修 士 論 文

多様な質問を受け付ける質問応答システムの回答選択に関する研究

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

横川 裕太

2015 年 3 月

修 士 論 文

多様な質問を受け付ける質問応答システムの回答選択に関する研究

指導教員　白井 清昭 准教授

審査委員主査　白井 清昭 准教授
審査委員　飯田 弘之 教授
審査委員　東条 敏 教授

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

1310075 横川 裕太

提出年月: 2015 年 2 月

概 要

質問応答とは、ユーザが自然言語で質問文を入力とし、その質問に対する回答を文書集合から抽出するタスクである。このタスクの処理を行うシステムは質問応答システムと呼ばれる。質問応答システムに関する初期の研究は、事実を表す固有名詞や短い文を回答とするファクトイド型の質問応答が主流であったが、近年は理由や方法を答える質問（ノンファクトイド型質問）など様々な質問に答えることのできる質問応答システムへの期待が高まっている。一般的な質問応答システムは、文書集合から回答の候補をいくつか抽出し、また入力された質問文からユーザが求めている情報（人物、場所、理由など）は何かを分類・特定した上で質問が求めている情報の種類に対応するものを回答として選択する。このような分類を回答タイプ（または質問タイプ）と呼ぶ。しかし、定義されていない回答タイプの質問や、回答タイプが曖昧な質問に対して、正しい回答を返すことは難しい。また回答タイプを事前に網羅的に定義することは困難である。本研究では、明確な回答タイプの分類を行わず、多様な質問を受け付けることのできる質問応答システムを提案する。提案システムでは、回答候補の適切さを「内容の関連度」と「回答タイプの整合度」で評価する。内容の関連度は、質問と回答候補の内容やトピックの類似性を評価する。一方、回答タイプの整合度は、質問と回答候補の回答タイプが一致しているかを評価する。ただし、回答タイプの分類は定義せず、単に質問文と回答のタイプが一致するかを判定する。本研究では、回答選択手法として、上記の 2 つのスコアの最適な組み合わせ方を提案する。また、回答タイプを明確に定義せず、質問に合う回答を判別する方法として、質問・回答それぞれの文の形式・特徴を素性とし、回答タイプの一致を判定する分類器を機械学習する手法を提案する。また、回答タイプの一致判定を行う二値分類器のための最適な訓練データの作成方法も検討する。

提案システムはウェブ文書を対象とした質問応答を行う。質問と回答の内容の関連度は、質問文と回答候補を自立語を素性とする単語ベクトルで表現し、それらのコサイン類似度から計算する。ただし、単純なコサイン類似度では、質問と関連のない回答候補の関連度が高く見積もられる可能性があるため、重要な語に対して高い重みを与えること、回答候補の周辺の文に出現する語も単語ベクトルに加えること、回答候補を含む文書のウェブ検索時のランクをスコアに反映することなどの手法を考案した。

回答タイプの整合度は、まず質問と回答候補の組に対して両者の回答タイプが一致しているかを判定する分類器を教師あり機械学習する。学習アルゴリズムはロジスティック回帰を採用する。そして、分類器の学習ツール LIBLINEAR が出力する確率値を回答タイプの整合度とする。機械学習のための素性には、質問文中に現れる疑問表現、疑問表現を含む 3-gram、文末の単語列、回答文中に現れる付属語列、節末の単語列、節末の単語列の組み合わせの 6 つを用いる。分類器を学習するための訓練データは、インターネット上の質問ポータルコミュニティサイトの Yahoo!知恵袋における 76,782 組の質問と回答の組から作成する。正例は、元の質問と回答の組から作成する。一方、負例は元の質問と違う

質問の回答を組にして作成する。このとき、回答を無造作に選ぶと回答タイプが同じ質問と回答の組を誤って負例としてしまう可能性があるため、回答タイプが元の質問と類似していない質問に対応する回答を選択する。質問間の類似度は、回答タイプの一致判定器を学習する際に用いた素性のうち、質問文から得られるもののみからなる素性ベクトルのコサイン類似度とする。

訓練データ中の正例と負例の比は、質問応答システムの文書検索モジュールによって実際に得られる回答候補集合における正例と負例の比に近いことが望ましい。そのため、Yahoo!知恵袋の質問・回答のデータ集合内で検索のシミュレーションを行い、最適な正例と負例の比を求める。まず質問から得られる回答タイプ一致判定のための素性からなるベクトルを用いて、質問と回答の組をクラスタリングする。次に、各質問に対し回答データ集合内で文書検索をして、検索上位 N_d 件の回答のうち、元の質問と同じクラスタに属する (≒同じ回答タイプである) 回答の割合を求める。N_d は文書検索モジュールによって取得する文書数と同じ値とする。この割合の全ての質問に対する平均を最適な正例と負例の比とみなす。実験の結果、正例と負例の比は 1:5.9 と推定された。

回答選択の際に「内容の関連度」と「回答タイプの整合度」から回答候補のスコアを決める手法として、「フィルタリング方式」と「加算方式」の 2 つを提案する。フィルタリング方式は、回答タイプ一致判定で不一致と判断された回答候補を取り除き、残された回答候補を内容の関連度でランキングする方法である。一方、加算方式は、内容の関連度と回答タイプの整合度から計算された各スコアを合計することで最終的なスコアを求める方法である。2 つのスコアの重み付き和を最終スコアとすることも考えられるが、重みの最適化は難しいことから、それぞれのスコアの回答候補集合における最大のスコアとの比 (相対スコア) を合計する。

提案手法の評価実験について述べる。まず、回答タイプ一致判定器の性能を評価した。前述の方法で作成した正例と負例の集合をテストデータ 10%と訓練データ 90%に分けて回答タイプの一致判定の正解率などを測った。正例と負例の比を 1:5.9 にしたとき、判定の正解率は 86.00%であった。また学習素性の集合を変えて分類器を学習し、素性の有効性を評価したところ、回答よりも質問の文から得られる素性の方が有効であることが分かった。

次に、質問応答システムの性能評価を行った。7 種類の質問を各 5 問、計 35 問の質問集を用意し、出力結果に対して、正答のランクの逆数の平均 (MRR)、上位 10 件の精度の平均 (AP')、出力上位 10 件の精度 (P_{top10}) を測って評価した。

本研究で提案する 2 つの回答選択手法は、「内容の関連度」だけで回答を選択するベースラインよりも評価値が高いことから、回答選択において回答タイプの一致を判定することは重要であることが分かった。また正例と負例の比を 1:5.9 にした訓練データから獲得された回答タイプ一致判定器を用いたときは、1:1 や 1:10 のときと比べて評価値が高かったことから、提案手法による正例と負例の比の決定方法の有効性が確認できた。2 つのスコアを組み合わせる手法を比較したところ、全体的には加算方式の方が結果がよく、MRR は 0.63、AP' は 0.53、P_{top10} は 0.27 となった。ただし、定義型やファクトイド型の質問に

対してはフィルタリング方式が優れていることが分かった。

目 次

第1章　序論

1.1　背景

質問応答とは、ユーザが自然言語で質問文を入力とし、文書集合から回答の情報を抽出する自然言語処理のタスクであり、このタスクの処理を行うシステムを質問応答システムと呼ぶ。図 1.1 に質問応答システムの概略を示す。

質問応答技術は、質問を理解し、適切な情報をユーザに回答するという点では自然言語理解とみなすことができ、質問という形式で表現されたユーザの情報要求に対して必要な情報を検索するという点では情報アクセス技術と位置づけられる。また、情報アクセスの対象となる知識源はウェブページのようなテキスト集合であり、そこから必要な情報を得るという点から見ればテキスト処理技術やウェブ応用技術と位置づけられる。

回答に用いられる知識源の分野や領域が限定されない質問応答はオープンドメイン質問応答と呼ばれる。オープンドメイン質問応答は、新聞記事やウェブページなどの巨大なテキスト集合から、回答となる単語、文または段落を取り出すことを目的としている。これは、情報検索におけるパッセージ (文書の一部分) 検索と位置づけられる。

質問応答の種類として次のようなものが挙げられる。

ファクトイド型質問応答

ファクトイド型の質問とは、回答が人名・地名・組織名・値段・日時・距離などの

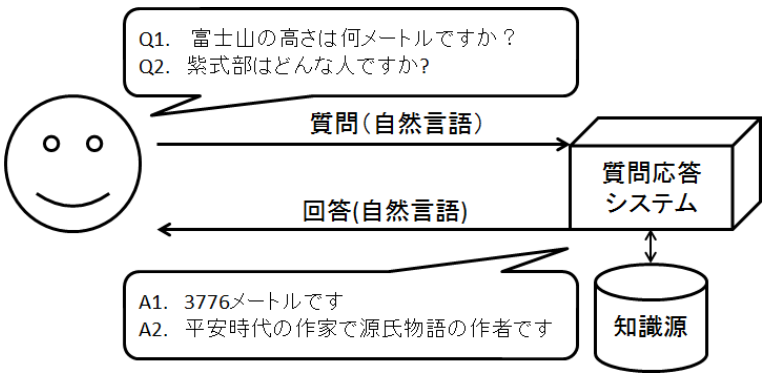

図 1.1: 質問応答システムの概略

短い固有名詞か名詞句で簡潔に表される質問である。

ファクトイド型質問の例

- ビッグマックは何カロリーありますか?
- 日本にメガネを伝えたのは誰?

ノンファクトイド型質問応答

ファクトイド型質問が人名や地名など短い語句で答えられるものを回答とするのに対し、ノンファクトイド型の質問は、定義や理由、方法など文や文章を回答とする質問である。

ノンファクトイド型質問の例

- 織田信長とはどんな人物ですか?
- どうして地震が起きるのですか?
- 蜂に刺された場合、どう対処しますか?

リスト型質問応答

リスト型質問は、複数の回答を要求する質問である。主にファクトイド型質問の変種として扱われる。

リスト型質問の例

- オリンピックの正式種目を８つ挙げよ
- 世界の大陸名を全て挙げよ

質問応答システムに関する初期の研究は、事実を表す固有名詞や短い文を回答とするファクトイド型の質問応答が主流であったが、近年は理由や方法など、文章で答える質問（ノンファクトイド型質問）など様々な質問に答えることのできる質問応答システムへの期待が高まっている。

適切な回答を出力するため、質問応答システムは質問文から利用者が求めている情報（人物、場所、理由など）を分類・特定する必要がある。このような分類を回答タイプ（または質問タイプ）と呼ぶ。以下に質問とそれに対する回答タイプの例を示す。括弧内が回答タイプである。

- NGN とはなんですか? (定義)
- 法隆寺はどこにありますか? (場所)
- 人力車を開発したのは誰? (人物)
- なぜインドとパキスタンは対立しているのか? (理由)
- エジソンはいつ生まれたのですか? (日時)

一般的な質問応答システムは、あらかじめ回答タイプの種類を定義し、与えられた質問文のタイプを判別し、知識源となる文書集合からタイプに合う単語や文を検索し、回答を返す。入力された質問の回答タイプを特定する際には、人手で定義されたパターン、もしくは機械学習によって得られた分類器が用いられる。しかし、あらかじめ定義されていない回答タイプの質問や、回答タイプが曖昧な質問に対しては、回答タイプを判定することが困難であるため、正しい回答を返すことが難しい。また、人間が問う質問は多種多様であり、ありとあらゆる質問の回答タイプをあらかじめ網羅的に分類し、定義しておくことも困難である。これらの問題点を解決する、回答タイプに依らない質問応答システムの開発が求められる。

1.2 目的

本研究では、明確な回答タイプの分類を行わない質問応答システムにおける回答選択手法を提案する。明確なタイプ分類を行わない質問応答システムでは、回答候補に対して、質問と回答候補の「内容の関連度」と「回答タイプの整合度」の 2 つの基準でスコア付けし、最良の回答を選択する。

内容の関連度は、質問と回答候補の内容やトピックの類似性を評価する。これは、例えば質問と回答に同じ自立語がどれくらい出現するかなどの指標で見積もることができる。一方、回答タイプの整合度は、質問と回答候補の回答タイプが一致しているかを評価する。例えば、理由を尋ねている質問に対し、何らかの理由を説明しているテキストは回答タイプが一致するが、言葉の定義を説明しているテキストは回答タイプが一致しないとする。ただし、回答タイプの存在を暗黙に仮定しているが、回答タイプはあらかじめ定義せず、単に回答タイプが一致しているかを評価する点に本研究の特徴がある。

本研究では、回答選択手法として、上記 2 つの基準によるスコアの組み合わせ方を提案する。具体的には、「フィルタリング方式」と「加算方式」の 2 つの手法を提案し、評価実験により比較する。回答タイプの整合度を算出するために、質問と回答候補の組が与えられた時、その回答タイプが一致しているか否かを判定する二値分類器を教師あり機械学習する。また、回答タイプの一致の判定に有効と考えられる素性を提案する。さらに、回答タイプ一致を判定する二値分類器を学習するための最適な訓練データの作成方法も検

討する。

1.3 論文の構成

本論文の構成は以下の通りである。2 章では質問応答システムに関する関連研究を紹介し、本研究との違いについて述べる。3 章では提案システムの詳細を述べる。特に回答タイプの一致を判定する分類器の学習、そのための訓練データの作成方法、「内容の関連度」と「回答タイプの整合度」の 2 つのスコアを組み合わせる方法について説明する。4 章では、実装した回答タイプの一致判定器と提案する質問応答システムの評価実験を行い、結果を考察する。最後に 5 章では、本研究のまとめと今後の課題について述べる。

第2章　関連研究

本章では本研究の関連研究について述べる。質問応答システムやその要素技術については様々な研究があるが、ここでは本研究と関連の深いものを以下の構成で紹介する。2.1節では、質問応答のための情報検索技術について紹介する。2.2節では、ファクトイド型質問応答システムにおける回答タイプの判定に関する先行研究を述べる。2.3節では、定義や理由についての質問に答えるノンファクトイド型質問応答について紹介する。2.4節では、明確なタイプ分類を行わない質問応答に関する研究を紹介する。最後に2.5節では、関連研究と本研究との違いを挙げ、本研究の特徴を述べる。

2.1　質問応答のための情報検索

一般に質問に対する回答は、質問内のキーワードを含む文の近くに出現する。したがって、キーワード検索により、入力された質問に関連する文書を取得することは、質問応答システムにおける重要な要素技術である。そのため、質問応答における文書検索について様々な手法が提案されている。

金井らは、ファクトイド型質問応答において、複数の異なるウェブ検索エンジンを組み合わせ、回答抽出の情報源に多様性を持たせる手法を提案している [4]。文書検索時に、複数の検索エンジンから文書集合を獲得し、検索エンジン別に回答候補を抽出・評価し、最終的に回答スコアを併合することで精度向上を図っている。評価実験より単独の検索エンジンを使ったシステムより、複数の検索エンジンを組み合わせたシステムの方が精度がよくなる傾向にあることが示された。

また、最初の検索で得られた回答候補をクエリに加えて再検索を行い、より関連の高い文書を見つける疑似関連フィードバックに関する研究もいくつか行われている。Prager らは、問合せ反転 (Question Inversion) という手法を提案している [9]。問合せ反転とは、ある質問に対して、回答候補が得られた時、回答候補から質問を問う質問を作り、再度検索を行う手法である。例えば、「カナダの首都は?」という質問に対して、「ワシントン DC」、「オタワ」といった回答候補が得られた時、「ワシントン DC はどこの国の首都?」、「オタワはどこの国の首都?」という質問を自動生成して、再検索して、つじつまの合う回答候補の組を選ぶ。実験では、問合せ反転を行うシステムの精度の向上が見られた。しかし、問合せ反転のためのパターンを回答タイプごとに用意しなければならないという問題点がある。また質問と回答の関係が唯一でないと問合せ反転は有効に働かないという問題点もある。例えば、「A さんの死因は?」という質問に対し、「交通事故」という回答が得

られた場合、問合せ反転を行うと「交通事故で死んだのは誰?」という質問が生成されるが、この質問に対する回答の数が多くなり、回答として「A さん」を見つけることが困難である。

