シリーズ型質問文に対する質問応答システム

村田 真樹　内山 将夫　井佐原 均
独立行政法人 情報通信研究機構　(murata @nict.go.jp)

1　はじめに

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．例えば，「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．本稿では，われわれの質問応答システムの説明をする．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行われた．本稿ではこの我々のシステムについて説明する．

2　質問応答システム

本節ではわれわれの質問応答システムについて述べる．詳細は文献 [1] を参照してほしい．このシステムは以下の三つの基本要素からなる．

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．例えば，入力の質問文が「日本の面積はどのくらいですか」だとする (図 1) 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．例えば，入力の質問文が「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．例えば，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逓減加点法という比較的新しい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 個のルールを作成した．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．

2.2　文書検索

文書検索のためのキーワードは ChaSen[2] 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なう．

1. まず以下の式で文書検索を行ない，上位 k_{dr1} 個の記事を取り出す．

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

   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で，$tf(d,t)$ は，記事 d に出現するキーワー

ド t の頻度で，$df(t)$ はキーワード t が出現する頻度で，N は記事の総数で，$length(d)$ は記事 d の長さで，Δ は記事長の平均である．k_t と k_+ は実験で定める定数である．この式はロバートソンの Okapi ウェイティング [3] の式に基づくもので，我々も情報検索でよく用いる式である [4]．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 の値としては大きな値を用いる．

2. 次に，以下の式で記事をリランキングし，上位 k_{dr2} 個の記事を取り出す．

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

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

ただし，T はキーワードの集合で，$dist(t1,t2)$ 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 $t2$ の関数で実験により定められる．

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，本システムでは式 1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あげる式を用いるので，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えるのである．この文書検索では上位 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いなかに基づく $Score_{sem}(c)$ 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．

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

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

表 1: 解の候補とオリジナルの得点 (「東京」が正解である場合)

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
...	...	...	...

表 2: 解の候補と逓減加点法に基づく得点 (「東京」が正解の場合)

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
...	...	...	...

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と特定する方法には 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」の場合に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 を最後に持つ候補に 1000 を与える．

2.4 複数記事の利用における逓減加点法の利用

さらに，われわれのシステムでは得点を減らしながら加算する比較的新しい逓減加点法と呼ぶ方法 [5] を利用している．本節ではこの方法について説明する．

例えば，「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，表 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．

表 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．

逓減加点法では複数の記事から与えられる得点を減じながら加算する．我々のシステムでは特に，細かいシステムの仕様として，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．その場合は，「東京」の得点は 4.3 となり ($= 3.2 + 2.8 \times 0.3 + 2.5 \times 0.3^2 + 2.4 \times 0.3^3$)，結果は表 2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．

2.5 解答が複数の場合への対応

QAC の問題では複数のものと一つの質問の答えになる場合がある．例えば，「常任理事国の国はどこですか」だと，「米国」「英国」「ロシア」「フランス」「中国」の 5 つが解答となる．このような複数のものが解答にな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 法というものを提案している．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という方法である．本稿ではこの割合のことを Rate と呼ぶ．例えば Rate を 0.9 として，最高得点の解の候補が 1100 であったとすると，990 の得点のものまで解として取り出す．

QAC の問題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．そういうときに対応して，どういう問題でも常に一個の解を解答とする Select-one 法という方法を提案している．さらにどういう問題でも常に二個の解を解答とする Select-two 法という方法も提案している．

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．例えば，「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（例えば「@」）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．

表 3: QAC1 での結果

	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

表 4: QAC2 での結果

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

例えば先の例であれば，「日本の首都は@ですか．昔はそこはなんと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．本稿ではこの方法のことを Conc1 と呼ぶ．

また，以上の他に，さらに現在の質問文以前の質問文の解答も追加してつける方法も利用している．例えば先の例で，「日本の首都はどこですか．」の解答としてシステムが「東京」と求めていた場合は，「日本の首都は@ですか．東京．昔はそこはなんと呼ばれていましたか．」のような，「東京」も追加した質問文を作成し，それを質問応答処理する．本稿ではこの方法のことを Conc2 と呼ぶ．

われわれはこれらの方法を QAC1 から利用している．

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 3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．評価は，MF と呼ばれる F 値の平均が用いられた．われわれの 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．

3.2 QAC2

NTCIR4 の QAC2 の Task-3(シリーズ型) の結果を表 4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

表 5: QAC3 での結果

RunID	Rate	シリーズ型	MF		
			Total	First	Rest
NICT1	0.95	Conc1	0.236	0.403	0.209
NICT2	0.90	Conc1	0.250	0.450	0.218
NICT3	0.95	Conc2	0.208	0.403	0.177

表 6: QAC3 での結果 (参照用データ)

RunID	Rate	シリーズ型	MF		
			Total	First	Rest
NICT1	0.95	—	0.305	0.403	0.289
NICT2	0.90	—	0.314	0.450	0.292
NICT3	0.95	—	0.305	0.403	0.289

計 251 個の質問文が用いられた．評価は，MF と呼ばれる F 値の平均が用いられた．表の RunID は提出システムの ID 番号である．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 の実験データの数は多かったが，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，オーガナイザー側で QAC2 のデータで評価を自動化する試みがなされている．

3.3 QAC3

NTCIR5 の QAC3 の結果を表 5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 50 個，合計 360 個の質問文が用いられた．評価は，MF と呼ばれる F 値の平均が用いられた．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．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，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．例えば，「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．」と質問する例だと，「昔は東京はなんと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．Total, Rest ともに参照用データ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，省略表現や照応表現を補って通常の質問文に変換したものの方が解きやすいことがわかる．また，参照用データにおいて，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，先頭の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，参照用データでの F 値の 0.75(=0.218/0.292) の精度を出している．

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．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．本稿ではこのシステムの説明を行った．

参考文献

[1] Masaki Murata, Masao Utiyama, and Hitoshi Isahara, Japanese question-answering system using decreased adding with multiple answers at ntcir 5, *Proceedings of the Fifth NTCIR Workshop*, (2005).

[2] Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Hiroshi Matsuda, and Masayuki Asahara, Japanese morphological analysis system ChaSen version 2.0 manual 2nd edition, (1999).

[3] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford, Okapi at TREC-3, *TREC-3*, (1994).

[4] 村田真樹, 内元清貴, 小作浩美, 馬青, 内山将夫, 井佐原均, 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).

[5] Masaki Murata, Masao Utiyama, and Hitoshi Isahara, Use of multiple documents as evidence with decreased adding in a Japanese question-answering system, *Journal of Natural Language Processing*, Vol. 12, No. 2, (2005).