

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

松村 真宏* 大澤 幸生* 谷内田 正彦*

*大阪大学大学院基礎工学研究科

概要

本研究では、FAQ 文書（よくある質問とそれに対する回答を集めたもの）を用いて、ユーザの質問に答えるシステムの作成を目指す。従来の研究にFAQファインダーがあるが、これはFAQ文書を知識をして利用してユーザの質問に答えるシステムである。しかし、FAQファインダーでは過去のどの質問にも当てはまらないような質問に対しては、ユーザを満足させる回答を返すことができなかった。そこで、本研究ではFAQ文書を用いてユーザが満足するような質問応答システムの構築を目指し、Answer Assembling Systemを提案する。これは、ユーザの質問がFAQ文書中のどの質問にも当てはまらない場合でも、FAQ文書中の回答を組み合わせることによりユーザが満足できる回答を自動作成するシステムである。

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

Naohiro Matsumura*, Yukio Ohsawa* and Masahiko Yachida*

*Graduate School of Engineering Science, Osaka University

Abstract

In daily 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 users' satisfaction.

1 序論

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ったとする。こ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、実際問題として電話が繋がらないことが多い。自分で調べるにしても、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きな問題ではない。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を解決するであろう。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予め用意しておき、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、元々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。

知識としてFAQ文書を持つことの利点は、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ことにある。しかも、元々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ことが多い。

しかし、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るから、書式を統一することは困難である。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が容易になるが、巨大なデータベースを作る際の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のだが、それと同時に質問に対する回答を見つけ出すことが難しくなる。また、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きりがなく、事実上不可能である。

2 従来の質問応答システム

知識としてFAQ文書を用いる質問応答システムの従来研究にFAQファインダー[2]がある。これは、ユーザが自然言語で質問文を与え、FAQ文書の中からユーザの質問に最も近い質問を探してその回答を出力する

システムである。FAQファインダーでは、ユーザの質問を決まった形式に分類し、各形式ごとにFAQ文書との照合の方法を用意している。例えば、“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?)という質問は「比較」という形式の質問と判断し、FAQファインダーはFAQ文書から同じ形式(比較)で同じ用語(purified waterとtap water)を含む質問を探す。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、エンジンに用いるオイルなのか料理に用いるオイルなのか判断できない時は、FAQファインダー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、ユーザが候補の中から選択する。このように、ユーザとシステムが対話形式で回答を絞り込んでいくのが従来のFAQファインダーの特徴である。

FAQファインダーはユーザの質問がFAQ文書中のどの質問にも当てはまらない場合には、回答を出力しない。また、得られた回答がユーザの望んでいた回答でない場合は、ユーザが質問を変えて望ましい回答を探すのであるが、概してこの操作は難しい。なぜならユーザの望んでいる回答が元から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文を変えれば望むべき回答が得られるのかがユーザには全く分からないからである。そういう場合を想定して、ユーザの質問に関連のありそうな回答の一覧を出力するという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、効率が悪い上にユーザに与える負担も大きい。

3 Answer Assembling System(AAS)

前述したように、従来の質問応答システムでは回答を得るまでに乗り越えなければならないハードルがいくつもあり、ユーザはなかなか望ましい回答を得ることができなかった。そこで本研究では、このハードルを低くするためにユーザの質問がFAQ文書中のどの質問にも当てはまらない場合の処理を改善することに主眼を置いた。

人間が情報収集を行なう際には、多少情報が欠落していても過去の経験を用いてそれを補う(推論すること)により、様々な状況に柔軟に対処することができる。同様に、計算機による質問応答システムでも過去の経験をFAQ文書に置き換えることにより情報の欠落を埋めることができるのではないかと考えられる。

しかし、計算機に人間が行なうような高度な推論を行なわせることは非常に難しいという問題がある。

そこで本研究では、質問応答システムにおいて情報の欠落を補えるようなアルゴリズムを考え、実際に実装した。その実装した質問応答システムを Answer Assembling System(AAS)と呼ぶ。そして、実験を行なうことによりアルゴリズムの性能を評価した。このAASが目指すのは、ユーザの質問が過去にされたどの質問にも当てはまらない場合でも、過去の回答を組み合わせるによりユーザの質問に対する回答を自動作成するシステムである。

