シリーズ型質問文に対して単純結合法を利用した遞減的 加減質問応答システム*

村田 真樹[†]・内山 将夫[†]・白土 保[†]・井佐原 均[†]

Decreased-Adding-Based Question-Answering System Using Simple Connection Method for Contextual Questions*

Masaki MURATA[†], Masao UTIYAMA[†], Tamotsu SHIRADO[†] and Hitoshi ISAHARA[†]

We participated in NTCIR QAC-1, QAC-2, and QAC-3, which were evaluation workshops for answering questions held by the National Institute of Informatics of Japan and studied question-answering systems for contextual questions. Contextual questions are defined as a series of questions with contexts. For example, the first question is “What is the capital of Japan?” and the one succeeding is related to the first such as “What was it called in ancient times?”. Contextual question-answering can be considered interactive. This paper describes our system for contextual questions. Our question-answering system used numerous effective methods to answer questions. We used a method of connecting question sentences to handle contextual questions. We also used a method of using multiple documents to obtain more accurate answers. We confirmed that these methods were effective in the experiments. Our question-answering system obtained the second best accuracy in QAC-1 and the best accuracy in both QAC-2 and QAC-3 for contextual question answering. It is thus a high-performance system.

1. はじめに

質問応答システムとは、与えられた質問に対してその答えを出力するシステムのことで、たとえば、「日本の首都はどこですか」という質問文が与えられると、「東京は日本の首都で、その国の最も大きく重要な都市であり、東京は日本の47都道府県のうちの一つである」という文をウェブや新聞記事などの電子テキストから探し出し、「東京」と答える。質問応答システムは情報検索の代りとして重要になるだろうし、また将来の人工知能システムの基本要素にもなるであろう重要なものである[1-3]。近年、多くの研究者がこの重要な質問応答システムの研究をしている[4-8]。質問応答システムに関する評価型ワークショップも開かれるようになり、米国ではTREC[9]が、日本ではNTCIRのQAC[10]がある。これらの評価型ワークショップは、複数のチームの性能

を同一の基準で比較することで技術の発展に貢献している。QACは、これまでにQAC1,QAC2,QAC3の三回が行われている。われわれは、早くから質問応答システムを研究し[11]、また、QAC[12]にも参加しながら質問応答システムの改良を続けてきた[2]。

QACは特にシリーズ型質問という興味深い研究トピックを扱っている。シリーズ型質問文とは、対話的な環境の質問応答を想定して、文脈的に関連する質問を逐次的に連続して行う質問文のことをいう。たとえば、「日本の首都はどこですか」と質問し、さらに「昔はそこはなんと呼ばれていましたか」と先に質問した内容に関連する質問をする。このシリーズ型質問応答は、人間と機械の対話につながる興味深い質問応答と思われる。また、一般の対話と異なり、逐次的に連続して行う質問文をあらかじめ作成する場合、それぞれの解答をあらかじめ作成しておくことが可能で、自動評価を行うことができる。科学的研究において評価は重要であり、シリーズ型質問では、自動評価ができるというメリットがある。われわれは国立情報学研究所の主催する評価型ワークショップ NTCIR QAC1,QAC2,QAC3に参加して、シリーズ型質問文に対する質問応答システムの研究を行ってきた[2,13,14]。本論文では、われわれの質問応答シス

* 原稿受付 2006年11月13日

[†] 独立行政法人情報通信研究機構 知識創成コミュニケーション研究センター Knowledge Creating Communication Research Center, National Institute of Information and Communications Technology; 3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, JAPAN

Key Words: question answering, contextual questions, simple connection, decreased adding, selection by ratio.

テムの説明をする。われわれの質問応答システムは、より正しい解答を得るために数多くの有効な手法を利用している。われわれのシステムでは、シリーズ型質問文を解くために過去の質問文を連結させた一つの長い質問文を作成しそれを元の質問文の代りに用いて質問応答処理をする。また、より正確な解答を得るために複数の記事の情報を組み合わせて利用する。われわれのシステムはQAC1, QAC2, QAC3のシリーズ型質問応答のコンテストにおいて、2位、1位、1位の精度をあげており、比較的性能の高いシステムである。シリーズ型質問応答はQAC1, QAC2ではtask3に対応する。QAC3ではシリーズ型質問応答のタスクのみが行われた。

2. 質問応答システムの構成

本節ではわれわれの質問応答システムについて述べる。このシステムは以下の三つの基本要素からなる[14]。

(1) 解表現の推定

システムは疑問代名詞の表現などに基づいて解表現（解がどのような言語表現か）を推定する。たとえば、入力された質問文が「日本の面積はどのくらいですか」だとすると「どのくらい」という表現から解表現は数値表現であろうと推測する。

