

和文題目 AAS:過去の回答の自動組み合わせに基づく質問応答システム

和文副題

和文発表者名 松村 真宏, 大澤 幸生, 谷内田 正彦

和文勤務先 大阪大学大学院基礎工学研究科

連絡先住所 大阪大学大学院基礎工学研究科 〒560-8531 大阪府豊中市待兼山町1-3

・電話番号 06-850-6363

E-mail matamura@yachi-lab.sys.es.osaka-u.ac.jp, osawa@sys.es.osaka-u.ac.jp, yachida@sys.es.osaka-u.ac.jp

あらまし

本研究では、従来の質問応答システムでは難しかったユーザの興味を深く満たせるような回答を見つけるシステムの作成を目指す。文書データベースにはFAQ文書（様々な質問とその回答を集めた文書）を用いる。より深く文書を理解するためのアプローチとして、まずFAQ文書を仮説知識とその重みで表す方法を提案する。そして、この仮説知識に対してコストに基づく仮説推論を行なうことにより、複数の回答を組み合わせて最適な回答を出力するシステム Answer Assembling Systemを提案し、実験による評価を行なう。

キーワード 質問応答システム、FAQ文書、コストに基づく仮説推論、仮説知識、コスト
説 説

英文題目 AAS:Question-Answering System based on Automatically Assembling Relevant Past Answers

英文副題

英文発表者名 Naohiro Matsumura, Yukio Ohsawa, Masahiko Yachida

英文勤務先 Graduate school of Engineering Science, Osaka University, Osaka

英文連絡先住所 Graduate school of Engineering Science, Osaka University, Osaka 560-8531, Japan

・電話番号 06-850-6363

E-mail matamura@yachi-lab.sys.es.osaka-u.ac.jp, osawa@sys.es.osaka-u.ac.jp, yachida@sys.es.osaka-u.ac.jp

Abstract

In dairy lives, we frequently encounter new and queer accidents. It is about such a queer matter that people want to consult a counselor. Previous memory-based systems which answer queries expressed in sentences, e.g., FAQ-finder cannot answer queer questions about these queer things, because they are referred only to past questions from which they find similar questions to the current user's one and returns the corresponding answers. In this paper, we present an Answer Assembling System, which combines paragraphs in past answers for answering newly asked questions. Empirical results are evaluated by the extent of user's satisfaction.

key words Question-Answering System, FAQ, cost-based hypothetical reasoning, hypothetical knowledge, cost

1 はじめに

日常生活において、例えばパソコンが急に起動しなくなるといった困った事態に陥ることはよくある。この例の場合はメーカーのサービスセンターに電話するのが最善の処置だが、著者らの経験によると電話は繋がらないことの方が多い。マニュアルを読んでも原因は分からないことが多いし、それで分かるような問題は大した問題ではないであろう。おそらく一番早い解決方法は、そのことに詳しい友人などに聞くことである。同じことを過去に経験して解決した友人がいれば、事もなげに問題は解決できるであろう。人が陥りやすい失敗は、多くの場合それまでに繰り返しされてきた失敗が多い。そこで、よくある質問とそれに対する回答をあらかじめ用意しておく、困った時はそこから回答を探すということが行なわれるようになった。この「よくある質問と、それに対する回答」の文書を集めたものをFAQ(Frequently Asked Questions)文書と呼ぶ。

FAQというメカニズムは、世界中で使用可能な電子情報サービスであるUSENET newsgroups[1]で考え出された。newsgroupsの投稿者が、よくある質問とそれに対する回答をFAQ文書としてまとめたのが最初である。本研究では文書データベース(以後、文書DBと略す)としてFAQ文書を用いるが、それは次の2つの理由による。

- 質問に対する回答が細切れの断片的な知識ではなく文書として存在するため、よくまとまっている
- 質問に答えた文書であるから、分かりやすい回答になっている

一般にFAQ文書は、様々な人の質問に対して様々な人が回答した文書を集めたものであり、本研究で用いたFAQ文書も特に書式は決まっておらず、自然言語(英語)で書かれている。文書の書式を決定することにより、必要な時に必要な情報を扱うことは容易になるが、巨大なデータベースを作る際には大きな障害となるため、FAQ文書の書式を統一することは実際には難しい。

