2D2-1

日本語オントロジー辞書システム Ontolopedia を用いた検索手法に関する一考察

A Note on Method for information retrieval using Japanese ontology dictionary “Ontolopedia”

宮城良征[*1] 當間愛晃[*2] 遠藤聡志[*2]
Yoshiyuki MIYAGI Naruaki TOMA Satoshi ENDO

[*1]琉球大学大学院理工学研究科情報工学専攻
Information Engineering Course, Graduate School of Engineering and Science, University of the Ryukyus

[*2]琉球大学工学部情報工学科
Department of Information Engineering, Faculty of Engineering, University of the Ryukyus

Semantic Web technologies such as RDF (Resource Description Framework) and ontology are required in order to make a useful search engine. I constructed a Japanese ontology dictionary “Ontolopedia” using the XML data of Wikipedia(Ja). This paper described a published API and its sample applications, especially “tonchi4u” to support search tasks.

1. はじめに

セマンティック Web 技術を利用した検索エンジンを作成する場合、RDF とオントロジーは必要不可欠なものであるが、日本語を処理するための公開された日本語オントロジーが存在しない。

本研究の基礎として、Wikipedia(日本語) の文章をコーパスとして用いて、汎用的な利用を想定した日本語オントロジー構築システム「Ontolopedia[*1]」を設計・構築した。日本語オントロジーを構築するにあたり、基礎データをあらかじめ作成し、修正をボランティアの力を用いて行う。そのために、ユーザが操作しやすいインターフェースの実装を行った [3]。

現在、システムの API を公開し、外部アプリケーションとの連携をはかっており、本稿ではサンプルとして Yahoo! 検索 Web サービス [4] を使用したアプリケーションについて述べ、今後のアプリケーション応用について説明する。

2. システム概要

Ontolopedia システム概要を示す。Wikipedia のダンプデータを解析し、ユーザーが Web インターフェースを介してデータベースを修正することで精度の高い日本語オントロジー辞書を構築する。データベースから XML ファイルを出力する事で、外部のアプリケーションから利用する事ができる。

2.1 Wikipedia データの解析

Wikipedia のコンテンツは全て GNU Free Documentation License の下にライセンスされており再配布や再利用のためにデータベース・データの提供が行われている。Wikipedia の提供しているダンプデータには数種類あり、本研究では 2006 年 9 月 27 日に作成された「jawiki-latest-pages-articles.xml.bz2」を使用した。これは、494,854 ページの記事で構成された、1.3GB の一つの XML ファイルである。この XML のデータ解析手順を以下に示す。

1. **XML データを各ページ 1 ファイルに分割：** Wikipedia の 1.3GB の XML ファイルをメモリに格納し、DOM(Document Object Model) で処理するには大量のメモリを消費する。また、Ruby のメモリ領域の容量が 2GB までなので、メモリ上に DOM ツリーを展開し、処理する事ができない。そこで、XML ファイルを分割し、個別に処理する方法で対処した。

連絡先: 宮城良征, 琉球大学大学院理工学研究科情報工学専攻,yosshi@eva.ie.u-ryukyu.ac.jp

*1 http://www.nal.ie.u-ryukyu.ac.jp/ontolopedia/

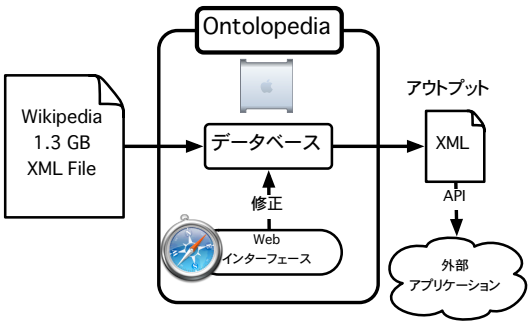

図 1: システム概要

2. **各ファイルの title タグに囲まれている語句を抽出 :**分割した各ファイルの title タグで囲まれている語句を抽出する。この抽出した語句を中心に概念構築を行う。

