QA コンテンツに対する別解の発見とランキング

高田　夏希† 　大島　裕明† 　小山　聡†† 　田中　克己†

† 京都大学大学院情報学研究科社会情報学専攻 〒 606-8501 京都府京都市左京区吉田本町
†† 北海道大学大学院情報科学研究科複合情報学専攻 〒 060-0814 札幌市北区北 14 条西 9 丁目
E-mail: †{takata,ohshima,tanaka}@dl.kuis.kyoto-u.ac.jp, ††oyama@ist.hokudai.ac.jp

あらまし　本稿では，与えられた質問回答コンテンツに対し，コンテンツ内の回答とは異なる別解情報を発見する手法及び別解情報のランキング手法を提案する．質問に含まれる語や回答に共通して現れる語を用いて生成した Web クエリを実行する．得られた Web 検索結果の各ページについて、ページが回答を含む可能性の高さを表すスコア付けと、別解を含む可能性の高さを表すスコア付けを行い，それらのスコアをもとにランキングを行う．

キーワード　QA コンテンツ, 別解情報検索

1. はじめに

近年，ユーザ同士が質問や回答をやりとりして情報を得ることのできる Yahoo!知恵袋(注1)や教えて!goo(注2)などの質問応答サイト（以下 QA サイト）の利用者が増えている．ユーザの投稿した質問や回答は QA サイトに蓄積されていくため，QA サイトには「質問，回答をする」以外に「過去に投稿された記事を閲覧する」という利用方法もある．株式会社リアルワールド リアルサーチのアンケートによると，多くのユーザは QA サイトを過去の記事を閲覧するために利用するという調査結果がでている(注3)．

過去に投稿された質問の内容と自分が疑問に思ったことが一致すれば，ユーザはその質問に与えられている回答を見ることで自らの疑問に対する何らかの情報を得ることができる．

しかし，QA サイトにおける質問回答コンテンツ（質問と 1 つ以上の回答の組，以下 QA コンテンツ）には次のような問題点があると考えられる．

- 回答を投稿できるのは QA サイトのユーザのみであるため，質問の内容によっては回答可能なユーザが少ない場合がある．この場合，質問に対して十分な回答情報のない QA コンテンツが生成される可能性がある．
- ある質問に対し有益な情報を持つユーザが存在していても，そのユーザが質問の存在に気づかなければ回答情報が与えられないという状況が起こりうる．Yahoo!知恵袋の場合，質問に対し回答の投稿可能な期限が設けられており，十分な回答情報が与えられないまま回答が締め切られてしまう可能性もある．

回答が不十分な QA コンテンツの質問には，その質問に対して既に投稿された回答以外にも（別解となり得る）回答（以下別解情報）が存在するケースがある．QA コンテンツから情報を得ようとするユーザに対し別解情報を提示することは，ユーザの疑問を解決するための情報の種類を増やすこととなり，ユーザの疑問解決の補助につながるものと考えられる．

そこで，本稿では QA コンテンツの質問に対する別解情報を含む Web ページを取得する手法を提案する．Web 上には Yahoo!知恵袋ユーザの知らない知識も存在するものと思われるため，Web ページに別解情報を求めることで Yahoo!知恵袋の QA コンテンツでは得られない情報が取得できると考えられる．

別解情報の取得は「検索クエリを生成する」,「Web ページが質問に対する回答を含むか判定する」,「回答を含むと判定された Web ページに別解情報が含まれるか判定する」という 3 つの段階を踏む必要がある．まず，QA コンテンツの質問と回答を用いて Web 検索クエリを生成し，検索エンジンを用いて検索結果のページ集合を得る．次に，各ページについて「質問に対する回答情報が含まれているか」を判定する．最後に，回答を含むと判定されたページについて「含まれる回答情報には QA コンテンツ内の回答と異なる情報が含まれるか」を判定し，別解情報を含むと判定されたページ集合を別解を含む可能性の高さを表すスコアをもとにランク付けして提示するというのが別解情報取得の大まかな流れとなる．

以降，2 節では関連研究を述べ，3 節で別解情報取得までの流れ，4 節で各段階における提案手法，5 節で本稿のまとめについて述べる．

2. 関連研究

Web 上の QA コンテンツを対象にした研究はすでにいくつか存在する．Berger らは FAQ アーカイブから回答を検索する手法を提案している [1]．質問文で用いられる語と回答文で用いられる語には隔たりがあることから回答検索は一般の文書検索とは性質が異なることが述べられており，ある語が回答文中に出現する時に質問文中ではどのような語が現れやすいかを質問と回答のペアから学習して回答検索に利用する手法が提案されている．

QA アーカイブからの質問検索に関しては言語モデルや言い換え確率モデルを用いた Jeon らの研究 [2] や Xue らの研究 [3] があげられる．意味的に類似した質問同士でも用いられる語彙が異なることで検索できないという問題が QA 検索には存在す

