連想構造を用いた WWW からの情報抽出と整理

前田 晴美・糀谷 和人・西田 豊明

奈良先端科学技術大学院大学 情報科学研究科

Extraction and Reorganization of Information from WWW Pages Using Associative Structures

Harumi MAEDA, Kazuto KOUJITANI and Toyoaki NISHIDA

E-mail: harumi-m@is.aist-nara.ac.jp

雑多で構造の不均質な HTML 文書の情報を抽出・整理する手法を提案する。基本となるアイディアとして雑多な情報をゆるやかに関連づけるための連想構造というデータ構造を用いる。我々は、連想構造を用いて WWW から情報を収集し、整理する過程を支援するシステム CM-2(Contextual Media version 2) を試作した。 CM-2 では、 URL で示される HTML 文書を入力して、利用者の整理項目に従って情報を切り出し、構造化して出力できる。人工知能研究者のホームページを題材として CM-2 を実験的に評価したところ、適合率において研究者毎で 90%、プロジェクト毎で 68%、また再現率において研究者毎で 83%、プロジェクト毎で 73% の確度で情報整理が正しく行われることが確認できた。

1 はじめに

インターネットには情報が溢れている。特に WWW(World Wide Web) 上では急速に情報が増加している。 WWW 上の膨大な情報の中から、個人が必要な情報だけを得られると便利である。これまでに、インターネットからの異質な情報源からの情報獲得の研究 [1][2][3][4][5] が報告されているが、これらは主に情報収集の手法に焦点があてられている。また、 Yahoo などの検索エンジンのように WWW から利用者の要求に応じて URL や HTML 文書を提示するシステムは数多くあるが、収集した HTML 文書から必要な情報を抽出して利用者の視点から整理できるシステムはこれまでにほとんどなかった。

HTML 文書では、情報の構造や表現方法が作者によって違うため、必要な情報を抽出することが難しい。インターネット上の雑多で構造が不均質な情報源から情報を抽出する研究として、浅い自然言語処理を用いて電子ニュースからダイジェストを作成する研究 [6] がこれまでに報告されている。この研究では、抽出したい情報に固有のドメイン知識や構造情報を組み込んだプログラムを作成するため、目標となる情報やドメインが変わるとプログラムを新たに作成しなくてはならず、人間の負荷が高い。また、利用者の入力から情報を整理するところまでは研究が進められていない。

本論文では、 WWW 上の HTML 文書のような雑多で構造の不均質な情報源から情報を抽出・整理する手法を提案する。基本となるアイディアとして雑多な情報をゆるやかに関連づけるための連想構造というデータ構造を用いる。連想構造を用いることにより、生データから必要な情報を抽出し、利用者の視点から簡単に整理できる。

我々は、連想構造を用いて情報を収集し、整理する過程を支援するシステム CM-2[1] を試作した。 CM-2 では、 URL で示された HTML 文書を入力して、利用者の整理項目に従って情報を切り出し、構造化して出力できる。

本論文は以下のように構成される。 2 節では連想構造と CM-2 の概要について述べる。 3、 4 節では情報抽出・整理手法について説明する。 5 節では実験結果を示し、有効性について議論する。

2 連想構造と CM-2

2.1 連想構造

連想構造は key と呼ぶいくつかのユニットと value と呼ぶいくつかのユニットの間に定義され、「key が与えられると value が想起される」というゆるやかな関連を表す。 key と value の間の連想関係は厳密に定義されたのではなく、多分に主観的であることを許している。これは既存の情報の雑多性、多様性に対応することを狙っている。

ユニットとは、 CM-2 における情報の基本構成要素である。ユニットは CM-2 の外部のテキストファイルやイメージファイルを表す外部参照データユニットと、 CM-2 において情報を関連付けるための内部的な対象である概念ユニッ

