

質問事項の抽出とその依存関係の特定

田村 晃裕[†] 高村 大也^{††} 奥村 学^{††}

[†] 東京工業大学 大学院総合理工学研究科 ^{††} 東京工業大学 精密工学研究所
aki@lr.pi.titech.ac.jp {takamura,oku}@pi.titech.ac.jp

1 はじめに

既存の質問応答システムの入力は、「日本の首相は誰ですか?」のような一文で構成される質問のみを想定しているといつてよい。これは、既存の質問応答システムを評価するためのワークショップである、TREC の QA-TRACK[2] や NTCIR の QAC[3] で扱われている質問が、全て一文で構成されていることから分かる。

一方 Web には、質問をのせておく他のユーザが答えてくれる Q&A サイト (Yahoo!知恵袋、教えて!goo など) がある。そこによせられる質問には、「デスクトップ上のアイコンやファイルが消えてしまいました。どうやったら元の状態にもどせますか?」のような複数文で構成される質問 (複数文質問) も数多くある。複数文質問には、前述した例「デスクトップ上の～」のように複数文で一つの事柄を尋ねる質問と、「夏に会社の同僚とポーランドに行くことになりました。有名な観光地はどこでしょうか? また、何語が話されているのですか? よろしくお願いします。」のように複数文で二つ以上の事柄を尋ねる質問の 2 種類ある。

これらの複数文質問 (両種類とも) は、既存のシステムでは想定外の入力で、既存の手法をそのまま適用できない。もしくは、適用できたとしても精度が極端に落ちてしまう。そこで、本研究では、これらの複数文質問にも対応できるシステムを構築するために必要な技術を提案する。今回扱う技術は、「質問事項の抽出」と「質問事項間の依存関係の特定」の 2 つである。「質問事項の抽出」とは、質問事項が何かを正しく把握するための処理であり、「質問事項間の依存関係の特定」とは、質問事項が複数あった場合、それらの質問事項同士の依存関係を把握する処理である。以降、2 節で「質問事項の抽出」について詳しく述べ、3 節で「質問事項間の依存関係の特定」について詳しく述べる。そして、最後に 4 節でまとめと今後の課題を述べる。

2 質問事項の抽出

複数文質問は、1 節に述べたように質問事項が一つとは限らないため、まず質問事項が何か、また何個あるかを把握する必要がある。そのための技術として「質問事項の抽出」を扱う。この処理は、複数文質問を構成する文から、質問事項一つにつきその質問事項を最も表す文 (以降、[1] 中の core sentence にあわせて核文と呼ぶ) を一文抽出することを目的とする。つまり、一つの質問事項は一つの核文で表されているという仮定を定め、質問事項の抽出を核文の抽出によって実現する。ここで、この仮定の下では質問事項の数と核文の数は等しくなるので、核文の数を数えることで、質問事項の数が把握できることを確認しておく。

具体的には、1 節の「夏に会社の～」という複数文質問から二文目「有名な観光地はどこでしょうか?」と三文目「また、何語が話されているのですか?」を核文として抽出することを目的とする。この例では、核文が二つなので質問事項数が二つと分かる。

ここで、この処理は一見、疑問形の文を核文として抽出すればよいように思える。しかし、核文には色々なタイプの文がなりうる。例えば「～を教えてください。」などの疑問形をしていない文や「～が分かりません。」のように暗に解答を求める文も核文になりうる。この点がこのタスクの難しいところである。

