Vol.2018-ICS-192 No.12
2018/7/7

ヘルプデスクの対応記録からのQAリストの半自動抽出

壹岐 太一[1,a)]　田嶋 隼平[1]　下沢 将啓[2]　比屋根 一雄[2]

概要：我々は対話システムに用いる一問一答型の QA リストの半自動抽出の手法を提案する．問い合わせ対話システムを構築する際に，FAQ などの元となる情報が十分ない場合も多く，QA リストの作成に多大な労力を要している．本研究では，実際の問い合わせへの対応記録から質問と回答をそれぞれ抜き出し，対話知識としてのふさわしさ・内容の類似性を考慮して QA リストを作成する技術を開発した．実際の IT ヘルプデスクの問い合わせ対応記録を用いて，QA リストの作成に本技術を適用した場合としなかった場合の比較についても報告する．

1. はじめに

我々は情報技術による，企業業務の支援を研究している．

近年，計算機は企業活動に浸透しており，多くの業務の遂行にあたり必要である．それに伴って，計算機の管理者と利用者の間では日々多量の質問と回答が行なわれている．利用者の個別状況を聞き出して回答することは計算機管理者にとって負担となり得る．また，管理者に質問することに抵抗を感じ，必要以上に利用者自身が調べてしまうことも業務の効率を低下させるおそれがある．

質問回答の自動化には，管理者への直接の問い合わせを減らしつつ，利用者が気軽に質問できるようにすることで，企業におけるこれらの問題を緩和する可能性がある．

実世界の質問回答タスクにおいては利用者の個別状況の聞き出しが重要であるため，チャットボットや音声対話システムなどの会話型のインターフェースが有利であると考えられる．実際に Kiyota らの自動質問応答システム [1] など，昔から研究が続けられている．会話型インターフェース自体は深層学習の近年の技術発展もあり，次第に水準が高まっている．

一方で，質問応答システムは質問回答のための知識とインターフェースの両者が揃って，はじめて効果的に活用できるようになるものだが，少なくとも企業の業務支援の文脈に関しては，知識に関する技術発展がインターフェースの発展に比べると十分ではないと考えられる．

非タスク指向の対話を対象とした対話知識の構築手法には，twitter をコーパスとした稲葉ら [2]，wikipedia を利用した太田ら [3]，杉本ら [4] らの研究がある．これらの web 上の大規模言語資源を用いる手法は，個人が一般的に話す内容や常識など普遍的な対話知識の構築には有効であると考えられる．しかし，企業という特殊環境で有用な対話知識を構築するためには，その環境に固有の情報が不可欠である．固有情報は必ずしも大量に用意できるものではないため，新しい知識の構築法が必要となる．

以上の背景から，我々は企業の既存の文章を用いて低コストで対話知識を構築する方法を検討している．本稿では，企業のヘルプデスク対応報告を対象として，抽出型の要約をベースとする対話知識抽出の適用方法を提示し，対話知識の抽出によるチャットボットの回答性能向上の可能性について報告する．

2. 問題の定式化

2.1 知識抽出の対象 — ヘルプデスク対応報告

本研究では対象のログとしてヘルプデスクの対応報告を扱う．ヘルプデスクの対応報告は，**図 1** に示す具体例[*1]のようにメモ書き風の自由形式であるが，問い合わせ内容と対応内容 (質問と回答) は比較的容易に分離可能である．

従って問題は，与えられた質問と回答それぞれの元文に関して本質的な部分を抽出する問題であると捉えることができる．直感的には，問い合わせ内容と対応内容の本質的な部分の決定には，相互情報を考慮する必要があると思われるが，本稿の範囲では，単純化を行い，問い合わせ内容と対応内容のそれぞれについての本質的な部分の抽出と考える．単純化によって，ログからの知識抽出の問題は問い合わせ内容と対応内容の要約に還元される．

QA 知識の抽出を要約問題として捉えた上での，本研究

[1] 株式会社 Nextremer　Nextremer Co., Ltd.
[2] 株式会社三菱総合研究所　Mitsubishi Research Institute, Inc.
[a)] taichi.iki@nextremer.com

[*1] 本稿の具体例は企業の情報を含むために ⟨place⟩ など適宜置換している．実際には情報がそのまま出力される．

【問合せ内容】メーリングリストの管理者を変更したいのだが、どうすればいいのか教えてほしい。【対応内容】<place>で、現行管理者が新しい管理者の社員番号に変更して頂くよう回答しました。

