RDFを利用した推論に基づくリスト型質問応答システム

名取拓也 * 佐藤晴彦 小山聡 栗原正仁
(北大情報科学)†

1 はじめに

インターネット関連技術の発達に伴い，現代は情報爆発の時代と呼ばれるようになってきている．黎明期のインターネットでは，コンピュータやWebに関する知識を有する者だけが情報を発信することができたが，いわゆるWeb 2.0[1]の到来により，そのような知識を持たない一般の利用者でも容易に情報を発信できるようになり，情報爆発を一層加速した．

そのような中，これらの豊富な情報の利用方法の一つとして，質問応答への知識源としての応用が近年注目を浴びている．質問応答は，主に自然言語を用いた質問文に対し，知識源の中から適切な回答を探し出し提示する技術であるが，その中でも「北海道出身の金メダリストは誰ですか？」といった，正答が複数存在しうるものをリスト型質問応答と呼ぶ．

一方で，更なる次世代のWebとして，セマンティックWeb（Web 3.0）が現在提案されている[2]．従来のWebは，文書などのリソースについて，人間がハイパーリンクをたどり，別のリソースに移動することができるというものである．それに対しセマンティックWebでは，リソースやリソース間のリンクに関するメタデータを充実させ，計算機がリソースの意味を解釈できるようにすることで，より高度な情報処理を目指している．

これらを踏まえ本研究では，Webから取得可能なRDFを背景知識として利用することで，リスト型質問応答において重要となる回答候補のスコアを改善することを目指す．

2 関連技術

2.1 リスト型質問応答システム

典型的なリスト型質問応答システムの処理の流れをFig. 1に示す．

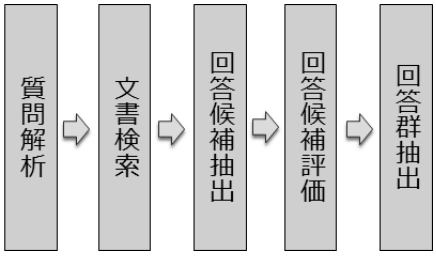

Fig. 1 一般的な質問応答システムの構成

回答候補評価部では，それぞれ回答候補に正答らしさを示すスコアを与え，スコアが高いものを優先して回答とする．通常の質問応答を行うシステム，即ち正答が単一となるようなシステムでは，最もスコアが高いものを回答として選択すればよく，システムの良し悪しは，知識源の生成や回答の抽出方法，質問文の解析，スコア付けのアルゴリズム等の性能から議論される．

しかしながら，リスト型質問応答システムにおいてはこの手法を適用することはできない．何故ならば，正答の数は必ずしも明確ではなく，更に正答が順に良いスコアを付与されているとは限らないためである．スコア付けされた回答候補からの回答選択の手法として，これまでに単純に上位から回答を採用していく方法[4]の他，スコア分布を用いる方法[5]，集合拡張(Set Expansion)を行う方法[6]などが提案されている．

しかしながら，Rzmaraらの研究[7]によれば，回答の良さはスコア付けの方法に大きく依存すると報告されている．そこでAqualog[8]では，RDFを用いることでノイズの多いスコアへの対応を試みているが，山本らの研究[9]では，同様にRDFを背景知識としているもののベイズ推定を行うことで，RDF特有のプロパティ選定の煩わしさからの脱却を実現している．

2.2 ベイズ推定

山本らの手法では，RDFを背景知識として利用するためにベイズ推定を行っている．本研究でもこの手法を用いるため，ベイズ推定について簡単に紹介する．

ベイズ推定は，事後分布は尤度と事前分布の積に比例するという性質(ベイズの定理; 式1)を利用し，事前分布から事後分布を推定する方法である．

$$\begin{aligned} P(A|X) &= \frac{P(X|A)P(A)}{P(X)} \\ &= \frac{P(X|A)P(A)}{\sum_A P(X|A)P(A)} \end{aligned} \tag{1}$$

ここに，Xはデータ，Aはパラメータであり，$P(X)$および$P(A)$はその確率分布である．事前分布$P(A)$は既に得られている知見などによる確率分布であり，$P(A|X)$は新たな観測値Xに基づいて訂正された事後分布である．

