

# ランキングアルゴリズムを用いた質問応答システムの試作

岩間 雄太<sup>1,a)</sup> 伊藤 孝行<sup>2,b)</sup>

概要：企業サービスの多様化によりサポートセンターへのユーザからの問い合わせが増加している。現在、サポートセンターではユーザから問い合わせに人手で対応しているが、ユーザ対応には多くの時間がかかり、迅速に、また、正確に対応することは困難である。そのため、サポートセンターが人手で行う作業の自動化が求められている。以上のような背景のもと本論文では、ユーザの問い合わせに対して適切な回答を提案するシステムを試作する。試作にあたり、問い合わせの回答となるドキュメントを検索、スコアリングをしユーザに回答として提供している。スコアリングはランキングアルゴリズムを使用することにより実装した。最後に本手法の効果を評価し確認する。

## 1. はじめに

企業にとって、サポートセンターの重要性が高まっている。サポートセンターとはユーザの問題に対して、適切な解決策を提案し、問題を解決に導きユーザ満足度を高めるものである。したがって、企業ではサポートセンターを用意し、商品やサービスの問い合わせに対するサポートを充実させユーザの満足度を高めようとしている。また、近年ではIT技術の発達により企業は多種多様な商品、サービスを提供するようになった。サービスの増加に伴い、サポートセンターへの問い合わせが増加し、内容も多岐に渡りようになってきている。上記の対応には、サポートセンターの担当者はすべて人手で対応してきたが、サービスの多様化に伴う問題によって、サポートセンターへの問い合わせが増加し、人手と対応にかかる時間が今までより求められるようになった。

サポートセンターでは、ユーザの問い合わせがあると過去のユーザとのやりとりやWebサイトのサポートページに書いてある「よくある質問」を参考に解決策を提示している。また、サポートセンターへの問い合わせの大半が「よくある質問」などで対応可能な質問である。過去の問い合わせ履歴のからの検索、または「よくある質問」などで対応可能な質問に対して多くの人手と時間をさくことは企業にとって負担になる。

また、ユーザについてもサービスの多様化にともない、サービスが複雑化するという問題も存在する。サービス

の多様化によりWebサイトのサポートページが膨大になり、ユーザが必要としている情報の場所がわかりづらいという点である。Webサイトがわかりづらくユーザがサービスに不満を持ったままではユーザの満足度は下がり、ユーザがサービス、商品を使わなくなるという問題点がある。

上記のようなサービスの複雑化や多様化といった問題に対応するために、サポートセンターが手動で行っていた過去の問い合わせ履歴のからの検索、「よくある質問」で対応可能な問題の回答などを自動でおこない、ユーザの必要な情報が一目でわかるインターフェイスを持ったアプリケーションが求められている。

そこで、本論文ではある固有のドメインに関するユーザからの質問に対して過去のユーザ対応や「よくある質問」からユーザが求めている解決案の候補とドキュメントを探し出し、探し出したドキュメントに対してランキングアルゴリズムを用いてスコアリングを行い、スコアリング結果に基づいて結果をユーザに提案するアプリケーションの試作について述べる。

以下に本論文の構成をのべる。1章で研究背景を述べる。2章でシステムの実装、機能について述べる。3章では、使用するデータや検索手法、スコアリングアルゴリズムを述べる。4章ではシステムの性能評価をし、既存システムに比べた有用性を述べる。5章では本論文のまとめと今後の課題を述べていく。

## 2. 質問応答システム

### 2.1 システムの実装

本システムの実装について説明する。本システムは、webアプリケーションであり、webサーバ上のサーバプログラ

<sup>1</sup> 名古屋工業大学工学部情報工学科

<sup>2</sup> 名古屋工業大学大学院産業戦略工学専攻/情報工学科

a) iwama.yuta@itolab.nitech.ac.jp

b) ito.takayuki@nitech.ac.jp



図 1 システム検索画面インターフェイス

ムとブラウザ上のクライアントプログラムから構成される。サーバサイドは Ruby on Rails(以下,Rails) により開発した。一般的な web アプリケーションでは、クライアントサイドとサーバサイド間でデータを通信するために多くのページ遷移必要となる。つまり、ユーザの操作が発生した後、ユーザによって作成されたデータ等の操作結果をサーバサイドに送信する際に、ページ遷移が発生するプロセスが採用されているということである。Rails は本プロセスを実現するために Model View Controller(以下, MVC) をデザインパターンとして採用している。しかし、近年では Ajax 技術の普及により、ユーザの操作がページ遷移を必要とせずにダイナミックに反映される UI が求められている。そこで Ajax 技術を使いダイナミックな UI を実現した。本システムのクライアントサイドで Ajax 技術を利用して非同期にサーバサイドとの通信を可能にした。すなわち、本来存在するはずのサーバ・クライアント間のデータ通信を、ユーザが意識することは無くなるということである。