【問い合わせ内容】着任されたばかりの派遣社員の方のメールの区分変更を実施し、本日<num>時以降，標準メールアドレスが利用できると通知を受領したが、メール区分変更後に何か作業をする必要があるか。利用する PC の OS は<product>とのこと。【対応内容】（一次対応）<time><product>は標準メールアドレスで認証しなおしていただく必要があるので、別途メールにて必要な作業を記載した資料を送付するのでそちらを確認いただき作業を実施いただくよう、説明（二次対応）<time>下記メールを<person>様宛に送付----------先ほどお問い合わせいただきました、メールアドレス区分変更後の対応につきまして下記 URL を参考にしていただき、作業を実施いただきますよう、よろしくお願いいたします。■ FAQ：メールアドレス区分変更後の対応を知りたい_<product><url>----------※作業を実施してみますとのこと

【問合せ内容】<product>の接続をしたが、エラーになる【対応内容】現在サーバにて認証等に時間がかかってエラーになっている可能性があるため、時間を空けて再度お試しくださいと、ご案内いたしました。

図 1 本研究が対象とするヘルプデスク対応報告の例

で目標とする出力を，図 1 の対応報告から抽出されるべき知識として**図 2** に例示する．

2.2 本質度の導入と本質度を用いた QA 知識の抽出

文の要約タスクは一般に，要約対象の文とどれくらいの長さに要約したいかが与えられたとき，出来るだけ要約対象の文が持つ意味が損なわれないように文を圧縮するタスクである．従って，抽出型要約では文間に何らかの類似度を定義し，文字長制約の下で要約前後での類似度の変化が出来るだけ少なくなるように抽出することが多い．一方で，QA 知識の抽出においてはチャットボット運用時のユーザ発話と QA 知識のマッチングがより正確に行われるよう，含まれる知識の要点が明らかとなるように圧縮することが望ましいと考えられる．

このような QA 知識抽出の特殊な条件に対応するため，我々は教師あり学習による抽出型要約を用いる．対応報告の各文を分割して得られるそれぞれの部分文字列に対して，問合せ内容あるいは対応内容として本質的である度合，すなわち，本質度が定まると仮定する．本質度の推定モデルは一定量の教師データを人手でアノテーションし教師あり学習で構築する．知識の抽出時には学習済みモデルで本質度を推定し，本質度が高い知識から優先的に抽出することで，要点のみを含んだ QA 知識を作り出す．

本研究での部分文字列への本質度の付与は，問合せ内容と対応内容で共通に主旨と条件の 2 観点から行う (それぞれの部分文字列に対して 2 つの本質度を推定する)．観点を複数設ける狙いは (1) 一つ目は推定モデルの過学習の抑制，(2)QA 知識とユーザ発話のマッチングでのメタ情報としての利用，の 2 つである．

Q: メーリングリストの管理者を変更したいのだが
A: ⟨place⟩ で, 現行管理者が新しい管理者の社員番号に変更してください

Q: 着任されたばかりの派遣社員の方のメールの区分変更を実施し, 標準メールアドレスが利用できると通知を受領したが, メール区分変更後に何か作業をする必要があるか, 利用する PC の OS は ⟨product⟩ とのこと
A: ⟨product⟩ は標準メールアドレスで認証しなおしていただく必要があるので, FAQ：メールアドレス区分変更後の対応を知りたい_⟨product⟩⟨url⟩

Q: ⟨url⟩ の接続をしたが, エラーになる
A: 現在サーバにて認証等に時間がかかってエラーになっている可能性があるため, 時間を空けて再度お試しください

図 2 抽出型の要約で図 1 の対応報告から抽出されるべき QA 知識

- **主旨**: やり取りの中で最も伝えたい部分．表層的には，問い合わせの場合は「〜してほしい」「〜ができない」など，対応報告の場合は「〜を案内」「〜と伝えました」などに現れる．
- **条件**: 主旨を修飾する条件の部分．表層的には「〜してから (, ・・・になった)」「今日は」「外出先だが」「〜ですので (, ・・・)」などに現れる．

なお，主旨と条件は日本語文法上の主節と従属節の関係に類似しているが，本研究ではあくまでも QA 知識としての主旨，条件の観点からアノテーションするものであり，必ずしも一致するものではないと考えられる．