2.3 RDF

RDF (Resource Description Framework)は，Web標準化団体であるW3C (World Wide Consortium)の提唱する，リソース間情報を記述するための枠組みである．

* ntry@complex.ist.hokudai,ac.jp
† 札幌市北区北14条西9丁目北海道大学大学院情報科学研究科

RDF では，あるリソースに関する情報が「主語」(Subject),「述語」(Predicate, Property),「目的語」(Object) の三つ組（トリプル）の形で表される。主語にあたるリソースと，目的語にあたるリソース（またはリテラル）について,「主語に対して目的語がどのような関係であるか」が述語によって表現されている．RDF の例として Fig. 2 では，”Foo Bar”という事物を Web 上の URI で定義し,「この URI で表されるリソース（Foo Bar）は人間である」ことを示している．

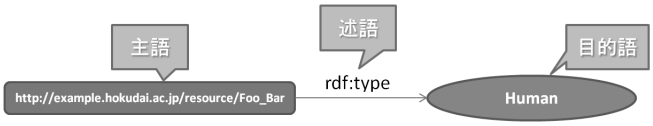

Fig. 2 RDF トリプルの例

2.4 Wikipedia

Wikipedia[3] は，フリーのインターネット百科事典であるが，その内容は誰もが自由に加筆訂正することができ，Web 2.0 代表的なのユーザーベースコンテンツとなっている．Wikipedia には分野の高い網羅性，コピーレフトの利用性，文書の構造性など多くの利点があり，かつスクレイピングが容易であるため，情報源としての利用を目指す研究が多数ある [10].

中でも，Wikipedia における種々の構造性に着目し，RDF トリプルを抽出しようとする試みは既にいくつかあり [11][12][13][14]，インフォボックスと呼ばれる統計情報や，カテゴリ階層，抄録，箇条書き情報などが生成の対象となっている．これらの構造的部分は，定まった書式で記述されているためデータの関係性が推測し易く，計算機に RDF トリプルを自動生成させることが可能である．Fig. 2 を RDF/XML と呼ばれる形式で表現すると，Fig. 3 となり，手作業で膨大な RDF データを拵えるのは現実的ではない.

