

## 画像や地図を用いて回答できる質問応答システム

佐藤 充

森 辰則

横浜国立大学 大学院 環境情報学府

横浜国立大学 大学院 環境情報研究院

E-mail: {mitsuru,mori}@forest.eis.ynu.ac.jp

質問応答の回答として、文字による解だけでなく画像や地図も表示できるシステムを提案する。既存の質問応答システムと画像検索、ジオコーディング、地図 API を組み合わせ、物に関する質問に対しては画像を、場所を問う質問に対しては地図を表示する。質問応答の解に対応する画像等に曖昧性があっても、質問文に含まれる情報を用いて適切なものを絞り込める。

評価実験によれば、質問応答の解が正しかったとき、画像は 0.70、地図は 0.85 の精度で適切なものを表示できることが分かった。

また、正解が複数存在するリスト型の質問においては、ユーザは並べられた画像を見ることでそれぞれの解を比較できたり、一つの地図上でそれぞれの解の位置関係を把握できるというように、質問応答システムにおける新しい回答の提示方法を示した。

## A Question-answering System that can Display Images and Maps as Answers

Mitsuru SATO and Tatsunori MORI

Graduate School of Environment and Information Sciences,  
Yokohama National University

We propose a system that can display not only the answer string of a given question but also images or maps related to the answer. The system consists of an existing Q/A system, an image search engine, a geocoding engine and a map API. Images are displayed when the question is about things like an animal, a plant or a person. Maps are displayed when the question is about place. Appropriate images or maps are selected by using the information in the question sentence even if the answer string is related to several different types of images or location.

The experimental results show that the accuracy of displaying images is 70 percent, and that of displaying maps is 85 percent under the condition that answer strings are correct.

Moreover, we show a new method that presents a result of the list-type question-answering on a display simultaneously. A list-type question may have two or more correct answers. Using the method, a user can compare the answers by seeing listed images, or a user can tell the positional relation among the answers by seeing a map in which all answers are marked.

### 1 はじめに

近年、文書情報にアクセスする技術として質問応答が注目され、国内外で盛んに研究が行なわれている。質問応答システムとは、ユーザが自然言語で与えた質問文に対し、知識源となる大量の文書集合から回答語句そのもの(解)を見つけて出力するシステムである。質問応答システムは他の文書処理技術の中で活用されることも期待されているが、本稿ではユーザがシステムを直接使う場面に焦点を当てる。

一般的な質問応答システムは文字による回答を返すが、質問によっては他のメディアを用いた方が分かりやすい場合がある。例えば、動植物や人物に関する質問の回答としてそれらの画像を提示することは、ユーザにとって有用と考えられる。場所を問う質問に対しては、場所を表すのに特化した画像ともいえる地図がある。そこで本稿では、Web上の画像検索エンジンや地図 API を質問応答システムと組み合わせ、画像や地図を用いた多面的な回答を提

示するシステムを提案する。

提案システムには次のような利点がある。解とともに提示された画像を見ることで、ユーザが解の妥当性を確認できる可能性がある。例えば「ライオンに似ていた首相は誰ですか」という質問に対して、解「浜口雄幸」とともにその画像を見ることで、少なくとも解的の外れでないことが分かる。ユーザが「浜口雄幸」を知らなかった場合には理解の助けになるであろう。

地図に関しても同様で、「ドイツの首都はどこですか」の回答として、解「ベルリン」とともに提示された地図を見ることで、ベルリンがドイツの中にあることが確認でき、どの辺りにあるのかといった情報も得られる。

また、提案システムは自然言語による画像検索・地図検索システムと捉えることもできる。従来のキーワードによる検索ではなく、「世界最大の花は何?」「本州最北端の町はどこ?」といった自然言語によ

るクエリ(検索質問)で画像や地図を検索できる。

提案システムを実現するための要件としては次の点が挙げられる。

1. 質問に応じて適切な回答方法(画像または地図)を選択できること。
2. 質問応答の解が多義語だったときでも適切な画像や地図を選択できること。
3. 地図の場合、適切なレベル(詳細度)で表示できること。

これらの問題には、質問文から得られるキーワード(以下、単に「キーワード」と呼ぶ)や質問文タイプ(人名や地名など、質問が問うている事物等の種類)といった情報を用いることで対処する。