以下、2.1 節で本タスクに対する提案手法を述べ、2.2 節で手法の有効性を実験を通して確かめる。

2.1 提案手法

本タスクに関連する研究として岩崎ら [4] が挙げられる。岩崎らは、コールセンターの対話データを文ごとに、SVM を用いて挨拶、質問、応答、返事、結論、その他の 6 種類に分類している。この手法は、質問を表す文を抽出することは可能であるが、本タスクは質問を表す文の中から質問事項を最も表す文 (核文) を抽出することが目的であるので、抽出時に必要な情報は異なり、適さないと考えられる。また、以前、我々は本タスクの前段階として、質問事項が一つである複数文質問に限って核文抽出を行っている [1]。この手法を基に、複数の質問事項を含む複数文質問にも対応する手法を提案する。提案手法の手順は以下の通りである。

step1 文分割

入力として与えられた複数文質問を文に分割する。

step2 一文ずつ核文であるか判定

step1 で文分割された一文ごとに SVM を用いて核文であるか判定する。

step3 核文の決定

step2 で SVM による判定の結果、SVM が核文と判定した文 (SVM が出力する分離平面からの距離が正となった文) を全て核文とする。

ここで、[1] と本提案手法との違いは step3 である。[1] では、入力を質問事項が一つである複数文質問に限ったので、核文は 1 つである。したがって、SVM により判定された分離平面からの距離を比較して核文を 1 つに決めている。しかし、今回は複数の質問事項を含む複数文質問も扱うため、核文も一つとは限らない。したがって、分離平面からの距離の比較は行わずに、距離が正となったもの全てを核文としている点が [1] との違いである。

step2 の SVM で核文かどうか判定する際の素性について述べる。素性は、形態素の unigram と bigram を

表 1: パターンマッチとの比較

手法	精度	再現率	F 値	正解率
パターンマッチ	0.745	0.862	0.800	0.567
提案手法	0.896	0.889	0.893	0.755

用いた．核文かどうかを判定する際は，判定する文の前にどのような文があるのか，後にどのような文があるのかを考慮する必要があるので，素性に判定文の前後の文の情報も使用した．この部分は，[1]と同様なので，詳しくはそちらを参照してもらいたい．以上より，素性ベクトルは（判定文より前にある文の unigram・bigram，判定文の unigram・bigram，判定文より後にある文の unigram・bigram）とした．

2.2 実験

2.1 節で提案した手法の有効性を実験を通して確認する．本研究の目的から，実験データとして Q&A サイトから収集した複数文質問 2387 個を用いる．Q&A サイトは，はてな¹と Yahoo!知恵袋²の 2 つを選んだ．この 2387 個の質問に対して予め核文を手でタグ付けしておいた．2387 個の複数文質問の平均質問事項数は 1.4 個，平均文数は 3.6 文であった．また，2387 個の内，質問事項が一つの場合は 1593 個（平均文数：3.4 文），質問事項が複数なのは 794 個（平均質問事項数：2.2 個，平均文数：4.0 文）であった．本実験の評価指標は，精度と再現率の調和平均である F 値と全複数文質問の中で過不足なく核文を正しく抽出できた質問の割合を示す正解率を用いる．ここで，精度とはシステムが抽出した核文の内どれだけ正しいかを示し，再現率とは実際の核文の内どれだけ抽出できたかを示すものである．評価は 2 分割交差検定により行う．SVM による学習には，SVM 学習パッケージ TinySVM³を使用した．また，文の形態素を抽出する際は，日本語形態素解析器 ChaSen⁴を用いた．

まず，本タスクが疑問形の文を抽出すれば済むような簡単なタスクではない事を示すと共に，本手法の必要性を確認するために，単純なパターンマッチによる手法と本手法との比較を行う．両手法で核文抽出した結果を表 1 に示す．ここで，パターンマッチによる手法は，文末が「？」で終わる文，「教えて」「探して」，「動詞の連用形＋たい（助動詞）」などを含む文を核文として抽出するものである．また，提案手法の素性には判定文の前後の文に関するものがあるが，前後の文として同質問内の全ての文を対象にした．つまり，「前後一文」のような範囲は設けていない．

表 1 より，本タスクはパターンマッチのような単純な手法で解決できるほど容易ではなく，SVM を用いた提案手法が有効であることが分かる．

次に，提案手法で用いた素性（unigram, bigram）の有効性や前後の文脈の有効性を調べるために，素性を変えて核文抽出実験を行った．結果を表 2 に示す．

表 2 より，素性は，unigram と bigram の両方を用いるのが効果的であることが分かる．文脈情報については，考慮した方がしないよりも結果がよいことが分かる．また，考慮する文脈の範囲を広げれば広げる程，結