2.2 回答タイプの判定

人物や場所など固有表現を答えるファクトイド型質問応答では、入力された質問の回答タイプを特定し、固有表現抽出により得られる回答候補の中から回答タイプが合致する回答を抽出する。回答タイプの自動分類については、人手によるルールや機械学習による手法が研究されている。

佐々木らは、機械学習技術を用いて、質問や回答のタイプを分類するファクトイド型質問応答システム SAIQA-II を提案している [10]。人手により経験的に作成されたルールやパターン、評価関数による回答選択では、未知の質問に対する正解率が高くなることを予測しながら改良することが困難である。そこで、このシステムでは、8 つの固有表現 (人名、地名、組織名、製品・タイトル名、日付、時間、金額、割合) を回答タイプと定義し、サポートベクターマシン（Support Vector Machine;SVM）を用いて機械学習を行い、質問解析・回答評価時に分類器を用いてタイプ判定を行う。SVM は二値分類器であるため、8 種類のタイプのそれぞれについて SVM を用意することで、多値のタイプ分類を実現している。訓練データとする質問文は、人手によって回答タイプが付与されており、質問解析用のタイプ分類 SVM は、それぞれ対応するタイプの質問から作成した素性ベクトルは正例、それ以外の質問から作成したものは負例として学習する。正解の回答タイプが付与された回答の集合から、回答評価のための SVM も 8 種類のタイプそれぞれについて学習する。質問と回答 (固有表現) の回答タイプが一致しているかをチェックし、一致した固有表現のみを回答として抽出する。実験では、質問文のタイプ分類の精度は 88%であった。質問応答システムの評価では、上位 5 位以内に正解が含まれる確率 (Top5 スコア) は 50%以上であったが、定義していない回答タイプの質問に対する Top5 スコアは 6.1%であった。

2.3 ノンファクトイド型質問応答

ファクトイド型質問応答に対して、ノンファクトイド型質問応答は、理由 (Why)、定義 (Definition)、方法 (How) など、文章による回答を問う質問を扱う。

諸岡らは、事実以外のことを問う質問や回答のタイプを判別するルールを人手で作成したノンファクトイド型質問応答システムを提案している [8]。表層パターンを用いて質問文のタイプ (Why 型、How 型、定義型) を判定し、回答抽出の際には、品詞と付属語によるパターンや、回答タイプの特定につながる名詞（手がかり語）を手がかりとすることで、ノンファクトイド型質問応答を実現している。しかし、人手により質問のタイプ判定のためのパターンや回答抽出のパターンを作成するためのコストが大きいこと、また定義されたパターンにマッチしない質問に対して正確に答えられない、といった問題点がある。

渋沢らは、Why 型 (理由型) 質問応答について、理由を表す文と事実にあたる文の文書中の位置関係に着目して回答文を抽出する手法を提案している [13]。まず、文書中に現れる事実を表す文を事実文、理由・原因を表す文を理由文と呼び、これらがどのような位置に現れるかを 4 つのケースに分類する。そして、理由文を特定する手がかりとなる単語やそのパターンに人手で重みを与え、回答をスコア付けすることで Why 型質問応答を実現している。回答位置特定の精度は 31.3%、再現率は 50.4%であったと報告している。しかし、パターンやその重みは人手で経験的に決定するため、調節が難しいと考えられる。

佐々木らは、How 型質問応答システムに Why 型質問応答システムを組み合わせることにより、回答の根拠も含めて回答評価を行う質問応答システムを実現している [12]。始めに、方法・行動を問う質問に対して、質問解析、文書検索、回答抽出などの処理を経て、回答集合を得る。得られた回答に対してその理由を問う質問を生成し、質問応答を行う。これら 2 回の質問応答の結果のスコアの積を最終的なスコアとし、根拠の伴う回答を選択することを実現している。ベースラインとする既存の質問応答手法との精度、再現率、F 値の比較では、提案手法の評価値が上回ることを確認した。

Han らは、質問対象のトピックと回答の記述表現の 2 つの観点から回答選択を行う定義型質問応答システムを提案している [2]。このシステムでは、一般的な言語モデル、質問のトピックに関する言語モデル、定義の記述に関する言語モデルを用意し、それらをもとに回答候補を評価することで、定義型質問応答を実現している。評価実験により、TREC2004 で高い性能を示したシステムに並ぶ性能であることが確認されている。

また、質問対象について複数の事柄を問う定義型質問や、複数の回答を求めるリスト型質問応答においては、回答候補の出現頻度や質問との関連度だけでなく、出力される回答集合にどれだけ多様な情報が含まれているかも重要となる。Achananuparp らは、回答候補集合に対して、互いの類似度に基づいてグラフを作り、回答を評価する DiverseRank モデルを提案している [1]。このモデルは、各回答候補を頂点とし、他の候補との類似度を計算し、それが閾値を超えていたら辺を与え、グラフを構築する。他のノードとつながっている辺が多い回答は頻出する回答となり、逆につながっている辺が少ないものは珍しい回答となる。つながる辺が多い回答を減点し、珍しい回答候補により高いスコアを与えることで、システムが出力する回答集合に多様な情報が含まれるようにする手法である。定義型質問について実験を行ったところ、回答候補を選択する際に質問のトピックとの関連度を重視する手法よりも若干の精度の向上が確認されている。

2.4 明確なタイプ分類を行わない質問応答

明確な回答タイプの分類を行わない質問応答は、質問と回答候補の「内容の関連度」と「回答タイプの整合度」の 2 つの基準で適切な回答を探す。また、システムの学習データとして大量に取得できるインターネット上の Q&A サイトのデータを利用する場合が多い。

佐々木は、質問応答を「質問文によりバイアスされた用語抽出問題」と捉えて、質問文と文書の特徴から回答となる用語または名詞句を抽出するファクトイド型質問応答手法を

提案している [11]。この手法では、回答候補を固有表現クラスに分類するのではなく、質問文に対応する回答であるかという観点で回答候補を分類する。あらかじめ固有表現のクラスを定義しないため、様々な固有名詞を問う質問に対応できるという意味で頑健である。評価実験では、回答タイプの分類に基づくファクトイド型質問応答システム SAIQA-II[10] に匹敵する精度を示した。

水野らは、Q&A サイトの質問・回答事例を学習データとし、質問と回答のタイプが一致しているかを判定する SVM を学習する手法を提案している [6][7]。この研究では、人手による回答タイプ分類を用いず、質問と回答の回答タイプが一致するかを単に判定することで、特定のタイプに限定することのない質問応答を実現している。この回答タイプ一致判定器は、SVM による二値分類器で実装されており、インターネット上の Q&A サイトから質問と回答のペアを収集して、学習データを作成している。機械学習の素性として、助詞、助動詞、感動詞、疑問表現、文末表現を用い、これらの素性を質問と回答から抽出して素性ベクトルを作成する。学習データを作成する際には、正例は質問と正解のペアとし、負例は質問とそれをクエリとして回答集合に対して文書検索したときのランクが 1 位ではあるが正解ではなかった回答とのペアとする。また、正例と負例の比がほぼ 1 : 1 となるように学習データを作成している。評価実験では、回答タイプの一致をチェックせずに、文書検索のみで回答を出力するベースライン比較してよい性能を示した。また、水野らは、回答タイプをあらかじめ定義して質問と回答のタイプを個別に判定し、両者の一致をチェックする手法も実装しているが、回答に対するタイプ分類は 1 割程度であり、質問と回答の回答タイプが一致するかを直接判定する手法の方が有効であると述べている。

石下らは、Q&A サイトのデータ集合から回答の記述スタイルを表す特徴的な単語 n-gram を動的に抽出し、「回答タイプの整合度」のスコアを算出する手法を提案している [3]。まず、質問文が入力された時に、質問文中の「疑問詞を中心とする単語 7-gram」の一致の度合いを類似度とし、入力された質問文と類似度が高い質問を Q&A データ集合から上位 N 件取得する。次に、取得した質問集合に対応する回答をデータ集合から取り出し、回答の特徴表現として、語のスキップ 2-gram（数語の間隔で隣り合う語の 2-gram）を取り出し、重みづけ関数を用いてその重みを決定し、回答の特徴表現とその重みのペアの集合を作成する。これを基に回答候補の「回答タイプの整合度」のスコアを算出し、このスコアと「内容の関連度」のスコアの重み積によって回答候補の順位付けを行うことで、明確な回答タイプ分類を行わない質問応答を実現している。評価実験では、内容の関連度のみのベースラインと比較して、提案手法の MRR の向上が見られた。しかし、特徴的な単語 n-gram を動的に求めるため、回答候補選択に要する計算時間が大きいという問題点がある。また 2 つのスコアの重みを変えた実験結果を示しているが、重みの最適化までは行っていない。

Soricut ら [14] や Komiya ら [5] は、統計的機械翻訳（Statistical Machine Translation;SMT）の技術を質問応答に利用することを提案している。SMT は、大量の対訳コーパスから翻訳器を学習する機械翻訳の技術である。対訳コーパスとは、同じ内容を表す 2 つの言語の文のペア（たとえば日本語と英語）を収集したものである。対訳コーパスから、翻訳前の

言語（源言語）から翻訳後の言語（目標言語）へ翻訳される確率を推定するモデルを学習する。SMT を応用し、質問を源言語、回答を目標言語に見立てることで、質問文から回答文へ翻訳する形で質問応答を実現する。対訳コーパスとなる質問・回答のペアは Q&A サイトから収集し作成する。この手法も回答タイプをあらかじめ定義していないが、「内容の関連度」と「タイプの整合度」の両方を同時に学習するため、大量の訓練データを必要とするという問題点がある。

2.5 関連研究と本研究の違い

本研究の提案システムは、2.2 節や 2.3 節で挙げた手法とは異なり、明確な回答タイプの定義を行わずに回答選択を行うという点に特徴がある。

本研究では、水野らの研究と同様に回答タイプを事前に定義せず、暗黙の回答タイプの一致を判定する二値分類器により質問応答システムを実現することを目的としている。しかし、水野らの手法では、回答タイプの一致を判定する分類器の学習データにおける正例と負例の割合を単に 1:1 に設定しているのに対し、本研究では正例と負例の比の最適化を試みている。さらに、負例となる質問と回答の組を選択する際に、回答タイプが似ていない事例を選ぶことによって、回答タイプが一致する質問と回答の組を誤って負例としない工夫を施す。

また本研究では、水野らや石下らのシステムのように「内容の関連度」と「回答タイプの整合度」の 2 つのスコアから最終的な回答スコアを計算する方法だけでなく、回答タイプの一致判定でタイプが一致しないと判定された回答候補を取り除く方法も実装し、2 つの回答選択手法を比較する。

第3章　提案システム

本章では本研究で提案する質問応答システムについて詳述する。

3.1　概要

質問応答システムは、基本的に質問解析、文書検索、回答抽出の3つのモジュールから構成される。

質問解析
ユーザが入力した質問文を解析し、キーワードを抽出して文書検索のためのクエリに変換する。また、ユーザがどのような回答を求めているか、すなわち質問の回答タイプを推定するための素性を抽出する。多くの質問応答システムでは、このモジュールで回答タイプを推定する。

文書検索
質問解析で得られたクエリに基づいて文書集合から回答を含む可能性のある文書を検索する。また、得られた文書を文や段落に区切り、回答候補の集合を得る。

回答抽出
得られた回答候補が、質問の回答としてどれくらい適切かを様々な情報に基づいて推測し、適切と考えられる回答を出力する。

本研究のシステムもこの3つのモジュールから構成される。提案システムの概要を図3.1に示す。本システムでは、知識源とする文書集合はウェブとする。回答評価モジュールでは、「内容の関連度」と「回答タイプの整合度」の2つの観点で回答候補を評価する。「内容の関連度」は、質問と回答候補の内容やトピックがどの程度類似しているかの評価である。一方「回答タイプの整合度」は、質問と回答候補の回答タイプが一致しているかの評価である。ただし、回答タイプはあらかじめ定義せず、単に質問と回答のタイプが一致するかを判定する。また、これら2つの観点によるスコアの組み合わせ方として、2つの手法を提案する。

提案システムはプログラミング言語Pythonを用いて実装した。Pythonを用いた理由は、HTML・XMLパーサのライブラリが存在し、ウェブ文書を比較的容易に処理できるためである。以降の節では、質問解析、文書検索、回答抽出の各モジュールについて説明する。

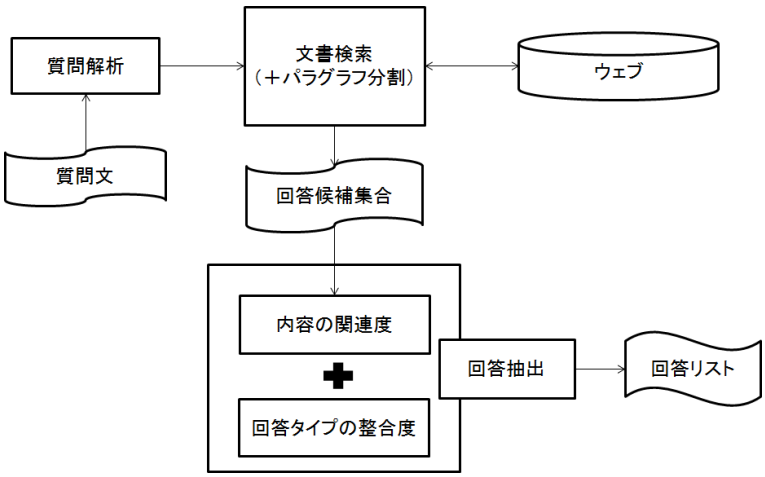

図 3.1: 本研究における質問応答システムの概要

```
人 ヒト ヒト 人 名詞-普通名詞-一般
の ノ ノ の 助詞-格助詞
骨 ホネ ホネ 骨 名詞-普通名詞-一般
は ワ ハ は 助詞-係助詞
何 ナン ナン 何 名詞-数詞
本 ポン ホン 本 接尾辞-名詞的-助数詞
あり アリ アル 有る 動詞-非自立可能 五段-ラ行 連用形-一般
ます マス マス ます 助動詞 助動詞-マス 終止形-一般
か カ カ か 助詞-終助詞
? ? 補助記号-句点
EOS
```

図 3.2: 例文「人の骨は何本ありますか?」の形態素解析結果

3.2 質問解析

質問解析モジュールでは、ユーザが入力した質問文を形態素解析し、単語 (形態素) に分割する。本システムでは、形態素解析ツールに MeCab[1]、形態素解析のための辞書は UniDic[2] を用いた。MeCab は他の形態素解析用の辞書も用いることができるが、以後、MeCab による形態素解析の時に用いる辞書は常に UniDic とする。図 3.2 に Mecab を用いた解析例を示す。1 行に 1 つの形態素の情報が記述され、各形態素は、表層形、発音、読み、原形、品詞の組で表される。また、句読点などの記号については、読みや発音は省略される。

形態素解析後、質問文から検索クエリとするキーワードを取り出す。基本的には、質問

[1]http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html
[2]http://sourceforge.jp/projects/unidic/

何，何故，何時，誰，何処，何所，どっち，いくら，いくつ，どう，どの，どれ，どんな，どなた

図 3.3: 疑問詞のリスト

質問文: iPS 細胞とは何ですか?
キーワード: iPS 細胞

質問文: 人の骨は何本ありますか?
キーワード: 人, 骨, 本

質問文: TPP についてどう思われますか?
キーワード: TPP, つく, 思う

図 3.4: キーワード抽出の例

文中の自立語をキーワードとする。助詞や助動詞、また品詞の第 1 階層が「動詞」でも第 2 階層が「非自立可能」である語は付属語として扱い、それ以外の語を自立語として抽出する。このとき、自立語の原形をキーワードとして抽出する。ただし、自立語であっても疑問詞はキーワードとして抽出しない。本研究では、図 3.3 に挙げた語を疑問詞とした。