3. 関連研究

自然文からの発話知識の抽出:

日本語の文から発話知識を抽出する試みは，twitter の文を用いる稲葉らの提案 [2] や wikipedia を用いる太田らの手法 [3]，杉本らの手法 [4] など，非タスク指向対話の対話知識を大規模な言語資源から効率的に入手することを目的に多数行われている．

このうち杉本らの手法では，はじめに記事から各文を取り出し，発話らしくするために語尾等を加工した後，完成文が実際に発話知識として使えるか否かをスコアリングし，閾値以上の場合は発話知識として採用する流れが取られる．なお，スコアは wikipedia 記事内での元文の位置情報を回帰することで算出する．この流れでは，最終段階でスコアリングするため，内容が適切か，加工結果が日本語として自然かの少なくとも 2 つの観点でスコアが決定し，前者は対象データのみに依存するが，後者は加工規則にも依存してしまう．そのため，加工規則の調整の度にスコアリングの学習データが必要になる可能性が高いと考えられる．

抽出による文の要約:

文の要約は代表的な自然言語処理のタスクであり，元文から文をコピーしてきて要約する抽出型 (extractive) と元文にはない語彙も用いて要約を作る生成型 (abstractive) の２種類に分かれる．

本研究で用いるモデルは日本語文を文字の系列として与えて本質度を出力する抽出型要約に属するが，近年では，ニューラル機械翻訳のモデルを応用した生成型の要約も研究されている [5][6]．深層学習を抽出型要約に用いた先行事例には，Nallapati らの Recurrent Neural Network(RNN) を用いた抽出型要約 [7] がある．彼らの提案では，単語列から文，文の列から文章の２段階の RNN を用いて，文が要約に含まれるか否かを分類する構造が用いられた．

抽出ベースの QA システムとの比較:

近年，深層学習を用いた抽出ベースの質問回答が活発に研究されている[*2]．抽出ベースの質問回答は自然文集合から直接，回答を抽出する．深層学習によって，もととなる自然文と質問，回答の教師あり学習から質問に対する回答箇所のパターンがモデリングされる．Chen らによるとパラグラフレベルの抽出ベース質問回答と関係パラグラフの検索を組み合わせて，Wikipedia を情報源とする質問回答システムを構築することもできる [8]．

原理的には，対応報告に基づく QA でもこの手法は適用可能であると考えられる．ただし，対応報告の記述スタイルは百科事典とは異なるため，パターンの再学習が必要となると予想される．通常，抽出ベースの質問回答の学習は数万件の自然文と質問，回答で行われ適用は容易ではない．

また，ボットの実運用では発話内容の制御性も重要になる．この手法は自然文のどこでも回答する可能性があり，発話の細かな調整は難しい．

4. 提案手法 – ニューラル要約型 QA 知識抽出

我々の提案手法の概要を**図 3** に示す．提案手法は対応報告を入力とし，分割，部分文字列の必要度の推定，部分文字列の抽出，文字列の加工の 4 段階を通してチャットボット用の QA 知識を抽出する．対応報告の問い合わせ内容と対応内容は分離された状態で手法に入力されるものとする．

先行研究を踏まえ，知識の事前抽出で制御性を確保し，学習が必要なモデルをできるだけデータの加工前に置くよう考慮した．また，部分文字列の本質度推定モデルには文字レベルの RNN を用いて，データが多量に用意できない状況での頑健性向上を狙う．

4.1 問い合わせ内容と対応内容の部分文字列への分割

提案手法では，はじめに，問い合わせ内容と対応内容それぞれを句読点，疑問符の位置で区切り部分文字列に分割する．問い合わせ内容と対応内容それぞれを部分文字列の単位で分割する理由は，それぞれを事前に分析した結果，報告の記述は多数の内容が盛り込まれた長い文で構成されることが多く，文単位よりも細かな粒度での抽出が適していると考えられたからである．

[*2] https://rajpurkar.github.io/SQuAD-explorer/ 代表的なデータセットである SQuAD での最新のランキングが見られる