(注1)：http://chiebukuro.yahoo.co.jp/
(注2)：http://oshiete.goo.ne.jp/
(注3)：http://japan.internet.com/research/20090909/1.html

る．そこで内容の似た質問のそれぞれに対して付けられた回答同士の類似性を利用して意味的に類似した質問を集め，それを利用して質問文の言い換えを学習することで 1 つのクエリから意味的に類似した質問を多く検索することを提案している．Wang らも QA サイト内にある質問中から類似した質問を検索するといった研究 [4] を行っているが，この研究は言語モデルではなく，質問文を構文木に変換したものを利用して質問同士の類似度を測っている．これらの研究 [1]〜[4] は類似する質問や回答を対象アーカイブ中から検索するというものであり，本研究の，Web 中の質問とそれに対する回答という形を取らない情報（Web ページ）から別解情報を検索するものとは異なっている．ユーザが入力した自然文形式の質問に対する回答文検索に関する研究としては，新聞記事を対象にした NTCIR ワークショップの QAC タスク [5] や Web 文書を知識源とするシステムの研究 [6], [7] などがある．研究 [6], [7] はいずれも自然文入力された質問からキーワードを抽出し，検索エンジンにより得られた結果から回答を抽出することに着眼したものである．佐藤らの研究 [6] では本研究と同様に質問に対する回答を Web から取得する方法を提案しているが，QA コンテンツへの補完ではなくユーザの入力した質問に対する回答取得が目的である点で本研究と異なっている．田村ら [7] は質問文に含まれる語からクエリを生成し回答文検索を行っているため，妥当な回答文にクエリ語が含まれていない場合はそのような回答を収集することは困難である．本研究では QA サイトの質問文だけでなく回答文からも特徴語を抽出して検索する手法を提案することで多くの回答情報を得ることを目指している．

島田らは一つの質問に対し複数の回答者がいる場合の回答者間の共起関係から興味分野の重なりを抽出し，他の QA コンテンツを関連付ける手法の提案を提案している [8]．この研究は QA コンテンツ内のみを対象としている点で本研究と異なっている．

3. 別解情報の取得問題の定式化

本節では，Yahoo!知恵袋に存在する QA コンテンツが与えられた際に，その QA コンテンツに対する別解情報を含む Web ページの取得する問題の定式化を行う．

まず，ユーザは，QA サイトにおいて，1 つの QA コンテンツを閲覧しているという状況を想定している．システムの出力として，その QA コンテンツのみが入力として与えられた時に，別解が含まれる Web ページを出力として返すのが理想的であると思われる．しかし，QA コンテンツの質問文には，質問の主要な部分に加えて余計な情報が多く含まれることや，複数の質問内容が含まれることなどもめずらしくなく，それらから，ユーザが現在どのような質問に対する別解を求めているのかを自動的に判断するのは困難である．そこで，今回は，閲覧中の QA コンテンツから，ユーザが現在別解を求めたいと考えている質問の主要な部分を指定してもらうこととする．このような入力から，最終的に別解を含むと考えられる Web ページを返すには，以下の 3 つの段階を経る．

Step 1. 入力された情報を基にして，Web 検索エンジンを利用して Web ページを収集する．

Step 2. 取得された Web ページに QA の回答となる情報が含まれているか判定する．

Step 3. 回答を含むと判定された Web ページに別解情報が含まれるか判定する．

はじめに与えられる入力は，QA コンテンツと，質問文からユーザが選択した部分である．QA コンテンツは，1 つの質問文と，複数の回答文からなる．質問文を q とし，また，それに対する回答文の集合を A とする．個々の回答文は，$a_1, a_2, ...$ とし，すなわち，$A = \{a_1, a_2, ...\}$ である．ユーザは，q の一部を選択するが，選択された文を u とする．以上の (u, q, A) の組が入力である．

[Step 1] 入力を基にして，Web 検索エンジンに対するクエリを作成し，検索された Web ページの収集を行う．取得された Web ページ集合を $P = \{p_1, p_2, ...\}$ と表すとすると，以下のような関数を作成することになる．

$$P = CollectPages(u, q, A) \tag{1}$$

[Step 2] 収集された個々の Web ページが，入力として与えられた質問に対する回答を含んでいるかどうかを表すスコア付けを行う．この段階では，書かれている回答が，QA コンテンツに既存の回答であるか，別解であるかは考慮しない．このスコアは，以下の関数で計算するものとする．

$$IncAns(u, q, A, p_i) \tag{2}$$

Step 1 で得られた P の要素それぞれに対してこの関数の値を取得する．

[Step 3] Web ページに対し，回答を含むかどうかを表すスコア付けを行った後は，そのページが，既存の回答にはない別解を含むかどうかについてのスコア付けを行う．そのようなスコアの計算のために，以下のような関数を作成する．

$$JudgeAlt(u, q, A, p_i) \tag{3}$$

ユーザに対する出力としては，ページが回答情報を含み、かつ別解を含む可能性の高い順に Web ページ集合の順位付けを行い，それらを提示するということが考えられる．そのため、回答を含む可能性を表す $IncAns(u, q, A, p_i)$ の値と別解を含む可能性を表す $JudgeAlt(u, q, A, p_i)$ の値を利用して P の順位付けを行うことが考えられる．本稿では，両者を考慮した以下のようなランキング関数を用いるものとする．

$$Rank(u, q, A, p_i) = \Phi(IncAns(u, q, A, p_i), JudgeAlt(u, q, A, p_i)) \tag{4}$$