MeCab による形態素解析では、英数字で表される単語はアルファベット 1 文字ごとに区切られてしまうので、英字、数字が連続して出現するときは、まとめて 1 単語として扱う。さらに英数字の後に自立語が続く場合は全体で複合語とみなして連結する。図 3.4 に質問文とそれから抽出されるキーワードの例を示す。

上述の操作で取り出したキーワードを検索クエリとし、文書検索モジュールに渡す。

3.3 文書検索・パラグラフ分割

文書検索モジュールでは、検索エンジンを用いて回答を含む文書集合を取得する。提案システムでは、文書をパラグラフ単位に分割し、回答候補集合を作成する操作もこのモジュールで行う。

まず、質問解析により取得した検索クエリを使い、ウェブ検索を行う。本システムでは、Bing のウェブ検索 API[3] を利用した。この API では、検索クエリに加えて検索件数を指定できる。本論文では、検索する文書の数を N_d と記す。

次に、各文書からテキストを取り出し、パラグラフ単位に分割する。まず、取得したウェブページの HTML ソースを取り出し、HTML の構造を解析し、<body>タグで囲まれる HTML ソースに対して以下の操作を行う。HTML の構造解析には Beautiful Soup[4] ラ

[3]http://datamarket.azure.com/dataset/bing/search
[4]http://www.crummy.com/software/BeautifulSoup/

イブラリを用いた。

テキスト以外の情報の除去

次のタグおよびタグで囲まれた部分は、スクリプトコードや画像情報などテキスト以外の情報であるので、取り除く。

```
<a>, <img>, <sub>, <b>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>,
<script>, <style>,<font>,<!-- … -->
```

パラグラフ境界の検出

ウェブ文書をパラグラフ単位に分割する前準備のため、以下のタグは#split#という記号に書き換える。

```
<div>, <p>, <dd>, <br><br>
```

これらのタグは、しばしばパラグラフをマークアップするときに用いられるので、パラグラフに分割するときの境界とする。また
タグは、1つのみだと改行を表すが、2つ以上連続して用いられる場合は、段落を変えて新しい話題を記述する傾向があるので、このときはパラグラフの区切りとした。

リスト・テーブルの連結

特に理由や方法など文章を回答する場合、リストや表は全体で回答を表すことが多い。したがって、リストや表は連結して1つのパラグラフとする。ただしタグで区切られているところは異なる文と扱うため、#kutouten#という記号に置き換える。

```
<li>,<td>
```

上記のタグの書き換え・除外処理が終わった後、HTML ソース中のタグを全て取り除いた文字列を得る。次に、#kutouten#を句点“。”に置き換え、#split#で文字列を分割する。分割によって得られたテキストの断片のうち、空列を除いたものをパラグラフとする。

本研究では上述の処理で得られたパラグラフを回答候補とする。回答候補の集合を次の回答抽出モジュールに渡す。

3.4 回答抽出

回答抽出モジュールでは、各回答候補のスコアを計算し、より適した回答を選択する。本システムでは、質問と回答候補の「内容の関連度」と「回答タイプの整合度」をそれぞれ算出し、この2つを組み合わせて回答候補のスコアとし、回答候補をランキングする。また2つのスコアの組み合わせ手法として「フィルタリング方式」と「加算方式」を提案する。

3.4.1 内容の関連度

質問と回答候補の関連度は、これらの内容がどれだけ類似しているかの度合いである。本論文ではこれを $Score_r$ と記す。$Score_r$ は、基本的には質問と回答候補からそれぞれ自立語の単語ベクトル $\vec{q}, \vec{a}$ を作り、それらのコサイン類似度 $sim_{cos}(\vec{q}, \vec{a})$ で算出する。自立語の抽出は、質問解析モジュールにおけるキーワード抽出と同様にして行う。

しかし、単純な単語ベクトルによるコサイン類似度では、質問と無関係な回答候補に高いスコアを与えてしまうことがある。そのため、質問とより関連の高い回答候補との関連度が高くなるように、以下の処理を行う。

質問中の重要な単語に高い重みを与える

質問文の中でも、主語などの特に重要な語 (重要語) は回答にも含まれている可能性が高い。そこで、単語ベクトル作成時に、重要語の重みを大きくすることで、その語が含まれている回答候補により高いスコアを与えるようにする。具体的には、質問中の助詞「が」、「は」、「の」の前にある語を重要語として扱い、その単語ベクトルにおける重みを2、それ以外の語の重みを1として単語ベクトルを作成する。重要語の抽出には文節の係り受け解析器 CaboCha[5] を用い、助詞「が」、「は」、「の」を含む文節内の語を重要語とする。また重要語と並列の助詞「と」を介して係り受け関係にある語も重要語とする。以下に重要語抽出の例を示す。

質問文: 人の骨は何本ありますか?
重要語: 人, 骨
単語ベクトルの重み: (人:2), (骨:2), (本:1)

文書中の前後のパラグラフを考慮

一般に、質問に対する回答は、質問文中のキーワードの近くに出現する。また、質問文中のキーワードは回答の中に出現するだけでなく、回答の周辺に現れることもある。回答

[5] https://code.google.com/p/cabocha/

候補に含まれる自立語のみから単語ベクトル $\vec{a}$ を作成すると、回答候補の周辺に質問文中のキーワードが出現するか否かを内容の関連度 $Score_r$ に反映させることはできない。質問のキーワードの周辺に位置する回答候補は質問との関連度が高いと考え、この情報を関連度計算に組み込むこととする。具体的には、文書中の回答候補の前後のパラグラフに含まれる語を重み 0.5 として単語ベクトル $\vec{a}$ に加える。

ウェブ検索結果のランキングを考慮

ウェブ検索 API は、検索クエリと関連する文書をランキングする。このとき、上位の文書に含まれる回答候補は質問との関連が高いと考えることができるため、これを関連度のスコアに反映させる。具体的には、ウェブ検索時のランクが高いほどスコアが高くなるようにコサイン類似度 $sim_{cos}(\vec{q}, \vec{a})$ を補正する。

質問と回答候補の内容の関連度 $Score_r$ の定義を式 (3.1) に示す。r は回答候補が含まれる文書のウェブ検索時のランク、N_d は検索で取得した文書数である。$\vec{q}$ は重要語に高い重みを加えた単語ベクトル、$\vec{a}$ は前後のパラグラフに出現する単語を加えた単語ベクトルである。ともにベクトルの大きさが 1 になるように正規化してからコサイン類似度を計算する。

$$Score_r = sim_{cos}(\vec{q}, \vec{a}) \times (1 - \frac{r}{N_d}) \tag{3.1}$$

3.4.2 回答タイプの整合度

回答タイプの整合度とは、質問の回答タイプと回答候補の回答タイプがどの程度一致するかを表す指標である。本論文ではこれを $Score_t$ と記す。回答タイプをあらかじめ定義する質問応答システムでは、質問と回答候補のタイプ判定を個別に行い、それぞれのタイプが一致するかを判定する。一方本システムでは、回答タイプの存在は仮定しているが、回答タイプの集合をあらかじめ明確に定義しない。質問・回答の文の記述形式や表現を手がかりに暗黙的な回答タイプが一致するかを判定する。

質問と回答候補間の回答タイプの一致判定には、機械学習による二値分類器を用いる。機械学習ツールとして LIBLINEAR [6]、学習アルゴリズムとして L2 正則化ロジスティック回帰を用いる。LIBLINEAR を用いたのは、大量の学習データを用いたときでも比較的速く実行できるためである。また L2 正則化ロジスティック回帰を用いたのは、LIBLINEAR で L2 正則化ロジスティック回帰を用いた場合、判定だけでなくテスト事例が正例である確率、負例である確率も出力できるため、これを回答タイプの整合度のスコアとして利用できると考えたからである。ここでの正例とは、質問と回答候補の回答タイプが一致している事例、負例とは一致していない事例を指す。回答タイプの整合度 $Score_t$ の定義を式

[6] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

```
1    1:1 39063:1 41241:1 67309:1 67317:1 67401:1 67522:1 67662:1
-1   1:1 39063:1 41129:1 41805:1 67310:1 67452:1 67677:1 67904:1
1    2:1 8:1 18:1 23:1 41130:1 67604:1
-1    2:1 8:1 18:1 23:1 42002:1 67307:1 72936:1
1    2:1 230:1 880:1 1819:1 13698:1 41130:1 44918:1 67626:1
-1    2:1 230:1 880:1 1819:1 13698:1 41129:1 41137:1 41177:1
1    1:1 15844:1 41159:1 42317:1 56390:1 67338:1 67482:1 67560:1
-1   1:1 15844:1 41129:1 41141:1 66686:1 67348:1 67359:1 85886:1
 :
 :
```

図 3.5: 訓練・テストデータの例

(3.2) に示す。

$$Score_t = (\text{質問と回答候補の回答タイプが一致する確率}) \tag{3.2}$$

すなわち $Score_t$ は L2 正則化ロジスティック回帰により算出される正例である確率とする。

LIBLINEAR に用いる訓練・テストデータは、1 行に 1 つの事例 (質問と回答候補の組) を「クラスラベル　素性番号:重み 素性番号:重み …」という書式で記述する。「クラスラベル」は正例のときは 1、負例のときは −1 とし、「素性番号」は学習素性の ID、「重み」はその素性の重みである。本研究の回答タイプ一致の判定では、素性の重みは常に 1 とする。図 3.5 に LIBLINEAR のフォーマットで記述されたデータファイルの例を示す。

回答タイプの一致判定器を学習するためには、訓練データとなる大量の質問と回答のデータが必要となる。インターネット上の質問ポータルコミュニティサイト (Q&A サイト) には、投稿された多くの質問とその回答が蓄積されているので、これを訓練データ作成に利用する。本研究では、回答タイプ一致判定の分類器を学習するための訓練データは、Q&A サイト Yahoo!知恵袋 [7] の質問と回答の組から作成する。本研究で使用する Yahoo!知恵袋のデータは、投稿された質問とそれに対する回答の中で最も良いと質問者が判断した回答 1 つ (ベストアンサー) を収録したものであり、91,404 個の質問と回答の組から構成される。質問の内容に応じて 15 のカテゴリ (趣味，コンピュータなど) が設定され、OC01 〜OC15 という名称で分類されている。このうちカテゴリ OC07 に該当する質問はデータには含まれていないため、カテゴリの数は 14 である。

3.4.2.1　回答タイプの素性

回答タイプの一致の判定器を学習するための素性は、質問と回答候補の記述の特徴・スタイルとする。素性抽出のための前処理として、質問と回答候補を MeCab によって形態素解析する。以下、機械学習に用いる素性を説明する。素性の種類は 6 種類あり、$\mathbf{f}_{\mathbf{x}}^{\mathbf{q}}$ もし

[7] http://chiebukuro.yahoo.co.jp/

(a) 疑問を表す語

ナニ, ダレ, ドコ, ドンナ, ナン, イツ, ドッチ, イクラ, イクツ, ナゼ, ドウ, ドウシテ, ドノ, ドレ

(b) 回答タイプの判定に有効と考えられる語

リユウ, ユライ, シカタ, テイギ, ジョウケン, イミ, ホウホウ, ゲンイン

図 3.6: 疑問表現のリスト

くは $\mathbf{f}^{\mathbf{a}}_{\mathbf{x}}$ と表記する。$\mathbf{f}^{\mathbf{q}}_{\mathbf{x}}$ は質問から抽出される素性、$\mathbf{f}^{\mathbf{a}}_{\mathbf{x}}$ は回答候補から抽出される素性を表す。

$\mathbf{f}^{\mathbf{q}}_{\mathbf{in}}$：質問に現れる疑問表現

疑問表現とは、ここでは「何」、「誰」などの疑問を表す語と定義する。本研究では疑問表現のリストを独自に設定する。文書中に現れる疑問表現は表記に揺れがあるため、疑問表現の発音表記のリストを用意し、文中の語の発音がリスト内のいずれかと一致した時その語を疑問表現の素性として抽出する。疑問表現のリストを図 3.6(a) に示す。また、疑問を表しているわけではないが、回答タイプの判定に有効と考えられる語も疑問表現と同じように扱う。具体的には図 3.6(b) に示す語を疑問表現として扱う。

また、これらの疑問表現が含まれていない質問には、疑問表現なしを表す素性として `wh_no` を抽出する。表 3.1 に質問とそれから取り出される $\mathbf{f}^{\mathbf{q}}_{\mathbf{in}}$ の例を示す。疑問表現にあたる語は【】で囲んで表す。

$\mathbf{f}^{\mathbf{q}}_{\mathbf{in3}}$：質問文に現れる疑問表現を含む 3-gram

疑問表現を含む単語の 3-gram を質問文から取り出す。ただし、疑問表現以外の自立語は品詞に置き換える。自立語の品詞は UniDic における品詞体系の最上位の分類 (名詞、動詞など) とする。また、付属語はひらがなで表記する。表 3.1 に質問とそれから取り出される $\mathbf{f}^{\mathbf{q}}_{\mathbf{in3}}$ の例を示す。

分類器の訓練データとする Yahoo!知恵袋における質問の中には、「誰か知っていますか?」というような回答タイプに関係なく問いかけを行う表現がいくつか出現した。このような表現から抽出される素性は、回答タイプの一致判定の妨げになると判断し、除去する。具体的には、3-gram の中に「【誰】_か」を含む素性は抽出しない。

表 3.1: 素性 $\mathbf{f^q_{in}}$ と $\mathbf{f^q_{in3}}$ の例

質問	抽出される素性
iPS 細胞とは何ですか?	【何】, と_は_【何】, は_【何】_です, 【何】_です_か
寝癖を直すにはどうすればいいですか?	【どう】, に_は_【どう】, は_【どう】_すれ, 【どう】_すれ_ば
日本で一番高い山はどこですか?	【どこ】, < 名詞 >_は_【どこ】, は_【どこ】_です, 【どこ】_です_か
宇宙人は存在しますか?	wh_no

方法, 原因, 理由, 対処, 意味, 情報, 内容, 語源, 必要, 言葉, 大丈夫, 本当, 可能, 如何, 変, 普通, 好き, 嫌い, せい, 無理, 問題, 失礼, 方, 違い, 名前, 場所, アドバイス, コツ, レシピ, 仕方, 誰, どれ, 何方, 如何, 何故, 思う, 言う, 読む, 書く, 就く

図 3.7: 回答タイプ指示語のリスト

$\mathbf{f^q_{end}}$: 質問の文末表現

文末表現とは、ここでは質問文の末尾に現れる「自立語」+「付属語の列」と定義する。正確には、$\mathbf{f^q_{end}}$ は「(自立語の品詞) +付属語の列」もしくは、「(回答タイプ指示語) + 付属語の列」で表す。自立語の品詞は UniDic における品詞体系の最上位の分類である。付属語は発音 (カタカナ) で表記する。また、回答タイプ指示語とは、「方法」「原因」「理由」「対処」など、回答タイプを特定する手がかりとなると考えられるキーワードである。本研究では回答タイプ指示語のリストを人手で定義した。これを図 3.7 に示す。

また、MeCab による形態素解析では名詞や動詞として分類される語でも、形式名詞や形式動詞のようにそれ自身は明確な意味を持たない単語は付属語として扱う。本研究では図 3.8 に示すリスト中の語は例外的に付属語として扱う。$\mathbf{f^q_{in3}}$ と同様に、回答タイプに関係なく問いかけを行う表現から素性を抽出することを避けるため、「どなたカ」「方イマス (カ)」「イラッシャイマス (カ)」を含む素性は抽出しない。

また、「教えてください」や「〜が分かりません」といった文末表現は、動詞を含む表現であるが、これらは質問でよく使われる表現であり、回答タイプ判定の手がかりとなる語はこれらの動詞の前に出現することが多い。そこで「教えてください」のような表現を『END(教えて)』、「分かりません」のような表現を『END(分かりません)』という特殊な記号に置き換える。それぞれは以下の正規表現で定義される。