3.1 AASの基本となるアイデア

本研究の中心となるアイデアは、「ユーザの質問に対する回答がFAQ文書中に存在しない時でも、既にある回答を組み合わせるにより適切な回答を作成できることは多い」である。

例えば、「喉が痛くて熱がある」という症状の原因について知りたい時、もしFAQ文書中にこれと同じ質問がなければ、従来の質問応答システムではユーザが満足する回答を出力することは出来なかった。しかし、FAQ文書中に同じ質問がない場合でも、「喉が痛い」ことに関する質問と「熱がある」ことに関する質問がFAQ文書中に存在すれば、その2つの質問に対するそれぞれの回答をひとまとめにしてユーザに提示することにより質問に答えることができる。

3.2 AASの処理の流れ

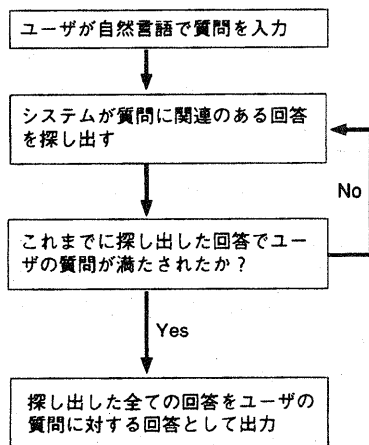


図 1: AAS の処理の流れ

ユーザは AAS に対して質問を自然言語で与える。その質問に対して、AAS は FAQ 文書の中から最もふ

さわしい回答を1つ探す。この時点で、AAS は探し出された回答によりユーザの質問が十分に満たされたかどうかを判断する。もしまだ満たされていないと判断したら、その満たされていない質問の部分を探め、その部分を満たすような回答を探す。この操作を繰り返すことにより、最終的にユーザの質問を満たすような回答をいくつか選び、それをひとまとめにして出力する。

3.3 質問の与え方

システムに質問を与える手段として、従来からキーワードと呼ばれる文章の主張を表す語句を複数与えることが行なわれてきた。しかし、これにはユーザが質問したい内容をキーワードだけで表現することは難しいという問題がある [3]。

ユーザにとって最も自然なインターフェイスは自然言語による質問であると考えられる。そこで AAS では、質問は自然言語で与えることにし、その質問文からキーワードを自動抽出することにした。また、適切なキーワードが思いついた時は、キーワードを折り混ぜての入力も可能である。

3.4 キーワード抽出法

自然言語で書かれた FAQ 文書を計算機が処理できるようにするために、FAQ 文書に含まれる全ての質問と回答のそれぞれに対してキーワードをいくつか抜き出す必要がある。文書からキーワードを抽出することを人が行なうのは大変な労力を伴うので、計算機によるキーワード自動抽出に関する研究が従来から行なわれてきた。本研究ではその中から Information Retrieval 分野で広く用いられている TFIDF[4] を参考にし、また扱う文書が FAQ 文書をいう点を考慮して評価関数 $Value()$ (式 (1)) を作成した。

文書から抜き出したキーワードの候補となる単語それぞれに対し評価関数 $Value()$ を用いて評価値を計算し、その値の高い単語をいくつか(無指定時は 10 個)選び、それをキーワードとした。なお、キーワードの候補とは、文書に含まれる単語から、“the” や “have” などのキーワードにはなりえない単語(ノイズワード)を除いた単語のことを指す。なお、ノイズワードは予め AAS に知識として与えてあり、キーワードの候補として選ばれることはない。

この評価関数は、少数の文書に集中して現れる単語や、質問とその回答の文書に共通して現れる単語に対しては高い評価値を割り当てる。

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

ここで、term は文書に出てくるキーワードの候補、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、weight(term) は term の重みを表している。FAQ 文書は質問とそれに対する専門家の回答が対になっており、その対には一貫した共通の主張があると考えられる。従って、FAQ 文書の質問と回答のペアにおいて、質問文にも回答文にも現れる単語はキーワードになる可能性が高いと考え、そのような単語には weight = 2、それ以外の単語には weight = 1 を与えた。