また、システムの性質上ユーザが入力した質問をインクリメンタルに解析し、その都度リアルタイムに質問に対する回答の提示が求められる。したがって、本システムでは、Rails は Model と Controller に重点をおいた設計にし、View 側はクライアントサイドで担うように設計した。

本システムのクライアントサイドでは CoffeeScript を使用して実装を行っている。CoffeeScript とはコンパイルすることで JavaScript になるプログラミング言語である。CoffeeScript を使うことによって Rails で使われている Ruby 言語に近いシンタックスでプログラミングできるようになる。

## 2.2 システム機能

本システムの検索画面インターフェイスを図 1 に示す。図 1 の中央部①にある検索フォームを使用しユーザが質問を入力する。検索フォームの下に、ユーザからの質問にあった解決案をシステムで設定したスコアの高い順に提案していく。提案されたデータをクリックすると、データ

>Q&A

### サービス1でアカウントを変更したい

サービス1でアカウントを変更したい場合

サービス1(無償版)のホームページアドレスは、あとから変更ができません。

なお、サービス1(無償版)は、ひとつのメールアドレスでも複数のアカウントを登録できますので、お手数をおかけしますがご希望のアカウント名にて、ご登録ください。

また、現在のアカウントについては、ご不要な場合は以下の手順にて削除可能となっております。  
※Jimdoに登録後、一度もログインがなく14日間が経過した場合には自動的に削除されます。  
お急ぎでなければ、そのままの状態にしてしばらくお待ちいただければと存じます。

[アカウントの削除手順]

1. サービスにログインします。
2. 画面右にある管理メニューより「設定」アイコンをクリックします。
3. 表示されたメニューより、「ユーザー」の項目にある「アカウント削除」のアイコンをクリックします。
4. 「ホームページを削除します」のボタンをクリックしていただければ、ご登録いただいたアカウントは削除されます。

Category: アカウントを変更したい

関連する質問

> 登録を一度解除した後、再度同じアカウントで登録しようとしたが、登録完了のメールが届きません。何か制限があるのでしょうか?同じアドレスを使用したいのですが。

図 2 詳細ページ

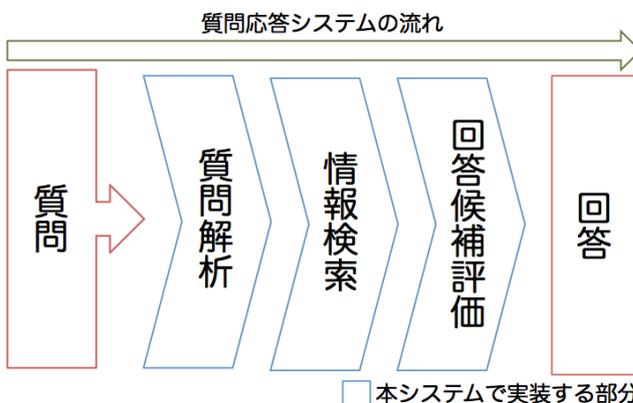


図 3 検索機能の概要

に付いての詳細ページに飛ぶことができる(図 2)。このとき、画面遷移はなく非同期にデータを取得するように設計してある。システムにはユーザ、サポートセンター担当者の両方が質問をし解決案を得るために以下のような機能が実装した。実装した機能は検索機能、ログイン機能、文章変更機能、および変更履歴確認機能である。以下、それぞれの機能について詳述する。

### 【検索機能】

検索機能の概要図を図 3 に示した。以下で検索機能についての概要を詳述する。システムのユーザが検索フォームを使用し質問を記述する。質問を解析し、システムで使用できるフォーマットに変換する。解析結果から、データベース中あらかじめ保存してある「過去のやりとり」や「よくある質問」の中から解決案の候補となるデータを取得する。取得したデータをスコアリング関数を使いスコアリングし評価する。最後にスコアの高い順にユーザに提案する。

ユーザの質問を形態素解析する際にオープンソース形態素解析エンジンである MeCab を用いた。また、システムで使うフォーマットに変更するために現状のシステムでは、質問中に存在する的名詞を切り出すようにしている。質問中に存在した名詞を切り取ったものを利用してデータ