3. **各ファイルの title に関連する語句を抽出：** 各ファイル毎にページタイトルに関連する語句として「wiki 形式の太文字」「wiki 形式のリンク」「Category タグ」の 3 種類を抽出する。

   - **Wiki 形式のリンクを抽出：** page と page を接続するために、Wiki では ‘[[...]]’ の間に page タイトルを記述する。この page タイトルとは、title タグではさまれている語句のことである。このリンクが張られている語句は、page タイトルにとって重要な語句だと考えることができる。しかし、この語句がどの概念にあたるのか、コンピュータに判断させるのは難しいので、「未分類 (重要)」に分類する。
   - **Wiki 形式の太文字を抽出 :**リンクを張られていない言葉でも、強調した言葉は重要な場合がある。言

葉を強調する場合、Wiki ではシングルクオーテーション 3 つで語句を囲む。これも wiki 形式のリンクと同様に重要だと考え、「未分類 (重要)」に分類する。

- **‘Caegory:’ から始まる Wiki 形式のリンクを抽出 :**Category へのリンクがある場合、この語句はページに対する上位概念だと考える事ができる。そこで、‘Category:’ または ‘category:’ が含まれる場合、これを上位概念に当てはめるようにする。

4. **抽出した語句を種類ごとにデータベース (MySQL) へ登録 :**ページタイトル、太文字、リンク、Category タグの 4 種類をデータベースに登録し、基礎情報を作成を行う。

2.2 概念

図 2 に概念構造を示す。中心にあるのは、Wikipedia のページ名である。この語句を中心に概念を形成していく。「上位概念」「下位概念」「類義語」「部分材料」「動作概念」「属性」「環境」の 7 つから構成される。また、図 2 の各矢印上の英単語は、XML 出力時の各概念のタグになる。

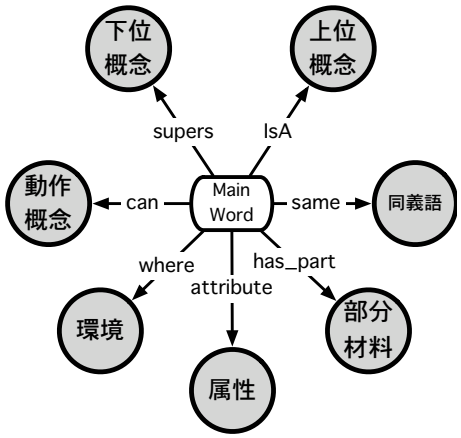

図 2: Ontolopedia 概念構造

2.3 各概念解説

構成する各概念について解説する。図 2 の中心に位置する ‘Main word’ に関する概念を形成するために、以下に述べる概念に分類する。また、実際に分類した例として「飛行機」について概念を形成する場合を考える。

- **上位概念:** ‘Main word’ の上位にあたる語句。
「飛行機は hoge の一つだ。」と表現できる物 (機械、飛行物体)

- **下位概念:** ‘Main word’ の下位にあたる語句。
「hoge は飛行機の一つだ」と表現できるもの (戦闘機、輸送機、旅客機、F-22、ジャンボ...)

- **同義語:** ‘Main word’ のと同義の語句。
「飛行機と hoge は動議である。」と表現できるもの (航空機....)

- **部分材料** ‘Main word’ のを構成する語句。
「飛行機を構成する要素」(エンジン、主翼、尾翼、胴体....)

- **属性:** ‘Main word’ のがどのような様子か。
「飛行機がどのような様子か」(便利、重い、難しい、言葉がたくさん、楽しい....)

- **環境:** ‘Main word’ はどのような環境にあるか、どのような環境で使用されるか。
「飛行機がある場所」(空、空港)

- **動作概念:** ‘Main word’ は何をするのか、されるのか。
「飛行機に関係する動作。飛行機を hoge する。飛行機に hoge する。」(操縦する、乗る....)