[1] “CM” とは我々の長期的研究目標である Contextual Media の略語である。超並列計算機 CM-2 とは無関係である。

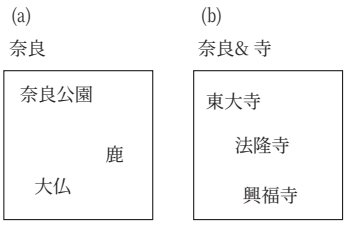

図 1: 連想構造

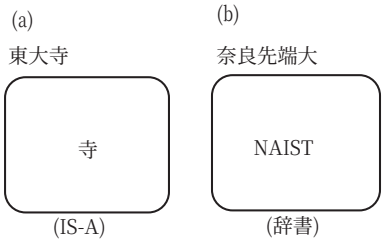

図 2: 連想構造の特別な形態

トに大別される。

本論文では CM-2 における連想構造を図 1のように記述する。図 1(a) は key「奈良」が与えられると value「奈良公園」「大仏」「鹿」が想起されることを示す。図 1(b) は key「奈良」と「寺」から value「東大寺」「法隆寺」「興福寺」が想起されることを示す。

連想構造には、特別な形態として、 IS-A 構造 (図 2(a)) と辞書構造 (図 2(b)) を定義できる。 IS-A 構造は推論に使用される。 IS-A 構造においては、クラス、サブクラスなどを区別することなく、上位概念をクラス、下位概念をインスタンスと呼ぶ。例えば、図 2(a) の IS-A 構造では、「東大寺」がインスタンスで「寺」がクラスである。辞書構造は、概念ユニット間の変換に使用される。例えば、図 2(b) の辞書構造がある場合、利用者が「奈良先端大」と入力すると「NAIST」というユニットに変換が可能である。

2.2 CM-2

CM-2 は、連想構造を用いて WWW から情報を収集し、整理する過程を支援する。

図 3に CM-2 の処理の流れを示す。情報抽出過程は、利用者からの URL 入力で示される HTML 文書を取り込み、CM-2 の連想構造を生成する。情報整理過程は、 CM-2 の連想構造を統合し、利用者からの整理項目に従って情報を整理し、表や箇条書形式で出力する。

2.3 例題

例えば、自分の興味のある分野の研究者やプロジェクトに関連する情報を知りたいが適当な書籍やデータベースが存在しないとする。このような時に、 WWW のページを収集して、必要な情報だけを得ることができると便利である。

しかし、 HTML 文書では、情報の構造や表現方法が作者によって違うため、必要な情報を抽出することが難しい。例えば、ある人は、ページの最初に写真や名前を載せ、次に

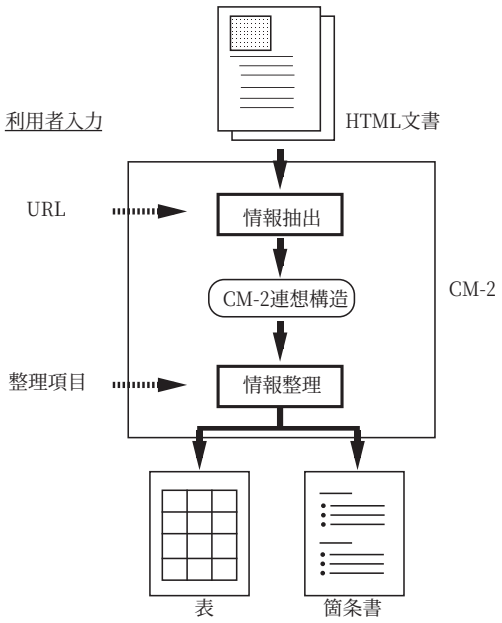

図 3: CM-2 の処理の流れ

履歴や研究に関する情報を通常の文章として書いている。またある人は、名前、 e-mail アドレス、肩書などを箇条書で書いている。さらに、作者によって同じ概念に違う名前をつけたり、違う概念に同じ名前をつけることもある。

CM-2 では、連想構造を用いてこれらの構造の異なる HTML 文書からゆるやかに情報を取り込み、利用者の入力に従って必要な情報を切り出し、構造化して出力する。

図 4に、人工知能研究者のホームページの URL を入力として HTML 文書を収集し、利用者の入力に応じて情報を整理した結果を示す。図 4(a) は、利用者がキーワード「reasoning」と、整理項目「研究者」「e-mail」「プロジェクト」「大学」を入力して、研究者毎に表形式に情報を整理して WWW ブラウザに出力した結果を示す。同様に図 4(b) は、「プロジェクト」「研究者」「e-mail」「大学」を整理項目として入力してプロジェクト毎に箇条書形式で出力した結果を示す。