(2) 文書検索

システムは質問文からキーワードを取り出し、これらのキーワードを用いて文書を検索する。この検索により、解が書いてありそうな文書群を集めることになる。たとえば、入力された質問文が「日本の面積はどのくらいですか」だとすると、「日本」「面積」がキーワードとして抽出され、これらを含む文書を検索することになる。

(3) 解の抽出

システムは解が書いてありそうな文書群から、推定した解表現に適合する言語表現を抽出し、それを解として出力する。たとえば、入力された質問文が「日本の面積はどのくらいですか」だとすると、文書検索で検索した「日本」「面積」を含む文書群から、解表現として推定した数値表現にあたる言語表現を解として抽出する。

2.1節から2.3節で、それぞれをより詳しく説明する。また本システムでは、複数の記事の情報を組み合わせて利用する通減加減法という方法も利用しており、2.4節でそれを説明する。また、本システムは解答が複数の場合への対応、シリーズ型質問文への対応の手法も備える。それぞれ、2.5節、2.6節で説明する。

2.1 解表現の推定

人手で作成したヒューリスティックルールを使って解表現を推定する。39個の規則を作成した。そのいくつかを以下に示す。

- 質問文に「誰」という表現がある場合、解表現は人名である。

- 質問文に「いつ」という表現がある場合、解表現は時間表現である。
- 質問文に「どのくらいの」という表現がある場合、解表現は数値表現である。
- 質問文に「どこ」という表現があり、質問文が「場所はどこ」「地域はどこ」などの形式の場合、解表現は地名である。
- 質問文に「どこ」という表現があり、質問文が「会社はどこ」「組織はどこ」などの形式の場合、解表現は組織名である。
- 質問文に「どこの国」という表現がある場合、解表現は国名である。
- 質問文に「首都はどこ」という表現がある場合、解表現は首都名である。
- 質問文に「読みは何」という表現がある場合、解表現は平仮名表現である。

そのほかに「X曜日」の形式を解表現とするもの、「X+名詞」の形式を解表現とするものなどの規則がある。

本研究では、固有表現抽出の技術を利用することで、文章中の固有表現（人名、地名、組織名、時間表現など）を抽出できる。また、国名辞書、首都名辞書をあらかじめ人手で作成して用意しており、文章中の国名、首都名を抽出できる。また、字種や文字列パターンを使って平仮名表現や「X曜日」「X+名詞」などの表現を抽出できる。本研究の解表現の推定規則は、これらの技術により取得可能な解表現をもとに、作成している。

2.2 文書検索

文書検索のためのキーワードはChaSen[15]により取り出し、付属語などはキーワードから除外した。

文書検索は以下のように行う。

- (1) まず以下の式で文書検索を行い、上位 k_{dr1} 個の記事を取り出す。

$$Score(d) = \sum_{\text{term } t} \left(\frac{tf(d,t)}{tf(d,t) + k_t \frac{length(d) + k_+}{\Delta + k_+}} \times \log \frac{N}{df(t)} \right) \quad (1)$$

ただし、 d は記事で t は質問文から取り出したキーワード、 $tf(d,t)$ は記事 d に出現するキーワード t の頻度、 $df(t)$ はキーワード t が出現する記事の数、 N は記事の総数、 $length(d)$ は記事 d の長さ、 Δ は記事長の平均である。 k_t 、 k_+ はそれぞれ0.00001、20である。これらの値は、QAC1の際のシステム構築用に与えられた質問文と解答のデータと、われわれが独自にシステム構築用に作成した質問文と解答のデータを利用して定めた。この式はロバートソンのOkapiウェイトニング[16]の式に基づく

もので、われわれも情報検索でよく用いる式である [17]. ただし、質問応答では多くの種類のキーワードがマッチすることが重要なので k_t の値としては極めて小さい値を用いる.

- (2) つぎに、以下の式で記事をランキングし、上位 k_{dr2} 個の記事を取り出す.

$$Score(d) = \max_{t1 \in T} \sum_{t2 \in T3} w_{dr2}(t2) \log \frac{N}{2dist(t1, t2) * df(t2)} \quad (2)$$

$$T3 = \{t | t \in T, 2dist(t1, t) \frac{df(t)}{N} \leq 1\} \quad (3)$$

ただし、 T はキーワードの集合、 $dist(t1, t2)$ はキーワード $t1$ と $t2$ の間の距離で、便宜上 $t1 = t2$ のとき $dist(t1, t2) = 0.5$ としている. $w_{dr2}(t2)$ は $t2$ が動詞のときに 0.5、それ以外の自立語のときに 1 にしている. $w_{dr2}(t2)$ は、QAC1 の際のシステム構築用に与えられた質問文と解答のデータと、われわれが独自にシステム構築用に作成した質問文と解答のデータを利用して定めた.