図 4 ログイン後のシステムインターフェイス



図 5 変更機能のシステムインターフェイス

ベース中に存在するデータを取得する。検索方法については 3.3 節の検索方法にて後述する。検索結果となったデータに対してスコアリング関数を使ってスコアリングをする。スコアリング関数についても、3.4 節で後述する。

#### 【ログイン機能】

本システムのログイン機能とは、ある特定のユーザに対して文書の変更権限を与えることや、変更履歴を閲覧、変更できるようにする機能のことである。ログイン機能によりサービスのユーザとサポートセンターの担当者が同じインターフェイスのアプリケーションを共用することが可能になる。

また、権限にも種類を与えることにより担当者の中にも変更権限が持っているユーザと持っていないユーザや、変更履歴みてを元のデータに戻せるユーザと閲覧のみができるユーザといったような、より実用的な機能を実現することができる。ログイン後のインターフェイスを図 4 に示す。ログイン後には図 4 の上部のバーの①と②を見るとわかるように変更履歴を確認することができる History ボタン、文書変更機能である Edit ボタンといった管理者権限が必要な処理が表示されるようになっている。

#### 【文書変更機能】

本システムの文書変更機能とは、企業が提供するサービ

スの内容が変わった際、システムで提案するデータも変更する必要があるさいにその文書を変更する機能である。本システムではデータをデータベースに保管してあるのでデータベース上のデータを変更することになる。変更するためのインターフェイスとして、文書変更機能を実装した。管理者が変更することができるのは現状では「よくある質問」の質問とそれに対する回答である。例えば、アカウントの変更方法のサービス内容を変更した場合には、管理者が本機能を使ってアカウントの変更法について記述してあるデータを変更する必要がある。図 5 に変更機能のインターフェイスを示す。図 5 の①にデータの質問を編集するフォーム、図 5 の②にデータの質問に対して答えを編集するフォームがある。編集終了後、右下部にある Update Document ボタンをクリックすることで変更を完了する。

#### 【変更履歴確認機能】

本システムの変更機能とは文書変更機能で述べた文書変更した際の変更履歴を保持しておき、任意の地点に戻ることができる機能のことである。また、本機能には変更時刻、変更者及び、変更点のメモも記述できるようになっている。メモとはなぜ変更を行ったを記述する欄である。メモと変更者を書くことにより誰がどのような意図で変更を行ったからわかるようになり、サポートセンターの担当者が変更した際にも円滑に引き継ぎが行われる。また、本機能は変更履歴を全て残しておくことにより変更点に問題がある場合にすぐに、任意の履歴に戻れることを可能にしている。すべての変更を残しておくことにより、なぜ変更したかの意図がより明確になる。

### 3. ランキングアルゴリズム手法

#### 3.1 使用データ

本システムで使用しているデータについての説明する。本システムは共同研究先の企業の協力を得て、共同研究先企業が行っているサービスで使用しているデータを利用している。具体的には、共同研究先企業が行っているサービスのヘルプページで実際に使用されているデータ及び、サポートセンターで実際にユーザとのやりとりから得られたメールデータを使用している。

本研究では次の 2 種類のデータを用いる。1 つ目のデータはテンプレートデータである。2 つ目のデータはメールデータである。テンプレートデータは企業側がまとめた、「よくある質問」といったユーザが頻繁にする質問とそれに対応する答えのペアの FAQ データである。メールデータはサポートセンターの担当者がユーザとの実際のメールのやりとりをして得られたデータである。また、テンプレートデータ、メールデータのことを合わせてドキュメントデータと定義する。以下にテンプレートデータ、メールデータについて詳述する。

### 【テンプレートデータ】

テンプレートデータについて詳述する。テンプレートデータは2つのデータを用いて構成している。1つ目は質問データである。2つ目は回答データである。質問データとはサポートページに書いてあるような「よくある質問」や典型的な質問のことである。回答データとは質問への解決案が記してあるデータのことである。上記の質問とそれに対する回答を一つのデータとしてまとめたものをテンプレートデータという。また、本データはカテゴリという属性も保持している。カテゴリとは本データをいくつかのグループにわけたときに、それぞれのグループに対してつける名前をつけることのできる属性である。例えば「アカウント名を変更したい」といった質問は「アカウント」というカテゴリに分類することができる。本来、カテゴリ属性は本データには付与されていないのであらかじめ人手によりカテゴリ分類を行っている。現在、本データを73種類のカテゴリに分類している。本データは本システム内で非常に有用なデータとして利用している。ユーザの頻繁にする質問に対しての回答なので本データを利用することで回答データの提案精度が向上すると考えられるためである。よって、回答の提案精度を向上させるために本データはメールデータに比べ上位に提案するようにシステムを設計した。また、本データは全てで310件のデータを用いて構成している。