3 情報抽出

URL で指定される HTML 文書から情報を取り込み、 CM-2 の連想構造を生成する情報抽出過程について述べる。アルゴリズムの概要は以下のとおり。

- **step 1** URL 解析による HTML 文書の取得
- **step 2** 形態素解析と HTML 構造解析を用いたユニットと連想構造の生成
  - **step 2.1** 形態素解析を用いたユニットの生成
  - **step 2.2** HTML 構造解析を用いた連想構造の生成
- **step 3** クラス判定ヒューリスティックを用いた IS-A 構造の生成と連想構造の修正

(a) 研究者毎に表形式で出力　　(b) プロジェクト毎に箇条書形式で出力

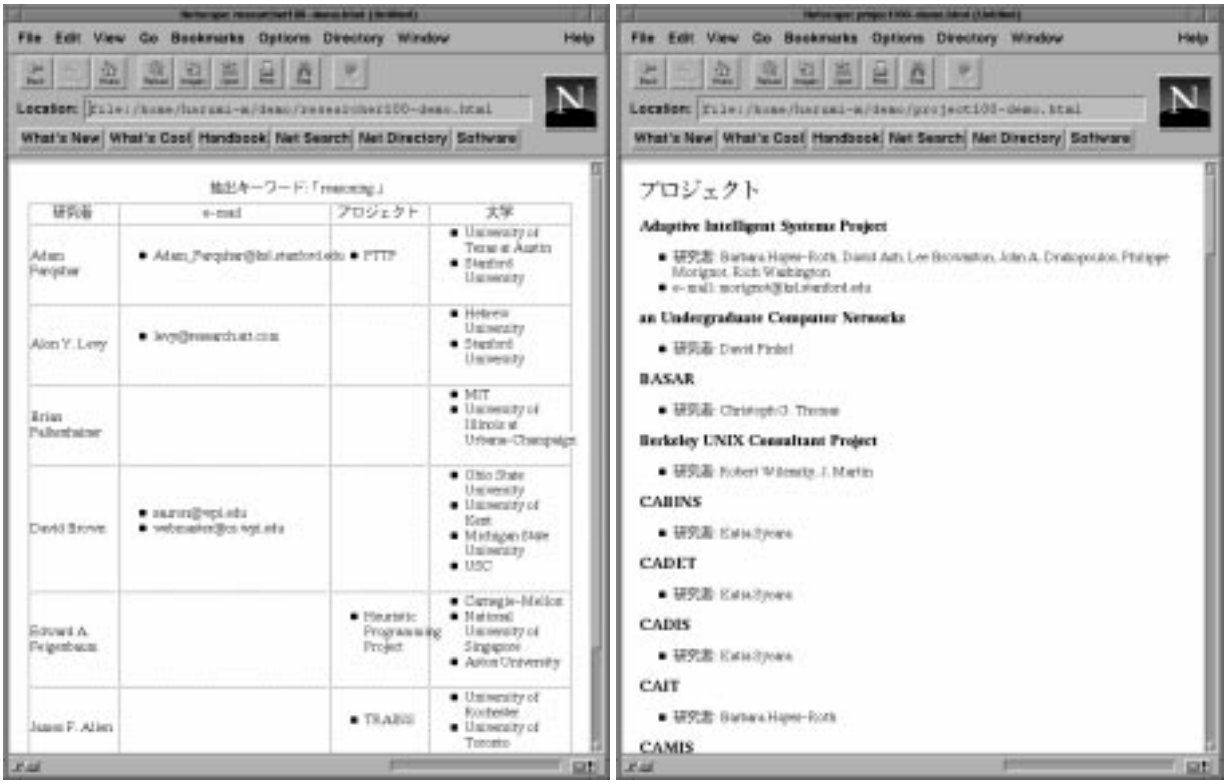

図 4: 人工知能研究者のホームページの HTML 文書を入力とした CM-2 の出力例

3.1 形態素解析と HTML 構造解析を用いたユニットと連想構造の生成

3.1.1 形態素解析を用いたユニットの生成

まず文書から HTML のタグ表現などの自然言語以外の情報を除去する。ただし、形態素解析では処理できないが有用だと想定される、電話番号、 e-mail、 URL などは事前に抽出し、概念ユニットとして登録する。