各段階においては，様々な実装が考えられる．次節では，別解情報を含むページの取得とランキングを行うための各段階における実装について，我々が考案した手法を説明する．

4. 別解情報を含む Web ページの発見手法の実装

4.1 入力を基にした Web ページの収集

我々が今回対象とした QA サイトは，Yahoo!知恵袋である．

入力として与えられる質問文 q とそれに対する回答文集合 A は，Yahoo!知恵袋において解決済となっている QA コンテンツから得られる．入力として，それに加えて q からユーザが選択した部分 u がある．我々は，この入力から，質問を表現するような語集合を取得し，それらを AND 条件とするクエリを作成することで，Web 検索エンジンを用いた Web ページ収集を実現する手法を提案する．

質問を表現するような語は，質問文 q やユーザが指定した部分である u に含まれていると考えられる．よって，クエリの作成に用いる語は全て，q や u に含まれている語とする．

ここで，文から語集合を抽出する機能が必要となる．実装では，形態素解析器 MeCab(注4)を用いて文の形態素解析を行い，名詞と動詞のみを取得した．文 $sentence$ から抽出された語集合を以下のように表現する．

$$ExtractWords(sentence) \tag{5}$$

q や u から抽出された語集合は，$ExtractWords(q)$ と $ExtractWords(u)$ と表される．

q や u に現れるような語の中でも，回答文集合 A でよく出現するような語は，質問を特徴付ける語としてより適したものであると考えられる．しかし，同時に，q には（u にも），質問文でよく使われるような語（例えば,「質問」「教える」など）も現れており，また，q と A において，一般的な語（例えば,「する」）も現れているものと考えられる．そこで，q や u に含まれる語について TF-IDF による重み付けを行うことで，質問をよりよく表現するような語の取得を試みる．

TF 値の取得には，q と A を考慮する．$k \in ExtractWords(q)$ または，$k \in ExtractWords(u)$ であるような語 k の TF 値 tf_k は以下のように計算される．

$$tf_k = count(k, q) + \sum_{a \in A} count(k, a) \tag{6}$$

ただし，$count(k, sentence)$ は，文 $sentence$ 中に現れる語 k の回数を表すものとする．

IDF 値の取得には，対象となる語 k が現れる文書の数 df_k と，総文書数 N が必要である．今回，IDF を計算する際に対象とする文書は，Yahoo!知恵袋の全ての QA コンテンツとする．そこで，df_k は Yahoo!知恵袋 Web API を用いて，語 k で検索を行ったときのヒットカウントを用いる．また，$N = 35,108,527$ とする．これは，1 月 9 日時点において Yahoo!知恵袋のトップページに記載されている QA コンテンツの総数である．語 k の IDF 値は次の式で表される．

$$idf_k = \frac{N}{df_k} \tag{7}$$

以上で得られた tf_k,idf_k を用いて k の TF-IDF 値 $v(k)$ は以下のように計算される．

$$v(k) = tf_k \cdot log(idf_k) \tag{8}$$

(注4)：http://mecab.sourceforge.net/

(9) 式による評価値の高い語を質問を表現するような語とみなす．今回は $v(k)$ 値の高い順に 3 語取得したものを質問を表現する語集合 K とし（$K = \{k_1, k_2, k_3\}$），それらを AND 条件とするクエリを Web 検索エンジンに与えて別解ページ候補となる Web ページ集合 P を取得する．

4.2 回答情報を含むページの判定

ページ集合 P の各要素 p_n が，入力として与えられる質問 q に対する回答情報を含むかを，回答を表現する際に共通して現れる語の集合 $C = \{c_1, c_2, ...\}$ を用いて判定する手法について述べる．

本稿では，ある質問に対する回答集合において回答を表現する際に共通して現れる語の存在を仮定している．たとえば,「二日酔いの解消方法は？」という質問に対する回答情報の周りには二日酔いの原因である“ 酒 ”という語が出現することが考えられる．また，二日酔いの解消方法として「ウコンを飲む」,「大量の水を飲む」というものがあるがこれらの回答には“ 飲む ”という語が共通して現れている．このような，回答に共通して現れる語を取得し，ページ内にその語が含まれるかを調べることでページに回答情報が含まれるか否かを判定することができると考えられる．

