factoid 型 WebQA における クエリ拡張に基づく複数情報源の組合せの効果

金井 明[†]　　石下 円香[†]　　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: {a-kanai,mitsuru,ishioroshi,mori}@forest.eis.ynu.ac.jp

1　はじめに

質問応答 (QA) は，質問に対して関連文書を探して提示するだけではなく，直接質問の答を提示してくれるシステムである． また, 近年では，新聞記事集合のような静的かつローカルな文書群ではなく, 文書が豊富で日々追加・更新される Web 文書を情報源とした質問応答システム (Web QA) が研究されている．通常， Web 検索エンジンを質問応答用に独自に用意するのは非現実的なので，既存の Web 検索エンジンが利用される．

我々は, 人名, 地名といった名称や数量を問う factoid 型質問応答において, 情報源となる文書の多様性を増し, 異なる文脈における解候補を見つけることを積極的に行うことにより, 求解精度が向上するのではないかと考え, 検討を行っている．特に, 投票手法のように頻度に基づく解候補の重みづけ手法において, 有効であると考えている．

例えば, 金井ら [7] において，複数の異なる検索エンジンが Web QA に使用できることに注目し，異なる Web 検索エンジンの出力する検索結果を組合せる手法を提案している．評価実験によれば，同手法により情報源の多様性を増す事ができ,Web QA の精度向上が確認された．しかし，同手法では，異なる検索結果 (snippet) を出力する複数の検索エンジンを必要とするという制約がある．そこで，本稿では，関連性フィードバックに基づくクエリ拡張により，検索エンジンへ入力する異なるクエリを生成することにより，検索エンジンの数を抑制しつつ，検索結果の多様性を増すことを試みる．特に，質問応答処理を 2 回行ない,1 段目の処理により得られた解候補を 2 段目の処理における文書検索クエリに追加する手法を提案する．さらに，同手法により得られた結果を組み合わせることによる求解精度の向上について調査を行う．

2　関連研究

初の Web QA システムの 1 つである START の最近の版では，複数の情報源を活用している [3]． Radev ら [6] は Web QA における確率に基づくアプローチを提案しているが，そこでは 3 つの主要な Web 検索エンジンを組合せて上位 40 文書を得ている．

これらの研究では異なる情報源から得られた文書を使用してはいるが，文書検索よりも後の段階においては，情報源の違いは考慮していない．これに対して，節 4で述べる我々の手法では，複数の検索エンジンの出力 (snippet) や，異なるクエリ拡張による検索結果 (snippet) を利用することにより得られた異なる情報源の間でのデータの冗長性を活用しようとしている．

一方, 関連性フィードバックによるクエリ拡張は情報検索の基本的な技法として広く利用されている [4]． 同手法において,1 段目の検索結果より関連語を抽出し, これを元の検索クエリに加えて 2 段目の検索を行う．追加された語が新たな文脈として機能するので, 初期検索クエリの中に含まれている語の意味の曖昧性が低減するとともに, 付け加えられた語に関連する文書が獲得されるので, 再現率の向上が期待される．

これに対して, 我々の手法においては, 関連語のうち, 質問応答タスクで最も重要である, 質問に対する解の候補のみを利用することにより, 質問文中の語と解候補が同時に現れる文書を積極的に獲得し, 解候補の優劣の判断に役立てることを試みている．

3　基本となる WebQA システム

本研究で使用する Web QA システムは、 Web 検索エンジンの出力を利用し, 日本語の factoid 型質問に対し, 日本語で答を返すシステムである． また, 数百もの Web 文書をダウンロードすることは，非常に時間がかかる処理であるので, 応答時間の短縮のために Web 検索エンジンによる短い抜粋出力である snippet を Web QA の情報源として用いている．

図 1にその Web QA システムの構成を示す．質問文解析部は利用者から質問文を受け取り，キーワードのリストや質問文の型などの情報を抽出する．キーワードのリストが検索質問として Web 検索エンジンに入力され， snippet 群が検索される．文照合部は，文書集合から抽出された文集合をパッセージ抽出部から受けとり，それらを処理する．ここで得られた各文を本論文では**検索文**と呼ぶ．各検索文中の各形態素が一つの解候補として扱われ，それらに対して，次節に述べる方法によりスコアが与えられる．