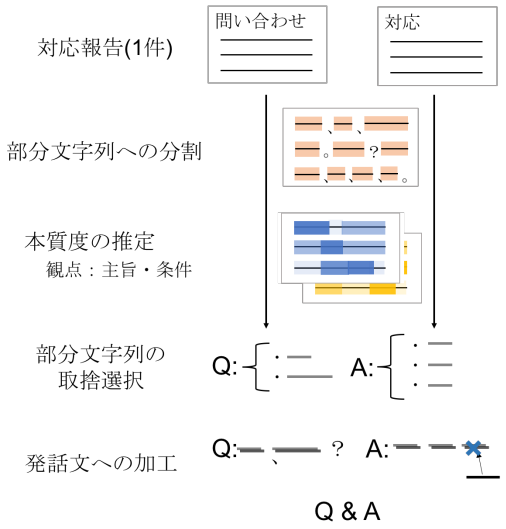

図 3 提案手法の処理概要

表 1 必要度推定モデルのニューラルネットワーク構造

層	構造	出力次元	活性化関数	備考
入力	文字列	—	—	文字単位
1	文字埋め込み	256	—	—
2	LSTM	1024	tanh	dropout
3- 6	Highway	1024	tanh	dropout
7	全結合	2	sigmoid	dropout

* dropout は層の入力前，ratio=0.5

4.2 部分文字列の本質度の推定

分割したそれぞれの部分文字列に対して本質度を推定する．本研究ではニューラルネットワークを教師あり学習させて本質度推定のモデリングを行う．

4.2.1 ニューラルネットワークの構造

ニューラルネットワークは文字列を入力とし，2 つの実数を出力する．文字列は文字単位で埋め込みベクトルに変換し，1 層の LSTM でエンコードを行う．LSTM にすべての文字列を入力したときの LSTM の隠れ層を，4 層の Highway Networks[9] に入力し，全結合層で 2 つの実数値に変換する (**図 4**)．各層の詳細は**表 1** にまとめて示す．

なお，文字埋め込みは日本語版 wikipedia の全記事ダンプの本文を対象とし，gensim ライブラリ [10] に実装されている Word2vec[11][12] を用いて事前に学習させた．その際の条件は skip-gram 手法，ベクトル次元 256 次元，窓幅 4，頻度 100 回以上の文字を使用，特定文字の除去は行わない，とした．獲得した文字の埋め込みベクトルは任意の２ベクトル間の cos 類似度が保存するように，ベクトルの l^2-norm の標準偏差による規格化のみを施した．

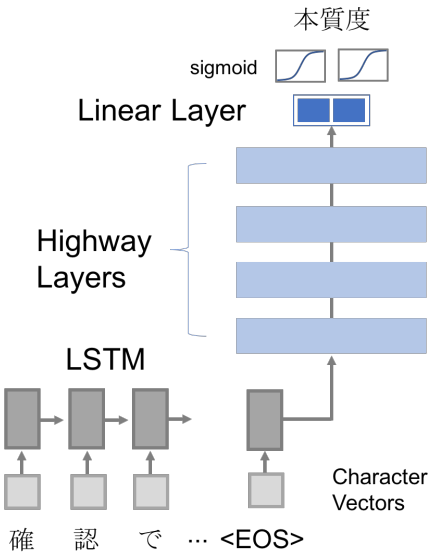

図 4　本質度の推定モデルの構造図

4.3　本質度に基づいた部分文字列の取捨選択

それぞれの部分文字列に対して必要度を推定したのちに，知識として抽出するかしないかの決定を行う．

現状では，貪欲に抽出するように設計してある．1 件の業務報告に対して，質問と回答それぞれに文字数の上限を設け，必要度が高い順に上限を超えない限りで抽出する．文字数の上限は質問，回答ともに 100 文字と設定する．

4.4　選択した部分文字列の発話文への加工

抽出を終えた文は，質問あるいは，ボットの回答発話として適切になるよう加工を行う．

質問は，抽出結果がもともと質問文に近い形をしていることが多いため，抽出した部分文字列に対する特別な加工は行わず，複数の部分を読点を挟んで連結する．ただし，知識管理者が定めるタグ語彙[*3]が回答には含まれ，質問には含まれない場合には，先頭に「〜について,」と追加する．

回答は，複数の部分を読点を挟んで連結したのち，語尾を正規表現でマッチングし，できるだけ案内の表現に近づくよう置換を行う．なお，置換には次のような正規表現を用いる．

```
ご?(?:確認 | かくにん)(?:頂 | いただ)
 (?:く | ける)(?:様 | よう | 旨 | むね) を?ご?
 (?:回答 | 案内 | 説明 | 連絡 | 解答)(?:致 | いた)?
 (?:済 | 済み | ずみ | ズミ
  | した | しました | し | して | しまして)?$
→ ご確認ください
```