この TFIDF を参考にしたキーワード抽出法は簡単な計算によってキーワードを求めることができる利点がある。

3.5 回答生成のアルゴリズム

AAS はユーザの質問を FAQ 文書と機械的に照合することにより回答を探す。照合は文書の主張をベクトル空間で表現したキーワードベクトルを用い、そのキーワードベクトル同士のなす角度に基づいて行なう。なお、キーワードベクトルはそれぞれの文書に対してキーワード抽出法により取り出されたキーワードとその評価値を用い次のように定義される。

$$\begin{pmatrix} \text{Value}(\text{keyword}_1) \\ \text{Value}(\text{keyword}_2) \\ \text{Value}(\text{keyword}_3) \\ \vdots \end{pmatrix} \quad (2)$$

Value(keyword_i) : keyword_i の評価値

キーワードベクトルは文書の主張をベクトル空間で表現していると考えられる。つまり、それぞれのキーワードベクトルの向きと大きさは、それぞれ文書の主張とその強さを表す。したがって、キーワードベクトル同士のなす角度が小さいならそのキーワードベクトルが表す文書の主張はお互いに似ている、もしくは関連があると判断できる。したがって、FAQ 文書のそれぞれの文書のキーワードベクトルの中からユーザの与えた質問文から作成したキーワードベクトルと最もなす角度が小さいキーワードベクトルを探すことにより、ユーザの質問に対する回答を見つけることができる。図 2 にその関係を図示する。図中の Query は

ユーザの与えた質問文のキーワードベクトルを表し、Answer は回答に選ばれた文書のキーワードベクトルを表す。点線のベクトルは、回答として選ばなかった文書のキーワードベクトルを表す。

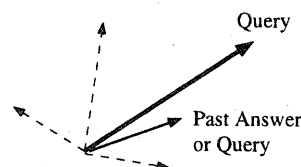


図 2: 回答探索

このようにして取り出された回答によってユーザの質問が十分に満たされたかどうかは、ユーザの質問のキーワードベクトルと取り出した回答のキーワードベクトルとの差分ベクトルの大きさから判断できる。というのもこの差分ベクトルはユーザの質問の満たされていない部分を表すと考えられるからである。したがって、この差分ベクトルの大きさが小さいとユーザの質問は満たされたと判断できる。また、差分ベクトルの大きさがある値より大きい、すなわちユーザの質問が満たされていない時は、この差分ベクトルを新たな質問のキーワードベクトルと見なし再び回答を探す。

図 3 にユーザの質問のベクトルと取り出した回答のベクトルからユーザの質問のまだ満たされていない部分を求める様子を図示する。ここで Unsatisfied Query はユーザの質問のまだ満たされていない部分を表すベクトルである。

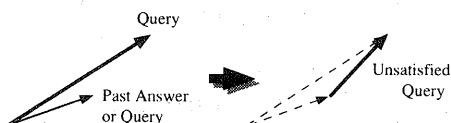


図 3: まだ満たされていない質問を表すベクトル

以上の動作をユーザの質問が満たされるまで、つまり差分ベクトルの大きさが小さくなるまで繰り返すことによりユーザの質問を満たす回答を幾つか選ぶことができる。

また、従来の質問応答システムでは、ユーザの質問と FAQ 文書中の質問との照合を行なうが、AAS では FAQ 文書中の質問と回答のそれぞれに対して照合を行なう。これは、FAQ 文書中の質問文が曖昧な場合や、回答にのみ現れるキーワードにも対応できるようにするためである。