一般には質問応答システムでは、質問文から取り出した複数のキーワードが近くに出現することを保証するために記事を段落などの小さい単位に分割するが、本システムでは (2) 式のリランキングによりキーワードが近くにある場合に得点を増やす式を用いるので、記事を分割する必要がなく記事をそのまま文書検索に使うことができる.

本研究では k_{dr1} として 300 を k_{dr2} として 20 を利用した. この文書検索で上位 20 記事を取り出し、それを次の解の抽出で利用する.

2.3 解の抽出

文書検索で得た記事から、名詞、未知語連続を取り出しそれらを解の候補とする. それぞれの候補には、解の候補とキーワードの近さに基づく得点 $Score_{near}(c)$ と、解表現の意味制約を満足しているか否かに基づく $Score_{sem}(c)$ の二つの得点を与え、その合計点が最も大きい候補を解とする.

$Score_{near}(c)$ は以下の式で与えられる.

$$Score_{near}(c) = \sum_{t2 \in T3} w_{dr2}(t2) \log \frac{N}{2dist(c, t2) * df(t2)} \quad (4)$$

$$T3 = \{t | t \in T, 2dist(c, t) \frac{df(t)}{N} \leq 1\} \quad (5)$$

ただし、 c は解の候補であり、 $w_{dr2}(t2)$ は実験で定められる関数である.

解表現の意味制約に基づく得点 $Score_{sem}(c)$ は、人手で作成した規則により与えられる. 45 の規則を作成した. そのいくつかを以下に示す.

- (1) 推定した解表現 (人名や地名など) と一致する候

補に 1000 を与える. 解の候補が人名か地名かを特定する方法には SVM に基づく固有表現抽出技術を利用した.

- (2) 解表現が「国名」の場合、解の候補が国名のときに 1000 を与える.
 (3) 質問文が「何+名詞 X」の場合、名詞 X を末尾に持つ表現の候補に 1000 を与える.
 (4) 解表現が平仮名表現の場合、解の候補が平仮名表現のときに 1000 を与える.

そのほかに「X 曜日」の形式を解表現とする場合に「X 曜日」の形式に加点するものや、もの、数字表現を解表現とする場合に数字表現に加点する規則などがある.

本研究では、固有表現抽出の技術を利用することで、文章中の固有表現 (人名、地名、組織名、時間表現など) を抽出できる. また、国名辞書、首都名辞書をあらかじめ人手で作成して用意しており、文章中の国名、首都名を抽出できる. また、字種や文字列パターンを使って平仮名表現や「X 曜日」「X+名詞」などの表現を抽出できる. 本研究で用いる意味制約の規則は、これらの技術により取得可能な解表現をもとに、作成している.

これらの規則では基本的に適合すると、1000 という大きな得点を与える. これは、解の候補とキーワードの近さに基づく得点 $Score_{near}(c)$ と解表現の意味制約に基づく得点 $Score_{sem}(c)$ のうち、意味制約に基づく得点 $Score_{sem}(c)$ の方を絶対的に重要な情報として扱うためである. 本システムでは、意味制約をより満足する候補を優先し、意味制約の満足度が同じ候補については、解の候補とキーワードの近さに基づいて定める方法をとっている. $Score_{near}(c)$ の値はたかだか数十程度のため、 $Score_{sem}(c)$ の得点を 1000 にしておくことで、意味制約の情報を、解の候補とキーワードの近さの情報よりも優先的に利用できる.

2.4 複数記事の利用における逓減加点法の利用

「日本の首都はどこですか」という質問文が与えられたとしよう. このとき、得るべき答えは「東京」である. 一般的な質問応答システムは、Table 1 のように、解の候補と得点をリストとして出力でき、また、解の候補を取り出した記事を指し示す記事番号も出力することができる.

Table 1 の例だと、最も得点の大きい候補は「京都」であり、誤った解を出力することになる.

解の候補の得点を単純に加算する方法は、すでに提案されている [18]¹. 先の例だと、この方法は Table 2 の結果を得て、正しく「東京」を解として出力することができる. この方法は、複数の記事の情報を利用することで正しい解を得ることができた.

¹ほかに先行研究としては、複数記事から得た解の候補の頻度をその解の候補のスコアとして利用する方法 [19] がある.