[*3] 見出し語とその同義語を列挙して定義する．例えば，「インターン: intern，インターン，インターンシップ」のような定義を行う．

5.　実験の設定

我々は提案手法を，部分文字列の必要度推定モデル単体，手法全体で得られる知識に関して評価した．

必要度推定モデル単体の評価では，アノテーションした教師データ (詳細は後の節で示す) のうち 8 割を学習用データとし残り 2 割を評価用データとして用いることで，正解率と混同行列を作成した．ただし，必要度推定モデルの正解率の計算は，モデルの推定スコアと評価用データのスコアを閾値 0.5 で必要・不要に分け，二値分類として行った．

手法全体の評価には，抽出知識に基づくチャットボットを作成し，事前に用意した想定質問と模範回答に関する正答率を用いた．QA 知識の抽出元には必要度推定モデルの学習と手法の調整に使用したのとは異なる期間のメール・メーリングリスト関係とタグ付けされた対応報告を使用した．比較対象として，対応報告を抽出せず直接に知識とした場合も作成した．

5.1　部分文字列の必要度推定モデルの学習

5.1.1　教師データ

対応報告の問い合わせ内容と対応内容を提案手法の章で説明した手法で分割し，それぞれの部分文字列を著者の一人がアノテーションした．100 対応報告を分割して得られた 1916 件の部分文字列に関して，主旨らしさ，条件らしさの 2 観点からそれぞれ 4 段階評価した．

5.1.2　ニューラルネットワークのパラメータ最適化

ニューラルネットワークのロス L は観点ごとに教師データとの sigmoid cross entropy をとった．ただし，4 段階評価の結果を取り入れて学習を行うために，評価を悪い方から 0, 0.33, 0.66, 1.00 の確率値と解釈し，この値に近づくよう学習させている．具体的には，入力 i に関するネットワークの推論結果を $\hat{p}_i$，教師データの確率値を p_i として，次式でロスを定義した:

$$L = \sum_i \{p_i \log \hat{p}_i + (1 - p_i) \log(1 - \hat{p}_i)\} \qquad (1)$$

このようなロスを用いた学習は，Hinton らによってネットワークの蒸溜の文脈で soft targets による正則化として深層学習に適用され，汎化性能の向上が報告されている [13]．

学習の回数は 1 エポックをすべての学習データを一巡する間隔として，25 エポック行った．パラメータの最適化には SGD を用い，10 エポックごとに学習率を 1.0 から半減させた．ミニバッチの大きさは 64 を用いた．Soft targets 以外の正則化として Gradient clipping(norm=1), Weight decay(rate=1e-6), Dropout(ratio=0.5), 文字の未知化 (ratio=0.2) を行った．He らの標準化によって初期化した，初期パラメータの異なる，同一構造の 3 つのニューラルネットワークを同じデータで学習させ，アンサンブル

表 2　必要度推定モデルの評価用データに関する混同行列

(主旨)	Prediction +	Prediction −	合計
Actual +	56	38	94
Actual −	53	236	289
合計	109	274	383

(条件)	Prediction +	Prediction −	合計
Actual +	74	44	118
Actual −	47	218	265
合計	121	262	383

平均を推論時のスコアとした．なお，ネットワークの実装とパラメータの最適化には chainer[14] を用いた．

5.2　評価に用いたチャットボット

評価に用いたチャットボットは，ユーザから発話を受け取ると，投入されているすべての知識の質問とユーザ発話の間の類似度を計算し，類似度が最も大きい知識の回答を発話として返答するものとした．類似度には知識の質問とユーザ発話のそれぞれを文字の集合とみなし，集合に関する Jaccard の類似指標を用いた．

6. 結果と考察

6.1　部分文字列の必要度の推定モデル

部分文字列の必要度の推定モデルの評価用データに関する混同行列を**表 2** に示す．

必要と不要それぞれを推定しており，学習ができていることが確認できる．また，正解率，適合率，再現率は，主旨に関してそれぞれ 0.76，0.51，0.59，条件に関してそれぞれ 0.76，0.61，0.62 であり，主旨と条件の正解率に関して極端な差はなく，同程度に学習できていることがわかる．しかし，両者とも適合率，再現率はそれほど高くない．