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:dc="http://purl.org/dc/elements/1.1/">
    <rdf:Description rdf:about=http://en.wikipedia.org/wiki/Tony_Benn>
        <dc:title>Tony Benn</dc:title>
        <dc:publisher>Wikipedia</dc:publisher>
    </rdf:Description>
</rdf:RDF>
```

Fig. 3 RDF/XML による表現

加えて，Wikipedia の各記事は固有の URI を持っており，容易に RDF のリソースとして充当することが出来る.

3 システムの提案

3.1 概要

本研究では，山本らの提案したシステムを参考に，自動的な Wikipedia からの RDF データ生成を行い，更にリスト型質問応答において背景知識として利用し推論を行うシステムを提案する.

3.2 システムの構成

本システムの処理の流れを Fig. 4 に示す．本システムでは，予め回答候補とスコアの組み合わせが与えられるものとし,「RDF ストア生成部」および，Fig. 1 における「回答群抽出部」を実装するものである.

以下では，システムの各モジュールの設定や詳細について述べていく.

3.2.1 RDF ストアの生成

Web が知識減であるため本来なら動的に RDF を生成するシステムが望ましいが，Wikipedia ではクロールが禁止されている．従って，システムを動作させる準備として事前に Wikipedia よりダンプデータを取得し，先述した構造性を利用して RDF を抽出しデータベース化しておくことが望ましい．また，有用な RDF ストアのエンドポイントが公開されている場合には，これを利用するのも良い.

3.2.2 RDF データからの情報抽出

回答候補として与えられたリソースを含む RDF を，SPARQL を用いてデータベースより検索し抽出する．SPARQL (SPARQL Protocol and RDF Query Language) は，RDF を検索するのに用いられるクエリ言語であり，ワイルドカードを含んだ検索が可能である．これを全ての回答候補に対して行い，RDF トリプルのパターンを得る.

3.2.3 関係マトリックスの生成

各回答候補について，抽出した RDF トリプルのパターンを持つか否かを下記の関係マトリックス（以下 RM）で表す.

$$RM = \begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & & r_{2,n} \\ \vdots & & \ddots & \vdots \\ r_{m,1} & r_{m,2} & \cdots & r_{m,n} \end{pmatrix}$$

$$r_{i,j} = \begin{cases} 1 & \text{回答候補}_i\text{が RDF パターン}_j\text{を持つ} \\ 0 & \text{回答候補}_i\text{が RDF パターン}_j\text{を持たない} \end{cases}$$

ここで，m は入力される回答候補の数，n は抽出された RDF パターンの数である．ここで言う RDF パターンとは，検索対象が一要素として出現する RDF トリプルの残りの要素の組み合わせを指す.

3.2.4 ベイズ推定

ここでは，RDF データより生成した RM と，回答候補に与えられているスコアを用いて正答らしい RDF パターンの推定を行う．RDF ストアに質問に関わる正しいデータが蓄積されているならば，正答となる回答群が持つ特定の RDF パターンが共通していると考えられる.

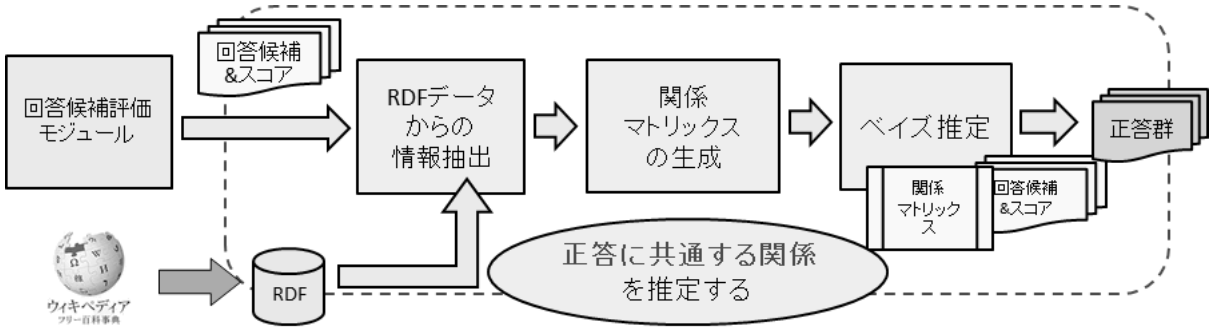

Fig. 4 提案システムの構成

そこで，ある R_j が正答群に共通する RDF パターンである事後確率 $P(R_j|S)$ をベイズ推定によって求める．$S = \{s_1, s_2 \ldots s_m\}$ であり，s_i は，i 番目の回答候補のスコアを示す．ベイズの定理にこれらを代入すると式 2 のようになる．

$$P(R_j|S) = \frac{P(S|R_j)P(R_j)}{P(S)} = \frac{P(S|R_j)P(R_j)}{\sum_{k=1}^{n} P(S|R_k)P(R_k)} \quad (2)$$

また，式 2 のうち分母は正規化の役割であり，事後分布は尤度と事前分布の積に比例するということを考慮すると，式 2 は式 3 のように変形される．

$$P(R_j|S) = \alpha P(S|R_j)P(R_j) \quad (3)$$

ただし，α は正規化のための係数であり，$0 \leq \alpha \leq 1$ である．

また，それぞれの j において s_i は $r_{i,j}$ に対し独立であると仮定すれば，式 3 の尤度関数は式 4 のように導かれる．

$$P(S|R_j) = \prod_{i=1}^{m} P(s_i|r_{i,j}) \quad (4)$$

今，尤度関数 $P(s_i|r_{i,j})$ は，i 番目の回答候補に対する j 番目の RDF パターンの有無が与えられたとき，s_i を観測する尤もらしさであり，R_j が正答特有のパターンであるならば，$r_{i,j} = 1$ となる場合には高いスコアが，$r_{i,j} = 0$ となる場合には低いスコアが与えられるはずである．従って，s_i と $r_{i,j}$ の値の差は必然的に小さくなると考えられるため，$P(s_i|r_{i,j})$ はこれらの差に対し単調減少であることが望ましい．例として，ユークリッド距離（式 5）やシグモイド関数（式 6）などが挙げられる．