2.4 各テーブル解説

各テーブルのカラム、リレーションを図 3 に示す。

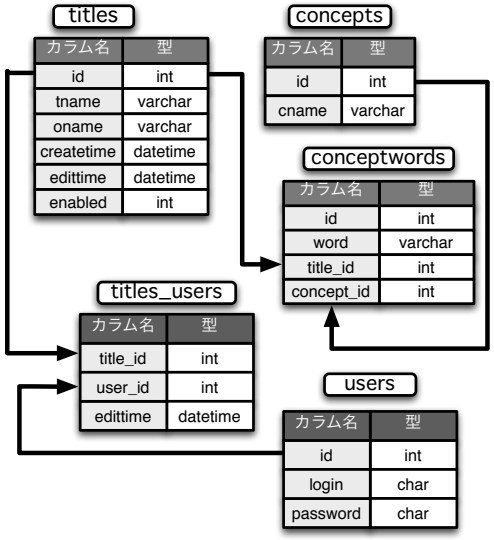

図 3: データベーステーブル構造

- **titles :** Wikipedia のダンプデータから抜き出した各ページの title を格納する。id をプライマリキーとする。tname が表示される ‘Main word’ を、createtime は最初に作成された時刻を格納する。edittime はユーザーが変更した時間を格納する。enabled にて、削除フラグを設定する。

- **conceptwords :** 各ページから抽出した語句を格納する。word に抽出した語句を格納する。title_id には、どの title に関連しているのかを示すために、titiles テーブルの id を格納。concept_id には、どの概念なのかを示す数字を入力する。この数字は、concepts テーブルの id に対応している。

- **concepts :** このテーブルは、「上位概念」「下位概念」等の概念名を保存する。プライマリキーは id である。

- **users :** ユーザー情報を格納する。login がユーザー名、password は ハッシュ関数を通して出力された値が登録される。

- **titles_users :** user_id と titles_id に任意の値を格納する事で、users テーブルと titles テーブルを繋げるている。これは、多対多のリレーションシップである。この

テーブルには、他のテーブルと違い、id 列が無い。外部キーはそれぞれのテーブルのプライマリキーである。プライマリキーが 2 つあり、それぞれの組み合わせは一つだけなので、行は一意に定義される。

2.5 ユーザインターフェース

本システムでは、ユーザは語句の検索、表示、編集を行うことができる。Flash を使用して作成した画面でユーザは語句の概念グラフ (図 4) を見ることができる。これは、データベースから XML を出力してこ読み込むようにしている。編集は、語句の一覧からボタンを使用し、容易に概念を分類することができる。

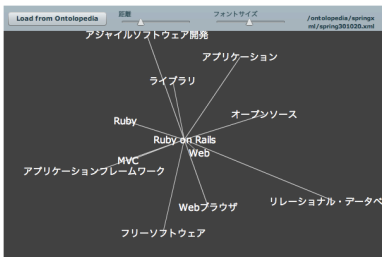

図 4: 概念グラフ

図 5 に項目の概要画面、図 6 にその上位概念を示す。

図 5: 要約表示 (WikipediaAPI):人工知能

3. Ontolopedia API

外部アプリケーションとの連携を目指し、API を公開した。この API は HTTP を使用した REST(Representational State Transfer) 型 API であり、URL を指定してデータのリクエストを取得する。Ontolopedia が返すデータは XML 形式となっている。

現在、データを取得するための 3 種類の API を提供している。これらの API は Ontolopedia の情報を引き出すのに使用される。データの登録・変更等のデータベース更新用 API に関しては、今後実装を予定している。

- **概念検索 :**ページタイトルから部分一致で語句を検索することができる。出力は検索結果の語句のリストで返す。
- **完全一致検索 :**「概念検索」と同様にページタイトルから検索するが、出力は検索に一致した語句の概念を返す。
- **データ取得 :** 語句の ID を指定して概念データを取得する。図 6 に示す結果を得ることができる。

例えば琉球大学について概念検索*2・完全一致検索*3・データ取得*4をするには各々脚注に示す URL を指定することで XML 形式で参照結果を取得することができる.