また、FAQ文書が膨大になればなるほど様々な質問に対応できるようになるのだが、考えられるあらゆる質問に対してあらかじめ回答を用意しておくことは現実には不可能である。そこで、有限の文書DBを最大限に利用して質問に答えるシステムが必要となる。

2 従来の質問応答システム

この章では、文書DBにFAQ文書を用いる従来の質問応答システムの代表としてFAQファインダー[2]、

一般的な質問応答システムの代表としてベクトル空間モデルシステム(VSMS)、そして本研究の基となった旧AAS[3]を紹介し、その特徴と問題点を検討する。

2.1 FAQファインダー

FAQファインダーとその特徴

FAQファインダーでは、あらかじめFAQ文書を幾つかの形式(Q-HOW-TO形式、Q-COMPARISON形式など)に分類する。ユーザの質問も同じように分類し、FAQ文書の中から同じ形式の文書と照合することにより回答を探す。

例えば、ユーザが”Is purified water better than tap water for my houseplants?”(訳:鉢植えの植物には、水道水よりも純水の方がふさわしいですか?)という質問をした場合、「Q-COMPARISON形式」という形式の質問と判断し、FAQ文書の中から、同じ形式で同じ用語(purified waterとtap water)を含む質問を探し、その質問に対する回答を出力する。

また、”Is expensive oil worth it?”(訳:高いオイルは価値がありますか?)という質問のように、エンジンオイルなのか料理に用いるオイルなのかを判断できない時は、その選択をユーザに任せる。つまり、用語の解釈に不明瞭な候補があれば、ユーザに候補の中から選択させる。

FAQファインダーの問題点

[フリーテキストを扱えない]

FAQファインダーが扱うことのできるFAQ文書は、上記のように幾つかの形式に分類する必要がある。一般的にFAQ文書は様々な人の手によるフリーテキスト(書式が決まっていない文書)で書かれているので、この分類を計算機で行なうことは難しく、人の手に委ねざるを得ない。しかし、これには多大な人的・時間的コストが伴うので、膨大な文書を扱う際には現実的ではない。

[回答の絞り込みが難しい]

FAQファインダーの出力した回答がユーザの質問を満たさない時は、ユーザが質問を変えることにより再び回答を探す。この操作は概して難しい。なぜなら、ユーザの質問を満たす回答がFAQ文書中に存在しないかもしれないし、もし存在するにしても、どういう具合に質問を変えれば所望の回答を得られるのかがユーザに全く分からないからである。ユーザの質問に関連のありそうな回答の一覧を表示する方法も考えられるが、目的の回答が得られるまで出力された全ての回答に目を通さなければならないので効率が悪く、またユーザに与える負担も大きい。

2.2 VSMS

VSMS とその特徴

ここでいう VSMS は、検索にベクトル空間モデルを用いるシステムの総称である。ベクトル空間モデルとは、あらゆるキーワードを各軸に持つ空間を仮定して、文書をその空間に配置するモデルである。このモデルの特徴は、ベクトル空間内に配置された文書の位置ベクトルの向きが、その文書の内容・主張を表していると見なすことができる点にある。したがって、ユーザの質問の位置ベクトルと最もなす角度が小さい位置ベクトルを持つ文書を探すことにより、回答を見つけることができる。

VSMS の問題点

VSMS は、キーワードがそのまま検索結果に反映される。膨大な文書を扱うためにはキーワードを自動抽出することが望まれるが、現在のキーワード自動抽出技術では、どうしてもノイズワードが含まれてしまう。したがって、ある程度の成果は望めるものの、それ以上の成果は望めないことが問題となる。

2.3 旧 AAS

旧 AAS とその特徴

旧 AAS とは、著者らが昨年開発した本研究の基となったシステムである。旧 AAS ではベクトル空間モデルを採用しているが、VSMS の問題点で指摘した検索精度の限界を打開することを目標としており、その手段として複数の回答を組み合わせるといった方法を取った。