### 【メールデータ】

メールデータについて詳述する。メールデータはテンプレートデータと同様に2つのデータを用いて構成している。1つ目は質問データである。2つ目は回答データである。質問データとは、ユーザがサポートセンターに送った質問のことである。回答データとは、サポートセンターに寄せられたユーザからの、質問に対してサポートセンターの担当者が返信した回答のことである。上記のユーザからの質問とそれに対するサポートセンター回答を一つのデータとしてまとめたものをメールデータという。また、本データはラベルという属性も保持している。

ラベルとはユーザからの質問に対して、サポートセンターが回答を作成する時に参考にしたテンプレートデータのidを保持する属性である。ユーザからの質問は同じような質問に偏っていたり、過去に対応したことのある質問が非常に多い。そこで、サポートセンターではユーザからの質問に対する回答を作成する際にはテンプレートデータと過去の対応を参考にして回答を作成して。参考にしたテンプレートデータのidをラベル属性として付与した。付与は人手で行っている。参照が多いテンプレートデータはユーザにとって、有用なデータと考えることができる。ラベル属性により、参照が多いテンプレートデータが判別が可能になった。また、テンプレートデータでは表現しきれないような特殊な質問にも対応が可能になる。本データは

全てで4935件のデータを用いて構成している。

### 3.2 データベース

本システムで使用しているデータベースの構成について述べる。本システムでは、テンプレートデータとメールデータをあらかじめ形態素解析をし、それぞれのデータについての転置インデックスを作成している。転置インデックスとは、単語ごとに、その出現位置を記録した出現位置索引構造のことである(図6)。転置インデックスを使用することで、検索語が与えられた時に素早く検索結果を返すことができる。本システムでは形態素解析した結果の名詞を単語とし、単語がどのドキュメントに出現したかと、出現したドキュメントのどの位置に出現したかを記録した出現位置リストを保持している。

$T_1 = \{\text{今日はいい天気}\}$   
 $T_2 = \{\text{明日の天気は晴れ}\}$   
 $T_3 = \{\text{今日と明日の天気}\}$

"今日" =  $\{(1,1),(3,1)\}$   
"いい" =  $\{(1,2)\}$   
"天気" =  $\{(1,3),(2,2),(3,3)\}$   
"明日" =  $\{(2,1),(3,1)\}$   
"晴れ" =  $\{(2,3)\}$

図6 転置インデックスの例

### 3.3 検索方法

本システムの検索方法について述べる。本研究ではの検索方法は以下の3つのアプローチが存在する(図7)。

#### 【テンプレートデータを検索】

検索方法1はテンプレートデータを検索する方法である。テンプレートデータを検索するときには、ユーザの質問の解析結果をもとにして検索を行う。質問の解析結果は名詞を使用しているので、テンプレートデータに解析結果である名詞が存在する場合は当該テンプレートデータを検索結果とする。検索結果にテンプレートデータを使用するのは、テンプレートデータは質問に対して一意な解が存在するためユーザにとっては有用であるためである。

#### 【メールデータを検索】

検索方法2はメールデータを検索する方法である。本検索方法は1つ目の検索方法で説明した検索方法と同様で、ユーザの質問の解析結果がメールデータ中に存在する場合には当該メールデータを検索結果とするものである。メールデータを検索結果として表示する目的は、テンプレートデータでは回答不可能な問題に対して実施の事例を使用して対処するためである。しかし、メールデータは実際の

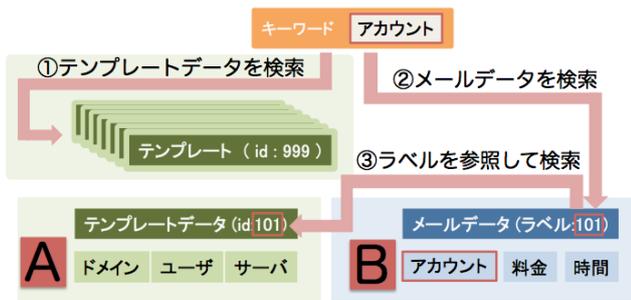


図 7 検索方法の概要

ユーザとのやりとりの記録なので個人情報などには十分注意する必要がある。そのため、本データではメールデータの全ての個人情報を削除したデータを使用してプライバシーの問題に対応している。