Table 1 Candidate answers with original scores, where “Tokyo” is the correct answer

順位	解の候補	得点	記事番号
1	京都	3.3	926324
2	東京	3.2	259312
3	東京	2.8	451245
4	東京	2.5	371922
5	東京	2.4	221328
6	北京	2.3	113127
⋮	⋮	⋮	⋮

Table 2 Candidate answers with simply added scores, where “Tokyo” is the correct answer

順位	解の候補	得点	記事番号
1	東京	10.9	259312, 451245, 371922, 221328
2	京都	3.3	926324
3	北京	2.3	113127
⋮	⋮	⋮	⋮

しかし、この方法には、高頻度の解の候補を取り出しやすいという問題がある。これは、特に性能が高いシステムで深刻な問題であり、もともと性能が高いシステムでは、システムの出力した元の得点の方が単純に加算した得点よりも信頼できる場合が多く、単純に加算する方法はしばしばシステムの性能を劣化させることになる。

この問題に対処するために、われわれはQAC2のときに得点を減らしながら加算する方法を開発した。解の候補の得点を単純に加算する代りに、下位の順位の解ほど得点を大きく減らす重みをつけて得点を加算するのである。この方法は、高頻度語を取り出しやすいという悪い効果を減じ、なおかつ複数記事の情報を組み合わせることによるシステムの性能向上の効果を持つ。

この方法の有効性を示す例をあげる。「日本の首都は西暦 1000 年の時はどこでしたか」と質問が与えられ、システムは Table 3 の出力をしたとしよう。正解は「京都」であり、もともとこのシステムは正解を出力している。

しかし、単純に加算する方法を用いるとその結果は Table 4 のようになり、間違った解の「東京」をシステムの解としてしまう。

ここで得点を減らしながら加算するわれわれの新しい方法を利用する。細かい手法の仕様として、得点の高い順にソートしたうえ、 i 番目の候補の得点に $0.3^{(i-1)}$ を乗じることとする。その場合、「東京」の得点は 2.8 であり $(= 2.1 + 1.8 \times 0.3 + 1.5 \times 0.3^2 + 1.4 \times 0.3^3)$ 、システムの結果は Table 5 のようになり、「京都」の方がまだ点が高いので正解の「京都」を解として正しく出力することができる。

われわれの方法は最初の例（「日本の首都はどこですか」）でも正しい解を得ることができる。最初の例に適

Table 3 Candidate answers with original scores, where “Kyoto” is the correct answer

順位	解の候補	得点	記事番号
1	京都	5.4	926324
2	東京	2.1	259312
3	東京	1.8	451245
4	東京	1.5	371922
5	東京	1.4	221328
6	北京	1.3	113127
⋮	⋮	⋮	⋮

Table 4 Candidate answers with simply added scores, where “Kyoto” is the correct answer

順位	解の候補	得点	記事番号
1	東京	6.8	259312, 451245, 371922, 221328
2	京都	5.4	926324
3	北京	1.3	113127
⋮	⋮	⋮	⋮

Table 5 Candidate answers obtained by decreased adding, where “Kyoto” is the correct answer

順位	解の候補	得点	記事番号
1	京都	5.4	926324
2	東京	2.8	259312, 451245, 371922, 221328
3	北京	1.3	113127
⋮	⋮	⋮	⋮

用すると、「東京」の得点は 4.3 となり $(= 3.2 + 2.8 \times 0.3 + 2.5 \times 0.3^2 + 2.4 \times 0.3^3)$ 、結果は Table 6 のようになり、「東京」が最も高い得点となり解として正しく出力される。

Table 6 Candidate answers obtained by decreased adding, where “Tokyo” is the correct answer

順位	解の候補	得点	記事番号
1	東京	4.3	259312, 451245, 371922, 221328
2	京都	3.3	926324
3	北京	2.3	113127
⋮	⋮	⋮	⋮

以上述べたように、得点を減らしながら加算するわれわれの方法は、例にあげたどの例も正しく解を求めることができる。この方法は、高頻度の解の候補を取り出しやすい欠点を減じながら、なおかつ複数記事の情報を利用し精度向上を実現できる。

ここでは i 番目の候補の得点に $0.3^{(i-1)}$ を乗じることとしているが、この 0.3 がよいことは予備実験において確かめている [20,13]。

われわれのシステムではこの方法も利用する。ただ

し、このシステムでの実際の実装では、解表現の意味制約に基づく得点 $Score_{sem}(c)$ は、絶対的に重要な情報として扱うために、得点の加算は解の候補とキーワードの近さに基づく得点 $Score_{near}(c)$ のみで行う。すなわち、 $Score_{sem}(c)$ が同じ値であり解答が同じ候補において、 $Score_{near}(c)$ を上述のように得点を減らしながら加算し、その加算によって求めた値と $Score_{sem}(c)$ の値を足しあわせて最終的な候補の得点を求める。