以上をまとめたのが、次の回答生成アルゴリズムである。

回答生成アルゴリズム

以下にキーワードベクトル同士のなす角度に基づいた回答生成のアルゴリズムを示す。

- 1) FAQ 文書中の全ての質問と回答それぞれに対して、キーワードベクトル $\overrightarrow{\text{Document}}_{(i)}, i = 1, 2, 3, \dots$ を生成する。回答文書集合 $S = \phi$ とする。
- 2) ユーザの入力した質問文からキーワードベクトル $\overrightarrow{\text{Query}}$ を生成する。
- 3) $\overrightarrow{\text{Query}}$ と $\overrightarrow{\text{Document}}_{(i)}, i = 1, 2, 3, \dots$ とのなす角度 θ を式 (3) により求め、 θ が最小となる文書の番号を k とおき、文書 k を S に加える。このとき文書 k が質問の文書なら、その質問に対する回答の文書を加える。

$$\theta = \cos^{-1} \left(\frac{\overrightarrow{\text{Query}} \cdot \overrightarrow{\text{Document}}_{(i)}}{|\overrightarrow{\text{Query}}| |\overrightarrow{\text{Document}}_{(i)}|} \right) \quad (3)$$

- 4) 次式 (4) を用いて、ユーザの質問の満たされていない部分を表すベクトル $\overrightarrow{\text{Unsatisfied Query}}$ を求める。

$$\overrightarrow{\text{Unsatisfied Query}} = \overrightarrow{\text{Query}} - \overrightarrow{\text{Document}}_{(k)} \times \frac{\text{Que_Value}(x)}{\text{Doc_Value}(k, x)} \quad (4)$$

ここで、 x は $\overrightarrow{\text{Document}}_{(k)}$ の中で評価値最大のキーワード、 $\text{Que_Value}(x)$ は $\overrightarrow{\text{Query}}$ のキーワード x の評価値、 $\text{Doc_Value}(k, x)$ は文書 k のキーワード x の評価値である。

関数 $\text{Que_Value}()$ と関数 $\text{Doc_Value}()$ は、差分ベクトルの成分がマイナスにならないように、取り出した文書のキーワードベクトルを補正するための関数である。

- 5) $|\overrightarrow{\text{Unsatisfied Query}}|$ が設定したしきい値以上なら、 $\overrightarrow{\text{Query}} = \overrightarrow{\text{Unsatisfied Query}}$ として 3) へ。そうでなければ 6) へ。
- 6) S を出力する。

3.6 AAS の出力について

AAS は、FAQ 文書中の回答をいくつか取りだし、それらをひとまとめにして出力する。自然な回答の作成を目指すのなら、取り出した回答を一旦バラバラにし

てから最構成して一つの回答に作り直すということも考えられる。しかし、FAQ 文書の回答はそれ自体で完結していて良くまとまっているのが特徴であるから、再構成させると知識ベースとして FAQ 文書を用いた利点が生かせない。従って、回答の文書をバラバラにして再構成させる必要はないと考えている。

4 実験に基づく AAS の評価

AAS を、Pentium-Pro200MHz(128MB) の DOS/V(Linux) 上に perl5.0 を用いて実装した。また AAS の知識ベースとして用いた FAQ 文書には、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” [5] から 1220 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、実験を行なう際にはその点に注意を払いつつ行なった。

4.1 AAS の動作例

AAS の動作を具体例を用いて説明するために、次のような質問文を与えた。実際の AAS への入力画面は、図 4 のようになる。

“If I injected, snorted or smoked heroin, marijuana and the other drugs, what risk would I have?”

(訳：もしヘロインやマリファナやその他の麻薬に手を出せば、どのような危険がありますか?)

AAS は、この質問に対する回答として

- 「マリファナに関する文書」
- 「ヘロインに関する文書」
- 「麻薬が危険であることについて述べた文書」

の 3 つの文書を組み合わせてユーザに提示した。

AAS は、ユーザの質問に対する回答として、まず「マリファナに関する文書」を選び、この回答によりユーザの質問が満たされたかどうかを判断する。その結果、AAS は「ヘロインや他の麻薬が危険であること」に関する質問がまだ満たされていないと判断し、次にその質問を満たすような回答を新たに探す。AAS は、新たに取り出された「ヘロインに関する文書」によりユーザの質問が満たされたかどうかを再び判断し、「その他の麻薬が危険であること」に関する質問の部分がまだ満たされていないと判断する。そして、その質問を見たような「麻薬が危険であることについて述べた文書」を取り出す。ここで初めて AAS は