事, 物, 者, 人, どう, 如何, 何, いい, よい, 悪い, よろしい, 正しい, 無い, 存知, 易い, 気, 駄目, 所, 知り

図 3.8: 例外的に付属語として扱う語

- END(教えて)
  (教えて|教授)(付属語)+ $

- END(分かりません)
  (分かり|解かり|判かり|分り|解り|判り|思い出せ)(付属語)+ $

$は文末を表す記号である。これらは付属語として扱い、これらの前に出現する自立語もしくは回答タイプ指示語までを連結したものを素性とする。

表 3.2 に質問とそれから抽出される $\mathbf{f^q_{end}}$ の素性の例を示す。

表 3.2: 素性 $\mathbf{f^q_{end}}$ の例

質問	抽出される素性
どの動物が一番速く走りますか?	< 動詞 > マスカ
北陸新幹線についてどう思いますか?	思いマスカ
エジソンについて教えてください	ついテ END(教えて)
京都駅までの道が分かりません	< 名詞 > ガ END(分かりません)

$\mathbf{f^a_{cl}}$: 回答の節末表現

節末表現は、ここでは回答候補の文を節で区切ったとき、その末尾に現れる表現と定義する。節末表現は文末に出現する表現も含むことに注意していただきたい。節の境界は、「(用言) +接続助詞」もしくは「(サ変可能名詞) +接続助詞」の直後とする。$\mathbf{f^a_{cl}}$ は、「(自立語の品詞)+付属語の列」もしくは「(動詞)+付属語の列」で表す。付属語は発音 (カタカナ) で表記する。また、$\mathbf{f^q_{end}}$ と同様に図 3.8 に示す語は付属語として扱う。表 3.3 に回答候補の文とそれから取り出される $\mathbf{f^a_{cl}}$ の例を示す。/ は節の境界を表す。

表 3.3: 素性 $\mathbf{f^a_{cl}}$ の例

回答	抽出される素性
エジソンは電球を開発しました。	< 名詞 > シマシタ
果物は多年生植物ですが、/ 野菜は一年生植物です。	< 名詞 > デスガ, < 名詞 > デス
この層のことをオゾン層と呼びます。	呼びマス

$\mathbf{f^a_{cl\text{-}all}}$：回答の節末表現の組み合わせ

回答候補の文から節末表現を抽出し、それを文中での出現順に '_' で連結したものを素性とする。1 文中に節末表現の素性が 2 つ以上取り出せたときのみ抽出する。表 3.4 に回答候補の文とそれから取り出される $\mathbf{f^a_{cl\text{-}all}}$ 素性の例を示す。

表 3.4: 素性 $\mathbf{f^a_{cl\text{-}all}}$ の例

回答	抽出される素性
まず早寝して、/ 生活のリズムを整えるのが大切です。	＜名詞＞シテ_＜名詞＞デス
果物は多年生植物ですが、/ 野菜は一年生植物です。	＜名詞＞デスガ_＜名詞＞デス
夏は、活動が活発ですが、/ 冬はあまり見かけないと思います。	＜形状詞＞デスガ_思いマス

$\mathbf{f^a_{func}}$：回答中の付属語列

回答候補文中に現れる付属語の 1 単語以上の列を素性とする。付属語は発音 (カタカナ) で表記する。表 3.5 に回答候補の文とそれから取り出される $\mathbf{f^a_{func}}$ 素性の例を示す。

表 3.5: 素性 $\mathbf{f^a_{func}}$ の例

回答	抽出される素性
彼は、1893 年に生まれました。	ワ, ニ, マシ_タ
寝癖を防ぐには、髪をよく乾かす必要があります。	オ, ニ_ワ, ガ_アリ_マス
富士山の標高は 3,776 メートルです。	ノ, ワ, デス

質問と回答候補の組から上述の 6 種類の素性を抽出し、素性ベクトルを作成する。図 3.9 に素性抽出の例を示す。

分類器の訓練データとする Yahoo!知恵袋では、質問も回答も複数の文で記述される。このデータ集合の中には、質問の中に疑問文ではない文が、一方回答の中に疑問文が出現するものも含まれている。しかしながら、質問応答システムへの質問として平叙文が入力されることは稀である。また、回答の中の疑問文から取り出される素性は回答タイプ一致の判定に悪影響を与えると考えられる。そこで、質問から疑問文ではない文は除去し、回答からは疑問文を除去したうえで素性を抽出する。ここでは、疑問表現を含む文もしくは文末が以下のいずれかである文を疑問文と判定する。

ですか, でしょうか, ますか, ましたか, でしたか, ませんか,?

図 3.10 に例を示す。この例では、質問は `Q_S1`〜`Q_S3` の 3 文であり、回答は `A_S1`〜`A_S4` の 4 文である。質問のうち `Q_S1` は疑問文ではないので除去する。回答のうち `A_S2` と `A_S3` は疑問文なので除去する。残された `Q_S2`,`Q_S3`,`A_S1`,`A_S4` から素性を抽出する。

例 1:質問: 江戸幕府を開いた人を教えてください。
例 1:回答: 徳川家康が江戸幕府を開いた。

$\mathbf{f^q_{in}}$ wh_no

$\mathbf{f^q_{end}}$ <動詞>タヒトオ END(教えて)

$\mathbf{f^a_{cl}}$ 開いタ

$\mathbf{f^a_{func}}$ ガ, オ, タ

例 2:質問: そばとうどんの違いは何ですか?
例 2:回答: そばはそば粉で作られますが、うどんは小麦粉から作られます。

$\mathbf{f^q_{in}}$ 【何】

$\mathbf{f^q_{in3}}$ <名詞>_は_【何】, は_【何】_です, 【何】_です_か

$\mathbf{f^q_{end}}$ 違いワナンデスカ

$\mathbf{f^a_{cl}}$ 作らレマスガ, 作らレマス

$\mathbf{f^a_{cl\text{-}all}}$ 作らレマスガ_作らレマス

$\mathbf{f^a_{func}}$ ワ, デ, レ_マス_ガ, カラ, レ_マス

図 3.9: 素性抽出の例

3.4.2.2 訓練データの作成

回答タイプ一致判定の分類器を学習するための訓練データは、Yahoo!知恵袋内の質問と回答の集合から作成する。回答タイプの一致判定器の訓練データは正例 (回答タイプが一致する質問と回答の組) と負例 (一致しない組) から構成される。図 3.11 に示すように、正例は元の質問と回答の組から作成し、負例は元の質問と違う質問の回答との組から作成する。ただし、負例とする回答をランダムに選ぶと回答タイプが同じ質問と回答の組を誤って負例としてしまう場合がある。これを避けるため、元の質問と似ていない質問に対応する回答を選ぶ。以下、訓練データ作成の詳細な手続きを示す。

1. 正例の作成
   Yahoo!知恵袋のデータ集合から質問と回答の組 (q, a) を選び、これを正例とする。ただし、質問もしくは回答から素性が抽出できなかった場合、その組は訓練データに含めない。

2. 負例の選択

×Q_S1 姪が入院しました。
Q_S2 小学生の女の子にお見舞いを送りたいのですが、何がいいでしょうか?
Q_S3 今の小学生は何が好きですか?

A_S1 携帯、プリクラ、洋服が好きですよ。
×A_S2 ご病気でしょうか?
×A_S3 長い入院になるんでしょうか?
A_S4 パジャマっぽくないパジャマが重宝しました。

図 3.10: 質問、回答に出現する疑問文の処理

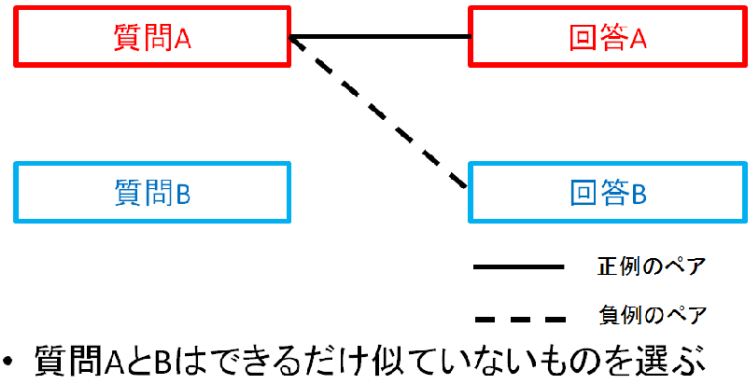

図 3.11: 正例・負例の作成

Yahoo!知恵袋における同じカテゴリ[8]の別の質問から、回答タイプの一致という観点で q との類似度の小さい質問 q'_i を N_{neg} 個選択する。N_{neg} は正例と負例の比を表す。質問間の類似度は、3.4.2.1 で述べた質問の素性 ($\mathbf{f^q_{in}}$, $\mathbf{f^q_{in3}}$, $\mathbf{f^q_{end}}$) から成る素性ベクトルのコサイン類似度で測る。

3. 負例の作成
元の質問 q と q'_i に対応する回答 a'_i の組 (q, a'_i) を負例とする。ただし、質問もしくは回答から素性が抽出できなかった場合、その組は訓練データに含めない。

上述 1〜3 の操作をデータ集合内の全て質問に対して行い、訓練データを作成する。

Yahoo!知恵袋のデータは、質問と回答の文をそれぞれ MeCab で形態素解析し、その結果を XML 形式で表したものである。この XML ファイルから回答タイプの素性を自動的に抽出し、素性ベクトルを作成する。XML の解析には、文書検索モジュールでも用いた Beautiful Soup ライブラリを用いた。抽出した素性の異なり数を表 3.6 に示す。素性 $\mathbf{f^a_{cl}}$,

[8] 3.4.2 項の冒頭で述べたように、Yahoo!知恵袋における質問は 14 種類のカテゴリに分類されている。

表 3.6: 素性の異なり数

素性	異なり数
$\mathbf{f^q_{in}}$	74
$\mathbf{f^q_{in3}}$	11,990
$\mathbf{f^q_{end}}$	29,064
$\mathbf{f^a_{cl}}$	26,178 (112,724)
$\mathbf{f^a_{cl\text{-}all}}$	4,507 (106,841)
$\mathbf{f^a_{func}}$	26,721 (87,092)

$\mathbf{f^a_{cl\text{-}all}}$, $\mathbf{f^a_{func}}$ は素性の種類が非常に多く、素性ベクトルの要素数も多くなり、機械学習の際に過学習を起こしたり、後述するクラスタリングの実行効率が落ちる可能性がある。このため、出現頻度の小さい素性を取り除き、ベクトル作成に用いる素性の数を減らす。$\mathbf{f^a_{cl}}$ はデータ集合内での文書頻度が 3 以上の素性を用いる。素性の文書頻度とは、ここではその素性が出現する回答の数である。$\mathbf{f^a_{cl\text{-}all}}$ は文書頻度が 2 以上の素性を用いる。$\mathbf{f^a_{func}}$ はデータ集合における文書頻度が 3 以上かつ 11,100 以下の素性を用いる。高頻度の付属語列の素性も除去しているのは、これらがどの回答にもよく出現し、回答タイプの一致判定に有効ではないと判断したためである。なお、文書頻度の閾値は経験的に定めた。表 3.6 の括弧内の数値は削除前の素性数である。

素性ベクトル作成の実装

素性ベクトルを作成する詳細な手続きについて述べる。前述のように、訓練データの負例は質問と回答を組み合わせて作成する。そこで、処理の効率化を図るため、質問から得られる素性ベクトルと回答から得られる素性ベクトルを別々に作成し、両者をマージして最終的な素性ベクトルを作成する。

1. 素性リストの作成

   種類別に素性のリストを作成する。具体的には、素性を $\mathbf{f^q_{in}}$+$\mathbf{f^q_{in3}}$, $\mathbf{f^q_{end}}$, $\mathbf{f^a_{cl}}$, $\mathbf{f^a_{func}}$, $\mathbf{f^a_{cl\text{-}all}}$ の 5 つのグループに分け、それぞれについて素性のリストを作成する。各素性リストでは、素性は出現頻度の高い順に並べられ、また個々の素性にはユニークな素性番号が割り当てられている。$\mathbf{f^q_{in}}$ と $\mathbf{f^q_{in3}}$ を 1 つのグループとして取り扱っているのは、実装の都合上これらの素性を同時に抽出しているためである。また、$\mathbf{f^a_{func}}$ と $\mathbf{f^a_{cl\text{-}all}}$ の順序が 3.4.2.1 での説明の順序と異なるのは、$\mathbf{f^a_{cl\text{-}all}}$ の素性を一番最後に考案したためである。以下、それぞれの素性のリストを $L(\mathbf{f^q_{in}} + \mathbf{f^q_{in3}})$, $L(\mathbf{f^q_{end}})$ などと記す。

2. 質問の素性ベクトルの作成

   以下の手続きにしたがって、全ての質問について素性ベクトルを作成し、保存する。図 3.12 はこの手続きを図解したものである。

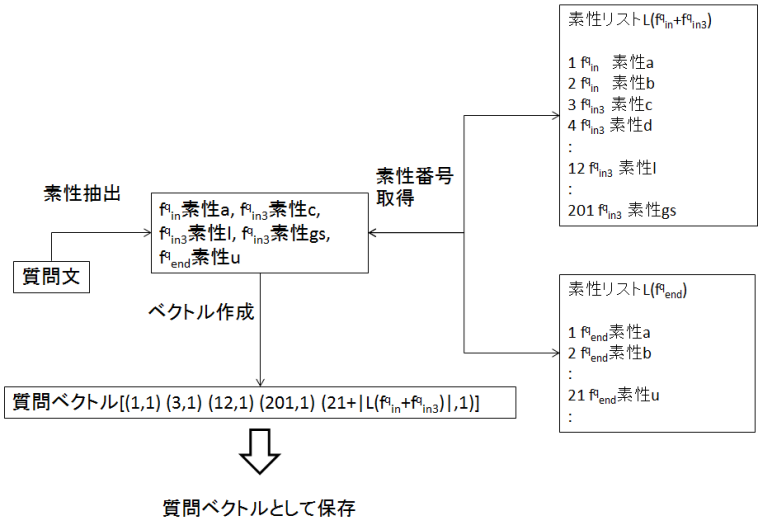

図 3.12: 質問の素性ベクトルの作成

(a) 質問から抽出された $\mathbf{f^q_{in}}$ と $\mathbf{f^q_{in3}}$ の素性に対して、$L(\mathbf{f^q_{in}} + \mathbf{f^q_{in3}})$ を参照し、対応する素性番号 n_1 を得る。この素性を質問の素性ベクトルにおける n_1 番目の次元に対応させる。

(b) 質問から抽出された $\mathbf{f^q_{end}}$ の素性に対して、$L(\mathbf{f^q_{end}})$ を参照し、対応する素性番号 n_2 を得る。この素性を質問の素性ベクトルにおける $|L(\mathbf{f^q_{in}} + \mathbf{f^q_{in3}})| + n_2$ 番目の次元に対応させる。

(c) (a),(b) で抽出した素性の次元の重みは 1、それ以外の次元の重みは 0 として、質問の素性ベクトルを作成する。

3. 回答の素性ベクトルの作成

以下の手続きにしたがって、全ての回答について素性ベクトルを作成し、保存する。図 3.13 はこの手続きを図解したものである。

(a) 回答から抽出された $\mathbf{f^a_{cl}}$ の素性に対して、$L(\mathbf{f^a_{cl}})$ を参照し、対応する素性番号 n_3 を得る。この素性を回答の素性ベクトルにおける n_3 番目の次元に対応させる。

(b) 回答から抽出された $\mathbf{f^a_{func}}$ の素性に対して、$L(\mathbf{f^a_{func}})$ を参照し、対応する素性番号 n_4 を得る。この素性を回答の素性ベクトルにおける $|L(\mathbf{f^a_{cl}})| + n_4$ 番目の次元に対応させる。

(c) 回答から抽出された $\mathbf{f^a_{cl-all}}$ の素性に対して、$L(\mathbf{f^a_{cl-all}})$ を参照し、対応する素性番号 n_5 を得る。この素性を回答の素性ベクトルにおける $|L(\mathbf{f^a_{cl}})| + |L(\mathbf{f^a_{func}})| + n_5$ 番目の次元に対応させる。