3.1　解候補のスコア付け

基本となる QA システムでは，解候補に対する複合的な照合スコアを採用している．

$$S(AC, L_i, L_q) = Sb(AC, L_i, L_q) + Sk(AC, L_i, L_q) + Sd(AC, L_i, L_q) + St(AC, L_i, L_q) \quad (1)$$

本論文ではこのスコアのことを**原スコア**と呼ぶ．このスコアは， i 番目の検索文 L_i にある解候補 AC に対し，質問文 L_q に関する以下の 4 つの部分スコアを計算し，その線形結合を求めたものである．

1. $Sb(AC, L_i, L_q)$: 文字 2-grams の観点で計算した照合スコア
2. $Sk(AC, L_i, L_q)$: キーワードの観点で計算した照合スコア
3. $Sd(AC, L_i, L_q)$: 解候補とキーワードの間の依存構造の観点で計算した照合スコア

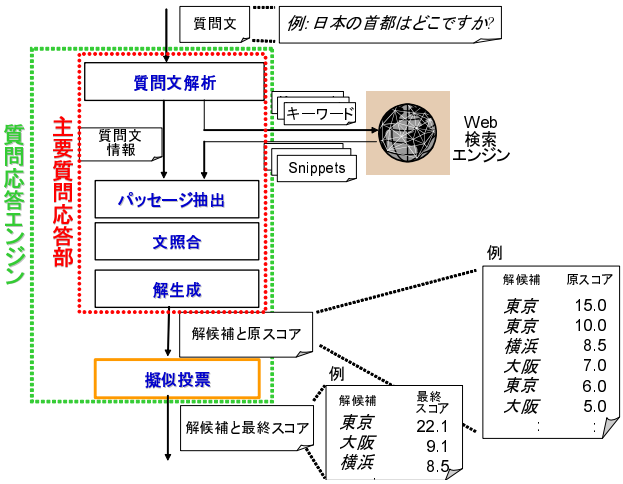

図 1: 基本となる WebQA システム

4. $St(AC, L_i, L_q)$: 質問文の型の観点で計算した照合スコア

なお, 計算量の削減のために, 上記スコア計算には A* に基づく探索制御が導入されている. この制御手法により, システムは最も有望な解候補のスコア計算を優先的に行ない, それ以外の解候補のスコア計算を遅延させることが可能となり, 解候補の n-best 探索ができる.

3.2 探索制御の枠組みにおける疑似投票手法

既存の多くの QA システムでは解候補に関する大域的な情報を利用している. 特に冗長性は最も基本的であり, かつ重要な情報である. 例えば, 文書中に複数回出現する解候補に対し, そのスコアを増加させるという投票手法がある [1].

一方で, 我々の QA システムのように解候補の探索に基づく枠組においては投票手法をそのまま利用できない. なぜならば, 同枠組では上位 n 件の解候補が見つかると検索をそこで終了してしまうので, 文書集合全体における解候補についての正確な頻度情報が得られないためである. そこで, 投票手法の一つの近似として, 以下に述べる疑似投票手法を導入している. まず, 上位 n 件の解が必要な場合には, **表層表現の異なる**解候補が n 個見つかるまで探索を継続していることに注意されたい. そのため, 探索の過程において, すでにスコア計算の終了状態に至っている解候補と全くおなじ表層表現を持つが別の解候補が新たに終了状態に至ることが有り得る. よって, 探索の過程において終了状態に至ったすべての解候補を記録することで, 解候補の頻度情報を部分的に利用できる. 本論文では, 解候補 AC に対する疑似投票スコア $S^v(AC, L_q)$ を次のように定義する.

$$S^v(AC, L_q) = (\log_{10}(freq(AC, AnsList)) + 1) \cdot \max_{L_i} S(AC, L_i, L_q) \quad (2)$$

ここで, $AnsList$ は n-best 探索において終了状態に至った解候補のリスト, $freq(x, L)$ を L における x の頻度とする. また, 本論文では, この疑似投票スコアを**最終スコア**と呼ぶことにする.