次に形態素解析 (日本語:juman、英語: Brill's Rule Based Tagger[7]) を用いて品詞情報を獲得し、名詞と、名詞が連続して出現する品詞群 (間に前置詞や接続詞、冠詞を含んでもよい) を概念ユニットとして取り込む。

なお、ストップワード辞書を用いて、不要語や非重要語を除去できる。

3.1.2 HTML 構造解析を用いた連想構造の生成

まず <h1> や <h2> などの見出しがある場合、また <dl>、<ul>、 などのリスト表現を見出しとして用いている場合にタイトル – スコープを設定する。また、 ( ) の直前の概念と中に含まれる概念の関係、 : の直前と直後の概念の関係、異なる段落 (<p> で分割されているもの) に含まれる概念の集合は、それぞれ異なる連想構造の value で表す。さらに見出し – 本文の構造が入れ子になっている時はそれぞれの value の含まれるブロックの各 key の集合を key とする。

前段で設定した概念とタイトル – スコープをもちいて、それらの概念がどのタイトル – スコープに含まれているかを元に連想関係を生成する。同一スコープに属する概念はそのスコープのタイトルに含まれる概念の集合を key とする連想関係の value を構成する。概念が複数のスコープに属している時は、それぞれのスコープのタイトルに含まれる概念の和集合を key トとする連想関係の value を構成する。

3.2 クラス判定ヒューリスティックを用いた IS-A 構造の生成と連想構造の修正

最初にクラス判定ヒューリスティックに基づき、 IS-A 構造を生成する。

クラス判定ヒューリスティックは、 people 判定ヒューリスティックや project 判定ヒューリスティックなどの、個別クラス判定ヒューリスティックから構成される。ユニットに対して、個別クラス判定ヒューリスティックを順番に適用し、最初の判定結果をそのユニットのクラスとする。個別クラス判定ヒューリスティックは、与えられたユニットの名前に含まれる文字列からクラスを推論する。

以下に people クラス判定ヒューリスティックと project 判定ヒューリスティックの概要を述べる。

people クラス判定ヒューリスティックは、あるユニットの名前に、知識として与えた人間の名前が含まれる場合にそのユニットを people であると判定する。

project クラス判定ヒューリスティックは、あるユニットが、 (1) ユニットの名前に、知識として与えたプロジェクトの名前が含まれる場合 (2) ユニットの名前に文字列「project」

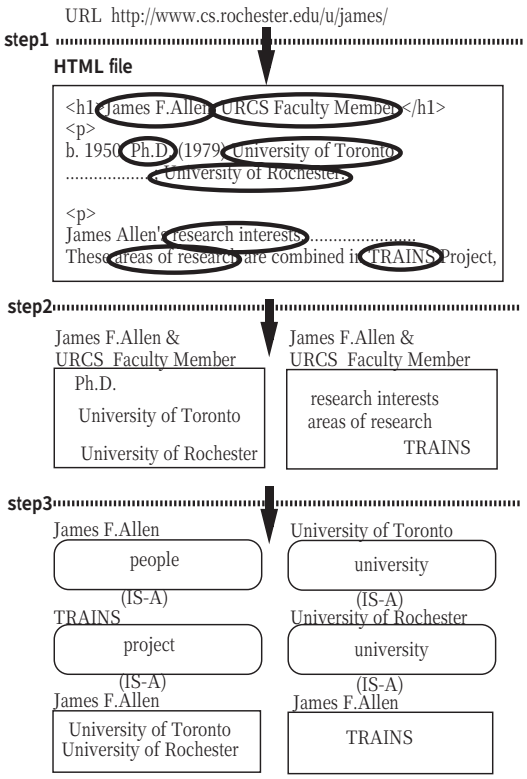

図 5: 情報抽出のアルゴリズム動作例

が含まれる場合 (3) ユニットの名前がすべて大文字で、 3 文字以上で知識として与えた不要語を含まない場合のいずれかに、そのユニットは project であると判定する。