ここで，q に対する回答を述べる際に共通して現れる語（以下 共通語）の集合 C の取得方法について述べる．理想的には，q に与えられている回答集合 A において出現回数の多い語が共通語集合の要素となるはずである．しかし，一般に QA コンテンツの回答は短い文章で構成されることが多い為，一つの QA コンテンツに存在する回答集合だけでは語数が少なく共通語を取得するのが困難である．そのため，本稿では q と A を含む文書を収集することで回答集合を拡張する事を考える．回答に共通して現れる語は，質問を表すような語集合と回答 a_m を表すような語集合を共に含む文書集合中に多く現れると考えられる．そこで，そのような文書を以下の手順で収集し，C を取得する．

1. 入力として与えられる回答集合 A の各要素 a_m について a_m を表すような語の集合 $W_{a_m} = \{w_1^{a_m}, w_2^{a_m}, ...\}$ を取得する
2. 4.1 で得られる質問を表す語集合 K と Wa_m の各要素を AND 条件とするクエリで文書検索を行う．この操作を A の要素数回繰り返し，文書集合を得る
3. 得られた文書集合において出現回数の多い語を C として取得する

以下に，各手順について説明する．

[1.]C を取得するためには各回答を表すような語を取得する必要がある．回答を表す語の取得には，馬らの研究における語の共起関係を用いた話題構造抽出手法 [9] を用いる．馬らは主題度・内容度という指標を用いてテキストの主題語・内容語を抽出するということを行っている．

QA コンテンツにおける主題は質問文の内容とみなすことが出来るため，QA コンテンツの主題語は 4.1 で得られる質問を表す語集合 K と考えられる．馬らの研究 [9] では主題についてテキストが何を述べているかを表す語が内容語となっている．QA コンテンツでは，質問を主題として各回答がそれぞれ主題について述べているものとみなせる．よって，馬らの研究 [9]

における内容語集合を本研究における回答を表す語集合 W_{a_m} に置き換えることができる．

馬らは内容語を求めるための内容度を主題語との無向共起度で求めるということを行っている．語 w_x と語 w_y の無向共起度 $cooc(w_x, w_y)$ は以下の式で定義されている．

$$cooc(w_x, w_y) = \frac{df(w_x AND w_y)}{df(w_x) + df(w_y) - df(w_x AND w_y)} \quad (9)$$

これは，語 w_x と語 w_y が文書中で同時に用いられやすいかを表す指標であり，$df(w)$ は語 w を含む文書数を表す．本稿では文書数を Yahoo!知恵袋内の QA コンテンツ数に置き換え，Yahoo! 知恵袋 Web API を用いて $df(w)$ を取得する．

回答 a_m に含まれる語 w が a_m において内容語である度合いを表す内容度 $con(w)$ は (10) 式を用いて以下のように計算することができる．

$$con(w) = \sum_{k_i \in K} cooc(w, k_i) \quad (10)$$

ただし $w \in ExtractWords(a_m)$ である．(11) 式による評価値の高い語が a_m を表す語（内容語）の集合 W_{a_m} の要素となる．本論文では $con(w) w \in CollectWords(a_m)$ 値の高い順に 2 語を W_{a_m} の要素とする．($W_{a_m} = \{w_1^{a_m}, w_2^{a_m}\}$)

[2.] 質問 q を表すような語集合と各回答 a_m を表すような語集合を共に含む文書集合を収集することで，回答集合 A を拡張することを考える．拡張を行うことで，回答表現に共通する語を取得しやすくすることが目的である．ここで，q を表すような語集合は 4.1 で得られる質問文から生成されるクエリ K とみなすことができ，各回答を表すような語集合は先ほどの 1. で得られる W_{a_m} とみなすことができる．両者を共に含む文書集合として，2 種類の集合が考えられる．一つ目は Web ページ集合であり，もう一つは QA コンテンツの回答集合である．以下に，それぞれの集合を収集する方法を述べる．

[2.1]**Web ページ集合による拡張**

4.1 で得られる K と先ほどの 1. で得られる W_{a_m} の各要素を AND 条件とする検索クエリを生成し，Web 検索エンジンに与えて両者を共に含む文書集合のスニペット集合を得る．このスニペット集合を回答集合の拡張とみなす．語集合 W を引数にとり，引数を AND 条件とするクエリを Web 検索エンジンに与えて検索結果のスニペット集合を得る関数を $CollectSnipets(W)$ とおくと，K と W_{a_m} を共に含む文書集合のスニペット集合 S は以下の式で表すことができる．

$$S = \bigcup_{a_m \in A} \bigcup_{i=1}^{2} CollectSnipets(K \cup w_i^{a_m}) \quad (11)$$

[2.2]**QA コンテンツの回答集合による拡張**

QA コンテンツによる拡張の場合，K と W_{a_m} を含む QA コンテンツの回答のみを収集する．これは，回答表現に共通する語を取得することが拡張の目的であり，そのような語は QA コンテンツ全体よりも QA コンテンツの回答に現れやすいと考えられるためである．先ほどと同様に，語集合を引数にとり，引数を AND 条件とするクエリを知恵袋内の検索機能に与えて検索結果の回答集合のみを収集する関数を $CollectAnswers(W)$ とすると，K と W_{a_m} を共に含む QA コンテンツの回答集合 A' は以下の式で表すことができる．

$$A' = \bigcup_{a_m \in A} \bigcup_{i=1}^{2} CollectAnswers(K \cup w_i^{a_m}) \quad (12)$$