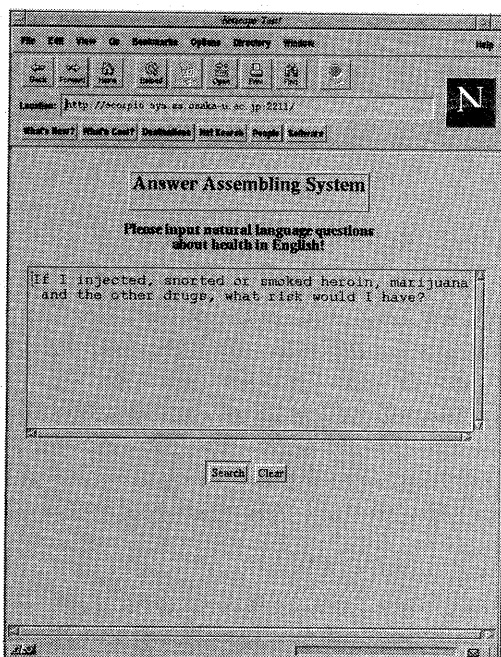


図 4: AAS の入力例

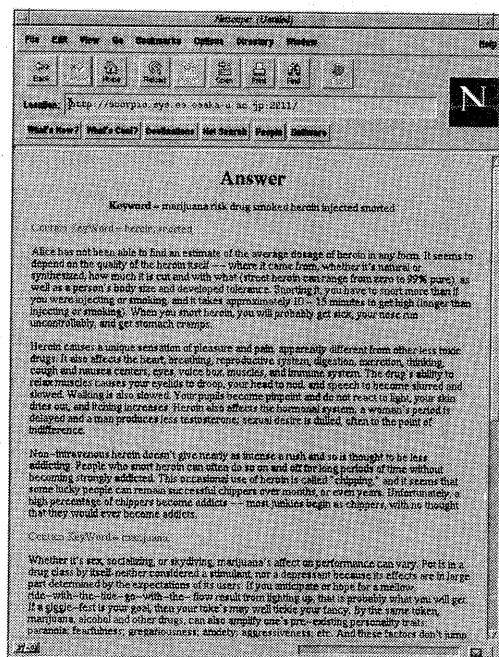


図 5: AAS の出力例

ユーザの質問が満たされたと判断し、取り出した回答をユーザに提示する。

AAS の出力画面を、図 5 に示す。

4.2 AAS の実験に基づく評価

18 問の健康に関する質問を用意して、AAS と従来の質問応答システムの出力を比較する実験を行なった。なお、AAS が複数の文書を取り出して回答を作成するのに対して、比較している従来の質問応答システムが 1 つしか文書を選ばないのは条件が平等でないと考えられるので、比較する質問応答システムが取り出す文書を 1 つとはせず、AAS が取り出す文書と同じ数だけ文書を取り出すようにした。

実験の評価には、Recall(再現率)や Precision(適合率)がよく使われるが、AAS や質問応答システムの性能を評価するのにこの値は適切ではない。というのも、これらのシステムにおける回答は個々の文章ではなく、一連の出力全文本(図 5 参照)でユーザの意図を満たしているかどうか重要であって、これは Recall や Precision の値では分からないからである。したがって、評価の基準をユーザの質問の意図が満たされたかどうかを設定して実験の評価を行った。実験結果を以下の表 1 に示す。

表 1: AAS と従来の質問応答システムの実験結果

	満足な回答が得られた	不十分
質問応答システム	9	9
AAS	13	5

表 1 から分かるように、従来の質問応答システムでは 18 問中 9 問しかユーザが満足する答えを返すことができなかったのに対し、AAS では 18 問中 13 問に対して満足のいく結果を出力することができた。

