発時代に対応するコンテンツ融合と操作環境融合に関する研究」(研究代表者: 田中克己，A01-00-02，課題番号: 18049041) ならびに計画研究「情報爆発に対応する新 IT 基盤研究支援プラットフォームの構築」(研究代表者: 安達淳，Y00-01，課題番号: 18049073) および，文部科学省研究委託事業「知的資産の電子的な保存・活用を支援するソフトウェア技術基盤の構築」，異メディア・アーカイブの横断的検索・統合ソフトウェア開発（研究代表者: 田中克己）および，平成 19 年度文部科学省科学研究費補助金若手研究(B)「Web からの履歴情報の発見とその呈示方式の研究」（研究代表者：小山聡，課題番号 19700091）によるものです．ここに記して謝意を表します．

[文献]

[1] Wikipedia: http://en.wikipedia.org/
[2] Wiktionary: http://en.wiktionary.org/
[3] R.Goldman and J.Widom: “WSQ/DSQ: a practical approach for combined querying of databases and the Web”, Proc. of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD2000), pp.285-296 (2000).
[4] 大島裕明，小山聡，田中克己:「Web 検索エンジンのインデックスを用いた同位語とそのコンテキストの発見」情報処理学会論文誌(トランザクション)データベース, Vol.47, No. SIG19, TOD32, pp.98-112 (2006).
[5] M. J. Cafarella, O. Etzioni: “A search engine for natural language applications”, Proc. of the 14th International Conference on World Wide Web (WWW2005), pp.442-452 (2005).
[6] P. D. Turney: “Mining the Web for synonyms: PMI-IR versus LSA on TOEFL”, Proc. of the 12th European Conference on Machine Lerning (ECML2001), pp.491-502 (2001).
[7] M. Baronim, S. Bisi: “Using cooccurrence statistics and the web to discover synonyms in a technical language”, Proc. of the 4th International Conference on Language Resources and Evaluation (LREC2004), pp.1725-1728 (2004).
[8] R. L. Cilibras, P. M. Vitanyi: “The Google similarity distance”, IEEE Transactions on Knowledge and Data Engineering, Vol.19, No.3, pp.370-383 (2007).
[9] Microsoft SQL Server: http:www.microsoft.com/sql/
[10] MeCab: http://mecab.sourceforge.net/
[11] Y. Tsuruoka, J. Tsujii: “Bidirectional inference with the easiest-first strategy for tagging sequence data”, Proc. of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP2005), pp.467-474 (2005).

大島 裕明 Hiroaki OHSHIMA
京都大学大学院情報学研究科社会情報学専攻特任助教. 2007 年京都大学大学院情報学研究科博士後期課程修了．博士（情報学）．主にウェブ，情報検索，データベースの研究に従事．情報処理学会，電子情報通信学会，日本データベース学会，ACM 各会員．

小山 聡 Satoshi OYAMA
京都大学大学院情報学研究科社会情報学専攻助教．2002 年京都大学大学院情報学研究科博士後期課程修了．博士（情報学）．主に機械学習，データマイニング，情報検索の研究に従事．電子情報通信学会，情報処理学会，人工知能学会，日本データベース学会，IEEE，ACM，AAAI 各会員．

田中 克己 Katsumi TANAKA
京都大学大学院情報学研究科社会情報学専攻教授．1976 年京都大学大学院修士課程修了．博士（工学）．主にデータベース，マルチメディアコンテンツ処理の研究に従事．IEEE Computer Society，ACM，人工知能学会，日本ソフトウェア科学会，情報処理学会，日本データベース学会等各会員．