旧 AAS は、まず全ての FAQ 文書をベクトル空間内に配置する。次にユーザの質問もベクトルで表し (\vec{Q})、VSMS と同じ方法で回答を探す。しかし VSMS と違う点は、ユーザの興味が満たされるまで \vec{Q} を更新しながら再帰的に回答を探るところにある。 \vec{Q} の更新は、取り出した回答と \vec{Q} との差分ベクトルを新たな \vec{Q} とすることにより行なう。

旧 AAS の問題点

旧 AAS では、回答生成時において 2 番目以降に選ばれる回答の内容がユーザの質問から離れていく傾向があった。これは、後になればなるほど \vec{Q} が短くなるため、内積の角度を計算する際の誤差が大きくなるためだと考えられる。また、ベクトル空間モデルでは、キーワードは互いに独立しているという仮定に基づいて各軸 (キーワード) は直交しているとしているが、キーワードには意味的な繋がりもあるため、この仮定は成り立たないという報告もある。

3 Answer Assembling System(AAS)

前章で述べたように、従来の質問応答システムではなかなかユーザの興味が満たす回答を得ることができなかった。しかし、著者らは旧 AAS において、幾つかの回答を読み進めるうちに徐々に理解が深まり、結果的にユーザの興味が満たせる場合があることを確信した。つまり、一つの回答だけでは不十分な場合でも、幾つかの回答を組み合わせることにより、より深くユーザの興味が満たせる回答を作成できるのである。

そこで本研究では、FAQ 文書の回答を幾つか組み合わせることにより、より深くユーザの興味が満たす回答を出力するシステム Answer Assembling System (AAS) を提案する。旧 AAS との違いは回答を選ぶ手法にあり、FAQ 文書を仮説知識とその各要素の重みで表し、コストに基づく仮説推論を適用することにより、ユーザの興味に沿った文書を最小の数だけ取り出すことを狙う。詳しくは 3.3 節で述べる。

3.1 質問の入力

ユーザの質問をシステムに入力する手段として、

1. 自然言語で入力する
2. キーワードで入力する

の 2 種類が考えられる。

1. は、人にとって最も自然なインターフェースは自然言語による質問であることに基づいているが、それが計算機にとっても自然なインターフェースかというと、そうではない。計算機には、計算機にしか理解できないフォーマットがあり、それに合わせる必要がある。本研究では、文書を効果キーワード、土台キーワード (後述) で表現する都合上、計算機へのフォーマットはキーワードになる。自然言語で入力された質問からキーワードを自動抽出することも考えられるが、この時にキーワードとしてふさわしくない語 (ストップワード) を選んでしまうと、的を射ない回答を選んでしまう原因となる。

一方、十分とは言えないかもしれないが、ユーザの手により直接入力されたキーワードはストップワードを含まないので、自動抽出したキーワードを用いるよりユーザの興味が集約されていると考えられ、その結果、回答の精度も良くなると期待できる。したがって、ユーザの抱えている興味がキーワードだけで表現することは難しい [4] という問題はあがあるが、本研究ではユーザが入力する質問を 2. による幾つかのキーワードによって行なうことにした。

3.2 コストに基づく仮説推論

仮説推論とは、必ず真である背景知識と、真かどうかは自明ではなく、矛盾を含むこともある仮説知識 H を用いて、ある事象（ゴール G ）を証明する仮説知識の無矛盾な集合を見つける推論である。

本研究で用いる仮説推論は、コストに基づく仮説推論と呼ばれるもので、各要素仮説に数値的の重みを与え、要素仮説の重みの和（コスト）が最小となる仮説知識の無矛盾な集合 h を求める推論である。

背景知識、仮説知識は、 a, b, c のような単独の記号（アトム）と左向きの矢印（ \leftarrow ）を用いて、 \leftarrow の左辺に結論、右辺に前提となるアトムを置くことにより、次のように表記する。

$$\begin{aligned} G &\leftarrow a. \\ a &\leftarrow b, c. \\ &\vdots \end{aligned}$$