## 2 関連研究

### 2.1 テキストと画像の対応付け

藤井らは、事典検索サイト Cyclone[1]において、見出し語の説明テキストに対して画像の対応付けを行なった[2]。見出し語が「ハブ(装置/蛇)」のような多義語であったとき、画像の多義性を解消する必要がある。そのためまず、画像のリンク元の HTML ファイルにはその画像を特徴付ける語が多く存在すると考え、それらの語を画像の索引とする。次に、見出し語の、ある語義に対応する説明テキストを入力として、類似度の高い候補を上記の HTML ファイル群から検索、そこからリンクされている画像を説明テキストと対応付ける。画像周辺の語に着目して画像を検索するという手法は我々の提案システムでも使用する。

Cyclone は自然言語による検索も可能であるため、応用として「テキストによる質問に対して画像で回答すること」が可能であるとも述べられている。しかし、Cyclone に自然言語による質問文が入力された場合、入力文と適合する説明テキストが検索され、そこから質問文タイプに適合する見出し語が逆引きされるため、見出し語は必ずしも質問の答えにはならない<sup>1</sup>。一方、提案システムがベースとする質問応答システム[3]では、組織化されていないプレーンテキストから回答語句そのものを抽出するという点が異なる。

また、Cyclone では予め収集した画像とそのリンク元 HTML ファイルの対に対して事前に索引付けを行なっておくことを前提としている。システムの応答時間の面では有利であるが、この作業はコストが高い。提案システムでは既存の画像検索エンジンを利用することで、手元に画像データを持つことなく、質問が入力された時点で画像の検索・絞り込みを行なう。

<sup>1</sup>説明テキスト中に回答が含まれていても、回答が見出し語になっているとは限らない。

### 2.2 住所探索

提案システムにおいて地図で回答する際、処理の途中で対象の住所を調べる必要がある。佐藤[6]は、以下の手順に従って与えられた名称に対する住所情報を Web から探し出すシステムを構築した。

まず、既存の検索エンジンを用いて「名称 AND (“ドメインレーダー”<sup>2</sup> OR “所在地” OR “住所”)」というクエリを入力し、検索結果のページを収集する。ただし、「名称」には住所を調べる対象の名称が入る。得られたページの中から情報抽出を行なう領域を特定し、そこから正式名称、住所、郵便番号、電話番号などの情報を抽出する。住所の抽出には独自に作成した辞書を用いている。最後に、収集した住所情報の同一性を判定し、同じ対象を表していると思われる住所情報を統合する。

上記の処理にはある程度の時間を要するが、Web上に住所情報が多数記載されているものならば、多くの場合、システムは正しい住所情報を出力できる。

提案システムにおいては、実時間での応答が求められること、および住所探索の後のジオコーディングが細かい住所まで対応できないことなどを考慮して、上記の手法を簡略化して用いる。具体的な手法については第3章で述べる。

## 3 提案システム

提案システムでは、質問応答の解を文字で出力することに加え、人物や動植物など物に関する質問の場合はその画像を表示し、場所に関する質問の場合は地図でその位置を示すようにする。こうすることで、視覚的に理解しやすい回答を提示でき、第1章に示したようなメリットが得られると考えられる。

なお、質問文としては factoid 型(事実型)の質問のみを想定する。本章では各処理の詳細について述べ、システムの動作例を紹介する。

### 3.1 回答方法の決定

画像・地図のどちらを用いて回答するかは、質問文タイプに基づいて決定する。質問文タイプとは、人名や地名など質問が問うている事物等の種類であって、表1に示すような種類がある。質問文タイプは、疑問詞などに着目した規則に基づいて決められる。

表1: 質問文タイプの種類

質問文タイプ	種別	質問例
person	人名	～は誰
place	地名	～はこの国
organization	組織名	～はこの会社
数量(lengthなど)	数値・数量	～は何メートル
none	不明	～は何

<sup>2</sup><http://www.nda.co.jp/domain/>「ドメインレーダー」とはドメインと組織名の対応表を提供している Web サイトの名称である。

この中で、'person'は人物に関する質問であるから、その名前とともに画像を出力する。これは顔と名前が一致しないといった場合に効果的である。'place'は場所に関する質問であるから、地図を表示する。'place'のサブタイプとして'country'(国)、'prefecture'(都道府県)などがあり、これらの情報も後の処理で使用する。'organization'は組織に関する質問であるが、企業や大学など、組織は場所の情報を持つことが多いから、この場合も地図でその位置を示すことにする。数量は、解が「～メートル」「～個」といった数量表現になる質問である。この場合は画像や地図を表示するのは適切ではないと考え、どちらも出力しないことにする。「～は何ですか」というような質問のとき、質問文タイプが判定できずに'none'となる。この場合、解が名詞句であれば画像を表示するようにする。