ここで，S と A' はそれぞれ A の拡張とみなすことができ，$\bigcup_{i=1}^{2} CollectSnipets(K \cup w_i^{a_m})$ と $\bigcup_{i=1}^{2} CollectAnswers(K \cup w_i^{a_m})$ は各回答 a_m の拡張とみなすことができる．

[3.]2. で得られたスニペット集合 S または回答集合 A' において出現回数（$\sum_{s \in S} count(c, s)$ または $\sum_{a' \in A'} count(c, a')$）の多い語 c を回答に共通して現れる語 C の要素とする．本論文では出現回数の多い順に 10 語を C の要素として取得する．出現回数の値を v_c とおくと，C は以下のように定義される．

$$C = \{c | v_c > \phi\} \quad (13)$$

4.1 で得られたページ集合 P の各ページ p_n について，ページ内に C の要素がより多く含まれていればそのページには q に対する回答情報が含まれる可能性が高くなる．そこで，p_n 内のリンク文字列以外のテキストに C の要素がいくつ含まれるかを数えた値を $IncAns(u, q, A, p_i)$（式 (2)）と再定義する．ここでリンク文字列を対象外とするのは，そのページ自身に回答が現れるかを判定するためである．たとえば,「二日酔いに飲むと良いものはこちら」といったリンク文字列がある時,“ 飲む ” という語は含まれているが，何を飲めば良いのかはそのページ内ではなくリンクをたどった先に書かれていることが考えられるためである．ページ p の本文テキストからリンク文字列を除いたテキストを $Nolink(p)$ とおくと式 (2) は以下のように表すことができる．

$$IncAns(u, q, A, p_i) = \frac{\sum_{c_i \in C} count(c_i, Nolink(p_n))}{N} \quad (14)$$

$$IncAns(u, q, A, p_i) = \frac{v_{c_i} * \sum_{c_i \in C} count(c_i, Nolink(p_n))}{N} \quad (15)$$

N は $Nolink(p_n)$ に出現する語の総数を表す．式 (15) は v_c を語 c の重みとみなし，その重みを $IncAns(u, q, A, p_i)$ に反映させたものである．

4.3 別解情報を含むページの判定とランキング手法

Web ページが QA コンテンツ内の回答情報とは異なる別解情報を含むか判定する方法について述べる．我々は大島らの研究 [10] で用いられている文書間の非類似度という考え方を利用して別解を求めたい QA コンテンツに存在する既存の回答集合 A と Web ページとの非類似度を求めることでページに別解が含まれるかどうかを判定する．大島らは同一カテゴリに属する文書集合の各々に特有な内容を表す特徴ベクトルと，与えられたある文書を表す特徴ベクトルとのコサイン類似度を求め，その値から非類似度を求めている．本論文では，各回答 a_m を表す特徴ベクトルと Web ページを表す特徴ベクトルをそれぞれ作成し，そのコサイン類似度を利用して非類似度を計算する．特

徴ベクトルは大島らと同様に TF ベクトルを用いるものとする．

まず，Web ページ内のテキストからリンク文字列を除いたものを形態素解析して TF ベクトルを得る．ページ p_i の TF ベクトルは以下のように表すことができる．

$$\mathbf{t_{p_i}}(k) = count(k, Nolink(p_i)) \quad (16)$$

次に，各回答 a_m を表す TF ベクトルを作成する．しかし，4.2 節でも述べたように，QA コンテンツの回答は短い文章で構成されることが多く，TF ベクトルを作成するのに適していない場合がある．よって，4.2 節で用いた回答集合の拡張をここでも用いる．A 全体の拡張は式 (11) または (12) で表すことができる．回答または回答集合 X の拡張を $Expand(X)$ とおくと，各回答 a_m の特有な部分を表す TF ベクトルは以下のようになる．

$$\mathbf{t_{a_m}}(k) = max(count(k, Expand(a_m)) - count(k, Expand(A)), 0) \quad (17)$$

式 (17) は回答 a_m を表す TF ベクトルから全回答 A を表す TF ベクトルを引くことで a_m に特有な部分を表す T ベクトルを求めるものであり，値が負になる場合はその値を 0 に置き換えるという操作を行っている．

各回答についてこの TF ベクトルを求めたら，次は $\mathbf{t_{p_i}}$ と $\mathbf{t_{a_m}}$ とのコサイン類似度（$cos(\mathbf{t_{p_i}}, \mathbf{t_{a_m}})$）を計算する．$A$ のすべての要素についてコサイン類似度を計算し，それらのうち最も高い値をページ p_i と QA コンテンツの回答集合 A との類似度（$Sim(p_i, A)$）とする．

$$Sim(p_i, A) = max(cos(\mathbf{t_{p_i}}, \mathbf{t_{a_1}}), ..., cos(\mathbf{t_{p_i}}, \mathbf{t_{a_m}})) \quad (18)$$