最後に、IS-A 構造が生成された後、 step 2 で生成された連想構造の中から、 IS-A 構造で連結されていないユニットを削除し、連想構造を修正する。

3.3 例

例えば、 James Allen 氏のホームページの URL [2]が与えられた場合どのような処理が行われるのか説明する (図 5参照)。

URL 解析により HTML ファイル取得後、形態素解析を用いて、名詞句とそれに準じるものを概念ユニットとして抽出する。見出し表現 <h1>...</h1> に囲まれる範囲から生成されたユニット「James F.Allen」、「URCS Faculty Member」を key とし、段落表現 <p>...<p> で囲まれる範囲から生成されたユニット「Ph.D.」「University of Toronto」「University of Rochester」を value とする連想構造を生成する.

「James F.Allen」は「James」という英語名の単語を含むため「people」であると推論し、 IS-A 構造を生成する。「University of Rochester」「University of Toronto」は「university」という単語を含むため「university」であると推論し、 IS-A 構造を生成する。

「James F.Allen」と「URCS Faculty Member」を key とする連想構造から、 IS-A 構造によって連結されていない「URCS Faculty Member」と「Ph.D.」を削除して、連想構造を修正する。

[2]http://www.cs.rochester.edu/u/james/

表 1: 情報整理 (step 1) の主要ヒューリスティック

同名概念統合	同名の概念ユニットを統合する
辞書参照概念統合	辞書構造を参照して概念ユニットを統合する
包含的連想生成	ある概念ユニットの名前が他の概念ユニットの名前に含まれている時、その概念ユニットを key として他の概念ユニットを value とする連想構造を生成する
文脈参照連想統合	ある連想構造と他の連想構造の key が同じ時、連想構造を統合する

4 情報整理

CM-2 の連想構造を統合し、利用者の整理項目入力に従って情報を整理して提示する情報整理過程について述べる。以下に基本的なアルゴリズムを示す。

- **step 1** ヒューリスティックを用いたユニットと連想構造の統合
- **step 2** 利用者の項目入力による、経路探索を用いた連想構造の生成と結果の出力

4.1 ヒューリスティックを用いたユニットと連想構造の統合

ヒューリスティックを用いて、情報抽出過程で別個に生成されたユニットと連想構造を統合する。利用者が選択可能なヒューリスティックの中で、主要なものを表 1に示す。

4.2 利用者の項目入力による、経路探索を用いた連想構造の生成と結果の出力

利用者の項目入力に従って、経路探索を用いて連想構造を生成し、結果を表や箇条書形式で出力する。概略は図 6のとおりである。

基本的には、項目 1 で入力された概念を、利用者の視点として設定する。項目 2 以降の入力は、視点に関する属性を設定する。視点として設定されたユニットから IS-A 構造で連結されるユニット (インスタンス) に関して、属性情報を整理する。

キーワードが入力されている場合には、上記のインスタンスの中から、キーワードの文字列が含まれているユニットのみを抽出し、それらに関する属性情報を整理する。

4.3 例

例えば、利用者が「reasoning」に関連のある研究者とプロジェクトについて知りたいとする。「reasoning」というキーワードと、整理項目 1 「研究者」、整理項目 2 「プロジェクト」を入力した場合の動作例を示す (図 7参照)。

```
begin
  入力項目 1 と同名の概念ユニット item1 から IS-A 構造で連結
  されたユニット (インスタンス) を最初の key の候補集合
  key1-candidates とする;
  if item1 が辞書構造で連結されている
    then
      辞書構造で連結されたユニットから IS-A 構造で連結
      されたユニット (インスタンス) を key1-candidates
      に加える;
  if キーワードが入力されている
    then
      キーワードの文字列を含む概念ユニットを探し連想をたどって
      key1-candidates の中から最初の key の集合 key1s を確定
      する
    else key1s := key1-candidates;
  for すべての key1 := key1s do
    for すべての item := 2 番目以降の入力項目
      do begin
        key2 := item と同名の概念ユニット;
        key2 から IS-A 構造で連結されたユニット (インスタンス) を
        value の集合 value-candidates とする;
        if key2 が辞書構造で連結されている
          then
            辞書構造で連結されたユニットから IS-A 構造で連結
            されたユニット (インスタンス) を value-candidates
            に加える;
        key1 から連想をたどって value-candidates の中から value
        の集合 values を確定する;
        key1, key2 を key とし, values を value とする連想構造を
        生成する
        end;
  結果を表または箇条書形式にして HTML ファイルを作成して
  ブラウザに表示する
  end.
```