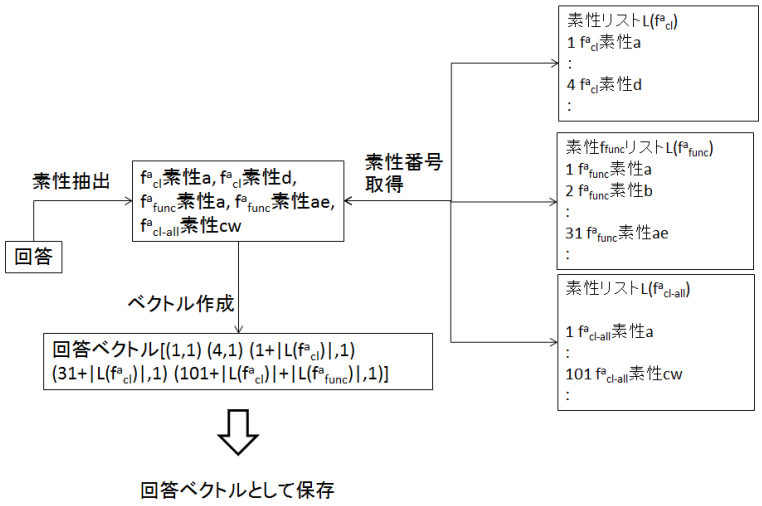

図 3.13: 回答の素性ベクトルの作成

(d) (a)〜(c) で抽出した素性の次元の重みは 1、それ以外の次元の重みは 0 として、回答の素性ベクトルを作成する。

4. 質問と回答の素性ベクトルの統合

   質問と回答を組み合わせて正例または負例を作成する際、対応する質問の素性ベクトルと回答の素性ベクトルを併合して、質問・回答の組に対する素性ベクトルを得る。このとき、回答の素性ベクトルの n 番目の次元は、併合後の素性ベクトルでは、n に質問から抽出される素性の総数 $|L(\mathbf{f^q_{in}} + \mathbf{f^q_{in3}})| + |L(\mathbf{f^q_{end}})|$ を足した次元に対応させる。図 3.14 はこの手続きを図解したものである。

図 3.9 の例 2 の質問と回答の組に対し、素性ベクトルを作成する過程を図 3.15 に示す。図中の L は素性リスト内における素性番号であり、「VEC」は素性ベクトルにおける次元を表わす。$\mathbf{f^q_{end}}$ の素性は、リスト内の素性番号に $L(\mathbf{f^q_{in}} + \mathbf{f^q_{in3}})$ の要素数 12,064 を加えた次元に対応させる。同様に、$\mathbf{f^a_{cl}}$, $\mathbf{f^a_{cl\text{-}all}}$, $\mathbf{f^a_{func}}$ に対しても、リスト内の素性番号にそれ以前の素性の総数を足した次元に対応させる。$*$ はその素性が素性リスト L 内に存在しないことを表す。これは、頻度が低いまたは高いために除外された素性である。このような素性は素性ベクトルでは無視される。

3.4.2.3 訓練データ中の正例・負例の比の決定

一般に、教師あり機械学習のための訓練データにおける正例と負例の比は適切に決定する必要がある。負例が少すぎるときは負例を正例と誤って判定することが多くなり、逆に負例が多すぎるときは正例を負例と誤って判定することが多くなる。

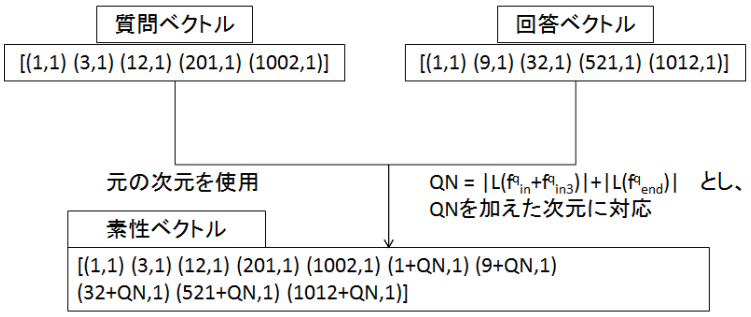

図 3.14: 最終的な素性ベクトルの作成

種類	素性	L	VEC
$\mathbf{f^q_{in}} + \mathbf{f^q_{in3}}$	【何】	2	2
	<名詞>_は_【何】	11	11
	は_【何】_です	21	21
	【何】_です_か	8	8
$\mathbf{f^q_{end}}$	違いワナンデスカ	76	12140
$\mathbf{f^a_{cl}}$	作らレマスガ	16034	57162
	作らレマス	3299	44427

種類	素性	L	VEC
$\mathbf{f^a_{cl-all}}$	作らレマスガ_作らレマス	*	—
$\mathbf{f^a_{func}}$	ワ	*	—
	デ	*	—
	レ_マス_ガ	653	67959
	カラ	4	67340
	レ_マス	80	67386

素性ベクトル
[(2,1) (8,1) (11,1) (21,1) (12140,1) (44427,1) (57162,1) (67310,1) (67386,1) (67959,1)]

図 3.15: 素性ベクトルの作成例

回答タイプ一致判定器を学習するための訓練データにおける正例と負例の比は、実際に文書検索モジュールによって得られる回答候補集合における正例と負例の比に近いことが望ましい。そこで本研究では、Yahoo!知恵袋の回答集合に対して疑似的な文書検索を行い、回答タイプが質問と一致する回答が占める割合を調べることで最適な正例と負例の比を推定する。3.4.2.2 で説明したように、訓練データを作成する際、1 つの質問 (正例) に対し N_{neg} 個の負例を生成する。したがって正例と負例の比は 1:N_{neg} と表せる。

正例と負例の比を決定する手法の概要を図 3.16 に示す。以下に詳細を述べる。

1. 質問・回答のクラスタリング
   Yahoo!知恵袋のデータ集合内の質問・回答の組をクラスタリングし、回答タイプが同じ質問・回答のクラスタを作る。質問・回答の組は、3.4.2.1 で述べた素性のうち、質問から抽出される素性 $\mathbf{f^q_{in}}$, $\mathbf{f^q_{in3}}$, $\mathbf{f^q_{end}}$ から成るベクトルで表現する。すなわち、クラスタリングするデータは質問・回答の組であるが、データ間の類似度は質問のみの類似性で測る。クラスタリングのアルゴリズムは Repeated Bisection 法を用いる。

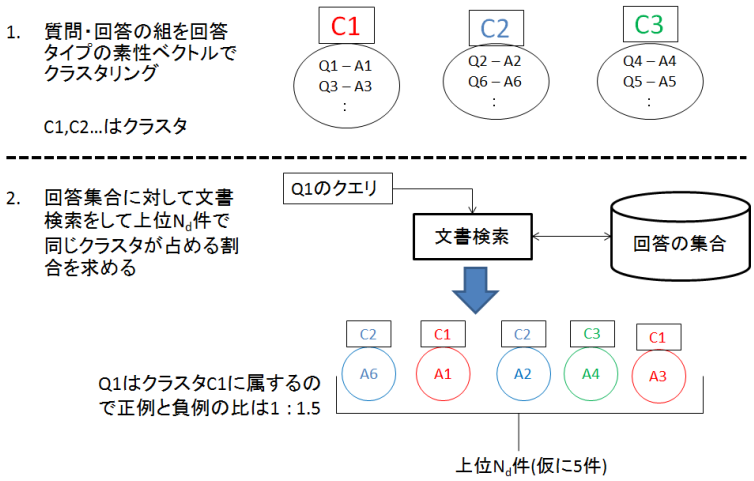

図 3.16: 正例と負例の比の求め方

クラスタ数は 15 に設定する。クラスタリングは CLUTO[9] というツールを利用して実行した。CLUTO でクラスタリングを行うと、各質問・回答の組に対して対応するクラスタ番号（ここでは 0〜14）が割り当てられる。ただし、ベクトルを構成する素性の文書頻度が極端に低いなどの理由でクラスタに分類できない質問・回答の組に対しては −1 が割り当てられる。

2. 回答検索

Yahoo!知恵袋における 1 つの質問をクエリとし、同じく Yahoo!知恵袋の回答集合に対して文書検索を行い、関連度の高い上位 N_d 件の回答を取得する。N_d は提案システムの文書検索モジュールで取得する文書数である。ここでの狙いは、質問応答システムを実際に使用した時に得られる回答候補と同じ数の回答を Yahoo!知恵袋の回答集合から検索することである。しかし、質問応答システムが検索する文書数は N_d と決まっているが、提案システムでは文書集合をパラグラフに分割し、そのパラグラフを回答候補としているため、回答候補の数は N_d よりも多い。しかし、1 つのウェブ文書からいくつのパラグラフが得られるかを予測するのは困難である。そこで、ここでは近似的に取得する回答の数を N_d と設定する。文書検索は、質問と回答をそれぞれ単語ベクトルで表現し、それらのコサイン類似度により回答をランキングする。単語ベクトルは自立語を素性とするベクトルとする。3.4.1 項で述べた内容の関連度の計算と同様に、質問の単語ベクトルは、助詞「は」「が」「の」の前に出現する自立語を重要語とし、その重みを 2 とし、それ以外の自立語の重みを 1 とする。一方、回答の単語ベクトルにおける語の重みは全て 1 とする。両ベクトル共に大きさが 1 になるように正規化する。

[9]http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview

3. 正例数と負例数の集計
   文書検索で得られた回答集合の中で、クエリの質問と同じクラスタに属する質問、すなわち回答タイプが同じであるとみなせる質問の回答を正例、そうでない回答は負例とみなし、集合内の正例と負例の数を求める。このとき、どのクラスタにも分類されなかった (クラスタ番号として -1 が割り当てられた) 質問については、同一のクラスタに属する回答は存在しないため、回答集合は全て負例とみなす。

Yahoo!知恵袋の質問・回答の組を 14 個のカテゴリ毎に分割する。まず、それぞれのカテゴリについて 1 のクラスタリングを行う。2 と 3 の処理もカテゴリ毎に分割した質問・回答の組の部分集合に対して行う。全ての質問について 2 と 3 の処理を行い、正例と負例の総和を求めて、正例と負例の比を決定する。

4 章で述べる評価実験ではウェブ検索で得る文書数 N_d を 100 件としている。ここでは $N_d = 100$ という条件の下、上述の手続きに従って正例と負例の比を推定した。文書検索は、Yahoo!知恵袋における 91,404 組の質問のうち、回答タイプの素性を抽出できないものは除き、残りの 78,233 組 [10] のデータを用いた。結果を表 3.7 に示す。この表では、14 個のカテゴリ並びにデータ集合全体について、質問数、正例と判定された回答の総数、負例と判定された回答の総数、正例の占める割合を示した。「分散」は、(OC07 を除く)OC01〜OC15 の各カテゴリの行については質問ごとに求めた正例の占める割合の分散を、「全体」の行では 14 個のカテゴリごとに求めた正例の占める割合の分散である。

データ集合全体での正例の割合は 0.1443 となり、正例を 1 としたときの負例の数 N_{neg} は $(1 - 0.1443)/0.1443 = 5.930$ となる。すなわち、1 つの質問に対して正例 1 個、負例 5.9 個を用意するのが適していると推測された。したがって、回答タイプの一致判定器を学習する訓練データは正例と負例の比がこの割合になるように作成する。実際の手続きでは、全体のうち 1 割の質問については 1 つの質問に対して負例を 5 個、残りの 9 割の質問については負例を 6 個作成し、全体で正例と負例の比が 1:5.9 になるようにする。

3.4.3 回答候補のスコア

本節の冒頭で述べたように、回答抽出モジュールでは、質問と回答候補の「内容の関連度」と「回答タイプの整合度」によって回答候補をランキングし、適切な回答を選択する。本研究では、この 2 つの基準によるスコアの組み合わせ方法として「フィルタリング方式」と「加算方式」の 2 つの方法を提案する。

[10] 4.1 節で述べる回答タイプ一致判定器の評価では、素性を抽出できる質問・回答の組の数は 76,782 である。ここで用いるデータの数と異なるのは、正例と負例の比を決めるときは図 3.10 に示した疑問文の処理を行っていないため、素性を抽出できない事例の数が少なくなっている。これは、正例と負例の比を実験により推定した後に図 3.10 の処理を考案したためである。本来、正例と負例の比を決めるときと回答タイプの一致を判定するときは同じ手続きで素性を抽出するべきである。今後の課題としたい。

表 3.7: 正例と負例のシミュレーション結果

カテゴリ	質問数	正例数	負例数	正例の割合	分散
OC01	9119	123866	788034	0.1358	0.009990
OC02	8095	106758	702742	0.1319	0.01105
OC03	2349	31961	202939	0.1361	0.009342
OC04	2059	28042	177858	0.1362	0.01035
OC05	3017	42870	258830	0.1421	0.01030
OC06	6048	96621	508179	0.1598	0.01463
OC08	6620	97595	564405	0.1474	0.01235
OC09	13379	193043	1144857	0.1442	0.01237
OC10	4102	54147	356053	0.1320	0.008940
OC11	1679	21497	146403	0.1280	0.009425
OC12	6232	86901	536299	0.1394	0.008959
OC13	3008	40541	260259	0.1348	0.008611
OC14	10489	172319	876581	0.1643	0.01465
OC15	2037	33316	170384	0.1636	0.01406
全体	78233	1129477	6693823	0.1443	0.0001369

3.4.3.1 フィルタリング方式

この方式では、回答タイプの一致判定で不一致と判定された回答候補を取り除く。その後、残された回答候補について質問との「内容の関連度」のスコア $Score_r$ を計算し、これを回答候補の最終的なスコアとする。フィルタリング方式における回答候補 i のスコアの定義を式 (3.3) に示す。

$$Score_{answer}(i) = \begin{cases} Score_r(i) & \text{if} \quad \text{回答タイプが一致} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

図 3.17 は、フィルタリング方式による回答選択を図示している。ここでは 5 つの回答候補があるが、2 番目と 4 番目の候補は回答タイプが一致しないと判定されたため除去される。残りの回答候補については質問との内容の関連度を求め、これを最終のスコアとする。最後にこのスコアの降順に回答候補を並べて出力する。

3.4.3.2 加算方式

この方式では、2 つの基準によって計算されたスコアを合計することで最終的なスコアを求める。各回答候補に対して、「内容の関連度」のスコア $Score_r$ と「回答タイプの整合度」のスコア $Score_t$ をそれぞれ算出する。2 種類のスコアを組み合わせるときは、両者の

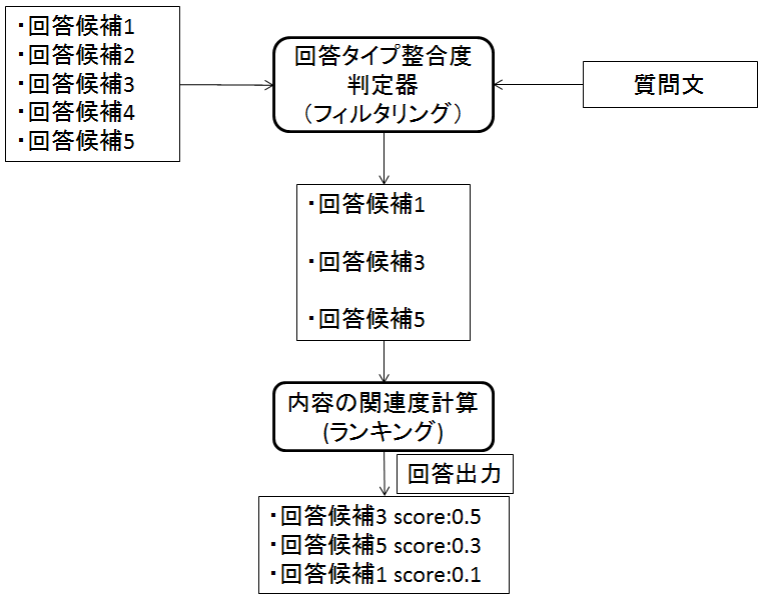

図 3.17: フィルタリング方式

重み付き和を最終的なスコアとすることが一般的である。しかし、2 つのスコアの重みをどのように最適化するかが問題となる。開発データを用意して最適化することも考えられるが、多様な質問を含む開発データを用意することは現実的には難しい。そこで、本研究では、$Score_r$ と $Score_t$ を相対スコアに変換し、それらを合計したものを最終的なスコア $Score_{answer}$ とする。相対スコアとは、回答候補集合におけるスコアの最大値に対する比と定義する。また、2 つの相対スコアの重みはともに 0.5 とする。$Score_{answer}$ の値の範囲は $[0, 1]$ となる。回答候補 i に対するスコアの定義を式 (3.4) に示す。

$$Score_{answer}(i) = (Score_r(i)/Score_{rmax}) \times 0.5 + (Score_t(i)/Score_{tmax}) \times 0.5 \qquad (3.4)$$