コサイン類似度の最大値は 1 であるため，1 から式 (18) の値を引くことで非類似度（$Dissim(p_i, A)$）を表すこととする．この非類似度が，ページが別解を含む可能性を表す値（式 (3)）となる．

$$Dissim(p_i, A) = 1 - Sim(p_i, A) (= JudgeAlt(u, q, A, p_i)) \quad (19)$$

以上が QA コンテンツに対する別解を含む Web ページの取得方法である．この手順を図 1 に示す．図 1 は，回答集合の拡張を QA コンテンツの回答で行った場合を示している．

最後に，ランキングスコアの計算式 (4) を以下のように定義する．

$$Rank(u, q, A, p_i) = IncAns(u, q, A, p_i)^{\alpha} * JudgeAlt(u, q, A, p_i)^{(1-\alpha)} \quad (20)$$

5 節では，各提案手法についての評価実験について述べる．

5. 評 価 実 験

5.1 テストセット

本節では各手法についての評価実験を行ったのでそれについて述べる．実験のためにまず知恵袋内の 10 種類の中カテゴリからそれぞれ一つずつ QA コンテンツを用意した．この QA コンテンツは 1．回答数が 3 つ 2．質問が一つの事を尋ねているもの　の条件を満たすものを人手で選択した．1．の条件は回答数を統一するためで，2．の条件は，質問が複数の事を尋ねている場合，QA コンテンツ内の回答がどの質問内容に対して回答しているかの判別が必要になるためである．

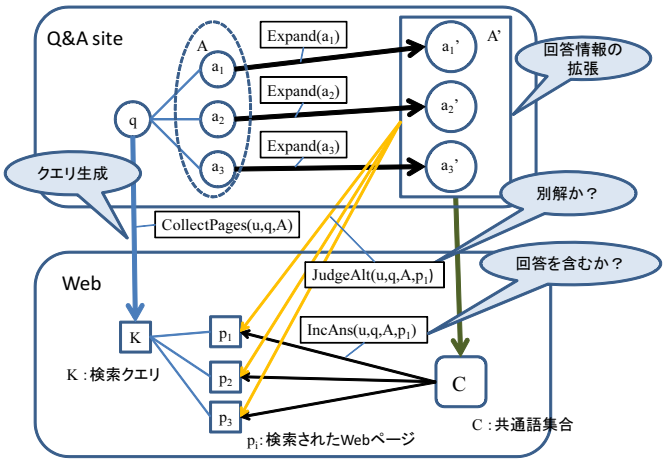

図 1　手順の概要図

5.2 生成された Web 検索クエリの妥当性

4.1 節について，QA コンテンツの質問文 q からのクエリ生成（以下，手法 (Q)）とユーザが指定した部分 u を用いてクエリ生成を行った場合（以下，手法 (U)）について比較した．本実験では，ユーザが指定した部分として q から質問を表すような文章を 1〜2 文我々が選んだものを用いた．用意した 10 個の QA コンテンツについて，手法 (Q) または (U) を用いてクエリ生成を行い，それぞれのクエリで Web 検索を行った．Web 検索結果の上位 20 ページについて，各ページが q に対する回答情報を含むかどうかと別解情報を含むかどうかを人手で判定した．10 個の QA コンテンツ各々において検索結果の 20 ページ中に各情報を含むページがいくつあったかを数え，その平均を計算したものを表 1 に示す．表 1 より，手法 (U) で生成したクエリの方が，より多くの回答情報または別解情報を含む Web ページを取得できることが分かった．これはつまり，ユーザに質問部分を指定してもらうことでより多くの別解候補ページ集合を得ることのできる Web 検索クエリを生成できるということである．以降の実験では手法 (U) で生成された Web 検索クエリ K を用いて Web 検索を行い，検索結果の上位 20 ページを取得してこのページ集合を実験の対象とする．

5.3 回答を含むかどうかのスコアの評価

式 (14) と (15) の値の妥当性について評価を行った．これらの値は Web ページが対象とする QA コンテンツの質問 q に対する回答情報を含むかどうかを表すものである．我々は 4.2 節