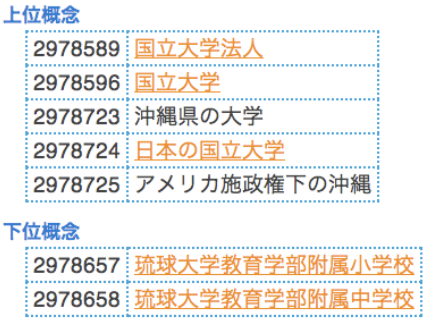

図 6: 概念表示 (上位概念):琉球大学

日本語を使用して検索する場合は、検索キーワードを URL エンコード (UTF-8) する必要がある。本システムのドキュメントにて、この API を使用するための Ruby クラスのサンプルを公開している。

3.1 API を使用して作成した応用例

Ontoloepdia の提供する API と Yahoo! JAPAN Web サービス (Yahoo!検索 Web サービス) を利用して関連語検索アプリケーションを作成した。前述した Ruby クラスを使用して Ontolopedia のデータを取得している。システム画面を図 7 に示す。

このアプリケーションは、ユーザが入力したキーワードで Ontolopedia から概念データを取得し、ユーザーの入力したキーワードと、各概念にて Yahoo!検索 Web サービスを利用して Web 上の情報を AND 検索している。

これには、完全一致検索 API を使用している。しかし、キーワードに対して完全に一致しなければ検索結果を出力することができない。これを回避するために部分一致で検索を行い、一番最初のデータを使用して検索結果を出力している。

4. API を使用しての今後の展開

今後の Ontolopedia の展開として、「情報検索支援システム tonchi4u」を計画している。

4.1 tonchi4u

インターネットやコンピュータネットワークが普及するに伴い、多くの人が自由に情報を発信、取得できるようになったが、ネットワーク上には多くの情報が流通し、その中から自分の求めている情報を検索・取得する作業が難しくなった。特に、情報を扱う技術に不慣れ、苦手意識を持つ人々は目的とする情報にたどり着けないことが多い。また、正しい情報だけで

*2 http://www.nal.ie.u-ryukyu.ac.jp/ontolopedia/api/search?keyword=琉球大学
*3 http://www.nal.ie.u-ryukyu.ac.jp/ontolopedia/api/matchfull?keyword=琉球大学
*4 http://www.nal.ie.u-ryukyu.ac.jp/ontolopedia/api/show/4657

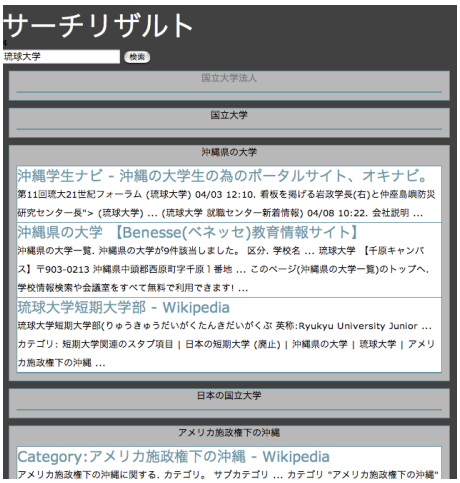

図 7: 関連語検索

なく、情報が欠落していたり、バイアスされた情報に偏っていたり、時事が不明瞭であったり、場合によっては情報が誤っていたりすることがある。このことから、一般ユーザにもそれなりの情報リテラシーが求められている。

明確な検索対象や検索スキルがそれなりにある場合でも、最初のキーワードを検索エンジンに入力するのに時間をかけて思考する [5]。検索エンジンを用いて情報を検索する行為は、キーワードに対する発想力・想像力 (イマジネーション) が求められる。それは、情報科学を学び、研究する者にとっても同様に言える。この問題を解決するために、情報検索支援システムを提案する。

エージェントプログラムがユーザの入力した言葉・語句・単語から検索に適したキーワードを生成し、ユーザの代わりに情報を収集し掲示する。ユーザはシステムが掲示する、検索された Web サイト、画像、動画を取捨選択することで、検索結果を絞り込む。検索結果をサムネイル形式で表示する。