また、仮説 $\eta \in H$ の重みは $w(\eta)$ とし、次のように表記する。

$$\begin{aligned} w(b) &= 1. \\ w(c) &= 2. \\ &\vdots \end{aligned}$$

本研究では、FAQ 文書から仮説知識とその要素仮説の重みを作成し、 h に含まれるコスト $\sum_{\eta \in h} w(\eta)$ を最小にするような無矛盾な集合 $h \in H$ を利用して回答を探すことを狙う。

3.3 AAS の概要

AAS を実現するためのアプローチとして、KeyGraph[7] で考案された以下の仮説を採用している。

文書は、著者の主張が込められた効果キーワードと、その主張を導くために重要な役割を果たす土台キーワードからなる

例として、「過去の回答の自動組み合わせに基づく質問応答システム」（つまり、この原稿）に関する文書を挙げる。この文書は、「文書 DB から仮説知識とその要素仮説の重みを定義し、コストに基づく仮説推論を適用することにより質問応答システムを実現する手法」に関して述べた文書である。この文書の場合、効果キーワードは「質問応答システム、回答の自動組み合わせ」であり、土台キーワードは、「コストに基づく仮説推論、FAQ 文書、仮説知識」などが挙げられる。

このように効果キーワード、土台キーワードを定義すると、何が基となって何を主張しようとしているのかという論旨の流れが明らかになり、従来のように文書からキーワードだけを定義するより、文書を詳細に記述することができるようになる。しかも、キーワードという形態をとっていることから、計算機にも非常に扱いやすい。効果キーワード、土台キーワードを抽出する具体的手法については、3.4節で述べる。

次に効果キーワード、土台キーワードをどのように用いると質問応答システムに利用できるかという問題に移る。仮説推論を適用するためには FAQ 文書を仮説知識で表さなければならないが、これは「土台キーワードによって効果キーワードが導かれる」と考えることにより、文書から仮説知識をシンボリックに作成することができる。つまり、

$$\text{効果キーワード} \leftarrow \text{土台キーワード}$$

とすることにより、文書を仮説知識で表現できる。

この土台キーワードの各要素に重みを与え、ユーザの与えたキーワードをゴールとしてコストに基づく仮説推論を行なうと、コスト最小となる土台キーワードの組み合わせが得られる。重みの与え方はいろいろ考えられるが、適切に与えることでユーザの質問を満たす、かつユーザの興味に沿った最小の文書集合を得ることができる。AAS の概要は、大まかに述べると以上の通りである。

まとめると、AAS のアルゴリズムは次の4つのステップから成る。

- 1) 効果キーワード、土台キーワードの抽出
- 2) FAQ 文書から仮説知識の作成
- 3) 仮説知識の各要素の重みの作成
- 4) 回答の探索（コストに基づく仮説推論）

以下の節で、各ステップについて詳細に説明する。

3.4 効果キーワードと土台キーワードの抽出

扱う文書が FAQ 文書という点を考慮すると、質問と回答のペア i (i は各文書に割り振られている番号) に共通して現れる語の集合 $\text{Keycommon}(i)$ (以後、共通語と呼ぶ) には、その文書における重要な意味が与えられていると考えられる。そこで、共通語と元の文書とを比較・検討した結果、共通語には土台キーワード、効果キーワードの両方が含まれていることが判明した。しかも、各文書に対し一意に効果キーワード、土台キーワードを決めることはできないことも分かった。それは、ユーザの抱いている興味によって、同じ語でも土台キーワード、効果キーワードのどちらにも

なり得るからである。このことを、「風邪の原因と症状」に関して述べた文書を例に挙げて説明する。この文書を「風邪の原因は何か?」という興味を持って読むと、効果キーワードは「風邪の原因」となり、土台キーワードは「ウイルス」、「不摂生」、「感染症」などとなるが、もしユーザが「風邪の症状」に興味をもって文書を読んだなら、効果キーワードは「風邪の症状」、土台キーワードは「咳」、「熱」、「鼻がつまる」といった具合になる。このように、効果キーワード、土台キーワードはユーザからキーワードが与えられるまでは一意に決定することはできないのである。