表 2: 素性変更時の核文抽出結果

	unigram	bigram	unigram+bigram
	F 値, 正解率	F 値, 正解率	F 値, 正解率
文脈なし	0.875, 0.704	0.881, 0.713	0.886, 0.726
前後一文	0.870, 0.703	0.877, 0.714	0.885, 0.730
前後二文	0.879, 0.729	0.879, 0.724	0.887, 0.738
前後全て	0.887, 0.756	0.882, 0.745	0.893, 0.755

表 3: 補正後の核文抽出結果

文脈情報	補正	F 値	正解率
文脈なし	なし	0.886	0.726
	あり	0.897	0.768
前後全て	なし	0.893	0.755
	あり	0.897	0.773

果は改善されていくことも分かる．これは，次の例のように，核文になりそうな文と核文が離れて存在することもあり，このような場合もお互いを考慮して，うまく核文決定ができるようになっていていることを示している．

（例）質問事項が一つの複数文質問

1 文目：核文になりそうな文（インストール方法が分かりません．）

2 文目～：説明（パソコンの環境など）

最後の文：核文（方法の分かるサイトを教えて．）

提案手法のままでは，抽出結果として複数文質問内に核文が存在しない場合も許してしまう．そこで，「入力には必ず質問である」という仮定と「複数文質問には核文が必ず一つはある」という事実を用いて，提案手法で核文を決定した結果，複数文質問内に一つも核文がない場合，最もふさわしそうな一文を核文とする補正を行った．ここで，SVM が出力する分離平面からの距離を核文としてのふさわしさと捉え，その値が最も大きい一文を核文とする．補正を行った場合の結果を表 3 に示す．

表 3 より，補正を行うことで結果が改善されている．これは，質問事項が一つの複数文質問に対する抽出精度の改善が大きく関係していると考えられる．つまり，補正前は一つも核文として認識されず間違いとなっていたものが，補正を行うことでうまく抽出でき，正解となったためと考えられる．また表 3 より，補正を行うことで，文脈情報を考慮した時とそうでない時との差が縮まっていることが分かる．この点について，質問事項が一つの複数文質問の精度に着目して考察する．文脈を考慮しない場合，補正前の F 値，正解率はそれぞれ 0.848, 0.726 で，補正後はそれぞれ，0.882, 0.790 に改善された．一方，全ての文を文脈として考慮した場合，補正前の F 値，正解率はそれぞれ 0.875, 0.785 で，補正後はそれぞれ，0.884, 0.814 に改善された．これより，補正前の抽出結果が，文脈を考慮しない場合の方が悪いことが分かる．そのため，文脈を考慮しない方がより補正の恩恵を受けやすく，結果もより改善されており，精度の差が縮まったと考えられる．