### 3.2 画像を用いた回答

3.1節で「画像を表示する」と判定された質問の解に対して、適切な画像を表示する手法について説明する。

#### 3.2.1 情報源と画像検索エンジン

画像データの情報源としては、膨大な画像データが存在する Web を対象とする。現在の実装では、Web 上の画像データへのアクセスには Infoseek<sup>3</sup>の画像検索エンジンを用いている。予備実験として、Infoseek 画像検索で曖昧性のない語句 50 語に対して 10 件ずつ画像を検索した。その結果、8 割以上の精度で適切な画像を検索でき、精度は十分実用に耐え得るものであった。また、クエリには「AND」や「OR」などの論理演算子を組み合わせて用いることができる。

#### 3.2.2 解の曖昧性

質問回答の解に対応する画像を検索する際、解が曖昧性のない語句であれば、画像検索エンジンにクエリとして解を入力するだけで、高い精度で適切な画像が得られる。

しかし一般に語は複数の意味を持つ。さらに、ある語の辞書的な意味が一つであったとしても、その語が例えば本や映画のタイトルとして使われれば、対応する対象は複数存在することになる。画像検索の対象になりやすいであろう人名も同姓同名の場合が考えられる。

検索エンジンで特定の語義に関する結果だけを絞り込みたいときは、検索語を追加することが有効である。しかし、画像検索の場合は検索語を追加すると結果件数が大きく減ってしまうことが多い。また、「X AND Y」というクエリでは、X と Y のどちらの(またはその両方が含まれる)画像が出力され

るかが分からない。すなわち、語義を絞り込む一方で解に対応しない画像が増えることになる。

#### 3.2.3 画像のリンク元テキストの解析による曖昧性解消

藤井らの手法 [2] では、Cyclone の見出し語の説明テキストを使って画像の多義性を解消していた。一方で提案システムにおいては、質問文を使って同様のことができると考えられる。

[2] で使われていた「ハブ(蛇/装置)」の例を考える。蛇のハブが解になる質問文と、装置のハブが解になる質問文は以下のように内容が異なると考えられる。

質問 1 「沖縄に生息する毒蛇は何？」

質問 2 「スター型 LAN で使われる集線装置は何？」

しかしながら、質問文からキーワードを取り出して、質問 1 なら「ハブ(沖縄 OR 生息 OR 毒蛇)<sup>4</sup>」をクエリとして画像を検索すればよいかというと、必ずしもそうではない。クエリにキーワードを追加することで、3.2.2節で述べたように無関係の画像(例えば「沖縄」の画像など)が混ざってしまう。

そこで、次の手順を提案する。まず、解(上の例では「ハブ」)だけをクエリとして画像を  $N$  件検索する。次に、それぞれの画像のリンク元の HTML ファイルを取得し、画像の周辺のテキストを切り出す(以下、「周辺テキスト」と呼ぶ)。周辺テキストは画像へのリンク(<IMG> や <A> タグ)の前後  $M$  文の範囲とする。最後に、質問文と周辺テキストの類似度を計算し、類似度が高い候補は質問の回答として適切な画像であると考え、これを出力する(図 1<sup>5</sup>)。

「解」と「質問文」をそれぞれ「見出し語」と「説明テキスト」に対応させれば、画像の対応付け方法は藤井らの手法 [2] と同じといえる。しかし、アプローチが異なる。[2] においては予めオフラインで画像データを蓄積・組織化するが、提案システムではユーザの質問が入力された時点で画像を検索する。そのため、最新の話題に関する質問に対応しやすいという利点がある。

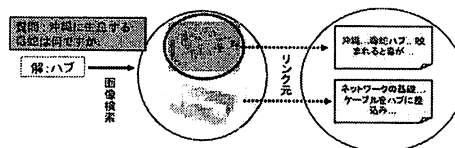


図 1: 画像の周辺テキストを解析

テキスト間の類似度の計算には種々の方法が考えられるが、提案システムではベクトル空間モデルを用いる。質問文と周辺テキストは内容語(名詞、動詞、形容詞)の重みを要素とするベクトルで表現され、2つのベクトルのなす角の cosine 値で類似度が

<sup>3</sup><http://search.www.infoseek.co.jp/>

<sup>4</sup>「AND」は省略し、スペースで代用している。

<sup>5</sup>著作権の都合により意図的に画像を不明瞭にしている。

計算される。重みは出現頻度を基本とするが、内容語の品詞の種類に応じて重みを増減させることも考えられる。また、質問文に含まれるキーワードの異なり数は少ないため、解が抽出された文脈中の語を追加するなどして調整を図ることも考えられる。これらの方法の比較は第4章で行なう。