コンピュータプログラム自身が言葉を理解することは困難だか、言葉と言葉の繋がりを利用して概念を知ることができる。この概念を利用して、人間が発想する代わりに、Ontolopedia より記号化された概念を取得し、概念の周辺語句を元に絞り込み、再検索を行う。

ユーザの公開している情報を学習データとして、パーソナライズ化した検索を行う。この公開されている情報とは、ユーザ個人が執筆しているブログ、ブラウザ等のブックマーク、ソーシャルブックマークサービス (はてなブックマーク、del.icio.us 等)、twitter 等のログを考えている。想定している学習データからユーザのプロファイルを構築し、嗜好に則した検索結果を提供する。

一方、情報技術に関して不慣れ、苦手意識を持つ人々に対しては、このようなデータを学習データとして利用できる可能性は低い。これを回避するために、職業、興味、趣味などの質問事項を入力し、それを学習データとして用いる。既存のユーザに嗜好が近いものがいれば、そのユーザのプロファイルを学習データとして利用する。

同様に検索履歴からもユーザの嗜好を学習させる。また、最近気になった話題を入力する事で意図的に学習データを与える事ができる。

検索結果は、既存の検索エンジンの表示方法とは異なり、サムネイル形式で一覧表示を行う。Web ページの場合はスクリーンショットを表示し、画像、動画の場合は縮小表示を行う。また、それぞれのサムネイル上において、そのページの要約文、キーワードをポップアップ等で表示する。

ユーザの嗜好をさらに学習させるため、検索結果を選択したかどうかも学習データとして利用する。サムネイル上には二つのボタンがあり、通常は「OK」ボタンをクリックする事で Web サイト等を閲覧するが、ユーザが期待した検索結果では無かった場合には「×」ボタンを押すようにする。

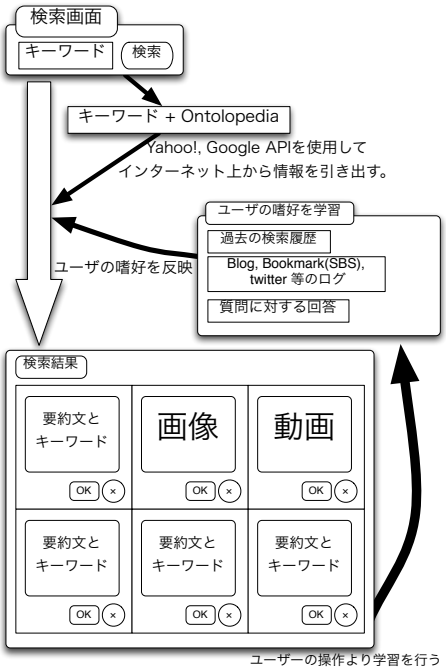

図 8: tonchi4u システム概要

5. まとめ

本研究では、Wikipedia の文章を基礎とした汎用的な日本語オントロジー辞書 Ontolopedia を構築し、API の公開を行った。公開した API を使用して関連語検索アプリケーションの作成を行い、有効性を示した。

また、情報検索支援システム tonchi4u の概要を述べ、今後の展開として実装を予定している。

参考文献

[1] 斉藤　孝 “意味論からの情報システム　ユビキタス・オントロジ・セマンティック” 中央大学出版部,2006

[2] 溝口理一郎, 古崎晃司, 來村徳信, 笹島宗彦 “オントロジー構築入門” オーム社,2006

[3] 宮城良征・當間愛晃 “日本語オントロジー辞書システム Ontolopedia の構築”, 第 60 回電気関係学会九州支部連合大会, 一般講演 (11-2A-09)　講演論文集 (p.384)

[4] Yahoo!デベロッパーズネットワーク,Yahoo Japan Corporation,http://developer.yahoo.co.jp/

[5] 情報検索に対する信頼性に関する調査および結果,2007 年 3 月, 情報 b 区初時代に向けた新しい基盤技術の研究: 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 [支援班],http://www.dl.kuis.kyoto-u.ac.jp/i-explosion/report/