4 クエリ拡張とその結果を組合せる手法

解候補によるクエリ拡張とそれに基づく質問応答の結果を組み合わせる手法として以下の手法 A ならびに B を検討する. なお, ベースライン手法として, 金井ら [7] による, 複数の検索エンジンから得られた質問応答処理の結果を組み合わせる手法を用意し, これとも比較検討をする.

ベースライン手法: 各検索エンジンから得られた文書集合をそれぞれ個別の主要質問応答部に送り, 疑似投票を行なう前に, 得られた解候補 (原スコアを有する) の各リストを併合する. その後に, 式 (2) を使って疑似投票を行ない, 各解候補に対する最終スコアを得て, 解候補の順位付きリストを出力する.

手法 A: まず, クエリ拡張を行なわず節 3で述べた WebQA システムで最後まで処理を行なう. この処理を「第 1 段目」と呼ぶ. そこで得られた上位 n 件の解候補を追加単語としてクエリを拡張する. クエリ拡張の方法としてはクエリに各位の解候補を 1 つのみ加えるものとし,n 個の異なる拡張済みクエリを得る. こうして得られた n 個の拡張済みクエリの各々について検索された文書集合をそれぞれ個別の主要質問応答部に送り, 疑似投票を行う前の解候補 (原スコアを有する) を得る. これを第 2 段目と呼ぶ. 得られた解候補の各リストを併合し, 式 (2) を使って疑似投票を行い, 各解候補に対する最終スコアを得て, 解候補の順位付きリストを出力する. 次の手法 B と異なり第 1 段目、第 2 段目の処理のいずれにも同じ検索エンジンが使われる.

手法 B: 手法 A と同じであるが, 第 1 段目で利用する検索エンジンと第 2 段目で利用する検索エンジンを異なるものにする.

図 2にクエリの拡張例を示す. 検索クエリを拡張してクエリ中の語が増えると, 検索件数が低下し, 質問応答処理の結果, 「解候補なし」となる設問数が増える. これを防ぐため, 手法 A,B いずれにおいても, 追加単語は 1 つのみでクエリ拡張を行なっている. これは例えば解候補「1 位～ 3 位」を使ってクエリ拡張を行った場合, 1 位と 2 位と 3 位の解候補を同時に含む文書は少ないため「解候補なし」となる設問数が増えるためである. さらに, 加える解候補の数を増やすと少なくなっていくと考えられる. また, 図 3,4,5にベースライン手法, 手法 A,B をそれぞれ示す. 各図において, 併合部は, 複数の入力から各々データのリストを受けとり, それらを単純に併合してリストを生成する部分プログラムである.

5 評価実験

前節で述べた手法を評価, 検討するために, 以下の評価実験を行なった. 特に, 解候補によるクエリ拡張後の結果の組合せの数の効果を調査するために, 質問応答における各種設定を同一にした状況において, i) 各 Web 検索エンジンを単独で使用した場合, ii) Web 検索エンジンを 2 つ組合せた場合 (ベースライン手法), iii) 手法 A,B の各々において 1 種類のクエリ拡張のみを行なった場合, iv) 手法 A,B の各々において, 複数種類のクエリ拡張を行い, その結果を組合せた場合, の各々で実験を行なった. ここで,(iv) はクエリ拡張なしの単独の結果に順位順に (iii) の結果を組合せていく方法をとった.