$Score_{rmax}, Score_{tmax}$ はそれぞれ回答候補集合における $Score_r, Score_t$ の最大のスコアである。

図 3.18 は加算方式による回答選択を図示したものである。質問と回答候補の集合が与えられたとき、それぞれの回答候補について $Score_r$ と $Score_t$ を求める。次に、内容の類似度のスコアの最大値は回答候補 3 の 0.8 であり、各スコアを 0.8 で割って $Score_r$ の相対スコアを得る。同様に、回答タイプの整合度のスコアの最大値は回答候補 5 の 0.9 であり、各スコアを 0.9 で割って $Score_t$ の相対スコアを得る。これらのスコアを 0.5 倍し、加算して、回答候補のスコアを得る。最後にこのスコアの降順に回答候補を並べて出力する。

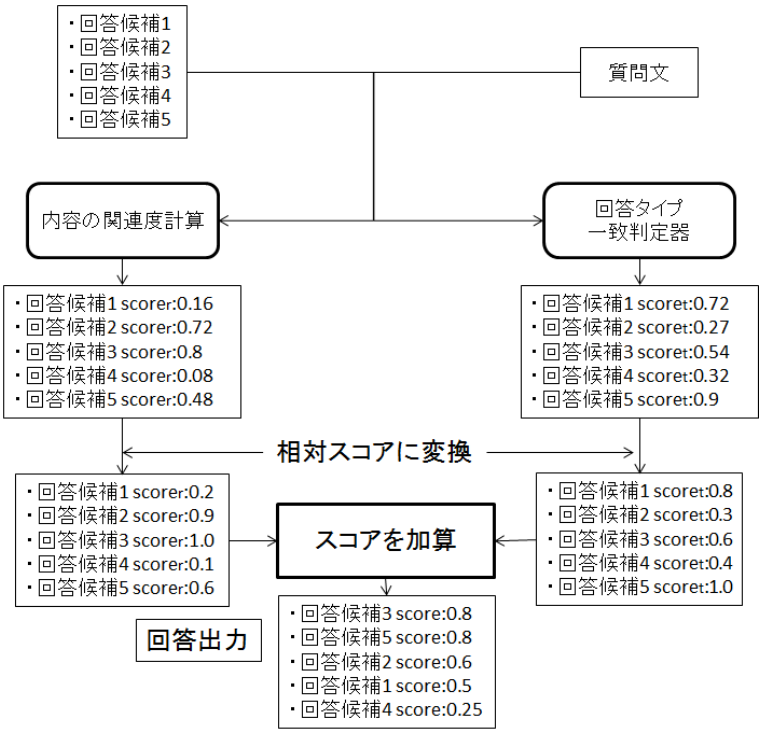
・回答候補1
・回答候補2
・回答候補3
・回答候補4
・回答候補5
質問文
内容の関連度計算
回答タイプ
一致判定器
・回答候補1 scorer:0.16
・回答候補2 scorer:0.72
・回答候補3 scorer:0.8
・回答候補4 scorer:0.08
・回答候補5 scorer:0.48
・回答候補1 scoret:0.72
・回答候補2 scoret:0.27
・回答候補3 scoret:0.54
・回答候補4 scoret:0.32
・回答候補5 scoret:0.9
相対スコアに変換
・回答候補1 scorer:0.2
・回答候補2 scorer:0.9
・回答候補3 scorer:1.0
・回答候補4 scorer:0.1
・回答候補5 scorer:0.6
スコアを加算
・回答候補1 scoret:0.8
・回答候補2 scoret:0.3
・回答候補3 scoret:0.6
・回答候補4 scoret:0.4
・回答候補5 scoret:1.0
回答出力
・回答候補3 score:0.8
・回答候補5 score:0.8
・回答候補2 score:0.6
・回答候補1 score:0.5
・回答候補4 score:0.25

図 3.18: 加算方式

第4章　評価実験

本章では提案手法の評価実験について述べる。4.1 節では、3.4.2 項で述べた回答タイプの一致を判定する分類器を評価する。4.2 節では、本研究の質問応答システムを評価する。

4.1　回答タイプ一致判定器の評価

4.1.1　実験内容

ここでは以下の 2 つの実験を行う。

- 正例と負例の比の決定方法の評価
  3.4.2.3 で述べた正例と負例の比を決める方法を評価する。提案手法で決めた正例と負例の比を 1:5.9 とした訓練データのほかに、比を 1:1, 1:3, 1:5, 1:7, 1:10 とした訓練データを用意した。また、テストデータにおける真の正例と負例の比は不明なので、テストデータについても比を 1:1, 1:3, 1:5, 1:5.9, 1:7, 1:10 としたデータを用意した。訓練データとテストデータの全ての組み合わせについて、回答タイプの一致を判定する分類器の学習と評価を行った。この実験では学習素性は 3.4.2.1 で述べた全ての素性を用いる。

- 素性の有効性の検証
  3.4.2.1 で提案した 6 種類の素性の有効性を評価する。具体的には、以下の素性集合を用いて分類器を学習し、その性能を比較した。

  ALL 全ての素性
  ALL $-$ $\mathbf{f^q_{in}}$ 質問中の疑問表現 $\mathbf{f^q_{in}}$ 以外の素性
  ALL $-$ $\mathbf{f^q_{in3}}$ 質問中の疑問表現を含む 3-gram $\mathbf{f^q_{in3}}$ 以外の素性
  ALL $-$ $\mathbf{f^q_{end}}$ 質問中の文末表現 $\mathbf{f^q_{end}}$ 以外の素性
  ALL $-$ $\mathbf{f^a_{cl}}$ 回答中の節末表現 $\mathbf{f^a_{cl}}$ 以外の素性
  ALL $-$ $\mathbf{f^a_{cl\text{-}all}}$ 回答中の節末表現の組み合わせ $\mathbf{f^a_{cl\text{-}all}}$ 以外の素性
  ALL $-$ $\mathbf{f^a_{func}}$ 回答中の付属語列 $\mathbf{f^a_{func}}$ 以外の素性
  ALL $-$ $\mathbf{f^q_{in}}$ $-$ $\mathbf{f^q_{in3}}$ 質問中の疑問表現 $\mathbf{f^q_{in}}$ とそれを含む 3-gram $\mathbf{f^q_{in3}}$ 以外の素性

1 つの素性を除いた素性集合を用いたときと全ての素性を用いたときを比べることで、その素性の有効性を検証する。ただし、$\mathbf{f_{in}^q}$ は素性数が少ないため、これを除いても **ALL** のときとの差があまり出ないと考えられることから、疑問表現の素性 $\mathbf{f_{in}^q}$ と $\mathbf{f_{in3}^q}$ を除く素性集合 $\mathbf{ALL} - \mathbf{f_{in}^q} - \mathbf{f_{in3}^q}$ を用いた分類器も評価した。この実験では、正例と負例の比はテストデータ、訓練データ共に 1:5.9 とした。

実験には Yahoo!知恵袋の質問・回答データ 91,404 組を用いた。これらのうち、質問または回答から素性を取り出すことができなかった組を除く 76,782 組を正例とした。次に、負例の作成方法の実際の手続きについて説明する。まず、訓練データにおけるそれぞれの質問について、訓練データにおける他の質問との類似度 (3.4.2.2 で記述) を求め、類似度の低い 20 個の質問を選択する。多くの場合、類似度が 0 となる質問が多数得られるので、その中から 20 個をランダムに選択する。次に、正例と負例の比が 1:x のとき、選択した 20 個の質問の中から類似度が低い順に x 個の質問を選択し、これらに対応する回答と元の質問を組み合わせて負例とする。ただし、回答から素性を抽出できなかったときは、それは負例とせず、次の質問を選ぶ。今回の実験では負例の割合が一番大きいのは 1:10 なので、1 つの質問につき類似度の低い 10 個の質問を検索すればよいが、回答から素性が抽出できない時もあるため、余裕を見て 20 個の質問を検索している。

このように作成した正例と負例の集合のうち、ランダムに選択した 1 割の事例をテストデータ、残りを訓練データとした。

4.1.2 評価基準

回答タイプ一致判定器の評価基準として、正解率並びに正例に対する精度、再現率、F 値を用いる。正解率 ($accuracy$) は、回答タイプ一致の判定結果が正解と一致した割合である。テストデータのデータ数を n、そのうち判定器が回答タイプの一致・不一致を正しく判定した数を r とすると、正解率は式 (4.1) で計算される。

$$accuracy = \frac{r}{n} \tag{4.1}$$

さらに、回答タイプの一致判定を正例、すなわち回答タイプが一致している質問と回答の組を検索するタスクとみなしたときの精度 ($precision$)、再現率 ($recall$)、F 値 ($F\text{-}measure$) も評価基準とする。テストデータ内の正例の数を C、出力結果において正例と判断されたデータ数を N、出力結果において正しく正例と判定された数を R とすると、3 つの評価基準は以下のようになる。

$$precision = \frac{R}{N} \tag{4.2}$$

$$recall = \frac{R}{C} \tag{4.3}$$

$$F\text{-}measure = \frac{2 \times precision \times recall}{precision + recall} \tag{4.4}$$

表 4.1: 回答タイプ一致判定器の正解率

		訓練データ					
		1 : 1	1 : 3	1 : 5	1 : 5.9	1 : 7	1 : 10
テストデータ	1 : 1	0.7510	0.6346	0.5606	0.5386	0.5254	0.5055
	1 : 3	0.6157	0.7947	0.7773	0.7692	0.7624	0.7525
	1 : 5	0.4820	0.8046	0.8435	0.8440	0.8408	0.8350
	1 : 5.9	0.4356	0.7867	0.8549	0.8600	0.8603	0.8566
	1 : 7	0.3914	0.7647	0.8588	0.8718	0.8770	0.8759
	1 : 10	0.3130	0.6935	0.8366	0.8685	0.8913	0.9084

表 4.2: 回答タイプ一致判定器の精度

		訓練データ					
		1 : 1	1 : 3	1 : 5	1 : 5.9	1 : 7	1 : 10
テストデータ	1 : 1	0.7679	0.9788	0.9942	0.9958	0.9976	0.9821
	1 : 3	0.3653	0.7359	0.9418	0.9724	0.9835	0.9399
	1 : 5	0.2041	0.3777	0.6793	0.8094	0.8892	0.9399
	1 : 5.9	0.1666	0.2699	0.5009	0.6360	0.7827	0.8887
	1 : 7	0.1364	0.1914	0.3300	0.4377	0.5921	0.8369
	1 : 10	0.0908	0.0931	0.1169	0.1361	0.1665	0.4089

テストデータは、ランダムに選択されるので、1 つの設定に対して評価を 5 回行い、各評価指標の平均を求めた。

4.1.3 結果と考察

4.1.3.1 正例・負例の比の実験

訓練データとテストデータの正例と負例の比を変化させたときの正解率、精度、再現率、F 値を表 4.1, 4.2, 4.3, 4.4 にそれぞれ示す。

表 4.1 から、テストデータと訓練データの比が同じ場合は、負例の比が大きくなるほど正解率が高くなっていることが分かる。一般に、正例と負例の数に偏りがあると、事例数の多いクラスを出力することで正解になる可能性が高くなる。この場合、負例の数が多いテストデータほど、システムが負例を出力すれば正解になる可能性が高くなるため、正解率も向上する。しかし、訓練データ中の負例の比がテストデータ中の負例の比よりも多い場合は、正解率が下がっていることが分かる。これは、訓練データ中の負例が増えると、学習した判定器が負例と判定することが多くなり、負例の多いテストデータに対しては高

表 4.3: 回答タイプ一致判定器の再現率

		訓練データ					
		1 : 1	1 : 3	1 : 5	1 : 5.9	1 : 7	1 : 10
テストデータ	1 : 1	0.7263	0.2730	0.1198	0.0807	0.0521	0.0113
	1 : 3	0.7265	0.2740	0.1219	0.0803	0.0515	0.0114
	1 : 5	0.7268	0.2767	0.1219	0.0824	0.0502	0.0116
	1 : 5.9	0.7237	0.2730	0.1203	0.0827	0.0514	0.0120
	1 : 7	0.7254	0.2728	0.1196	0.0816	0.0520	0.0118
	1 : 10	0.7260	0.2708	0.1206	0.0825	0.0515	0.0120

表 4.4: 回答タイプ一致判定器の F 値

		訓練データ					
		1 : 1	1 : 3	1 : 5	1 : 5.9	1 : 7	1 : 10
テストデータ	1 : 1	0.7465	0.4269	0.2138	0.1491	0.0991	0.0223
	1 : 3	0.4861	0.3993	0.2158	0.1484	0.0980	0.0226
	1 : 5	0.3187	0.3194	0.2067	0.1495	0.0952	0.0229
	1 : 5.9	0.2707	0.2714	0.1940	0.1464	0.0964	0.0236
	1 : 7	0.2296	0.2249	0.1756	0.1376	0.0957	0.0234
	1 : 10	0.1614	0.1385	0.1187	0.1027	0.0787	0.0234

い正解率を示すが、負例の少ないテストデータに対しては正例を誤って負例と判定する場合が増えるからだと考えられる。

また、表 4.4 の F 値に着目すると、訓練・テストデータの正例と負例の比が同じとき、負例の数を増やすと F 値が下がることが分かる。精度と再現率を見ると、負例の数が多くなると両方とも下がるが、再現率の方が精度に比べて大きく低下することが分かる。これは、負例の数が多いテストデータでは本来正例であるものを負例だと判定することが多くなったためと思われる。

同じ比の訓練データを用いて、テストデータの比を変化させた場合、すなわち表における 1 つの列に着目して評価値を比較した場合、負例が多くなると精度は下がるが、再現率はあまり変化しなかった。また、同じ比のテストデータに対して、訓練データの比を変化させた場合、すなわち表における 1 つの行に着目して評価値を比較した場合、負例の数が増えると精度は向上するが、再現率は下がっている。訓練データの負例を多くすると分類器が正例と判定する回数が減るためである。

質問応答システムでは、知識源に存在する全ての正答を取り出すことは要求されず、システムが出力した回答が正しいことが求められる。言い換えれば再現率よりも精度が重視される。回答タイプの一致判定でも、正例を網羅的に取り出すことより、正例と判定した時にその判断が正しいこと、つまり精度が高いことが重要だと考える。実験結果では、正例と負例の比が 1:10 のときの精度が最も高い。しかし、この時はほとんどの回答候補を負例と判定していることになり、回答タイプが一致している回答候補が抽出されにくいという問題点がある。

今回の実験では、提案手法で決めた正例と負例の比が最適であるという結論は得られなかった。しかし、実験データの正解は人手で判定したものではなく自動的に作成されたものであること、テストデータにおける真の正例と負例の比が不明であることなどから、実験設定自体に改善の余地がある。正例と負例の比を決定する方法については、4.2 節の質問応答システムの評価実験で改めて評価する。

4.1.3.2 素性の検証実験

学習素性を変えたときの判定器の評価結果を表 4.5 に示す。 実験結果より、全ての素性を用いた **ALL** は他の素性集合よりも高い正解率を示した。これにより、提案した 6 種類の素性は全て回答タイプ一致判定の正解率の向上に貢献するといえる。