2.5 解答が複数の場合への対応

QAC の問題では一つの質問の答えとして複数の解が存在する場合がある。たとえば、「常任理事国はどこですか」だと、「米国」「英国」「ロシア」「フランス」「中国」の五つが解となる。このように解が複数である場合を解決する方法として、われわれは QAC2 のときに、Select-by-rate 法（比率選択法ともよぶ）を提案している。これは、解の候補の最高得点に対しある割合以上の得点のものを解として取り出す方法である。本論文ではこの割合のことを Rate とよぶ。たとえば Rate を 0.9 とし、最高得点の解の候補が 1100 であったとすると、990 の得点のものまでを解として取り出す。

QAC1 の質問文では複数のものが解になる可能性があっても、解が一個の場合が多い時もあった。そういう場合に対応して、どういう質問文でも常に一個の解を出力する Select-one 法という方法を提案している [2]。さらにどういう質問文でも常に二個の解を出力する Select-two 法という方法も提案している [2]。

2.6 シリーズ型質問文への対応

シリーズ型質問文とは、対話的な環境の質問応答を想定して、文脈的に関連する質問を逐次的に連続して行う質問文のことをいう。たとえば、「日本の首都はどこですか」と質問し、さらに「昔はそこはなんて呼ばれていましたか」と先に質問した内容に関連する質問をする。

われわれのシステムではこのタイプの質問文への対応として、逐次的になされた質問文を現在の質問文まで次々と連結させて、その連結した文を一つの質問文として扱って、質問応答処理をする。ただし、現在の質問文以外の質問文での疑問詞はダミー文字（たとえば「@」）に置き換えて質問応答処理で疑問詞としてヒットしないようにしておく。

たとえば先の例であれば、「日本の首都は@ですか。昔はそこはなんて呼ばれていましたか」という質問文を作成し、それを質問応答処理する。本論文ではこの方法のことを Conc1 とよぶ。

また、以上のほかに、さらに現在の質問文以前の質問文の解も追加してつける方法も利用している。たとえば先の例で、「日本の首都はどこですか」の解としてシステムが「東京」と求めていた場合は、「日本の首都は@ですか。東京。昔はそこはなんて呼ばれていましたか」のように、「東京」も追加した文字列を質問文として作成し、

それを質問応答処理する。本論文ではこの方法のことを Conc2 とよぶ。

われわれはこれらの方法を QAC1 から利用している。本論文では Conc1, Conc2 を総称して単純結合法とよぶ。

3. コンテストでの結果

3.1 QAC1

NTCIR3 の QAC1 の Task-3（シリーズ型）の結果を Table 7 に示す。このワークショップでは、2 個の質問からなるシリーズ型質問文のセット 20 個が用いられた。評価には、MF とよばれる F 値の平均（各質問ごとに F 値を求め、その和を質問数で割ったもの）が用いられた。F 値は適合率と再現率の調和平均（逆数の平均の逆数）であり、適合率はシステムの出力のうちどのくらい正解したかを示し、再現率は本来の正解のうちシステムがどのくらい正解を出力したかを示す。QAC1 では通減的加点法は利用しなかった。われわれの MF は 0.17 で参加 7 チーム中 2 位であった。QAC1 の 1 位のチームは 0.19 であり、われわれの提出バージョンのシステムは劣っているが、提出後シリーズ型対応法を変更することで 0.23 の F 値という 1 位のチームを上回る精度を出している。ただし、合計 40 個の質問文しかなくデータ数が少なく結果の信憑性は低い。

Table 7 Results in QAC-1

	複数解選択	シリーズ型	MF
提出システム	Select-one	Conc1	0.167
提出後システム	Select-one	Conc2	0.225

3.2 QAC2

NTCIR4 の QAC2 の Task-3（シリーズ型）の結果を Table 8 に示す。このワークショップでは、5 から 10 個の質問からなるシリーズ型質問文のセットが 36 個、合計 251 個の質問文が用いられた。評価には、MF が用いられた。表の RunID は提出システムの識別番号である（当時のわれわれの組織の略称は CRL であった）。われわれのチームは 0.22 で参加 7 チーム中 1 位の精度であった。2 位のチームの精度は 0.20 であった。QAC2 の実験データの数が多かったが、評価を自動でできないデータであり、コンテスト後に実験を繰り返すことのできない不便なものであった。現在、オーガナイザ側で QAC2 のデータで評価を自動化する試みがなされている。

Table 8 Results in QAC-2

RunID	Rate	シリーズ型	MF
CRL1	0.95	Conc1	0.224
CRL2	0.97	Conc1	0.223

3.3 QAC3

NTCIR5 の QAC3 の結果を Table 9 に示す。このワークショップでは、5 から 10 個の質問からなるシリーズ

Table 9 Results in QAC-3

RunID	通減加点法	Rate	シリーズ型	MF		
				Total	First	Rest
NICT1	利用	0.90	Conc1	0.236	0.403	0.209
NICT2	利用	0.95	Conc1	0.250	0.450	0.218
NICT3	利用	0.90	Conc2	0.208	0.403	0.177
QAC3 の後のシステム	利用	0.95	Conc2	0.208	0.450	0.169