傾向をさらに詳しく知るために，実際の部分文字列に対してスコアリングを行った．**表 3** に 3 つの対応報告から分割された部分文字列への必要度の推定結果を示す．

「どうすればいいのか」「お問合せ後」と言った不要な部分には低いスコアが付与できていることがわかる．また，主旨に関しては「案内しました」「回答しました」などよく出現する言い回しは認識できている．しかし，主旨と条件のどちらが高くなるかはラベルと異なる場合が多い．どちらかが閾値を上回っているので抽出結果には影響はないものの，まだ改良の必要性があると思われる．改良には次のような方向性が考えられる．

(1) **教師データを増やす・アノテータを増やす**: 入力が文字レベルであるとはいえ，約 2000 文では少ない可能性が高い．また，アノテータを増やすことで，より細かく教師データのスコアを刻むことができるようになる．特に一人でアノテーションしているため，スコアの値が 0 または 3 に固定化されやすい．

(2) **評価の観点の曖昧性の解消**: 例えば，第 2 例の問合せの第 2 部分は今回は主旨としての本質度が高い部分としてアノテーションされているが，表層的に判断すれば第 3 部分の条件として捉えることも可能であり，評価の観点には曖昧性が残されている．この曖昧性を解消するためには，アノテーション規則の精密化や評価の観点の再設定などさらに踏み込んだ議論が必要である．

(3) **1 件の対応報告全体を同時に入力して，それぞれの部分文字列をスコアリングする**: 現実的には，問い合わせと対応の両者の相互作用で，ある部分の必要性は変化することが予想される．したがって，相互の情報を考慮して部分文字列の重要度を算出することが望ましいと考えられる．

6.2　抽出 QA 知識に基づいたチャットボット

メール・メーリングリスト関係とタグ付けされた 438 件の対応報告に対して知識抽出を行った結果，421 件の知識が抽出された．抽出した QA 知識，対応報告を問い合わせ内容と対応内容に切り分けただけの一覧，それぞれに基づいたチャットボットを作成して 562 件の想定質問を入力，チャットボットの回答を想定回答に対して**表 4** に示す基準で，著者のうち一人が評価した．結果は，我々の提案手法の方が僅かながら，高い正解率を示すものとなった (**表 5**)．

以降では，**表 6** を参照しながら具体例を検討していきたい．表ではチャットボットに入力した質問，それにマッチした知識の質問 (Q) 及び対応する回答 (A) が，提案手法，対応報告からの直接の使用ごとに示されている．

- **どちらも的確な回答**: ID1，2 のように，提案手法と直接作成の両者とも元とする対応報告は同一であることがほとんどであった．提案手法では，不要語句が取りのぞかれ，言い回しが加工されているため自然な回答となる場合が多かった．
- **どちらも不的確な回答**: チャットボットが Jaccard 類似指標を用いた単純な知識マッチングを行っているため，ID3 のように質問が短い対応報告に引きづられてしまう例がしばしば見られた．知識が短いことと抽出すべきか否かには必ずしも関連性はないため，抽出の工夫だけでこの問題に対応することは限界がある．より正確な知識のマッチングにはチャットボット側の改良が欠かせないと考えられる．
- **提案手法の方が的確な回答**: ID4，5 の質問は提案手法のみが正解できた回答の例である．それぞれ，直接作成の対応する知識では質問に「どうすればいいのか。[受入れ担当からの問い合わせ]」，「[受入れ担当からの問い合わせ]」が含まれていた．提案手法では，これらの不要な部分が除去されたため，該当知識の相対的な順位が上がり，望ましい回答が得られるようになった