1 種類の素性を除いた素性集合の中では、$\mathbf{ALL} - \mathbf{f^q_{in3}}$ が最も **ALL** との正解率の差が大きいことから、疑問表現を含む 3-gram の素性が最も有効な素性といえる。質問から取り出される素性の中では、$\mathbf{f^q_{in3}}$,$\mathbf{f^q_{end}}$,$\mathbf{f^q_{in}}$ の順に有効性が高い。しかし、$\mathbf{f^q_{in}}$ は素性の数が少なく、**ALL** との正解率の差が表れにくい。疑問表現の素性を除いた $\mathbf{ALL} - \mathbf{f^q_{in}} - \mathbf{f^q_{in3}}$ の正解率が大きく下がることから、疑問表現は回答タイプの一致判定に有効である。直感的にも「なぜ」「どうして」などの疑問表現は回答タイプを示唆すると考えられる。回答から取り出される素性の中では、$\mathbf{f^a_{func}}$ と $\mathbf{f^a_{cl}}$ はほぼ同等であり、付属語の列も節末表現もとも

表 4.5: 素性集合別の回答タイプ一致判定器の結果

素性集合	正解率	精度	再現率	F 値
ALL	0.8600	0.6360	0.0827	0.1464
$\mathbf{ALL} - \mathbf{f^q_{in}}$	0.8452	0.6562	0.1494	0.2434
$\mathbf{ALL} - \mathbf{f^q_{in3}}$	0.8438	0.6810	0.1186	0.2020
$\mathbf{ALL} - \mathbf{f^q_{end}}$	0.8440	0.6920	0.1157	0.1982
$\mathbf{ALL} - \mathbf{f^a_{cl}}$	0.8568	0.5805	0.0414	0.0774
$\mathbf{ALL} - \mathbf{f^a_{cl\text{-}all}}$	0.8603	0.6539	0.0813	0.1447
$\mathbf{ALL} - \mathbf{f^a_{func}}$	0.8562	0.5767	0.0340	0.0642
$\mathbf{ALL} - \mathbf{f^q_{in}} - \mathbf{f^q_{in3}}$	0.8413	0.6000	0.1443	0.2327

に回答タイプ一致判定に有効といえる。この 2 つと比べて $\mathbf{f^a_{cl\text{-}all}}$ の有効性は低いが、$\mathbf{f^a_{cl\text{-}all}}$ の素性数が少ないことも一因と考えられる。

精度、再現率、F 値を比較すると、 $\mathbf{ALL} - \mathbf{f^q_{in}}$, $\mathbf{ALL} - \mathbf{f^q_{in3}}$, $\mathbf{ALL} - \mathbf{f^q_{end}}$ は全ての素性を用いた ALL を上回る。また、 $\mathbf{ALL} - \mathbf{f^a_{cl\text{-}all}}$ の精度は **ALL** を上回る。これらの結果から、質問から取り出される 3 つの素性や $\mathbf{f^a_{cl\text{-}all}}$ は回答タイプの一致判定に悪影響を与えると言える。しかし、正解率の比較ではこれらの素性の有効性が確認できる。精度、再現率、F 値は正例に対する判定の評価であることから、これらの素性は主に負例の判定、つまり回答タイプが一致しない事例に対して不一致と正しく判定することに主に貢献すると考えられる。

4.2 質問応答システムの評価

実装した質問応答システムの評価実験を行う。特に、3.4.3 項で述べた回答候補のスコアを算出する 2 つの方式の比較や、3.4.2.3 で述べた正例と負例の比を決定する手法の有効性を検証する。

4.2.1 実験内容

まず、評価に用いる質問セットを用意した。本研究では様々な質問を受け付ける質問応答システムの構築を目指しているため、7 種類の質問を評価の対象にした。具体的には、表 4.6 に示すように、比較、事実確認、ファクトイド (固有名詞)、定義、理由、方法、意見・感想の 7 種類の質問を用いた。7 種類の質問のそれぞれについて 5 個、計 35 個の質問を用いた。これらの質問はインターネット上の雑学サイトなどから収集した。実験に用いた質問の一覧を付録 A に示す。文書検索の際に取得する文書数 N_d は 100 と設定した。ここでは以下の 3 つの評価実験を行った。

表 4.6: 評価に用いた質問の分類

比較	2 つの事柄の違いを問う質問	〜と … の違いは何?
事実確認	ある事実の真偽・様相を問う質問	〜はどうなっていますか?, 〜は本当ですか?
ファクトイド	固有名詞や短い言葉での回答を問う質問	〜した人は誰?, 〜がある場所はどこですか?
定義	事柄の定義を問う質問	〜とはなんですか?, 〜はどんな人物ですか?
方法	方法・手順を問う質問	〜はどうやりますか?, 〜するにはどうすればよいですか?
理由	現象や行動の理由を問う質問	なぜ〜なのですか?, どうして〜するのですか?
意見・感想	意見や感想など思考についての質問	〜についてどう思いますか?, 〜はどうでしたか?

- 回答候補選択手法の評価
  回答候補のスコアを決める「フィルタリング方式」と「加算方式」の 2 つの方法を比較した。また、回答タイプの整合度のスコアを用いず (回答タイプの一致を考慮せず)、内容の関連度のスコア $Score_r$ のみで回答選択する手法をベースラインとし、これと提案手法を比較した。。

- 正例と負例の比の決定方法の評価
  回答タイプ一致判定器の訓練データとして、正例と負例の比がそれぞれ 1:1, 1:5.9, 1:10 のデータを用意し、結果を比較した。この実験では、学習素性は 3.4.2.1 で述べた全ての素性を用いる。

- 素性の評価
  4.1.3.2 の実験では、正解率の比較では全ての素性について有効性を確認できたが、精度の比較では、$\mathbf{f^q_{in}}$, $\mathbf{f^q_{in3}}$, $\mathbf{f^q_{end}}$, $\mathbf{f^a_{cl\text{-}all}}$ についてはこれを除いた時の方が全ての素性を用いたときよりも高く、これらの素性が悪影響を与えることが示された。質問応答システムにおけるこれらの素性の有効性を検証するために、素性集合 $\mathbf{ALL}$,$\mathbf{ALL} - \mathbf{f^a_{cl\text{-}all}}$,$\mathbf{ALL} - \mathbf{f^q_{end}}$,$\mathbf{ALL} - \mathbf{f^q_{in}} - \mathbf{f^q_{in3}}$ によって学習した回答タイプ一致判定器を用いたシステムを比較した。質問応答システムの評価は人手で行うため、その作業コストを軽減するために、$\mathbf{f^q_{in}}$ と $\mathbf{f^q_{in3}}$ についてはこれをまとめて除いた素性集合を比較した。この実験では訓練データにおける正例と負例の比は 1:5.9 とした。

4.2.2 評価基準

評価基準は、正答のランクの逆数の平均 (MRR)、出力上位 10 件の精度の平均 (AP')、出力上位 10 件の精度 (P_{top10}) の 3 つとした。

MRR

MRR(mean reciprocal rank) は、質問応答システムが出力するランク付けされた回答のリストに対し、その出力の中で一番上位にある正答のランクの逆数 (reciprocal rank;rr) の平均である。例えば 10 問の質問があり、1 位で正答が得られた質問が 2

問、1 位で正解せず 2 位で正解した質問が 5 問、 2 位までに正解せず 3 位で正解したものが 3 問だったとき、$MRR = (1/1 \times 2 + 1/2 \times 5 + 1/3 \times 3)/10 = 0.55$ となる。本実験では、回答出力上位 100 位までを対象に正答を探し、100 位以内に存在しない場合、その質問の rr(reciprocal rank) は 0 とする。rr 及び MRR の定義を式 (4.5) と (4.6) に示す。

$$rr(k) = \begin{cases} \frac{1}{rank_k} & \text{if} \quad \text{出力 100 位以内に正答がある} \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

$$MRR = \frac{1}{N} \sum_{k=1}^{N} rr(k) \tag{4.6}$$

N は質問数であり、$rank_k$ は k 番目の質問における正答の最高順位である。MRR の値が高いほど、質問応答システムが出力できた正答のランクが高いことを示す。

AP'

AP'(Average Precision') は、各質問に対し、質問応答システムが上位 10 件の回答を順に出力したとき、正答が得られたときの精度の平均を ap' とし、質問集合全体の ap' の平均を求めた値である。例えば、ある質問に対するシステムの回答出力のうち、上位 10 件の中で 2 位、5 位、8 位が正答だった場合、$ap' = (1/2 + 2/5 + 3/8)/3 = 0.425$ となる。ap' 及び AP' の定義を式 (4.7) と (4.8) に示す。

$$ap' = \frac{1}{n} \sum_{i} P_i \tag{4.7}$$

$$AP' = \frac{1}{N} \sum_{k=1}^{N} ap'_k \tag{4.8}$$

n は質問における出力上位 10 件中の正答数であり、P_i は i 番目の正答が得られた時点での精度を表す。ap'_k は k 番目の質問の ap' である。AP' の値が高いほど、質問応答システムの出力の上位に正答が集まっていることを示す。

P_{top10}

P_{top10} は、質問応答システムが回答を 10 件出力したときの精度の平均であり、式 (4.9) で定義される。

$$P_{top10} = \frac{1}{N} \sum_{k=1}^{N} \frac{top10_k}{10} \tag{4.9}$$

$top10_k$ は質問 k に対する出力の上位 10 件に含まれる正答数である。P_{top10} の値が高いほど、出力上位 10 件の中により多くの正答が含まれていることを示す。

MRR は、質問応答システム、特にファクトイド型のシステムに対してよく用いられる評価基準である。MRR による評価では回答出力の最上位の正答しか評価の対象にならない。

しかし、質問によっては複数の回答が要求される場合もあり、どれだけ多くの正答が得られたかという観点からの評価も必要である。特に、意見・感想を問う質問に対しては、肯定的な意見、否定的な意見や様々な観点からの感想など、複数の情報をユーザに提供することが求められる。このようなときは最上位の正答のみを評価する MRR を用いるのは適切ではない。そこで、情報検索の評価に使われる平均精度 (average precision) を簡潔にした AP'、及び P_{top10} も評価基準とした。この 2 つの基準は、出力された回答の上位にどれだけ多く正答が含まれているかという観点からシステムを評価できると考えられる。

4.2.3 結果と考察

質問応答システムの MRR, AP', P_{top10} をそれぞれ表 4.7, 4.8, 4.9 に示す。表中の「フィルタリング方式」、「加算方式」、「関連度のみ」は回答候補のスコアを計算する手法を、「1:1」、「1:5.9」、「1:10」は回答タイプ一致判定器の訓練データにおける正例と負例の比を表す。表の各行は、7 種類の質問、および 35 個の質問集合全体の評価値を示している。

まず、提案手法とベースラインの違いについて論じる。内容の関連度と回答タイプの整合度の両方を考慮したフィルタリング方式や加算方式 (提案手法) と、ベースラインとなる関連度のみの回答選択手法の MRR や AP' を比較すると、フィルタリング方式や加算方式の方がよい結果であることが分かる。質問の種類ごとに見ると、定義、方法、意見・感想において提案手法の方が優れていることが分かる。また、本研究で提案する手法で定めた正例と負例の比を 1:5.9 としたときの訓練データを用いたシステムでは、比較やファクトイドの質問では提案手法はベースラインよりも MRR や AP' で劣っているが、P_{top10} では、加算方式がベースラインを上回った。以上から、提案手法はベースラインに優るといえる。提案手法とベースラインの違いは回答タイプの整合度を考慮するか否かである。実験の結果から、回答選択において回答タイプの一致を判定することは重要であると言える。

2 つの回答選択方式を比較すると、加算方式の方がフィルタリング方式よりも全体的に結果が良いことが分かる。これは、回答タイプの一致を判定する分類器の正解率が十分に高くなく、フィルタリング方式では回答タイプ一致判定によって不一致と判定した回答候補を除外することで正答を誤って取り除く場合が多いためと推察される。質問の種類ごとに正例・負例の比が 1:5.9 の時で両者を比較すると、定義やファクトイドの質問ではフィルタリング方式のほうが MRR や AP' が高いことがわかる。これらの事実を問う質問は訓練データによく出現することから、回答タイプの判別が比較的容易で、フィルタリング方式が効果的に働くためと考えられる。

正例と負例の比については、1:5.9 が 1:1 や 1:10 と比べて質問セット全体の評価値が高いことから、提案手法による正例と負例の比の決定方法の有効性が確認できた。訓練データ中の正例・負例の比が 1:1 の場合は、「フィルタリング方式」「加算方式」ともに比較、事実確認、ファクトイドの質問においてよい結果を示している。しかし、方法や定義といった文章で答えるノンファクトイド型の質問に対しては、他の比の実験結果より低いこ

とが分かる。

また、訓練データ中の正例・負例の比が 1:10 の場合、フィルタリング方式では、理由や意見・感想の質問に対する MRR が高いが、他の種類の質問及び質問セット全体の結果は他のシステムより低くなっていることが分かる。個々の質問に対する出力結果を見てみると、正解となる回答候補を全て取り除いてしまい、$rr = 0$ となっている場合がいくつかあることが分かった。つまり正例と負例の比が 1:10 の訓練データを用いた判定器は、ほとんどの回答候補に対して質問と回答タイプが一致しないと判定するため、実用的ではない。

加算方式で正例と負例の比が 1:10 と 1:5.9 の場合を比較すると、事実確認、方法、意見・感想の質問において、1:5.9 の場合の評価結果が優れていることが分かる。理由の質問については、MRR は 1:10 の方が高いが、AP' はほぼ同じであり、P_{top10} は 1:5.9 の方が高い。このように MRR でのみ評価値が高く、AP' や P_{top10} では評価値が低いということは、正例・負例の比が 1:10 の訓練データを用いる場合では、正答を多く取り出すのに不向きであると言える。

正例と負例の比が 1:5.9 の場合では、ファクトイドの質問に対して、1:1 や 1:10 のシステムより劣る傾向がみられる。出力を見てみると、1:1 のときに上位にあった正答を 1:5.9 のときでは回答タイプが一致しないと判断して除外していることが多かった。回答タイプ一致判定器のさらなる改良が必要といえる。

素性の有効性の検証

回答タイプ一致判定に用いる素性を変化させたときの評価結果を表 4.10, 4.11, 4.12 に示す。これらの表では $\mathbf{ALL} - \mathbf{f^a_{cl\text{-}all}}$, $\mathbf{ALL} - \mathbf{f^q_{end}}$, $\mathbf{ALL} - \mathbf{f^q_{in}} - \mathbf{f^q_{i3}}$ はそれぞれ $-\mathbf{f^a_{cl\text{-}all}}$, $-\mathbf{f^q_{end}}$, $-\mathbf{f^q_{in}} - \mathbf{f^q_{i3}}$ と略記する。

まず、**ALL** と $\mathbf{ALL} - \mathbf{f^a_{cl\text{-}all}}$ を比較してみると、加算方式では、$\mathbf{ALL} - \mathbf{f^a_{cl\text{-}all}}$ よりも **ALL** の方がよい結果となっていることが分かる。フィルタリング方式でも、全体の MRR はほぼ同等であるが、定義や方法などの質問に対する MRR や AP' では **ALL** の方が高い。このため、節末表現の組み合わせの素性である $\mathbf{f^a_{cl\text{-}all}}$ は、回答選択の性能を向上させるのに有効であると考えられる。

次に質問から得られる素性の有効性について考察する。**ALL** は、$\mathbf{ALL} - \mathbf{f^q_{end}}$ と比べて、MRR と AP' はほぼ等しく、P_{top10} は高い。一方、**ALL** は $\mathbf{ALL} - \mathbf{f^q_{in}} - \mathbf{f^q_{in3}}$ と比べて、加算方式での MRR と AP' はわずかに低く、P_{top10} では高い。また、フィルタリング方式では全ての指標で **ALL** の方が $\mathbf{ALL} - \mathbf{f^q_{in}} - \mathbf{f^q_{in3}}$ よりも上回る。全般的に、質問から取り出す素性を使うときと使わない時では、MRR と AP' はほぼ同等で、P_{top10} は質問の素性を使う方がよい。以上から、本研究で提案する素性は回答タイプの一致を考慮する質問応答システムにおいて有効に働くと結論付けられる。