Table 10 Results in QAC-3 (reference run 1 data)

RunID	通減加点法	Rate	シリーズ型	MF		
				Total	First	Rest
NICT1	利用	0.90	—	0.305	0.403	0.289
NICT2	利用	0.95	—	0.314	0.450	0.292

Table 11 Results in QAC-3 (reference run 2 data)

RunID	通減加点法	Rate	シリーズ型	MF		
				Total	First	Rest
NICT1	利用	0.90	Conc1, Conc2 とも利用せず	0.090	0.403	0.039
NICT2	利用	0.95	Conc1, Conc2 とも利用せず	0.099	0.450	0.042

型質問文のセットが50個、合計360個の質問文が用いられた。評価にはMFが用いられた。表のRunIDは提出システムの識別番号である（われわれの組織の略称はNICTである）。表のTotal, First, Restはそれぞれ合計、シリーズ型質問のセットの先頭の質問、残りの質問を意味する。QAC3のデータは自動評価が可能であり、ワークショップの後でもシステムの設定を変更した実験を行うことができ、研究を進めるうえで便利なものとなっている。

われわれのチームは0.25で参加7チーム中1位の精度であった。2位のチームの精度は0.19であり、われわれのチームは2位にある程度大きな差をつけて1位の精度を得たことがわかる。Firstの精度はRestよりも高く、質問文に省略や参照表現のないシリーズ型質問文のセットの先頭の質問文は比較的簡単で、それ以外の質問文が難しいことがわかる。NICT1とNICT3を比較すると、以前の質問の解答も質問文に加えるConc2の方法を利用するNICT3の方が性能が低かった。QAC3のデータでは、Conc2はあまり役に立たないことがわかる。

QAC3では参照用データ1も配付され、そのデータでも実験が行われた。このデータは、シリーズ型の質問文を、省略表現や参照表現を補って通常の質問文に変換したものである。たとえば、「日本の首都はどこですか」と質問し、さらに「昔はそこはなんて呼ばれていましたか」と質問する例だと、「昔は東京はなんて呼ばれていましたか」という質問文に変換されこの質問文が実験に用いられる。シリーズ型の実験結果と比較することで、シリーズ型の問題がどのくらい難しいかを示すものとなる。このデータの実験結果をTable 10に示す。Total, Rest

ともに参照用データ1の方が精度が高いことがわかる。このため、シリーズ型の質問文よりも、省略表現や参照表現を補って通常の質問文に変換した方が解きやすいことがわかる。また、参照用データ1において、FirstとRestを比較すると同じ通常の質問文でも精度が大きく異なり、先頭以外の質問文が先頭の質問文よりもかなり難しいことがわかる。

われわれのシリーズ型質問文への対処法は、以前の質問文も連結させて用いる単純なものであるが、Restでは参照用データ1でのF値の0.75(=0.218/0.292)の精度を出している。

QAC3では参照用データ2も配付され、そのデータでも実験が行われた。このデータでは、質問文は元のデータのままである。ただし、それ以前になされた文脈的につながりのある質問を利用することができない。たとえば、「日本の首都はどこですか」と質問し、さらに「昔はそこはなんて呼ばれていましたか」と質問する例だと、「昔はそこはなんて呼ばれていましたか」という質問文だけを利用して質問応答処理をする。このデータの実験結果をTable 11に示す。MFの値が極めて低いことに気づく。一番よいNICT2で0.099である。Restでは参照用データ1でのF値の0.14(=0.042/0.292)の精度である。一方、単純に質問文を連結する方法のConc1を利用した実験では参照用データ1でのF値の0.75(=0.218/0.292)の精度を出していた。この結果は単純に質問文を連結するわれわれの方法のConc1の有効性を示している。

本実験では、単純に質問文を連結する方法のConc1が有効であったが、これは以下の説明をつけることができる。シリーズ型質問文の場合、後続する質問文では省略

Fig. 1 Results in various adding methods

や参照表現が出現し、その質問文だけで問題を解く場合低い得点となる。省略表現や参照表現によってその質問文に記述されていない内容は、前方の質問文または質問文の解答に存在する。質問文の解答となる単語は前方の質問文から類推可能である。その類推を助ける前方の質問文の単語を現在の質問文とあわせて利用した文は、現在の質問文の省略表現や参照表現により記述されていない内容を補ったものとみなすことができる。質問文を連結する Conc1 の方法では、現在の質問文の単語に加えて、現在の質問文の省略表現や参照表現の内容を補う前方の質問文の単語を、文書検索のキーワードや解の抽出の際の解の近辺に出現するキーワードとして利用することができる。このため、質問文を連結する Conc1 の方法により、シリーズ型質問文をある程度解くことができたものと思われる。