表 3　必要度の推定例

		主旨		条件	
タイプ	部分文字列	label	推定	label	推定
問合せ	気づかずに	0/3	0.10	0/3	0.41
	いつの間にか ⟨product⟩ が「⟨product⟩」になってしまった	**3/3**	**0.40**	**0/3**	**0.71**
	どうすればいいのか	0/3	0.32	0/3	0.18
対応	**「⟨product⟩」はアンインストールし**	**3/3**	**0.50**	**0/3**	**0.45**
	「⟨product⟩」をインストールして頂くよう回答	**3/3**	**0.95**	**0/3**	**0.02**
	※お問合せ後	0/3	0.07	0/3	0.02
	「⟨product⟩」のインストーラーを ⟨place⟩ でお送りしました	**0/3**	**0.68**	**3/3**	**0.39**
問合せ	**現在社外にいるのだが**	**0/3**	**0.08**	**3/3**	**0.79**
	⟨product⟩ がほとんど繋がらない（全く繋がらないわけではない）状態なのだが	**3/3**	**0.65**	**0/3**	**0.23**
	何か問題でも発生しているのか	0/3	0.30	0/3	0.23
対応	お問合せ後	0/3	0.08	0/3	0.01
	基盤担当者（⟨person⟩）に状況を確認したところ	0/3	0.21	2/3	0.44
	同時間帯に一斉にアクセスが集中したことによる現象とのご連絡があったので	**2/3**	**0.32**	**0/3**	**0.67**
	ユーザーには状況をご説明し	**0/3**	**0.71**	**0/3**	**0.08**
	しばらく時間をおいてから接続を試して頂くよう案内しました	**3/3**	**0.85**	**0/3**	**0.11**

＊ ボールドはいずれかの推定値が0.5以上のため抽出される部分文字列

表 4　チャットボットの回答の評価基準

評価	説明
5	内容も文面も合っている.
4	内容はあっているが，回答の文面になっていない .
3	正しい回答内容を一部含んでいるが，余計な文章がある.
2	回答内容は違うがグループは同じ.
1	回答内容が全く違う.

表 5　想定質問に対するチャットボットの回答の正解率

	3 以上	4 以上
ログ直接使用	23.49 %	15.66 %
抽出結果使用	28.47 %	24.56 %

ものと推測される.

- **提案手法の方が不的確な回答**: 抽出が望ましくない影響を与える場合も見られた．ID6 では概要説明へのリンクのみが残り，質問に対して重要な部分が欠けてしまっている．しかし，両者はメーリングリストに関する他の質問 (「メーリングリスト名を編集したい」など) に対しても同じ知識で回答しており，抽出した文のほうが一般的であるためより正確に回答できていると言える場合もあった．知識を事前に作る形式でこの事例に対応するためには，一つの対応報告から，その報告の主題とは関係ないが有用である QA 知識も抽出できる必要がある.

7. おわりに

本研究ではヘルプデスク対応報告からの QA 知識抽出を手法を提案した．QA 知識の抽出を質問と回答それぞれに関する要約の一種とみなし，抽出時の部分文字列の必要度推定をニューラルネットワークの教師あり学習で行う.

提案手法で作成した QA 知識に基づくチャットボットは，ヘルプデスク対応報告を抽出せずに直接用いるものより高評価となり，要約型知識抽出が，ログからのチャットボットの QA 知識作成において有効である可能性を示した.

本稿の範囲では，質問と回答をそれぞれに関する要約問題として定式化したが，実際には質問と回答の相互情報を考慮して，必要度の推定を行うべきであると考えられる．そこで，今後の方向性として，質問と回答の相互情報を含めた必要度推定のモデリングを行うことが考えられる.

また，知識の抽出の工夫だけでは問い合わせに対して有用な情報にアクセスするのは難しい場合が明らかとなったため，問い合わせと知識のマッチングやマルチターン対話へと検討の幅を広げ，抽出された知識へのアクセス性の向上を目指していきたい.

参考文献

[1] Kiyota, Y., Kurohashi, S., and Kido, F.: *Dialog navigator: A question answering system based on large text knowledge base*, Proceedings of the 19th international conference on Computational linguistics, vol. 1, pp. 1-7 (2002).

[2] 稲葉通将, 神園彩香, 高橋健一: Twitter を用いた非タスク指向型対話システムのための発話候補文獲得, 人工知能学会論文誌, vol. 29, no. 1, pp. 21 (2014).

[3] 太田知宏, 島海不二夫, 石井健太郎: 発話生成を目的とした Wikipedia からの文抽出, 人工知能学会, vol. 23, pp. 1-4 (2009).

[4] 杉本俊, 植木拓, 林宏幸, ニコルズエリック, 中野 幹生: 特定ドメイン雑談対話システムのための Wikipedia を用いた発話文の生成, 人工知能学会言語・音声理解と対話処理研究会, vol. 81, pp. 98-99 (2017).

[5] Rush, A. M., Chopra, S., and Weston, J.: *A Neural Attention Model for Abstractive Sentence Summarization*, Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 379-389 (2015).