	回答情報を含むページ数	別解情報を含むページ数
手法 (U)	7.8	6.2
手法 (Q)	5.7	4.4

表 1　回答情報または別解情報を含む平均 Web ページ数

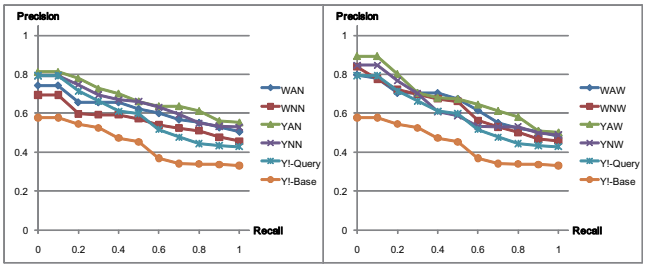

図 2　11 点平均適合率・再現率（式 (14)(15)）

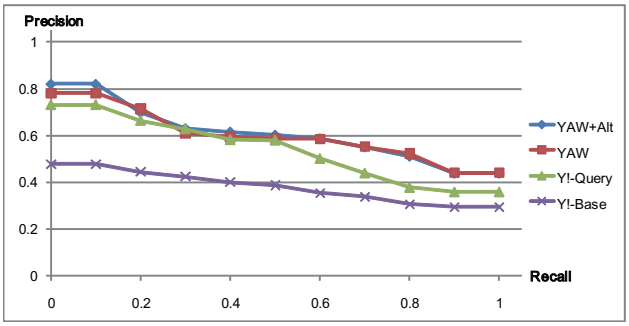

図 3　11 点平均適合率・再現率（式 (20)）

において，回答表現に共通する語集合 C の取得方法として以下の 2 種類の方法を提案した．

(WA) K と W_{a_m} をクエリとして Web 検索を行った検索結果のスニペット集合から C の要素を取得する

(YA) K と W_{a_m} をクエリとして Yahoo!知恵袋内の検索機能を用いて検索を行った結果の回答集合から C の要素を取得する

本実験ではこの 2 つの方法に加えて，比較手法としてさらに 2 つの C の取得方法を提案する．

(WN) K のみをクエリとして Web 検索を行った検索結果のスニペット集合から C の要素を取得する

(YN) K のみをクエリとして Yahoo!知恵袋内で検索を行った結果の回答集合から C の要素を取得する

これら 4 種類の方法で得られた C を用いて式 (14)，(15) の値を 5.1 節で得られた 20 個の Web ページについて計算した．上記の方法について，式 (14) を用いた場合をそれぞれ (WAN), (WNN), (YAN), (YNN) とおき，式 (15) を用いた場合を (WAW), (WNW), (YAW), (YNW) とおく．20 ページについてそれぞれの値を計算した後，値が降順となるようにページを並べ替えた．値が同値となった場合は検索エンジンの出力順を優先した．この操作を 10 個の QA コンテンツに対して得られた 20 ページについてそれぞれ行い，各手法の 11 点平均適合率・再現率を計算した．ここで，対象とする QA コンテンツの質問 q に対する回答情報を含む Web ページを正解ページとする．上記で述べた 8 つの手法に加えて，比較のためのデータとして

Baseline 1.(Y!-Query) クエリ K を Yahoo!検索エンジンに与えて得られた検索結果の出力順での再現率と適合率

Baseline 2.(Y!-Base)QA コンテンツからのクエリ生成を行わずに，ユーザが指定した部分 u に出現するすべての名詞，動詞，形容詞（ストップワードは考慮する）を AND 条件としたものをクエリとして Yahoo!検索エンジンに与えて得られた検索結果の再現率と適合率

の 2 種類についても同様の計算を行った．結果を図 2 に示す．図 2 より，すべての提案手法について，クエリ生成や回答を含むかどうかの指標を考慮しない (Y!-Base) の結果を上回ったことが分かる．また，C の重みを考慮した 4 手法については (Y!-Query) の結果も上回ることが分かる．すべての手法の中では (YAW) がもっとも良い結果となった．それぞれの手法の結果を比較すると，(WAN)，(WAW) 手法はそれぞれ (WNN), (WNW) 手法を上回り，(YAN)，(YAW) 手法はそれぞれ (YNN), (YNW) 手法を上回っていることが分かる．これは，回答表現に共通する語集合 C を取得する際に，質問 q の情報のみを用いる場合よりも q に加えて回答集合 A の情報も用いることでより精度よく C が抽出できたことを示すと考えられる．さらに，(W〜) 手法よりも (Y〜) 手法の方が結果が良いことから，回答集合の拡張を行う際には Yahoo!知恵袋の回答集合を用いる方が良いことが分かった．ただし，Yahoo!知恵袋の回答集合による拡張は，拡張に用いる回答集合を得る際に知恵袋の検索エンジンに与えるクエリが複雑であった場合に結果が得られないという問題が起こりうる．しかし，この問題が生じた場合は (YAW) 手法の代わりに (WAW) 手法を用いることで解決できる．