シリーズ型質問文の対処法としては、照応省略処理を行い現在の質問文の省略や参照表現の内容を特定し、その内容を補った完全な質問文を作成してその質問文を使って質問応答処理を行うものがある。省略箇所（ゼロ代名詞）の特定には格フレームを利用している。しかし、現在の自然言語処理の技術ではまだ高精度に省略や参照表現の内容を特定することは困難である。このため、省略や参照表現の内容の特定に失敗し精度が下がるケースが多い。したがって、本研究では照応省略処理を行わずに過去の質問を組み合わせた単純結合法を用いている。QAC3において省略や参照表現の内容特定の処理を行ったチームの最高の精度を Table 12 に示す¹。Table 9 のわれわれの NICT2 と比較すると Total, First, Rest とともにわれわれのシステムよりも性能が低い。また、Rest の精度の First の精度に対する比は、われわれのシステム (NICT2) が $0.48 (=0.218/0.450)$ で、省略や参照表現の内容特定処理を行ったチームが $0.38 (=0.13/0.34)$ である。省略や参照表現の内容特定処理を行ったチームの精度は、省略や参照表現の内容特定処理が必要な2問目以降の問題でとくに精度が下がっていることがわかる。

つぎに通減加点法について述べる。複数記事による加点の仕方をいろいろと変えた実験を行った。その結果を Fig. 1 に示す。図では、「加算無し」は通減加点法を利

Table 12 Results of a team using anaphora resolution in QAC-3

MF		
Total	First	Rest
0.16	0.34	0.13

用しない場合のもの、「頻度を利用」は賀沢らの方法 [19] を用い解答が出現した頻度を利用したものである。これは各記事における得点を 1 としてその解答が出現した記事分だけその得点を加算したものと等価である。「単純に加算」は高木らの方法 [18] を用い、各記事のスコアを単純に加えた値を利用するものである。その他のものは本論文の通減的加点法を利用したものである。k は通減的加点法の定数である。 $Score_{sem}(c)$ が同じ値であり解答が同じ候補において、i 番目の候補の $Score_{near}(c)$ に $k^{(i-1)}$ を乗じてから各記事でそれを加算し、その合計と $Score_{sem}(c)$ の和を最終的なその候補の得点とした。ただし、「頻度を利用」、「単純に加算」において賀沢らの方法、高木らの方法を用いる場合、われわれのシステムでの実装と同様、 $Score_{sem}(c)$ が同じ値の候補において、それぞれの方法を利用した。すなわち、「頻度を利用」では、 $Score_{sem}(c)$ が同じ値の候補において、解答が出現した記事の数（頻度）を求めてその数と $Score_{sem}(c)$ の和を最終的な候補の得点とした。また、「単純に加算」では、単純に $Score_{near}(c)$ を加算しそれを 100 で割ったものと $Score_{sem}(c)$ の和を最終的な候補の得点とした。Fig. 1 の棒グラフにつけた数字は Total の場合の MF の値である。k としては 0.3 がよいことは予備実験において確かめており [13,20]、QAC3 のわれわれの参加システムではその値を利用していた。Fig. 1 により QAC3 のデータの実験においても k としては 0.3 が最適であることがわかった。また、Fig. 1 により、通減的加点法は、通減的加点法を用いない方法、頻度を利用する方法、単純に加算する方法よりも性能が高いことがわかった。

つぎに Select-by-rate 法について述べる。Rate の値を 0.8, 0.85, 0.9, 0.95, 0.97, 0.99 と変える実験を行った。その結果を Fig. 2 に示す。Fig. 2 の棒グラフにつけた数字は Total の場合の MF の値である。QAC3 のデータでは、0.95 の場合が最もよいことがわかった。事前の研究 [13] で、Rate の値は 0.9 と 0.95 あたりがよいこ

¹ 精度の値は QAC3 のタスクオーガナイザ報告 [21] の棒グラフから読み取った。

とを確かめており、その結果とおおむね一致する結果となった。

Fig. 2 Results in various values of Rate

4. おわりに

われわれは国立情報学研究所の主催する評価型ワークショップ NTCIR QAC1, QAC2, QAC3 に参加して、シリーズ型質問文に対する質問応答システムの研究を行ってきた。シリーズ型質問文とは、対話的な環境の質問応答を想定して、文脈的に関連する質問を逐次的に連続して行う質問文のことをいう。たとえば、「日本の首都はどこですか」と質問し、さらに「昔はそこはなんと呼ばれていましたか」と先に質問した内容に関連する質問をする。本論文では、われわれの質問応答システムの説明をした。われわれのシステムは QAC1, QAC2, QAC3 のシリーズ型質問応答のコンテストにおいて、2位、1位、1位の精度をあげており、比較的性能の高いシステムである。われわれの質問応答システムは、より正しい解答を得るために以下の手法を利用している。一つは、シリーズ型質問文を解くために過去の質問文を連結させた一つの長い質問文を作成し、それをもとの質問文の代りに用いて質問応答処理をする方法である。もう一つは、より正確な解答を得るために複数の記事の情報を組み合わせて利用する方法である。実験を行い、これらの手法の有効性を確認した。