そこで本研究では、各質問と回答のペア i ごとに効果キーワード、土台キーワードの候補の集合を候補キーワード ($Key_{candidate}(i)$) という形で持つておく。候補キーワードには共通語を加えるが、文書によっては共通語が少ない場合や全くない場合もあり、これだけでは候補キーワードの数が足りない。また、共通語だけでは重要な語を網羅することはできないので、 $tfidf[5]$ による評価値の高い語の集合 $Key_{tfidf}(i)$ を候補キーワードに追加した。 $tfidf$ は自動キーワード抽出法の一つで、簡単に説明すると、文書からキーワードの候補 (文書中の全ての単語から、ストップワードを除いた単語の集合) に対し、以下の評価関数 (1) 式により評価値 $Value(term)$ を計算し、その値の高い単語をその文書におけるキーワードとする方法である。

$$Value(term) = TF(term) \times \log \left(\frac{N}{DF(term)} \right) \quad (1)$$

ここで、 $term$ はキーワードの候補、 $TF(term)$ は $term$ の一文書中の出現回数、 $DF(term)$ は $term$ が出現する文書数、 N は全文書数を表している。

以上のことをまとめると、候補キーワードは次のようにして与えられる。

[候補キーワードの作成]

$$Key_{candidate}(i) \subset \{Key_{common}(i) \cup Key_{tfidf}(i)\}$$

3.5 仮説知識の作成

仮説知識を作成する上で注意しなければならないことは、仮説知識間に生じるループである。ループとは仮説知識の因果関係において原因と結果が入れ替わっていることをいい、簡単な例を挙げると以下のようになる。

$$G \leftarrow a. \\ a \leftarrow b.$$

$$b \leftarrow a.$$

この仮説知識は、「ゴールは a 」、「 b ならば a 」、「 a ならば b 」という関係を表している。この仮説知識に基づいて仮説推論を行なうと、「ゴールを満たすのに必要なアトムは a 。 a を満たすアトムは b 、 b を満たすアトムは a 、 a を満たすアトムは...」と堂々巡りを繰り返し、解が得られない。ループを生じる仮説知識の組合せを明記することにより上記の問題は解決できるが、矛盾知識は推論を複雑にする主な原因の一つであり、推論時間を考えると、矛盾知識の使用はどのようにして避けたい。したがって、ループを生じないように仮説知識を作成する必要がある。

ここで、ループが生じる必要条件を考えると、効果キーワード、土台キーワードに同じキーワードが共通して現れることが原因となる。したがって、効果キーワード・土台キーワードに同じキーワードを共有させないことにより、ループを有しない仮説知識を作成する方法を提案する。

ユーザが入力した質問 (キーワード) の集合を $Query$ とすると、 $Query$ はユーザの興味を端的に表していると考えられるから、無条件に効果キーワードとする。すると、候補キーワード集合 $Key_{candidate}(i)$ のうち、 $Query$ 以外のキーワードは $Query$ を導くための重要な語だと見なすことができるので、これを土台キーワードとする。

以上の作業を全 FAQ 文書に対して行なうことにより、ユーザの興味に基づいた仮説知識を作成できる。このとき、土台キーワードに文書に割り振ったナンバー (i) を含めることで、解から直接文書を特定できるようにする。

以上のことをまとめると、次のようになる。

[仮説知識の作成]

for $\forall i$

$$G \leftarrow q_1, q_2, \dots, q_n.$$

$$q_1, q_2, \dots, q_m \leftarrow key_{i1}, key_{i2}, \dots, key_{ii}, i.$$

ここで、

q_1, q_2, \dots, q_n : $Query$ の要素

q_1, q_2, \dots, q_m : $Key_{candidate}(i)$ に含まれる $Query$ の要素

$key_{i1}, key_{i2}, \dots, key_{ii}$: $Query$ に含まれない

$Key_{candidate}(i)$ の要素

3.6 重みの与え方

仮説知識の要素仮説に与える重みについて考察する。重みは、与え方によりコストに基づく仮説推論の解の意味が変わってくる。例えば、文書の長さを重みに選んだ時の解は、「ユーザの質問を満たす文書集合のうち、全体の長さが最も短い文書集合が選ばれる」という意味を持つ。