3 質問事項間の依存関係の特定

質問事項（核文）が複数ある場合，後方の質問事項に答える際に，前方の質問事項の答えを考慮する必要が

¹<http://www.hatena.ne.jp/>

²<http://knowledge.yahoo.co.jp/>

³<http://chasen.org/~taku/software/TinySVM/>

⁴<http://chasen.naist.jp/hiki/ChaseSen/>

ある場合とそうでない場合がある．例えば、「福岡のおみやげで有名なものは何ですか？その値段もいくら位か教えてください．よろしくお願いします」という複数文質問の場合、二つ目の質問事項（二文目）の答えは一つ目の質問事項（一文目）の答えとなる“おみやげ”に依存する．そのため、質問事項を前から順に処理していき、二つ目の質問事項に答える際には一つ目の質問事項を考慮する必要がある．一方で、1節の「夏に会社の～」の例の場合、二つ目の質問事項（三文目）を答える際に、一つ目の質問事項（二文目）の答えとなる“観光地”には依存しておらず影響を受けないので独立で扱える．このような違いを判定することも、複数文質問に正しく答える上で重要であると考えられる．ここで、前者のように質問事項 A に答える際に、質問事項 B の答えが必要である時、“質問事項 A は質問事項 B に従属している”と呼ぶ．質問事項が二つの時は、それらが従属しているかどうかを判定するだけでよいが、質問事項が三つ以上ある場合には、どの質問事項がどの質問事項に従属しているかも解明する必要がある．そこで、B が A に従属している時、親ノードを A とし、その子ノードを B とした木で表すことにし、1つの複数文質問の従属関係を 1 つの木で表すことにする．したがって、本タスクの目的は、核文が複数あった場合、その核文間の従属関係を特定し質問を木構造化することになる．具体例を図 1 に示す．

フュメド・ボワソンを探しています。
核文1 スーパーに売ってますか？
核文2 売っているとしたら、値段はどの位ですか？
核文3 また、どういふコーナーに売ってるんでしょうか？

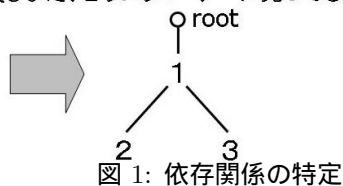


図 1: 依存関係の特定

以下、3.1 節で本タスクへの提案手法を述べ、3.2 節で手法の有効性を実験を通して確かめる．

3.1 提案手法

提案手法は SVM による手法を用いるが、訓練データとして与えられた質問の構造木から何を学習させるか（学習データの作り方）の違いにより、手法を 2 つ提案する．この 2 手法の比較は、3.2 節で行う．

まず一つ目は、手法 PARENT である．これは、SVM で親子関係を学習させるものである．具体的には図 2 を参照してほしい．この学習結果を用いてテストデータの木を構築する手順を以下に疑似コードで示す．前方の核文から順番にどの核文に従属するかを決め、木を構築していく．どの核文に従属するかは、前出の核文と一文ずつ従属関係にあるかを SVM を用いて判定していき、その結果を基に決める．ここで、従属先を決めている最中の核文を“判定文”，判定文の従属先として核文 A になるかを判定している時、核文 A を“比較文”と呼ぶことにする．

```

1: for  $i = 1, 2, \dots, n$  do
2:   for  $j = 1, 2, \dots, n - 1$  do
3:      $Likelihood(CS_i, CS_j) = SVM(CS_i, CS_j)$ 
4:   end

```

```

5: if ( $Likelihood$  の全ての値が負) then  $Parent(CS_i) = root$ 
6: else  $Parent(CS_i) = \operatorname{argmax}_{CS_k} Likelihood(CS_i, CS_k)$ 
7: end

```

ここで、 CS_i : i 番目の核文、 $SVM(CS_i, CS_j)$: 判定文を CS_i 、比較文を CS_j として SVM で従属判定し、分離平面からの距離を返す、 $Likelihood(CS_i, CS_j)$: 核文 CS_i の核文 CS_j に対する従属のしやすさ、 $Parent(CS_i)$: 判定文 CS_i の親ノードである．

二つ目は、手法 ANCESTOR である．これは、SVM で直接の親子関係だけでなく、祖先となるものを全て学習させるものである．具体的には図 2 を参照してほしい．この学習結果を用いてテストデータの木を構築する手順は、手法 PARENT の疑似コードの 3 行目を以下に置き換えることで実現できる．

```

3:  $Likelihood(CS_i, CS_j) = \sum_{CS_k \in Ancestor\{CS_j\}} SVM(CS_i, CS_k)$ 

```

ここで、 $Ancestor\{CS_i\}$: CS_i の祖先の集合を表す．これは、SVM の出力が従属させた際の正解ラベルに最も近いノードの下に従属させる事と等価である（証明は紙面の都合上省略する）．ここで、近さの指標としてコサイン距離を用いる．図 3 を参照してほしい．