謝 辞

本研究では NTCIR QAC により作成されたデータを利用した。これらのデータがなければ本研究を遂行することは困難であったと思う。データを作成された方々を含め、NTCIR QAC の運営に携わった方々に対してここに深く感謝いたします。

参 考 文 献

- [1] 村田, 内山, 井佐原: 類似度に基づく推論を用いた質問応答システム; 自然言語処理研究会 2000-NL-135, pp. 181-188 (2000)
- [2] M. Murata, M. Utiyama and H. Isahara: A question-answering system using unit estimation and probabilistic near-terms IR; *Proceedings of the Third NTCIR Workshop (QAC)* (2002)
- [3] 村田: 質問応答システムの現状と展望; 電子情報通信学会会誌, Vol. 86, No. 12, pp. 959-963 (2003)
- [4] J. Kupiec: MURAX: A robust linguistic approach for question answering using an on-line encyclopedia; *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (1993)
- [5] A. Ittycheriah, M. Franz, W.-J. Zhu and A. Ratnaparkhi: IBM's statistical question answering system; *TREC-9 Proceedings* (2001)
- [6] D. Moldovan, M. Pasca, S. Harabagiu and M. Surdeanu: Performance issues and error analysis in an open-domain question answering system; *ACM Transactions on Information Systems*, Vol. 21, No. 2, pp. 133-154 (2003)
- [7] Y. Sasaki: Question answering as abduction: A feasibility study at NTCIR QAC1; *IEICE Transactions on Information and Systems*, Vol. E86-D, No. 9, pp. 1669-1676 (2003)
- [8] T. Takahashi, K. Nawata, K. Inui and Y. Matsumoto: Effect of structural matching and paraphrasing in question answering; *IEICE Transactions on Information and Systems*, Vol. E86-D, No. 9, pp. 1677-1685 (2003)
- [9] TREC-8 committee: Trec-8 question answering track (1999) <http://www.re-search.att.com/~singhal/qa-track.html>
- [10] NTCIR-3 QAC task committee: Question Answering Challenge (QAC) home page (2001) <http://www.nlp.cs.ritsumei.ac.jp/qac/>
- [11] M. Murata, M. Utiyama and H. Isahara: Question answering system using syntactic information (1999) <http://xxx.lanl.gov/abs/cs.CL/9911006>
- [12] National Institute of Informatics; *Proceedings of the Third NTCIR Workshop (QAC)* (2002)
- [13] M. Murata, M. Utiyama and H. Isahara: Japanese question-answering system using decreased adding with multiple answers; *Proceedings of the NTCIR Workshop 4 (QAC)* (2004)
- [14] M. Murata, M. Utiyama and H. Isahara: Japanese question-answering system using decreased adding with multiple answers at NTCIR 5; *Proceedings of the Fifth NTCIR Workshop* (2005)
- [15] Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, H. Matsuda and M. Asahara: *Japanese morphological analysis system ChaSen version 2.0 manual 2nd edition* (1999)
- [16] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu and M. Gatford: Okapi at TREC-3; *TREC-3* (1994)
- [17] 村田, 内元, 小作, 馬, 内山, 井佐原: 位置情報と分野情報を用いた情報検索; 言語処理学会誌, Vol. 7, No. 2 (2000)
- [18] T. Takaki and Y. Eriguchi: NTT DATA question-answering experiment at the NTCIR-3 QAC; *Proceedings of the Third NTCIR Workshop (QAC)* (2002)

- [19] 賀沢, 加藤: 意味制約を用いた日本語質問応答システム; 情報処理学会 自然言語処理研究会 2000-NL-140 (2000)
- [20] 村田, 内山, 井佐原: 質問応答システムにおける逓減加
点法に基づく複数記事情報の利用; 情報処理学会 自然
言語処理研究会 2004-NL-160 (2004)
- [21] T. Kato, J. Fukumoto and F. Masui: An overview
of NTCIR-5 QAC3; *Proceedings of the Fifth NT-
CIR Workshop Meeting on Evaluation of Information
Access Technologies: Information Retrieval, Ques-
tion Answering and Cross-Lingual Information Ac-
cess*, pp. 361–372 (2005)