評価に用いる質問文集合としては, NTCIR-3 QAC1 [2] の質問文 200 問のうち, 検索 API のタイムアウト等

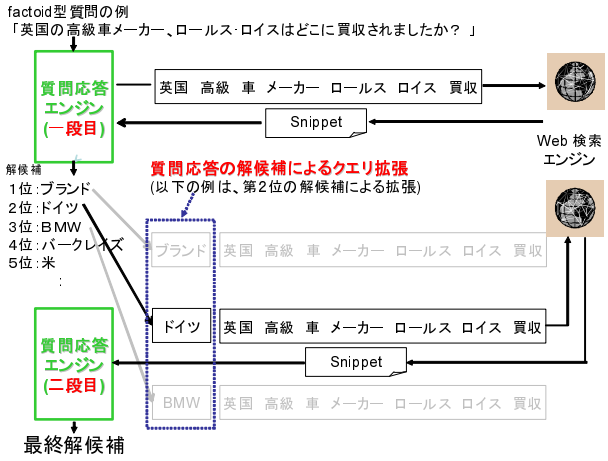

図 2: クエリ拡張例

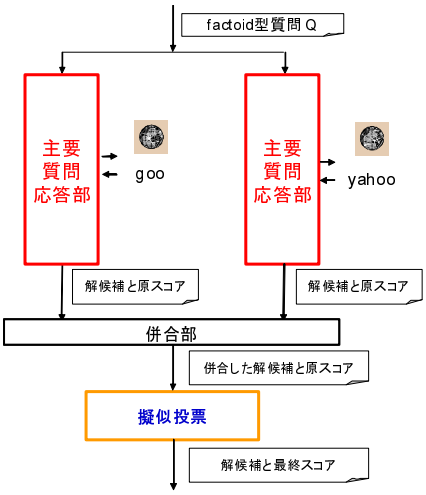

図 3: 複数の Web 検索エンジンを用いる QA システム: ベースライン

によりいずれかの QA エンジンが一定時間内に回答を返さなかった 34 問を除外した，計 166 問を評価に用いた． Web 検索エンジンとしては,goo, Yahoo! Japan, を使用した． 本実験の Snippet の取得は 2008 年 12 月 23 〜 25 日に行なった．

また, 本実験で設定した各種パラメタの値を表 1に示す．なお，一つのパッセージは隣接する 3 文から構成されている．

表 1: 評価実験における QA エンジンのパラメタ設定

パラメタ名	設定値	説明
a	10	検索すべき解候補の数
d	100	検索すべき文書 (snippet) の数
ppd	5	各文書 (snippet) から抽出するパッセージの最大数
p	30	求解で考慮すべきパッセージの最大数

ここで注意すべきことは，表 1の設定をいずれの実験においても共通に採用すると，組合せる結果の数を変化させた際に，検索される文書 (snippet) の延べ総数もそれに比例して変化する点である．ここにおいて何をもって公平な比較と考えるかは難しい問題である．もう一つの可能性としては，最終的にシステム全体で参照する文書 (snippet) の延べ総数が同じになるように，表 1のパラメタ d を調整することが考えられる．しかし, 表 2に

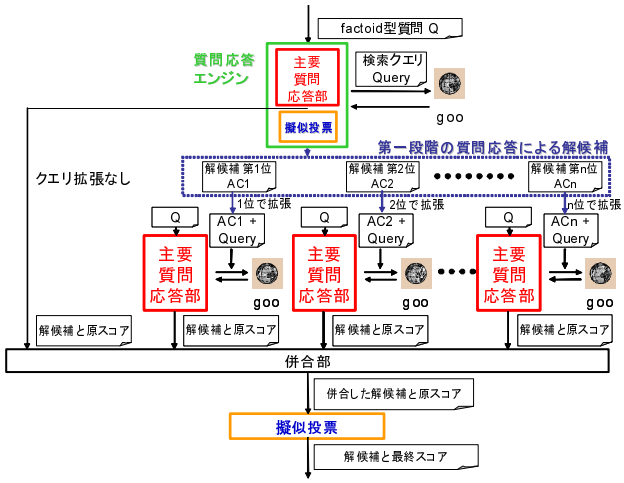

図 4: 1 つの Web 検索エンジン使ってクエリ拡張を行なった場合: 手法 A

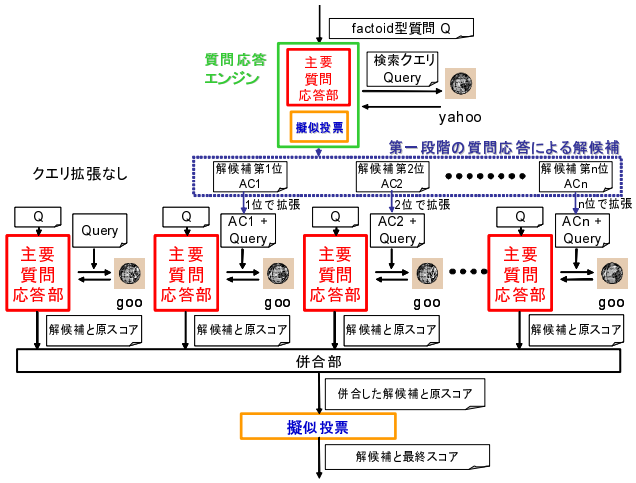

図 5: クエリ拡張に別の Web 検索エンジン使った場合: 手法 B

示す予備実験の結果より, 検索する文書数を増やすことが必ずしも求解精度の向上につながらないことが分かっている． これは，順位の低い文書まで参照することにより，その結果，原スコアは低いものの出現頻度が高い誤った解候補が出現しやすくなるためである．また, 同様のことが Mori[5] によって報告されている．