ユーザは、ある興味に基づいてキーワードを入力すると考えられる。したがって、回答もその興味に沿った、一貫した文書を選ぶべきである。しかし、そのような文書を選ぶためには、文書がユーザの興味に沿っているかどうかの判断基準が必要になる。そこで著者らは、土台キーワードの性質から次の関係を仮定した。

「土台キーワードを多く共有するほど、その文書間の話題は近くなる。」

これは、土台キーワードをなるべく多く共有するような文書を選ぶことが、ユーザと同じ興味に基づいた回答を選ぶことと等価になることを意味する。この仮説は、土台キーワードの全ての要素の重みを1とすることにより実現できる。また、このようにして重みを与えると、結果的に最小の文書数を選ぶことにもなる。

[重みの与え方]

仮説知識に含まれる土台キーワードの各要素の重みを1とする

3.7 回答の探索

3.5節で作成した仮説知識と3.6節で作成した重みから、コストに基づく仮説推論によりコスト最小となる土台キーワードを探す。コストに基づく仮説推論の計算複雑度はNP困難であり、仮説知識が複雑になると計算時間が問題となってくる。

現在実装しているシステムでは、全探索によりコスト最小となる解を探索しているが、ユーザの質問(キーワード)が5単語以上になると、数分から数十分かかる。そこで将来的には、コストに基づく仮説推論を0-1整数計画問題に帰着させることにより準最適解を多項式時間で求めることのできるネットワーク化バブル伝播法[6]の利用を予定している。

3.8 AASの動作シミュレーション

この節では、以下の文書1～3を例に挙げてAASの動作を順に追っていく。

文書1 : ブラウザでインターネットのwebページを楽しむことについて書かれた文書。
(候補キーワード: ブラウザ, インターネット, web, 文書1)

文書2 : 国内, 海外のwebページを見て回るネットサーフィンについて書かれた文書。

(候補キーワード: 海外, web, サーフィン, 文書2)

文書3 : サーフィンにもってこいの海外の波について書かれた文書。

(候補キーワード: サーフィン, 海外, 波, 文書3)

(ただし、ここではシステムの動作を追うことが目的なので、簡単化のため候補キーワードの数は4とする。)

ここで、ユーザが「インターネットでのサーフィンって何だろう?」と興味を持ち、「インターネット, サーフィン」とキーワードを入力した場合を考えると、仮説知識と重みは以下ようになる。

ゴール ← インターネット, サーフィン。
インターネット ← ブラウザー, web, 文書1。
サーフィン ← web, 海外, 文書2。
サーフィン ← 海外, 波, 文書3。