実験結果を詳細に検討してみると、従来の質問応答システムでは質問のうち最も重要なキーワードに関する回答を中心に選ぶことが多かった。また、AAS ではユーザの質問に含まれるキーワードを網羅するように複数の文書を取り出し回答を作成するので、質問応答システムに比べて回答に偏りが少なかった。

つまり、質問応答システムでは同じような内容の回答を複数選ぶことはできても、ユーザの質問を十分に満たすようには回答を選ぶことは少なかった。一方、AAS では質問応答システムでは選ぶことができなかった回答として有効な文書を取り出すことができた。それ故、AAS の方がよりユーザを満足させる結果を得

られたのだと考えられる。

4.3 問題点

前節の実験において、AAS でうまく結果が出なかった原因は、次の3つに集約される。

1. キーワード抽出の際の精度
2. 類義語、同義語の存在
3. 回答生成過程における2番目以降の回答の精度

まず最初の問題として、キーワード抽出方法の精度の問題がある。今回用いたキーワード抽出法では、キーワードを単語の3種類の出現頻度(TF,DF,weight)だけを手がかりにして求めたので、計算が簡単な反面、ノイズワードもキーワードに選んでしまうことがしばしばあった。そのため、キーワードベクトルが文書の主張を正確に表さなくなり、それ故AASがふさわしくない回答を選んでしまうことがあった。同じ単語でも文書によって意味や重要度は変わるので一概にノイズワードを決めることは難しい。これらの解決手法として、WORDNET[6]による文意の考慮や、SMART information retrieval system[7]による統計的情報抽出、Xerox taggerによる構文解析などがあるが、キーワード抽出の精度を上げることはそれだけで1つの研究分野になるほど難しい問題であり、この問題の解決には、まだしばらく時間がかかりそうである。

2番目の問題として、類義語、同義語の存在がある。ユーザが用いる語彙とFAQ文書の回答者が用いる語彙には少なからずギャップがあり、それによりユーザの意図したことがうまく回答に反映されないことがあった。

3番目の問題に、回答を取り出す過程において後半になるほど選び出す回答の精度が下がる傾向があった。これは、後半になるほど差分ベクトルが短くなるため、類似するキーワードベクトルを探す際の誤差が大きくなるからだと考えられる。また、ベクトル空間モデルにおいて、最も関連のある文書を取り出す際にキーワードベクトルの各軸(キーワード)は直交していると仮定しているが、実際にはキーワードには意味的な関係があるためこの仮定は成立しないという報告もある。現時点では回答を選ぶ際にベクトル同士のなす角度にしきい値を定めて、ある値以下なら回答として不適当であるという判断を加えているのだが、しきい値の理論的な判断基準はなく、AASではこのしきい値を筆者達の試行錯誤により定めている。

また、回答が根本的に存在しないとき、すなわちFAQ文書中の回答をどのように組み合わせても回答

が作成できない場合は、素直に回答を出力しないというのも賢い選択である。現在のシステムはその判断をキーワードベクトル同士のなす角度により判断しているのだが、この方法もまだ確立されてはおらず、改善の余地が残されている。

5 まとめ

本研究で提案し作成したAASは、回答の用意されていない質問に対しても、既にある回答を組み合わせることにより従来の質問応答システムに比べよりユーザが満足する回答を出力できた。また、AASではユーザの質問に含まれるキーワードを網羅するように回答を取り出せることが実験により明らかになった。これは従来の質問応答システムにはない、AASの大きな特徴であると言える。

今後の課題としては、回答の精度を上げる手法の開発、回答生成のアルゴリズムの改良を考えている。

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] 三輪眞木子: 「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
- [4] Salton, G. and Buckley, C.: "Term-Weighting Approach in Automatic Text Retrieval", Readings in Information Retrieval, pp.323-328, 1998.
- [5] ヘルスカウンセリングサービス
(URL) <http://www.columbia.edu/cu/healthwise/>
- [6] WORDNET
(URL) <http://www.cogsci.princeton.edu/~wn/>
- [7] Buckley, C.: "Implementation of the SMART Information Retrieval [sic] System", Technical Report 85-686, Cornell University, 1985.