[6] Chopra, S., Auli, M., and Rush, A. M.: *Abstractive*

表 6 想定質問とチャットボットの回答例

ID	質問	提案手法	直接
		どちらも的確な回答	
1	メール出来ない	Q: ⟨product⟩ でメールの送受信ができない（メールの切り替えは行なっていない） A: 自分宛にテストメールを送信して, 受信できるか確かめてください	Q: ⟨product⟩ でメールの送受信ができない（メールの切り替えは行なっていない） A: まずは, 自分宛にテストメールを送信して, 受信できるか確かめてください
2	メーリングリストは作成後, どのくらいで使用できるか	Q: 新規にメーリングリストを作成した場合, 実際に使えるまでにどれくらいの時間がかかるのか教えてほしい A: ⟨num⟩ 分〜⟨num⟩ 分ほどお待ちください	Q: 新規にメーリングリストを作成した場合, 実際に使えるまでにどれくらいの時間がかかるのか教えてほしい。 A: ⟨num⟩ 分〜⟨num⟩ 分ほどお待ち頂くようお伝えしました。
		どちらも不的確な回答	
3	メールについて知りたい	Q: ⟨person⟩ のメール移行について確認したい A: そちらに確認してください	Q: ⟨person⟩ のメール移行について確認したい A: 役員対応は ⟨person⟩ が担当されているので, そちらに確認してください。
		提案手法の方が的確な回答	
4	インターンにメールを使用させたい	Q: インターン用にメールの設定をしたいのだが A: インターンの方は ⟨product⟩ を使って頂くよう回答, 受入ガイドに記載があるので, そちらも確認してください	Q: sso パスワードに関して，インターンにメールを利用させるため，⟨product⟩ にアクセスしたが，認証が通らない。[受入れ担当からの問い合わせ] A: SSO パスワードが初期状態だったので, いったん変更して頂くよう案内しました。→その後，⟨ 省略 ⟩
5	インターンはメールアドレスを持っていますか	Q: インターンにメールを使わせようと思い, ⟨product⟩ のログイン画面を表示させたのだが, そもそもインターンのメールアドレスがわからない A: ⟨place⟩ でインターンのアドレスを確認してください	Q: ローカルメールアドレスを通常メールアドレスにする申請方法を教えてほしい A: 申請の手順をご案内いたしました。
		提案手法の方が不的確な回答	
6	誰でもメーリングリストを作成できますか？	Q: メーリングリストを作成したいのですが, 派遣社員でも申請できますか A: 詳細につきましては下記【メーリングリストサービス概要】をご確認ください, 【メーリングリストサービス概要】⟨url⟩	Q: メーリングリストを作成したいのですが, 派遣社員でも申請できますか？ A: 申し訳ございませんが, メーリングリストの作成は, 社員 ⟨ 省略 ⟩ のみが可能です。お手数ですが, 該当者にメーリングリストの作成をご依頼いただけますようお願いいたします。詳細につきましては下記【メーリングリストサービス概要】をご確認ください。【メーリングリストサービス概要】⟨url⟩

sentence summarization with attentive recurrent neural networks, Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 93-98 (2016).

[7] Nallapati, R., Zhai, F., and Zhou, B.: *SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents*, Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence(AAAI), pp. 3075-3081 (2017).

[8] Chen, D. and Fisch, A. and Weston, J. and Bordes, A.: *Reading Wikipedia to Answer Open-Domain Questions*, arXiv preprint arXiv:1704.00051 (2017).

[9] Srivastava, R. K., Greff, K., and Schmidhuber, J.: *Highway networks*, arXiv preprint arXiv:1505.00387 (2015).

[10] Rehurek, R., and Sojka, P.: *Software Framework for Topic Modelling with Large Corpora*, Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pp. 45-50 (2010).

[11] Mikolov, T., Chen, K., Corrado, G., and Dean, J.: *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781 (2013).

[12] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J.: *Distributed representations of words and phrases and their compositionality*, In Advances in neural information processing systems pp. 3111-3119 (2013).

[13] Hinton, G., and Vinyals, O., and Dean, J.: *Distilling the knowledge in a neural network*, arXiv preprint arXiv:1503.02531 (2015).

[14] Tokui, S., Oono, K. Hido, S., and Clayton, J.: *Chainer:*

a Next-Generation Open Source Framework for Deep Learning, Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS), vol. 5 (2015).