この手法では、最初に画像検索エンジンから得られた画像の中に適切な画像が含まれていなかった場合は正解に辿り着けない。このような場合はどの画像の周辺テキストも質問文との類似度が低くなることが予想されるため、適切な画像が見つからないことはシステム側で判断できる。その際はキーワードを追加して改めて画像検索エンジンで画像を検索するというように、類似度を見て手法を変更することが有効と考えられる。この方法についても第4章で実験を行ない、その有効性を検証する。

### 3.2.4 システム動作例 (画像)

画像を用いて回答する際のシステム動作例を図に示す<sup>6</sup>。図2は先程の「ハブ」が解になる質問での例である。それぞれ蛇のハブ、装置のハブが区別され、「ハブ」という解に対して曖昧性の解消が上手く働いていることが分かる。

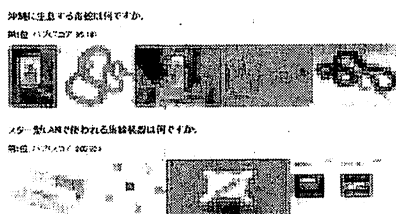


図 2: 画像を表示する例 (1)

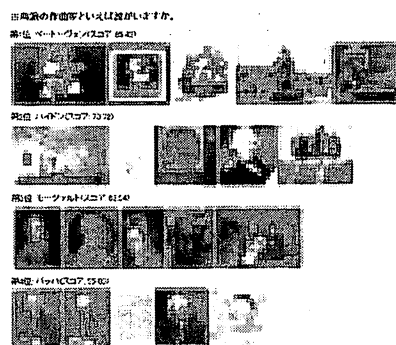


図 3: 画像を表示する例 (2)

図3は「古典派の作曲家といえば誰がいますか」という複数の正解を持つ質問を入力した場合の例で、「ベートーヴェン」「ハイドン」「モーツァルト」

<sup>6</sup>著作権の都合により意図的に画像を不明瞭にしている。

「バッハ」の画像が表示されている。何位までを正解とするかは、解のスコア分布を見ることでシステムが自動で判定する [4]。この場合は正解の画像をリストとしてまとめて見ることができ、ユーザは画像を見ることでそれぞれの解を比較できて便利である。また、「自然言語による画像検索」という見方をすれば、「～のメンバー」といった複数の対象を表す語句に対して画像検索が行なえることになる。これは従来の画像検索には無かった機能であり、提案システムの利点の一つである。

### 3.3 地図を用いた回答

3.1節で「地図を表示する」と判定された質問の解に対して、適切な地図を表示する手法について説明する。

#### 3.3.1 Google Maps API を用いた地図表示

地図の表示には Google Maps API<sup>7</sup>を用い、解が示す位置をプロットする。これを利用することで Web ページ上に地図を表示でき、ユーザはマウスの操作によって縮尺を変えたり、地図を動かして周囲に何があるのかを調べたりすることができる。

位置の指定には経緯度が用いられる。従って、プロットする対象の経緯度を求める必要がある。

#### 3.3.2 地図の表示レベル

国や都道府県、海や湖といったある程度の広がりをもった領域を示す場合は、その代表点に分かれればよい。一方、「東京ディズニーランド」などランドマーク (建物や施設などの地理オブジェクト) の位置を示す場合は、詳細な位置の情報が必要になる。この場合、直接ランドマークの経緯度を調べるのは難しいので、まずはランドマークの住所を調べ、得られた住所を経緯度に変換する (ジオコーディング) という二段階を踏むことにする。

#### 3.3.3 ランドマークの住所検索

日本のランドマークの住所は、佐藤の手法 [6] を簡略化したものを用いて求める。まず、検索エンジン (Goo<sup>8</sup>) に「ans (“住所” OR “所在地”）」というクエリを入力して Snippet (検索結果ページの要約) を取得する。ここで、ans は質問応答の解を表すものとする (以下同様)。

ans がランドマーク名ではなく住所の一部であり、詳細な住所を求めたいときは問われているランドマーク名およびキーワードをクエリに追加する (3.3.6 節参照)。

次に、得られた Snippet を整形し、ボタンマッチで「〇〇県××市△△x-y」といった住所表現を抽出する。ジオコーディングの際に「番地」までしか対象にならないことが多いので、号番号のみが異なる

<sup>7</sup><http://www.google.com/apis/maps/>

<sup>8</sup><http://www.goo.ne.jp/>

るものなどは一つにまとめる。そして、頻度が最も高かったものを出力する<sup>9</sup>。

### 3.3.4 ジオコーディング

ジオコーディングとは、住所を経緯度に変換することをいう。日本の住所を経緯度に変換するには、東京大学空間情報科学研究センターが提供する「CSIS シンプルジオコーディング実験」[5]を利用する。日本の住所を入力すれば、経緯度情報がXML形式で返される。また、変換の信頼度を表す iConf という値も得られ、この値が3以上のときは変換が成功したとみなせる。「神奈川県横浜市」というように住所の一部が入力された場合は、代表点としてその区域の重心の点が得られる。