表 2: 検索する文書 (snippet) 数と求解精度 (goo の場合)

	1 位	2 位	3 位	4 位	5 位	MRR
250 件	70	23	7	6	6	0.433
500 件	67	18	9	4	6	0.406
750 件	59	21	16	10	5	0.392

6 実験結果と考察

求解精度の尺度として MRR(正解が最初に現れた順位の逆数の全質問平均) を採用した. その値を表 3に示す． 本論文では紙面の都合上，手法 A,B において, 第 1 段目と第 2 段目の両者に goo を用いた場合と第 1 段目に Yahoo! Japan を利用し, 第 2 段目に goo を用いた場合について示す．また，表の見方を以下に示す．

yahoo : Yahoo! Japan 単独の場合.

goo： goo 単独の場合.

goo+yahoo： ベースライン手法.yahoo 単独と goo 単独の疑似投票を行なう前の質問応答処理により得られた, 解候補 (原スコアを有する) を併合し, その後, 疑似投票により最終的な解候補と最終スコアを得たもの.

ggn： goo の検索結果を用いる第 1 段目の質問応答処理から得られた解候補 n 位のものを使ってクエリ拡張を行ない， goo の検索結果を用いる第 2 段目の質問応答処理を行なったもの.

g+gg1 ～ n： goo， gg1， gg2, …， ggn による処理により，疑似投票を行なう前に得られた解候補 (原スコアを有する) を併合し, その後, 疑似投票により最終的な解候補と最終スコアを得たもの.

gyn： yahoo の検索結果を用いる第 1 段目の質問応答処理から得られた解候補 n 位のものを使ってクエリ拡張を行ない， goo の検索結果を用いる第 2 段目の質問応答処理を行なったもの.

g+gyn： yahoo， gy1， gy2, …， gyn による処理により，疑似投票を行なう前に得られた解候補 (原スコアを有する) を併合し, その後, 疑似投票により最終的な解候補と最終スコアを得たもの.

表 3: 各条件における, 解候補各順位の正解数と MRR

	1 位	2 位	3 位	4 位	5 位	MRR
yahoo	61	17	9	3	5	0.447
goo	59	11	7	11	4	0.424
goo+yahoo	72	15	7	8	5	0.511
gg1	63	14	5	6	4	0.446
gg2	44	16	3	3	6	0.331
gg3	43	16	9	5	6	0.340
gg4	57	10	8	3	1	0.395
gg5	43	22	6	2	4	0.345
gg6	44	14	12	1	2	0.335
gg7	45	16	6	4	5	0.343
gg8	49	10	6	9	4	0.356
gg9	46	15	7	6	3	0.349
gg10	48	11	8	6	8	0.357
g+gg1	61	11	8	10	7	0.440
g+gg1 ～ 2	63	13	13	4	5	0.457
g+gg1 ～ 3	61	18	10	6	5	0.457
g+gg1 ～ 4	65	16	10	6	5	0.475
g+gg1 ～ 5	63	19	11	5	7	0.475
g+gg1 ～ 6	63	19	12	6	5	0.476
g+gg1 ～ 7	64	19	9	5	7	0.477
g+gg1 ～ 8	68	19	7	4	4	0.492
g+gg1 ～ 9	69	19	7	4	5	0.499
g+gg1 ～ 10	68	21	8	6	4	0.503
gy1	68	11	3	1	5	0.456
gy2	50	16	8	3	2	0.372
gy3	54	17	7	3	8	0.405
gy4	54	14	5	4	3	0.387
gy5	51	11	2	9	3	0.362
gy6	44	11	14	3	4	0.336
gy7	52	11	4	5	2	0.364
gy8	47	16	9	7	0	0.360
gy9	52	6	4	5	3	0.351
gy10	47	16	7	1	2	0.349
g+gy1	64	20	9	4	2	0.472
g+gy1 ～ 2	67	20	8	3	4	0.489
g+gy1 ～ 3	70	17	12	6	3	0.510
g+gy1 ～ 4	75	14	11	6	6	0.532
g+gy1 ～ 5	75	14	10	6	5	0.529
g+gy1 ～ 6	77	13	10	5	5	0.537
g+gy1 ～ 7	77	15	10	1	5	0.537
g+gy1 ～ 8	76	16	11	1	6	0.537
g+gy1 ～ 9	75	20	8	1	7	0.538
g+gy1 ～ 10	76	19	8	3	5	0.542