図 6: 情報整理 (step 2) のアルゴリズム

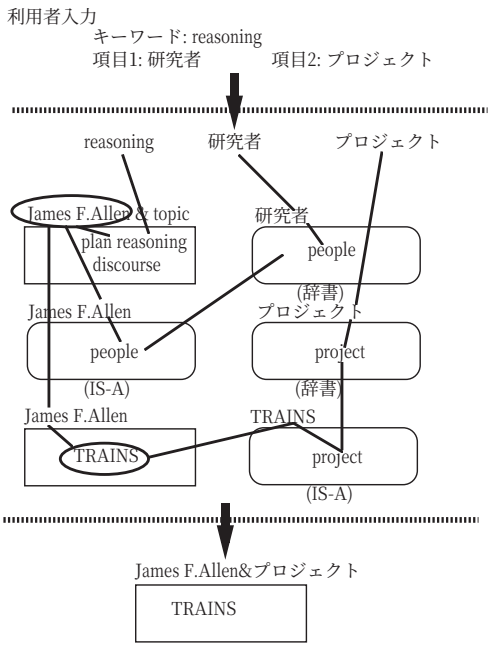

図 7: 情報整理 (step2) のアルゴリズム動作例

表 2: 実験結果

		適合率	再現率
テスト1	people	93%	85%
	people以外のクラス	89%	83%
	総合	90%	83%
テスト2	project	78%	82%
	project以外のクラス	65%	69%
	総合	68%	73%

適合率 : $\frac{\text{正しく生成された連想構造のユニット数}}{\text{生成された連想構造のユニット数}} \times 100(\%)$

再現率 : $\frac{\text{正しく生成された連想構造のユニット数}}{\text{生成されるべき連想構造のユニット数}} \times 100(\%)$

「reasoning」というキーワードを名前に含むユニット「plan reasoning」を検索する。「研究者」または「people」クラスのインスタンスであるユニットの中から、「plan reasoning」と連想構造でつながりがあるユニット「James F.Allen」を最初の key とする。項目 2 「プロジェクト」を 2 番目の key とする。「プロジェクト」または「project」クラスのインスタンスであるユニットの中から、「James F.Allen」と連想構造でつながりがあるユニット「TRAINS」を value として、連想構造を生成する。

5 実験と考察

人工知能研究者のホームページを示す 100 個の URL を入力して情報を抽出した後に、整理項目「研究者」「e-mail」「大学」「プロジェクト」「トピック」を入力して情報を整理するテスト 1 と、整理項目「プロジェクト」「研究者」「大学」「トピック」を入力するテスト 2 を行った。

生成されたユニットは 764 個、連想構造は 625 個、 IS-A 構造は 755 個である。クラス判定ヒューリスティックのために与えた個別知識は、 7 つのクラスに関して 288 個である。 people 毎に残りのクラスを整理するテスト 1、 project 毎に残りのクラスを整理するテスト 2 の結果を表 2に示す。

この結果、適合率において研究者毎で 90%、プロジェクト毎で 68%、また再現率において研究者毎で 83%、プロジェクト毎で 73% の確度で情報整理が正しく行われることが確認できた。 project 毎は、 people 毎に比べると若干率が悪いが、これはもともとのページが人間に関するページであり、 project クラスのインスタンスと他のクラスのインスタンスの間に直接的な連想関係がないためである。