これらの機能を利用して、ジオコーディングエンジンを以下のように利用する。

- ランドマークの住所を経緯度に変換する。
- 都道府県名や市区町村名を入力して代表点の経緯度を求める。
- 変換が成功すれば iConf が3以上になることを利用して、ある文字列が日本の住所かどうかを判定する。

### 3.3.5 経緯度の検索

海外の地名の場合は前述の住所検索やジオコーディングが行えない。そのため、Web を情報源として対象の経緯度を直接調べる。

フリー百科事典 Wikipedia<sup>10</sup>には多くの経緯度情報が記載されているため、まずはこれを利用する。「カイロ (エジプト / アメリカ)」というような多義語の場合を考慮して、最初に Goo で Wikipedia 内を検索する。クエリは「site:ja.wikipedia.org intitle:ans keywords」とし、得られた結果の上位から順に調べていく。ここで、keywordsはキーワードを「OR」で結合したものとす(以下同様)。例えば、「エジプトの首都はどこですか」の解「カイロ」の経緯度を検索するときのクエリは「site:ja.wikipedia.org intitle:カイロ (エジプト OR 首都)」となる。

質問文中のキーワードをクエリに追加するため、曖昧性を排除してページにアクセスできる。経緯度情報は英語版のページに記載されていることが多いため、日本語版のページからリンクを辿ってアクセスし、パタンマッチで「30° 03′ N 31° 22′ E」のような経緯度情報を抽出する。

Wikipedia で経緯度情報が得られなかった場合は、住所検索と同様に Snippet からの抽出を試みる。

「ans (“北緯” OR “南緯”) (“東経” OR “西経”）」というクエリを Goo に送り、経緯度情報をパタンマッチで抽出する。得られた経緯度が日本の領域外であれば、これを採用する。

<sup>9</sup>現状では候補が複数あった場合に対応できていない。後に述べる Wikipedia を用いた経緯度検索はこれに対応している。

<sup>10</sup><http://ja.wikipedia.org/wiki/>

### 3.3.6 地図表示までの流れ

以上を踏まえて、地図表示までの流れは以下のようになる。

1. 質問応答の解でジオコーディングを行ない、日本の住所 (iConf が3以上) かどうかを判定。
2. 日本の住所の一部のとき
  - (a) 質問文タイプが 'place.prefecture' ならば、得られた経緯度をそのまま採用。
  - (b) 質問文タイプが上記以外の場合は「landmarkはどこにあるか」といった質問であると考え、ランドマーク名 landmark を抽出する。そして「ans landmark keywords (“住所” OR “所在地”）」をクエリとして3.3.3節の方法で詳細な住所を求める。これをジオコーディングし、経緯度を得る。
3. 日本の住所の一部ではなかったとき
  - (a) 3.3.5節の方法で経緯度情報を得る。
  - (b) 経緯度が得られなかった場合、解は日本のランドマークであると考え、「ans (“住所” OR “所在地”）」をクエリとして3.3.3節の方法で住所を求める。これをジオコーディングし、経緯度を得る。
4. Google Maps API を用いて地図を表示し、得られた経緯度の点にプロットする。

### 3.3.7 システム動作例 (地図)

地図を用いて回答する際のシステムの動作例を図に示す。図4(左)は「2006年冬季五輪の開催地はどこですか」という質問を入力した場合の例で、解の「トリノ」の位置が示されている。図4(右)は「西郷隆盛の銅像はどこにありますか」という質問を入力した場合の例で、解の「上野公園」の位置が示されている。

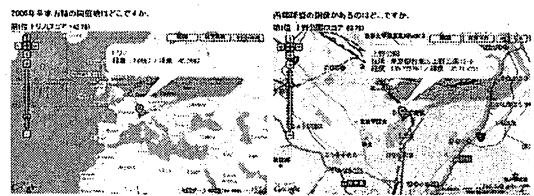


図4: 地図を表示する例(1)

また、図5は「世界三大夜景といたらどこのことですか」という複数の正解を持つ質問を入力した場合の例である。解のスコア分布から正解は3位までと判定され、「香港」「ナポリ」「函館」が解として出力される。このような場合は、1つの地図にまとめて解の位置を表示することができる。こうすることで、それぞれの位置関係が一目で把握できる