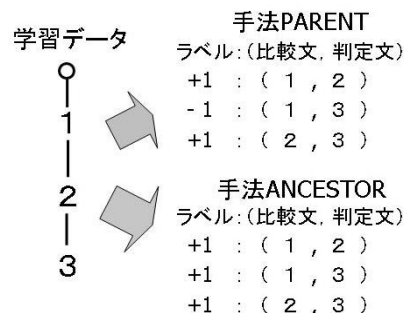


図 2: 手法 PARENT と ANCESTOR

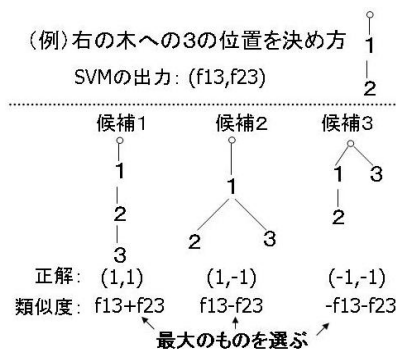


図 3: 手法 ANCESTOR による木の構築

次に素性について述べる．手法 PARENT, ANCESTOR 共に共通の素性を用いた．

判定文の素性 出現単語、接続詞有無、代名詞有無、連体詞有無、仮定形の単語有無、質問タイプ

比較文の素性 質問タイプ、判定文との共通単語（名詞・未知語）数

直前の文（比較文以外）の素性 判定文との共通単語（名詞・未知語）数

その他 判定文と比較文の間に文があるか、判定文と比較文の間に比較文と並列な核文があるか

質問タイプについては、[1] で定めたものを用いたので、そちらを参照してもらいたい．

表 4: 従属関係特定実験の結果

	手法 PARENT		手法 ANCESTOR		baseline
	提案素性	単語のみ	提案素性	単語のみ	
二項間一致率	75.5%	71.7%	77.2%	73.0%	71.5%
完全一致率	73.0%	70.4%	74.1%	70.5%	67.4%

単語のみ：素性として判定文の単語と比較文の単語を使用

表 5: 従属関係特定で有効な素性上位 5 個

手法 PARENT		手法 ANCESTOR	
除いた素性	完全一致率	除いた素性	完全一致率
判定文の単語	71.0%(-2.0%)	判定文の単語	71.0%(-3.1%)
比較文の質問タイプ	71.7%(-1.3%)	比較文の質問タイプ	71.2%(-2.9%)
仮定形の単語の有無	72.3%(-0.7%)	並列な核文の有無	73.3%(-0.8%)
判定文の質問タイプ	72.9%(-0.1%)	判定文の質問タイプ	73.3%(-0.8%)
並列な核文の有無	72.9%(-0.1%)	仮定形の単語の有無	73.3%(-0.8%)

3.2 実験

実験データは、2.2 節で使ったデータの内、質問事項が複数である複数文質問 794 個を使用した。794 個の内、質問事項が二つの質問が 637 個 (80%)、質問事項が三つの質問が 119 個 (15%)、質問事項が四つ以上の質問が 38 個 (5%) であった。予め、正解となる木は人手で作成しておいた。実験の際は、核文がどれかを与え、従属関係の判定のみに焦点をあてた。評価指標は、システムにより作成した木と正解の木が完全に一致した時に正解とする完全一致率と、木の二項間が正しい時に正解とする二項間一致率を用いる。

手法 PARENT と ANCESTOR で従属関係特定実験を行った結果を表 4 に示す。また、ベースラインとして、全ての二項間関係を従属ではないと判定した時の結果を示す。

表 4 より、SVM を用いた手法 (手法 PARENT, ANCESTOR) はベースラインよりいい結果が得られていることが分かる。また、素性として単純に単語だけを入れるよりも、今回のような素性を用いて情報を明示的に入れた方が結果が良いことも分かる。