$$P(s_i|r_{i,j}) = |s_i - r_{i,j}| \quad (5)$$

$$P(s_i|r_{i,j}) = \frac{1}{1 + exp\left[\lambda(|s_i - r_{i,j}| - 0.5)\right]} \quad (6)$$

また，$P(R_j)$ は事前分布であるが，事前確率の設定として一様分布（式 7）や，質問文中のキーワードが出現する頻度（式 8）などが考えられる．

$$P(R_j) = \frac{1}{n} \quad (7)$$

$$P(R_j) = \frac{t_j + \epsilon}{\sum_{k=1}^{n}(t_k + \epsilon)} \quad (8)$$

ただし，ϵ は $P(R_j) = 0$ となることを防ぐためのパラメータ，t_k はキーワードがある k 番目の RDF パターンに出現する数である．

こうして得られた事後分布 $P(S|R_j)$ を評価し，最も高い値を示す R_j を正答らしい RDF パターンとして，$r_{i,j} = 1$ となる i 番目の回答候補を出力する．

4 おわりに

本研究では，Wikipedia の構造性を利用した情報抽出と，リスト型質問応答システムへの応用手法を検討した．

今後の展望として，実データを揃えた上での実証実験や，アプリケーションの開発などに取り組んでいく．また，Wikipedia が持つ類義語や同音異義語への対応機能の利用を検討していきたい．

参考文献

[1] Tim O ' Reilly. what is web 2.0, http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

[2] T. Berners-Lee, J. Hendler, O. Lassila, et al. The semantic web. Scientific American, Vol. 284, No. 5, pp. 28-37, 2001.

[3] Wikipedia, http://wikipedia.org/

[4] Edward W. D. Whittaker, Matthias H. Heie, Josef R. Novak, and Sadaoki Furui. Trec 2007 question answering experiments at tokyo institute of technology. In TREC, 2007.

[5] 石下円香, 森辰則. 優先順位型質問応答の解スコア分布に基づくリスト型質問応答. 情報処理学会研究

報告. 自然言語処理研究会報告, Vol. 2005, No. 94, pp. 41-47, 2005-09-29.

[6] Richard C. Wang, Nico Schlaefer, William W. Cohen, and Eric Nyberg. Automatic set expansion for list question answering. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08 pp. 947-954, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[7] Majid Rzmara and Leia Losseim. Answering list questions using co-occurrence and clustering. Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC '08), Marrakech, Morocco, may 2008. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

[8] Vanessa Lopez, Victoria Uren, Enrico Motta, and Michele Pasin. Aqualog: An Ontology-driven question answering system for organizational semantic internets. Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 5, No. 2, 2007.

[9] 山本康貴 , 佐藤晴彦 , 小山聡 , 栗原正仁 . リスト型質問応答システムにおける RDF データの利用 . 情報処理北海道シンポジウム 2013 講演論文集 pp. 205-208 , 2013 .

[10] 中山浩太郎 , 伊藤雅弘 , Maike Erdmann, 白川真澄 , 道下智之 , 原隆浩 , 西尾章治郎 . Wikipedia マイニング : Wikipedia 研究のサーベイ . 情報処理学会論文誌 . データベース , No. 4, pp. 49-60, 2009.

[11] Soren Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. The Semantic Web, pp. 722-735, 2007.

[12] 玉川奨 , 森田武史 , 山口高平 . 日本語 wikipedia からのクラススキーマ階層の自動構築と利用 . 第 26 回人工知能学会全国大会 , 2C1-NFC-2-1, 2012.

[13] 藤原嵩大 , 吉岡真治 . Wikipedia の階層関係を分析するためのカテゴリパターンの提案 . 第 26 回人工知能学会全国大会 , 2C1-NFC-2-4, 2012.

[14] 柴木優美 , 永田昌明 , 山本和英 . 日本語語彙大系を用いた wikipedia からの汎用オントロジー構築 . 情報処理学会研究報告 . 自然言語処理研究会報告 , Vol. 2009, No. 4, pp. 1-8, Nov 2009.