表 4.7: 質問応答システムの MRR

	フィルタリング方式			加算方式			関連度のみ
	1 : 1	1 : 5.9	1 : 10	1 : 1	1 : 5.9	1 : 10	
比較	0.70	0.73	0.36	0.90	0.67	0.65	0.70
事実確認	0.71	0.55	0.60	0.65	0.81	0.72	0.57
ファクトイド	0.54	0.48	0.32	0.43	0.41	0.80	0.53
定義	0.36	0.67	0.16	0.42	0.54	0.57	0.40
方法	0.28	0.61	0.35	0.62	0.63	0.49	0.21
理由	0.32	0.25	0.41	0.14	0.43	0.47	0.39
意見・感想	0.38	0.60	0.65	0.46	0.90	0.55	0.21
全体	0.47	0.55	0.41	0.52	0.63	0.61	0.43

表 4.8: 質問応答システムの AP'

	フィルタリング方式			加算方式			関連度のみ
	1 : 1	1 : 5.9	1 : 10	1 : 1	1 : 5.9	1 : 10	
比較	0.65	0.56	0.35	0.75	0.62	0.47	0.64
事実確認	0.61	0.44	0.46	0.50	0.65	0.60	0.51
ファクトイド	0.54	0.39	0.25	0.42	0.39	0.70	0.53
定義	0.33	0.60	0.18	0.38	0.48	0.43	0.39
方法	0.36	0.54	0.31	0.49	0.54	0.40	0.26
理由	0.33	0.27	0.41	0.12	0.30	0.31	0.32
意見・感想	0.29	0.57	0.54	0.45	0.75	0.42	0.22
全体	0.44	0.48	0.36	0.44	0.53	0.48	0.41

表 4.9: 質問応答システムの P_{top10}

	フィルタリング方式			加算方式			関連度のみ
	1 : 1	1 : 5.9	1 : 10	1 : 1	1 : 5.9	1 : 10	
比較	0.34	0.30	0.18	0.36	0.36	0.32	0.32
事実確認	0.16	0.12	0.12	0.22	0.14	0.20	0.18
ファクトイド	0.16	0.14	0.16	0.22	0.22	0.26	0.18
定義	0.16	0.34	0.16	0.24	0.26	0.30	0.26
方法	0.24	0.28	0.12	0.26	0.28	0.22	0.20
理由	0.18	0.20	0.16	0.08	0.22	0.16	0.24
意見・感想	0.24	0.38	0.24	0.36	0.44	0.24	0.22
全体	0.21	0.25	0.16	0.25	0.27	0.24	0.22

表 4.10: 素性集合別の質問応答システムの MRR

	フィルタリング方式				加算方式			
	ALL	$-\mathbf{f}^{\mathbf{a}}_{\textbf{cl-all}}$	$-\mathbf{f}^{\mathbf{q}}_{\textbf{end}}$	$-\mathbf{f}^{\mathbf{q}}_{\textbf{in}} - \mathbf{f}^{\mathbf{q}}_{\textbf{in3}}$	**ALL**	$-\mathbf{f}^{\mathbf{a}}_{\textbf{cl-all}}$	$-\mathbf{f}^{\mathbf{q}}_{\textbf{end}}$	$-\mathbf{f}^{\mathbf{q}}_{\textbf{in}} - \mathbf{f}^{\mathbf{q}}_{\textbf{in3}}$
比較	0.73	0.66	0.73	0.70	0.67	0.67	0.68	0.69
事実確認	0.55	0.70	0.55	0.45	0.81	0.81	0.81	0.81
ファクトイド	0.48	0.48	0.48	0.45	0.41	0.49	0.41	0.50
定義	0.67	0.60	0.67	0.60	0.54	0.60	0.55	0.70
方法	0.61	0.44	0.61	0.41	0.63	0.53	0.63	0.53
理由	0.25	0.34	0.40	0.30	0.43	0.38	0.44	0.27
意見・感想	0.60	0.62	0.59	0.57	0.90	0.77	0.90	1.00
全体	0.55	0.55	0.57	0.49	0.63	0.61	0.63	0.64

表 4.11: 素性集合別の質問応答システムの AP'

	フィルタリング方式				加算方式			
	ALL	$-\mathbf{f}^{\mathbf{a}}_{\textbf{cl-all}}$	$-\mathbf{f}^{\mathbf{q}}_{\textbf{end}}$	$-\mathbf{f}^{\mathbf{q}}_{\textbf{in}} - \mathbf{f}^{\mathbf{q}}_{\textbf{in3}}$	**ALL**	$-\mathbf{f}^{\mathbf{a}}_{\textbf{cl-all}}$	$-\mathbf{f}^{\mathbf{q}}_{\textbf{end}}$	$-\mathbf{f}^{\mathbf{q}}_{\textbf{in}} - \mathbf{f}^{\mathbf{q}}_{\textbf{in3}}$
比較	0.56	0.43	0.53	0.56	0.62	0.64	0.58	0.51
事実確認	0.44	0.49	0.44	0.45	0.65	0.68	0.65	0.69
ファクトイド	0.39	0.37	0.38	0.36	0.39	0.33	0.35	0.46
定義	0.60	0.49	0.59	0.50	0.48	0.50	0.49	0.53
方法	0.54	0.43	0.60	0.39	0.54	0.49	0.54	0.53
理由	0.27	0.32	0.36	0.33	0.30	0.29	0.33	0.27
意見・感想	0.57	0.60	0.53	0.51	0.75	0.65	0.76	0.77
全体	0.48	0.45	0.49	0.44	0.53	0.51	0.54	0.54

表 4.12: 素性集合別の質問応答システムの P_{top10}

	フィルタリング方式				加算方式			
	ALL	$-\mathbf{f}^{\mathbf{a}}_{\textbf{cl-all}}$	$-\mathbf{f}^{\mathbf{q}}_{\textbf{end}}$	$-\mathbf{f}^{\mathbf{q}}_{\textbf{in}} - \mathbf{f}^{\mathbf{q}}_{\textbf{in3}}$	ALL	$-\mathbf{f}^{\mathbf{a}}_{\textbf{cl-all}}$	$-\mathbf{f}^{\mathbf{q}}_{\textbf{end}}$	$-\mathbf{f}^{\mathbf{q}}_{\textbf{in}} - \mathbf{f}^{\mathbf{q}}_{\textbf{in3}}$
比較	0.30	0.26	0.24	0.20	0.36	0.24	0.28	0.22
事実確認	0.12	0.18	0.12	0.08	0.14	0.12	0.14	0.12
ファクトイド	0.14	0.12	0.12	0.08	0.22	0.20	0.22	0.20
定義	0.34	0.34	0.34	0.28	0.26	0.24	0.26	0.24
方法	0.28	0.26	0.24	0.22	0.28	0.24	0.28	0.26
理由	0.20	0.24	0.22	0.24	0.22	0.22	0.22	0.22
意見・感想	0.38	0.40	0.28	0.34	0.44	0.36	0.42	0.32
全体	0.25	0.26	0.22	0.21	0.27	0.23	0.26	0.23

第5章　結論

5.1　まとめ

本研究では、多様な質問に答えられる質問応答システムにおいて、質問と回答の内容の類似度及び両者の回答タイプの整合度を基に適切な回答を選択する手法を提案した。回答タイプの一致判定では、回答タイプをあらかじめ定義するのではなく、質問と回答の暗黙のタイプが一致しているかを判定する。また、回答タイプの一致を判定する分類器を機械学習する際の訓練データにおける正例と負例の比を決定する手法も提案した。さらに回答選択手法として、回答タイプが一致しないと判定した候補を取り除くフィルタリング方式と、内容の類似度と回答タイプの整合度のスコアの合計で回答候補を選択する加算方式の2つを提案した。

実験では、回答タイプ一致判定器及び質問応答システム全体の性能を評価した。回答タイプ一致判定器の評価実験の結果、質問から抽出する素性の中では疑問表現の素性が、回答から抽出する素性の中では付属語列や節末表現の素性が特に有効であることが分かった。

質問応答システムの評価実験では、提案手法により正例と負例の比を決定し、その比に従って正例と負例の数を決めた訓練データを用いたシステムが最も高い性能を示した。2つの回答選択手法を比較したところ、フィルタリング方式よりも加算方式の方が優れていた。また、2つの手法共にベースラインを上回る結果が得られたことから提案手法の有効性を示すことができた。

5.2　今後の課題

最後に今後の課題について述べる。まず、回答タイプの一致の判定精度を向上させるために、現在提案されている素性の組み合わせや新しい素性を探究することが挙げられる。また、回答候補の長さを調節することも課題として挙げられる。本研究では、検索されたウェブ文書をHTMLのタグによって分割し、分割されたテキストを回答候補としているが、実際に得られた回答候補の中には極端に文字数が多いものもあった。あまりに長すぎる回答は、ユーザに掲示するのに不適であり、また回答となる文以外からも素性が多く取り出される可能性があり、回答タイプ一致判定の妨げとなる。これを解決するために、長い回答は要約する、もしくは回答候補抽出のためのパラグラフ分割アルゴリズムを再考する必要がある。

回答タイプ一致判定器の評価実験の結果から、質問から抽出される素性の中で、特に回答タイプとの関連が深いと考えられる疑問表現が正解率の向上に大きく貢献することが分かった。本研究で用いた疑問表現は、代表的な語を人手で列挙したものである。しかし、素性として有効な疑問表現を全て列挙できているかについては疑問が残る。実際、質問文中に疑問表現が存在せず、「疑問表現なし」という素性が抽出された場合も多かった。回答タイプの一致判定に有効な疑問表現を自動的にかつ大量に獲得することができれば、回答タイプ一致判定の正解率が向上すると考えられる。

今回、回答タイプ一致判定器の訓練データとして使用した Yahoo!知恵袋の質問には、様々な表現が使われていて、中には質問としてふさわしくない文も含まれていた。また、質問と組になっているベストアンサーも質問者が選んで選択しているため、必ずしも回答タイプに合った回答が選ばれているとは限らない。本研究でも、疑問文を検出する簡単なパターンマッチにより、質問文・回答文としてふさわしい文を選択しているが、この処理だけではおそらく十分ではない。より正確な回答タイプの一致判定を行うために、あらかじめ訓練データとして適切な質問・回答を選別する必要があると考えられる。また、質問と回答ともに文章の形式が自由なため、文語的な表現だけでなく口語や敬語など様々な表現が多かった。一方、質問応答システムの知識源としてウェブを用いているが、ウェブ上にも口語的な言い回しが使われることがあるものの、多くのウェブページは文語的な表現が多い。回答タイプ一致判定器の素性として、文末表現、節末表現、付属語列などを用いているが、訓練データ (本研究では Yahoo!知恵袋) と質問応答システムの知識源 (本研究ではウェブ) とで記述スタイルが異なると、これらの素性が一致せず、回答タイプの一致判定の正解率を下げる要因となる。したがって、質問応答システムの知識源と似た記述スタイルのテキスト集合を訓練データとして用意するか、ドメイン適用の技術を利用することも検討すべきであろう。

謝辞

本研究を行うに当たり、終始変わらぬ御指導を賜りました、主指導教員である白井清昭准教授に感謝の意を表します。また日頃よりご助言を頂きました、島津明教授に感謝いたします。自然言語処理講座の皆様には、研究生活を通して多くの知識や示唆をいただきました。この場をお借りしてお礼申し上げます。

参考文献

[1] Palakorn Achananuparp, Christopher C. Yang, and Xin Chen. Using Negative Voting to Diversify Answers in Non-Factoid Question Answering. *CIKM'09 Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 1681–1684, 2009.

[2] Kyoung-Soo Han, Young-In Song, and Hae-Chang Rim. Probabilistic Model for Definitional Question Answering. *SIGIR'06*, pp. 212–219, 2006.

[3] 石下円香, 佐藤充, 森辰則. Web 文書を対象とした質問の型によらない質問応答手法. 人工知能学会論文誌 24 巻 4 号 B, pp. 339–350, 2009.

[4] 金井明, 佐藤充, 石下円花, 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 14 回年次大会発表論文集, pp. 1013–1016, 2008.

[5] Kanako Komiya, Yuji Abe, Hajime Morita, and Yoshiyuki Kotani. Question answering system using Q & A site corpus Query expansion and answer candidate evaluation. *SpringerPlus*, 2013.

[6] 水野淳太, 秋葉友良. 任意の回答を対象とする質問応答のための実世界質問の分析と回答タイプ判定法の検討. 言語処理学会 13 回年次大会発表論文集, pp. 1002–1005, 2007.

[7] Junta Mizuno, Tomoyosi Akiba, Atsushi Fujii, and Katunobu Itou. Non-factoid Question Answering Experiments at NTCIR-6: Towards Answer Type Detection for Real World Questions. *NTCIR-6 Workshop Meeting*, pp. 487–492, 2007.

[8] 諸岡心, 福本淳一. 非 Factoid 型質問に対応した質問応答システム. 言語処理学会 13 回年次大会, pp. 958–961, 2007.

[9] John Prager, Pablo Duboue, and Jennifer Chu-Carroll. Improving QA Accuracy by Question Inversion. *Proceedings of COLING-ACL*, pp. 1073–1080, 2006.

[10] 佐々木裕, 磯崎秀樹, 鈴木潤, 国領弘治, 平尾努, 賀沢秀人, 前田英作. SVM を用いた学習型質問応答システム SAIQA-II. 情報処理学会論文誌, Vol. 45, pp. 635–646, 2004.

[11] 佐々木裕. 総合学習による質問応答システムの新しい構成法〜CLQA に向けて. 情報処理学会研究報告 NL-93, pp. 123–130, 2004.

[12] 佐々木智, 藤井敦. 回答の根拠を掲示する意思決定支援型の質問応答システム. 言語処理学会 17 回年次大会発表論文集, pp. 252–255, 2011.

[13] 渋沢潮, 林貴宏, 尾内理紀夫. Why 型質問の回答文を WEB から自動抽出するシステムの開発と評価. 情報処理学会論文誌, Vol. 48, pp. 1512–1523, 2007.

[14] R. Soricut and E. Brill. Automatic Question Answering Using the Web:Beyond the Factoid. *Journal of Information Retrieval Special Issue on Web Information Retrieval*, Vol. 9, pp. 191–206, 2006.

付 録A　評価用質問セット

質問応答システムの評価に用いた35個の質問の一覧を表A.1に示す。

表 A.1: 提案システムの評価用質問セット

比較	事変と戦争の違いは何ですか? うどんとそばの違いは何ですか? CD-R と CD-RW の違いは何? 菓子パンと総菜パンの違いは? 野菜と果物の違いは何ですか?
事実確認	毛を抜くともっと濃くなるのは本当ですか? ドライアイスを湯に入れるとどうなりますか? 顔の脂は燃えますか? 眼球の内部は空洞ですか? ヒトデに性別はありますか?
ファクトイド	人の骨は何本ありますか? 木の寿命は何年くらいですか? 富士山の高さはどのくらいですか? 恐竜が生きていたのはいつごろですか? 太陽系で一番高い山はどこですか?
定義	六法全書の六法とはどういう意味ですか? オゾン層とはなんですか? 紫式部とはどんな人ですか? iPS 細胞とはなんですか? ジョン・フォン・ノイマンはどのようなことをした人物ですか?
方法	絹糸はどうやって作りますか? 絵の具はどうやって作りますか? 寝癖を防ぐにはどうすればいいですか? 天体の質量はどう量りますか? IQ はどうやって測りますか?
理由	どうしてあくびが出ますか? カラスはなぜ黒いのですか? どうして地震が起こるのですか? どうして血は赤いのですか? なぜオリンピックは4年に1回なのですか?
意見・感想	TPP についてどう思われますか? 安倍政権についてどう考えますか? 日本の景気はよくなったと思いますか? 日本はカジノを公認するべきですか? 世界で一番美しい景色を持つ国はどこだと思いますか?