ようになり、これは複数の場所を解に持つ質問に対する効果的な回答の提示の仕方であるといえる。

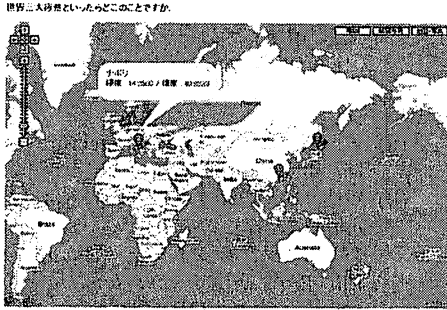


図 5: 地図を表示する例 (2)

## 4 評価実験

### 4.1 実験方法

質問応答システムが正しく解を出力できた場合、どの程度適切な画像や地図を出力できるかを評価する。実験に用いた質問文は全部で 100 問で、その内訳は表 2 の通りである。この質問群は内容が偏らないように 4 人で作成した。いずれも質問応答システムが正解を出力できた質問であり、正解は 1 つだけであるとする<sup>11</sup>。

表 2: 質問セットの内訳

	種類	数
画像	解が曖昧性のない語句になる質問 (例) 日本の国鳥は何ですか。	25
	解が曖昧性のある語句になる質問 (例) 1月の誕生日は何ですか。	25
地図	国に関する質問 (例) 世界最小の国はどこですか。	10
	解が海外の地名になる質問 (例) ブラジルの首都はどこですか。	10
	都道府県・市区町村に関する質問 (例) リンゴの出荷量が多い県はどこですか。	10
	解がランドマークになる質問 (例) 大相撲の聖地はどこですか。	10
	ランドマークの位置を問う質問 (例) 富士急ハイランドはどこにありますか。	10
	合計	100

### 4.2 画像を用いて回答する場合の評価

#### 4.2.1 評価方法

1 つの解に対して画像は複数件表示し、上位 3 件までを評価対象とする。質問は 50 問あるから、評価対象の画像は全部で 150 件となる。それぞれの画

<sup>11</sup> 正解が複数存在する質問では、そのうちの 1 つのみを評価対象とする。

像が質問の回答として適切であったかどうかは、次の 3 段階で評価する。

「○」: 適切。その画像に 1 点を与える。

「△」: 適切とはいえないが、間違ってもいないもの。対象以外の余計なものが目立つ場合や、対象が小さくて分かりにくい場合など。この場合はその画像に 0.5 点を与える。

「×」: 不適。この場合の得点は 0 である。

精度は次式で定義する。

$$\text{精度} = \frac{(\text{○の数}) + 0.5 * (\text{△の数})}{(\text{○の数}) + (\text{△の数}) + (\text{×の数})} \quad (1)$$

また、3 件の画像全ての評価が × であった場合は「失敗」とし、「失敗」となった質問の総数を「失敗数」とする。

#### 4.2.2 パラメタ設定

3.2.3 節の手法を実装する際、以下の要素をパラメタとして調整した。なお、質問文と周辺テキストの類似度を計算する際、それぞれを内容語の重みを要素とするベクトルに変換するが、前者を質問文ベクトル、後者を周辺テキストベクトルと呼ぶことにする。

- 画像の周辺テキストを解析する件数  $N$ 。
- 周辺テキストを切り出す範囲。画像へのリンク (<IMG> や <A> タグ) の前後  $M$  文。
- 質問文ベクトルと周辺テキストベクトルの要素の重みを、内容語の品詞の種類に応じて変えるかどうか。
- 解を抽出した文脈中の語も質問文ベクトルに加えるかどうか。
- 類似度が非常に低くなる場合、適切な画像が得られないと判断して手法を変更する。その際の類似度の閾値  $TH$ 。

これらのパラメタを変更しながら予備実験を行なった結果、以下のような傾向が見られた。

- 曖昧性がない質問に対しては、特に手を加えず解のみをクエリとして画像検索を行なった結果が最も精度が良い。
- 曖昧性がある場合の精度を上げるようにパラメタ設定を行なうと、曖昧性なし場合の精度が下がる。
- 画像の周辺テキストの解析件数を増やすと、曖昧性なしの精度は上がるが、全体としての精度は下がる。また、失敗数は少なくなるが、処理時間は大幅に増える。
- 周辺テキストを切り出す文の数は他のパラメタよりも精度に与える影響が大きい。
- 品詞の種類によって各ベクトルの要素の重みを変えても、結果にはあまり差が見られない。