$$w(\text{ブラウザ}) = 1.$$

$$w(\text{web}) = 1.$$

$$w(\text{文書1}) = 1.$$

$$w(\text{海外}) = 1.$$

$$w(\text{文書2}) = 1.$$

$$w(\text{波}) = 1.$$

$$w(\text{文書3}) = 1.$$

このときの知識ネットワークは図1のようになる。

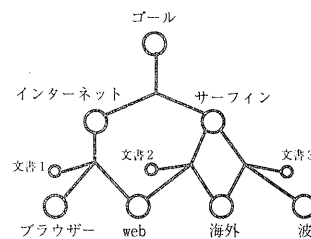


図1: 知識ネットワーク

この仮説知識に対して、ゴールを満たすようにコストに基づく仮説推論を行なうと、

文書1と文書2を選んだ時はコスト5

文書1と文書3を選んだ時はコスト6

となるので、回答は文書1と文書2になる(図2参照)。

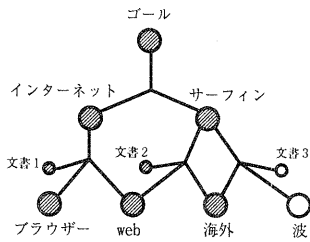


図2：仮説推論の結果（斜線のノードが解）

文書1と文書2は、ユーザが入力したキーワードには含まれていないが、ユーザの興味を理解する上で非常に重要な語である「web」を共通の土台キーワードとして持ち、話題も「web」を接点として一貫性がある。この例の場合は単一の文書では内容が不十分になるので、AASによる回答は妥当だと言える。

4 AASの実験による評価

AASを、Sun UltraSPARC 200MHz x 2(SunOS 5.5.1, 512MB)上にperl5.0, C言語を用いて実装した。またAASの文書データベースには、コロンビア大学のヘルスカウンセリングサービスである“Go ASK ALICE” [8]から1320枚のFAQ文書をダウンロードして用いた。このFAQ文書は様々な医療分野のほんの一分野をカバーしているに過ぎず、実験はその点に注意を払いつつ行なった。

4.1 評価方法

Information Retrieval分野では、システムの性能を評価する指標としてRecall（再現率）やPrecision（適合率）を用いることが多いが、AASを評価するには以下の理由から適切ではない。

- AASは、一つの文書だけでは回答として不十分な時に幾つかの回答を組み合わせた文書全体で回答とする点がユニークであり、また大きな特徴であるが、RecallやPrecisionではこの特徴を正確に評価することができない。
- RecallやPrecisionを用いるためには、ユーザの質問に対する正しい回答を膨大な文書DBの中から調べないといけないが、この作業は容易ではない。

著者らはAASを評価するための最初の段階として、「回答を構成するそれぞれの文書がユーザの興味をどの程度満足しているか？」を調べることが必要である

と考えた。そこで、回答を構成するそれぞれの文書に対し、以下の基準で○△×を用いて評価を行なった。

- 正に知りたいことが書かれてある（○）
- 知りたいことも少しは書かれてある（△）
- 的外れな文書である（×）

比較対象は、本研究のベースとなった旧AASと、一般の検索システムを代表してVSMSとした。また、平等な条件で比較するために、VSMSが出力する回答の数はAASと同数とした。FAQファインダーとの比較も興味深いだが、今回実験に用いたFAQ文書ではFAQファインダー方式は再現できないので諦めた。

4.2 実験

被験者に健康に関する質問をキーワードで入力してもらい、前節の評価法による判断を行なってもらった。その結果を以下に示す。

番号	ユーザの質問	AAS	旧AAS	VSMS
1	diet, food, exercise	○○	△××	△△
2	cancer, alcohol	○△	△○	△○
3	jogging, body	○	○×	○
4	drug, cancer, risk	○○	△×	△×
5	fat, alcohol, calorie	○	△×	△
6	breast, cancer	○	○×	○
7	tobacco, cancer, risk	○	○○	○
8	alcohol, limit, liver	○○	○×	○○
9	infect, AIDS, risk	○	△×○	○
10	throat, cancer, cold	△×	△△△	△△
11	throat, cold, medicine	○△	○×○	○○
12	muscle, training, eat	○	×○	×

4.3 考察

上記の実験結果からだけでは分からないが、より詳細に実験結果を検討してみると、それぞれのシステムには以下のような特徴があった。

AAS: 複数の文書を回答に選んだ時でも、お互いに足りない部分を補うような文書を選ぶ傾向があった

旧AAS: バラエティに富んだ回答を探し出しているが、2番目以降の回答があまり良くない。

VSMS: 複数の文書が、どれも同じような内容になる傾向があった

旧AASやVSMSでは、回答を逐次的の一つずつ探すため、回答が偏ったり、また見当違いな回答を選ぶ傾向があった。AASにそういった傾向が見られなかったのは、コスト最小となるような複数の回答を同