我々のアプローチでは、連想構造を用いることにより、特別な自然言語処理を行なわなくても、 HTML 文書からの重要なキーワード抽出と視点の変換がほぼ満足できる確度で行えると評価する。

本研究を佐藤ら [6] の研究と比較する。これらの研究では、あらかじめ抽出したい情報ターゲットを設定し、ドメイ

ン知識や構造情報を与えて浅い自然言語処理に基づくプログラムを作成するアプローチがとられている。本研究では、利用者にドメイン知識がない場合や最初からほしい情報がわからない場合も考慮して、HTML 文書ならどのようなドメインでも汎用的に取り込み可能な過程 (抽出 step 2) と、知識を与えてより有用な情報を取り込む過程 (抽出 step 3) から構成されるアプローチをとった。連想構造を用いた我々のアプローチの長所は以下のとおりである。

- 抽出 step 2 は汎用性があり、抽出 step 3 はプログラム作成が容易であるため、プログラミング負荷が軽減される。
- 利用者の項目入力による視点の変換と情報の整理、及び整理結果の出力が容易である。

短所は、現在は連想構造を生成する過程では知識をほとんど与えていないので、従来の方法と比べるときめ細かい情報抽出が難しく、情報の精度が低いことである。

この実験の主要な誤りは以下のように分類される。(1) 抽出 step 2 における名詞句生成失敗によるユニットの生成の誤り。(2) 抽出 step 2 における HTML 構造解析失敗による連想構造生成の誤り。(3) 抽出 step 3 における知識不足による IS-A 構造生成の誤り。

今後の課題として、(1) 抽出 step 2 において、名詞句生成のためのヒューリスティックを導入する、(2) HTML 構造解析の精度を向上させる、(3) 抽出中に得た知識を IS-A 構造として再利用できるしくみをつくる、(4) 実験量をふやして検証する (クラス、URL、他ドメイン、日本語など) などがあげられる。

6 おわりに

雑多で構造の不均質な HTML 文書の情報を抽出・整理する手法を提案した。基本となるアイディアとして雑多な情報をゆるやかに関連づけるための連想構造というデータ構造を用いた。我々は、連想構造を用いて WWW から情報を収集し、整理する過程を支援するシステム CM-2(Contextual Media version 2) を試作した。CM-2 では、HTML 文書を入力して、利用者の整理項目に従って情報を切り出し、構造化して出力できる。人工知能研究者のホームページを題材として CM-2 を実験的に評価したところ、適合率において研究者毎で 90%、プロジェクト毎で 68%、また再現率において研究者毎で 83%、プロジェクト毎で 73% の確度で情報整理が正しく行われることが確認できた。

我々のアプローチでは、連想構造を用いることにより、特別な自然言語処理を行なわなくても、HTML 文書からの重要なキーワード抽出と、利用者入力による視点の変換が可能な情報整理がほぼ満足できる確度で行えると評価する。

参考文献

[1] Levy, A. Y., Sagiv, Y. and Srivasava, D.: Towards efficient information gathering agents, *Working Notes of the AAAI Spring Symposium on Software Agents*, pp.64–70 (1994).

[2] Armstrong, R., Freitag, D., Joachims, T. and Mitchell, T.: A learning apprentice for the World Wide Web, *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, pp.6–12 (1995).

[3] Balabanovi'c, M. and Shoham, Y.: Learning information retrieval agents: Experiments with automated web browsing, *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, pp.13–18 (1995).

[4] Li, W.: Knowledge Gathering and Matching in Heterogeneous Databases, *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, pp.116–121 (1995).

[5] Iwazume, M., Takeda, H., Nishida,T.: Ontology-based approach to information gathering and text categorization, *Proceedings of International Symposium on Didital Libraries*, pp.186–193 (1995).

[6] 佐藤円, 佐藤理史, 篠田陽一.: 電子ニュースのダイジェスト自動生成, 情報処理学会論文誌, Vol. 36, No. 10, pp.2371–2379 (1995).

[7] Brill, E.: Some Advance in Transformation-Based Part of Speech Tagging, *Proceedings of the Twelefth National Conference on Artificial Intelligence (AAAI-94)*, (1994).