- 解を抽出した文脈中の語も質問文ベクトルに加えると、精度は悪化する。
- 適切な画像が得られていない場合、類似度は0に近い非常に小さな値を取ることが多い。

この結果を考慮して次のようにパラメタを設定した。画像の周辺テキストの解析件数は  $N = 40$  件、周辺テキストは画像の前後  $M = 30$  文ずつを切り出す。質問文ベクトルはキーワードのみで構成し、各ベクトルの要素の重みは、品詞の種類によらず一定とする。手法を変更する閾値は  $TH = 0.03$  とする。この条件の下で以下の本実験を行なった。

#### 4.2.3 手法

次の4つの手法を比較する。

**手法1** ベースライン1。「ans」をクエリとして画像検索を行なった結果をそのまま出力。

**手法2** ベースライン2。「ans keywords」をクエリとして画像検索を行なった結果をそのまま出力。

**手法3** 手法1で得られた画像の周辺テキストを解析し、質問文との類似度が高いものを出力。

**手法4** 手法3で類似度が  $TH = 0.03$  以下になったら、手法2へ移行する。

#### 4.2.4 結果と考察

結果を表3に示す。ここで、「○」「△」「×」の各列の数字は、その評価が与えられた画像の総数を表す。「失敗」は失敗数を表す。

表3: 画像を表示する場合の結果

手法		○	△	×	精度	失敗
1	曖昧性なし	60	9	6	0.86	0
	曖昧性あり	23	2	50	0.32	11
	合計	83	11	56	0.59	11
2	曖昧性なし	32	16	27	0.53	3
	曖昧性あり	33	6	36	0.48	4
	合計	65	22	63	0.51	7
3	曖昧性なし	56	10	9	0.81	0
	曖昧性あり	31	5	39	0.45	7
	合計	87	15	48	0.63	7
4	曖昧性なし	57	9	9	0.82	0
	曖昧性あり	40	6	29	0.57	1
	合計	97	15	38	0.70	1

手法1は曖昧性なしの場合の精度は最高であるが、曖昧性ありの場合の精度は最低であり、失敗数も多い。手法2では曖昧性ありの場合の精度が上がっているが、曖昧性なしの場合の精度が大きく下がってしまう。この結果より、画像検索のクエリを工夫するだけでは全体としての精度向上はあまり望めないことが分かる。

手法3は曖昧性なしの場合の精度をあまり下げることなく、曖昧性ありの場合の精度を上げることが成功している。これより、画像の周辺テキストと質

問文の類似度を計算する手法は、解の曖昧性の解消に役立つといえる。曖昧性なしの場合の精度が若干低下してしまう要因としては次のことが考えられる。画像検索エンジンが返す上位の画像は、周辺テキスト以外の情報も使って精度を向上させている可能性がある。そのため、周辺テキストだけの情報に頼って順位付け行なうと、精度が下がってしまう可能性がある。

手法4では、手法3から曖昧性なしの場合の精度を下げることなく、曖昧性ありの場合の精度だけを大きく向上させている。さらに、失敗数はわずか1となり、残りの49問においては何らかの関連のある画像を表示できたことになる。これより、類似度を見ることにより適切な画像が見つかるかどうかを判断することは有効といえる。類似度が低いときに手法2に移行するようにしたのは、曖昧性ありの場合は手法2は比較的精度が良いことと、周辺テキストの解析を行なわないため処理時間の増加がほとんどないことによる。

さらなる精度向上のためには、手法2をそのまま使うのではなく、得られた画像の周辺テキストを解析することや、手法2に移行する閾値の決め方の工夫が必要であると考えられる。

#### 4.3 地図を用いて回答する場合の評価

##### 4.3.1 評価方法

1つの解に対して地図を1件表示するので、評価対象の地図は全部で50件となる。それぞれの地図が質問の回答として適切であったかどうかは次の3段階で評価し、精度は式1で計算する。

「○」: 適切。その地図に1点を与える。

「△」: 正しい位置よりも少しずれている、住所は合っているが位置は間違っているといった場合。この場合はその地図に0.5点を与える。

「×」: 不適。この場合の得点は0である。

##### 4.3.2 パラメタ設定

検索エンジン(Goo)を利用する際、Snippetの取得件数は50件とした<sup>12</sup>。

##### 4.3.3 結果と考察

結果を表4に示す。

表4: 地図を表示する場合の結果

	○	△	×	精度
国名	10	0	0	1.00
海外の地名	8	1	1	0.85
都道府県・市区町村	9	1	0	0.95
ランドマークが解	7	2	1	0.80
ランドマークの位置	5	3	2	0.65
合計	39	7	4	0.85