時に選び出すという仮説推論の効果によるためだと考えられる。

また、一般にキーワードの入力方法として広く用いられている Boolean 表現 (AND, OR を用いた表記法) は、あらかじめキーワード間の論理関係を明確にする必要があり、人に負担がかかる上、その論理関係が正しいという保証もない。一方、提案手法では、コスト最小となるような最適なキーワードの組み合わせを選択するので、質問がキーワードのみの入力であるに関わらず、非常に柔軟に、かつ最適にキーワードを扱えるという利点も明らかになった。

次に、適切な回答が得られないケースの理由を考え、それぞれについて AAS と旧 AAS・VSMS とを比較してみる。

1. そもそも回答が存在しない
2. 回答を見つけられない
3. キーワードが不十分
4. コスト最小となる回答が複数存在する

1. は、既存の回答をどう組み合わせても回答を作成することが不可能な場合を指しており、この場合は素直に「回答は存在しない」と判断するべきである。今回の実験では 10 番がこれに当てはまると思われるが、AAS にはコストにしきい値、旧 AAS・VSMS にはベクトルのなす角度にしきい値を設けることによりこの問題は解決できると考える。

2. は回答が存在するにも関わらず誤った回答を出力する場合を指すが、AAS に関しては 2. を理由とする誤った回答は得られなかったので、AAS の回答を見つける能力は高いと思われる。しかし、旧 AAS・VSMS では根本的な問題なので、解決策は思いつかない。

3. はキーワードが十分、不十分かを一概に判断することは難しいことを意味している。AAS ではユーザの入力したキーワードの他に土台キーワードを重要な手がかりとして用いることによりキーワードの不十分さを補っているが、旧 AAS・VSMS ではユーザが補う以外に術がない。

4. は AAS 特有の問題だが、コストが同じ回答の優劣をどのように決めるのかということである。この問題に関しては、現在検討中である。

以上より、統計的評価は行っていないが、AAS は従来の質問応答システムに比べて、よりユーザの質問を満たすことができるシステムであると言える。

5 結論

FAQ 文書を仮説知識とその重みで表すことにより、ユーザの質問を満たす、かつユーザの興味に沿った文書を最小の数だけ取り出すシステム AAS を実装し、実験による評価を行なった。そして、適切な複数の文書を回答とすることがユーザの質問をより深く満たすことにつながることを実験により示した。

今後は、実験データを増やすと共に、よりふさわしい評価方法を考える。また、コストが等しい回答に対する対処も検討を重ねる予定である。

参考文献

- [1] Hammond, K.J., Burke, R. and Schmitt, K.: "A Case-Based Approach to Knowledge Navigation" In AAAI Workshop on Knowledge Discovery in Database. AAAI. 1994.
- [2] Burke, R., Hammond, K., Kulyukin, V., Lytinen, S., Tomuro, N. and Schoenberg, S.: "Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder System", University of Chicago, Department of Computer Science Technical Report, 1997.
- [3] 松村 真宏, 大澤 幸生, 谷内田 正彦: "AAS:過去の回答の自動組み合わせに基づく質問応答システム", 情報処理学会研究報告 SIG-ICS No.112, p.21-27, 1998.
- [4] 三輪真木子: 「データベースサーチャーの視点」. 情報処理学会誌, Vol.33, No.10, 1992.
- [5] Salton, G. and Buckley, C.: "Term-Weighting Approach in Automatic Text Retrieval", Readings in Information Retrieval, pp.323-328, 1998.
- [6] Yukio Ohsawa and Masahiko Yachida, : "An Extended Polynomial Solvability of Cost-based Abduction, Poster Session Abstracts", International Joint Conference of Artificial Intelligence (IJCAI'97) 79 (IJCAI'97), 1997.
- [7] 大澤幸生, Nels E. Benson, 谷内田正彦: 「KeyGraph: 単語共起グラフの分割・統合によるキーワード抽出」, 電子通信学会誌論文誌 J82-D1, No.2, 1999.
- [8] ヘルスカウンセリングサービス "Go ASK AI-ICE"
<http://www.columbia.edu/cu/healthwise/>