手法 PARENT と ANCESTOR を比べると手法 ANCESTOR の方がわずかながら結果がよい。この結果を考察する。結果が異なる要因は二つあると考えられる。SVM で学習したモデルをテストデータに適用した際の純粋な精度と、SVM での分類結果が誤っても、木を組む際に SVM の出力である分離平面からの距離を比較することで訂正される精度である。前者についてみると、手法 PARENT で正しく SVM が判定した正解率は 77.8%、手法 ANCESTOR では 78.3% になり 0.5% 結果がよい。このことより、本タスクにおいて提案素性を用いた学習では、親子関係より、祖先の方が学習しやすかったといえ、最終的な差につながったと考えられる。後者についてだが、前者による差は 0.5% だが、最終的な二項間一致率の差は 1.7% である。このことから、後者による影響でも手法 ANCESTOR の方が手法 PARENT よりも結果が良くなったと考えられる。

次に提案した素性の内どの素性がある有効であったかを、素性一つずつ抜いて実験を行うことで検証する。完全一致率を基に有効だった素性の上位 5 つを表 5 に示す。

表 5 より、両手法共に判定文の単語が一番重要である事が分かる。これは至極当然のことであるが、注目すべきは比較文の質問タイプが次に有効であったことである。この点を考察する。従属関係を質問タイプの移り変わりという視点で見ると、比較文の質問タイプが Yes-No 質問の場合、「16 歳ですけど、一ヶ月だけバイトできますか? いくらぐらい稼げますか?」の例

のように前の質問で確認をし、次にその事について質問をする従属関係が多くみられた。また、比較文の質問タイプが名詞になる場合の方がテキストになる場合よりも従属関係になりやすいことが分かった。これは、「好きな女性有名人は誰ですか? その方のプロフィールのあるページも教えて。」の例のように、前方の質問で名詞を尋ねると、次の質問でその名詞に関しての質問がしやすいためと考えられる。一方、前方で答えがテキストになる質問をしてしまうと、その答えとなるテキストに関する質問はし難いらしく、従属関係は少なかった。

4 まとめと今後の課題

本研究では、複数文質問にも対応できるシステムを構築するために必要な技術である“質問事項の抽出”と“質問事項間の依存関係の特定”について説明し、その手法を提案した。

“質問事項の抽出”では、SVM による手法で正解率 77.3% を実現できた。抽出の際は、前後の文脈を考慮することで精度が改善できることを示した。

“質問事項間の依存関係の特定”については、学習の仕方を変えた SVM による 2 手法で比較し、手法 ANCESTOR の方がわずかながら結果が良くなり、最高で完全一致率は 74.1% となった。また、従属関係を特定する際には質問タイプが重要な情報となっていることも示した。

“質問事項間の依存関係の特定”で扱ったデータには、核文が 3 つ以上のものが少なかったため、2 手法の差は現れにくかったと考えられる。今後はデータ量を増やしてより詳細に 2 手法の比較を行いたいと思っている。

謝辞

本研究を進めるにあたり、NTT コミュニケーション科学基礎研究所の平尾努氏にご指導、ご助言を頂きました。ここに、深くお礼を申し上げます。

参考文献

- [1] A. Tamura, H. Takamura and M. Okumura. “Classification of Multiple-Sentence Questions”, Proceedings of 2nd International Joint Conference on Natural Language Processing (IJCNLP 2005), pp.426–437, 2005.
- [2] E.M. Voorhees. “Overview of the TREC 2003 Question Answering Track”, TREC12, pp.54–68, 2003.
- [3] J. Fukumoto, T. Kato and F. Masui. “Question Answering Challenge (QAC1) Question answering evaluation at NTCIR Workshop 3”, Working notes of the Third NTCIR Workshop Meeting Part IV: Question Answering Challenge, National Institute of Informatics, pp.1–10, 2002.
- [4] 岩崎礼次郎, 荒木健治. “コールセンターの対話データを対象とした営業日報自動生成のための重要文抽出手法”, 第 19 回人工知能学会全国大会論文集, 2005.