<sup>12</sup>これ以上増やしても結果はあまり変わらない。

国名や海外の地名に関する質問は、対象の経緯度情報が Wikipedia に存在することが多かったため、精度が良くなった。上手くいかなかったのは「ガンジス川」「サハラ」という大きな領域を持つ対象であった。これらの代表点を決めるのは難しいため、付近の地名などを手がかりに位置を求める必要があると考えられる。

都道府県・市区町村に関する質問で唯一上手くいかなかったのは「青森県大間町」で、これはジオコーディングの失敗であった。

ランドマークが解になる質問で、住所検索は成功したがジオコーディングで詳細な位置まで求めることができなかった質問が2つあった。完全に失敗したのは「ムク美術館」という海外のランドマークである。経緯度を直接求めることができなかった海外の地名（もしくはランドマーク）は現状では失敗になってしまう。

ランドマークの位置を問う質問の精度はあまり良くなかった。ジオコーディングの問題で位置がずれてしまった他には次のような失敗があった。「さいたまスーパーアリーナはどこにありますか」という質問では形態素解析に失敗し、ランドマーク名を上手く抽出できなかった。「マザー牧場はどこにありますか」の解「千葉県内」をランドマーク名と誤判定してしまった。

上記の失敗以外は概ね意図した通りに動作した。位置がわずかにずれている場合でも、表示された地図はユーザが自由に動かせるため、それほど問題は生じないと思われる。海外のランドマークが解の場合の対処法、ランドマークの位置を問う質問の精度向上は今後の課題である。

## 5 結論

### 5.1 まとめ

本稿では、質問応答の回答として、文字による解だけでなく画像や地図も表示できるシステムを提案した。評価実験の結果、画像は0.70、地図は0.85の精度で適切なものを表示できることが分かった。また、複数の正解を持つ質問に対して対応する画像をリストにして表示したり、一つの地図に複数の解の位置をまとめてプロットするといった、質問応答システムにおける新しい回答の提示方法を示した。

### 5.2 今後の課題

評価実験は質問応答処理が上手くいった場合を仮定していたため、質問応答の精度を加味すると精度は十分とはいえない。また、場所を問う質問の処理の一部、住所検索などで解の曖昧性の解消が不十分な箇所があり、第1章の要件2を完全には満たしていない。その部分の工夫も必要である。画像・地図ともにそれほど複雑な処理は行っていないため、まだ精度向上の余地はあると考えられるが、実時間

での質問応答を行なうためには処理時間との兼ね合いが課題となる<sup>13</sup>。

それぞれの解に対して「画像や地図を表示すること」自体が適切であったかどうかの評価や、ユーザの満足度といったインタフェースとしての評価も必要であろう。表示された画像が適切であったかどうかの評価も主観が入るため、評価方法には検討の余地がある。

画像や地図だけでなく、他のメディアへの拡張も考えらえる。また、提案システムでは回答に画像や地図を用いたが、これらを質問に使うという方法も考えられる。現状では容易ではないと思われるが、例えば、ある人物の画像を入力して「この人は誰ですか」と質問したり、地図上で位置または領域を示して「この近辺の観光名所はどこですか」といった質問ができると便利である。

## 参考文献

- [1] Atsushi Fujii, Katunobu Itou, and Tetsuya Ishikawa. Cyclone: An Encyclopedic Web Search Site. In *The 14th International World Wide Web Conference (WWW2005)*, pp. 1184–1185, 2005.
- [2] Atsushi Fujii, Katunobu Itou, and Tetsuya Ishikawa. Image Retrieval and Disambiguation for Encyclopedic Web Search. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, pp. 1598–1599, 8 2005.
- [3] Tatsunori Mori. Japanese Question-answering System Using A\* Search and Its Improvement. *Transactions on Asian Language Information Processing (TALIP)* Vol. 4, No. 3, pp.280-304, ACM, 2005.
- [4] 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. 自然言語処理研究会報告 2005-NL-169, pp.41–47, 情報処理学会, 9 2005.
- [5] 東京大学空間情報科学研究センター. CSIS シンプルジオコーディング実験(街区レベル位置参照情報) <http://pc035.tkl.iis.u-tokyo.ac.jp/~sagara/geocode/modules/simple-geocode1/>.
- [6] 佐藤理史. ワールドワイドウェブを利用した住所探索. 情報処理学会論文誌 Vol.42 No.1, pp.59–67, 情報処理学会, 1 2001.

<sup>13</sup> 計算機は Sun Blade 2000(CPU:UltraSPARC III Cu 1GHz ×2, メモリ:8GB)と Sun Fire V40z(CPU:Opteron 852 2.6GHz ×4, メモリ:16GB)を用い、Perl5でCGIを実装している。現状では1問あたり、画像の場合で20秒前後、地図の場合で10秒前後の処理時間を要する。