表 3によれば,1 つの検索エンジンのみを利用する状況において,goo 単独の場合よりもクエリ拡張を行なってその結果を組合せた場合のほうが精度がよくなることがわかる. すなわち, 単一の検索エンジンしか利用できない状況においても, 提案手法に示すフィードバック手法を用いることにより, 精度を向上させることが可能であることがわかる. また, 第 1 段目で利用する検索エンジンと第 2 段目で利用する検索エンジンが同じである手法 A より, 第 1 段目で利用する検索エンジンと第 2 段目で利用する検索エンジンが異なる手法 B のほうが精度がよいことがわかる. 特に,g+gy1 ～ 4 以降においてはベースライン手法より精度がよいということがわかる.

また,1 種類のクエリ拡張のみを行なった場合をみると精度が低下している. 特に, 追加単語として順位の低い解候補を使用するほど精度が劣化することがわかる. しかし, これらの結果を組合せると精度が向上している. これは疑似投票手法による多数決処理が非常に良く効いているためだと考えられる.

7 おわりに

本論文では, 質問応答処理を 2 回行ない,1 段目の処理により得られた解候補を 2 段目の処理における文書検索クエリに追加する手法を提案し, その効果を検証した. 評価実験より, クエリ拡張を行なうことで検索結果の多様性を増すことができ, その結果を組合せることで精度が向上することが示された.

今後の課題としては，更に精度の良い質問応答を実現するために, a) Web 検索エンジンの出力の，より効果的な組合せ手法，b) snippet における表現の多様性を積極的に引き出すために， Web 検索エンジンに入力する検索要求を複合語を含む形や文節の形に整形する手法，などについて検討したい.

謝辞

評価型ワークショップ NTCIR の企画運営にご尽力されている皆様に感謝いたします.

本研究の一部は文科省科研費特定領域「情報爆発 IT 基盤」 (課題番号 19024033) によるものである.

参考文献

[1] Charles L.A. Clarke, Gordon V. Cormack, and Thomas R. Lynam. Exploiting redundancy in question answering. In *Proceedings of SIGIR '01: the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 358–365, 2001.

[2] Jun'ichi Fukumoto, Tsuneaki Kato, and Fumito Masui. Question Answering Challenge (QAC-1) — Question answering evaluation at NTCIR Workshop 3 —. In *Working Notes of the Third NTCIR Workshop meeting – Part IV: Question Answering Challenge (QAC1)*, pp. 1–6, 2002.

[3] B. Katz, M. Bilotti, S. Felshin, A. Fernandes, W. Hildebrandt, R. Katzir, J. Lin, D. Loreto, G. Marton, F. Mora, and O. Uzuner. Answering multiple questions on a topic from heterogeneous resources. In *Proceedings of TREC 2004*, 2004.

[4] Christopher D. Manning, P. Raghavan, and H. Schutze. Introduction to information retrieval. In *Cambridge University Press*, 2008.

[5] Tatsunori Mori. Japanese question-answering system using A* search and its improvement. *ACM Transactions on Asian Language Information Processing (TALIP)*, Vol. 4, No. 3, pp. 280–304, 2005.

[6] Dragomir R. Radev, Weiguo Fan, Hong Qi, Harris Wu, and Amardeep Grewal. Probabilistic question answering on the web. *Journal of the American Society for Information Science and Technology*, Vol. 56, No. 3, March 2005.

[7] 金井明, 佐藤充, 石下円香, 森辰則. factoid 型質問応答における異なる web 検索エンジンの組合せの効果. 言語処理学会第 14 回年次大会発表論文集, pp. 1013–1016, 2008.