5.4　別解を含むかどうかのランキングスコアの評価

5.2 節の結果より，回答集合 A の拡張には Web を用いるよりも Yahoo!知恵袋の回答集合を用いる方が良いことが分かった．そこで，本節の実験では A の拡張に式 (12) を用いた．5.2 節の実験結果で最も結果の良かった (YAW) 手法について式 (20) を計算し，先ほどの実験と同様に 5.1 節で得られたページ集合 P を式 (20) の値をもとに並べ替えた．ここで，(YAW) 手法とは，A の拡張を Yahoo!知恵袋の回答集合を用いて行い，さらに $IncAns(u, q, A, p_i)$ を C の重みを考慮した式 (15) で計算した手法である．式 (20) について，α の値を変化させて実験を行った結果，0.8 が最適となったため $\alpha = 0.8$ とする．先ほどの実験と同様に，各ページについて (YAW) 手法を用いてランキングの式 (20) を計算し（以降，(YAW+Alt)），値が降順となるようにページを並び替えたものについて 11 点平均適合率・再現率を計算した．この実験の場合は，q に対する別解情報がページの中に含まれていれば正解ページとみなす．比較のため，正解ページを別解情報を含むページとした場合の (Y!-Base)，(Y!-Query)，(YAW) 手法についても同様にして 11 点平均適合率・再現率を求めた．結果を図 3 に示す．

図 3 より，(YAW+Alt) 手法が他の手法をほぼ上回ることが分かる．そのため，回答情報を含むかどうかの指標である $IncAns(u, q, A, p_i)$ に加えて別解を含むかどうかの指標である $JudgeAlt(u, q, A, p_i)$ を考慮したランキング関数 (20) が有効であることが分かった．しかし，回答を含むかどうかを考慮しただけである (YAW) 手法との差が大きくならなかったことから，別解かどうかをより明確に表す指標が必要であることが判明した．

6. ま と め

QA コンテンツの質問に対する別解情報を含む Web ページを取得する手法とそのランキング手法について提案した．提案手法は大きく 3 つの段階に分けられる．まず，別解を求める QA コンテンツから Web 検索クエリを生成し別解を含む Web ページの候補集合を取得する．次に取得した各ページについて，質問 q に対する回答情報を含む可能性を表すスコアを計算する．最後に，各ページが q に対する別解情報を含むかどうかのスコア付けを行い，両スコアを考慮したランキング関数を用いてページ集合の要素をランク付けして提示する．実験の結果から，提案手法はベースラインとして用意した手法を上回ることが分かった．今後は，より明確に別解を含むページを判定する手法の考案が必要である．また，別解情報について，情報の質や多様性に考慮したランキングについても考えていく予定である．

謝　　辞

本研究の一部は，京都大学 GCOE プログラム「知識循環社会のための情報学教育研究拠点」，および，文部科学省科学研究費補助金特定領域研究「情報爆発時代に向けた新しい IT 基盤技術の研究」，計画研究「情報爆発時代に対応するコンテンツ融合と操作環境融合に関する研究」（研究代表者：田中克己，A01-00-02，課題番号：18049041），および，文部科学省科学研究費補助金若手研究（B）「オンデマンド利用を目的とする Web からの知識発見に関する研究」（研究代表者：大島裕明，課題番号：21700105），および，NICT 委託研究「電気通信サービスにおける情報信憑性検証技術に関する研究開発」（研究代表者：田中克己）によるものです．ここに記して謝意を表します．

文　　献

[1] A.Berger, R.Caruana, D.Cohn, D.Freitag and V.Mittal: “Bridging the lexical chasm: Statistical approaches to answer-finding”, Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 192–199 (2000).

[2] J.Jeon, W.B.Croft and J.H.Lee: “Finding similar questions in large question and answer archives”, Proceedings of the ACM Fourteenth Conference on Information and Knowledge Management, pp. 76–83 (2005).

[3] X.Xue, J.Jeon and W.B.Croft: “Retrieval models for question and answer archives”, Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 475–482 (2008).

[4] K. Wang, Z. Ming and T.-S. Chua: “A syntactic tree matching approach to finding similar questions in community-based qa services”, SIGIR, pp. 187–194 (2009).

[5] “http://www.nlp.is.ritsumei.ac.jp/qac/qac2/index-j.html”.

[6] 佐藤充, 石下円香, 森辰則：“Web 文書を情報源とする記述的な回答が可能な応答システム”, 言語処理学会第 14 回年次大会発表論文集, pp. 1009–1012 (2008).

[7] 田村元秀, 村上仁一, 徳久雅人, 池原悟：“Web 検索エンジンを用いた why 型質問応答システム”, 言語処理学会第 14 回年次大会発表論文集, pp. 1021–1024 (2008).

[8] 島田諭, 福原知宏, 佐藤哲司：“質問回答サイトにおける利用者行動に基づく記事の関連付け手法の検討”, 電子情報通信学会第 19 回データ工学ワークショップ DEWS2008 講演論文集 (2008).

[9] 馬強, 田中克己：“話題構造に基づく放送と web コンテンツの統合のための検索機構”, 情報処理学会論文誌, **45**, pp. 18–36 (2004).

[10] 大島裕明, 小山聡, 田中克己：“文書群を問合せとした兄弟カテゴリー文書の検索”, 電子情報通信学会論文誌 D, **J90-D**, 2, pp. 196–208 (2007).