#### 【メールデータのラベルを用いた検索】

検索方法 3 はメールデータのラベルを用いた検索方法である。メールデータのラベルとは、メールデータにはサポートセンターがユーザの質問を回答する際に参考にしたテンプレートデータを保存する属性のことである (3.1 章【メールデータ】)。本検索方法はメールデータが参照したテンプレートデータを検索結果とする検索方法である。

具体的には、検索方法 1 はユーザの質問の解析結果をもとにテンプレートデータを検索する検索方法である (図 7 の①)。検索方法 2 はユーザの質問の解析結果をもとにメールデータデータを検索する検索方法である (図 7 の②)。検索方法 3 は、検索方法 1 では検索結果とならなかったテンプレートデータが存在するとする (図 7 の A, 以下 A)。検索方法 2 で検索結果になったメールデータが存在するとする (図 7 の B, 以下 B)。B のラベルが A を参照していた時、A も検索結果にするという検索方法である。つまり、検索方法 3 は、検索方法 1 で検索結果とならなかったテンプレートデータが検索方法 2 で検索結果となったメールデータに参照されていたら、そのテンプレートデータも検索結果とする。

### 3.4 スコアリング関数

本システムで使用されているスコアリング関数について述べる。3.3 章で行った検索方法を使用して検索結果となったデータに対してスコアリング関数を適応する。つまり、検索結果となったものをスコアリング対象とみなし、スコアリング対象となったデータに対してスコアリング関数を使用して検索結果のデータがユーザの質問に対する回答にどの程度ふさわしいかを評価する。本研究では評価の指標として単語の頻度、文書中で単語の出現位置、カテゴリとの一致度、およびテンプレートデータの参照重みの 4 つの指標を用いてスコアリング関数を作成した。

#### 【単語の頻度】

1 つ目の指標である「単語の頻度」について述べる。本指標は、ユーザが入力した質問の解析結果がドキュメント、つまりテンプレートデータとメールデータ中で出現するときの出現頻度を評価する指標である。本指標はテンプレートデータとメールデータをととも使用することが可能である。本指標を関数  $frequency$  として定義すると、ドキュメントの単語数が  $N$  で、ユーザが入力した質問の解析結果の単語の出現数が  $n$  とすると、式 (1) のように表すことができる。この時、 $i$  はドキュメントデータの id を表し、 $j$  はユーザの質問の解析結果の  $j$  番目の単語とし、解析結果の単語数を  $k$  とする。式 (1) はユーザが入力した質問を形態素解析し、いくつかの単語に分けて、それぞれの単語について、本指標を用いていることを表している。

$$frequency_i = \sum_{j=1}^k \frac{n_j}{N_i} \quad (1)$$

#### 【文書中で単語の出現位置】

2 つ目の指標である「文書中で単語の出現位置」について述べる。本指標は、ユーザが入力した質問の解析結果がドキュメント中のどの位置に出現するかを解析して評価するものである。本指標はテンプレートデータとメールデータとも使用することが可能である。具体的には、質問の解析結果となった単語がドキュメント中でどの程度最初に出てくるかを評価する。つまり、解析結果の単語がドキュメント中においてはじめの方に出てくれば出てくるほど本指標の評価値は高くなる。本指標を関数  $location$  として定義すると、ドキュメント中の単語数が  $N$  で、ユーザが入力した質問の解析結果の単語の出現位置を  $index$  とすると、式 (2) のように表すことができるこの時、 $i$  はドキュメントデータの id を表し、 $j$  はユーザの質問の解析結果の  $j$  番目の単語とし、解析結果の単語数を  $k$  とする。式 (2) はユーザが入力した質問を形態素解析し、いくつかの単語に分けて、それぞれの単語について、本指標を用いていることを表している。

$$location_i = \sum_{j=1}^k \left( 1 - \frac{index_j}{N_i} \right) \quad (2)$$

#### 【カテゴリとの一致度】

3 つ目の指標である「カテゴリとの一致度」について述べる。本指標は、ユーザが入力した質問の解析結果の単語とカテゴリがどの程度一致しているかを評価する指標である。本指標で使われているカテゴリとはテンプレートデータに付与されているどのようなカテゴリの質問か、という属性のことである。本指標はテンプレートデータには使用することが可能だが、メールデータはカテゴリのような属性を保持していないため適応することができない。本指標を関数  $category$  として定義すると、式 (3) のように表すこ

とができる．この時， $i$  はドキュメントデータの id を表すものとする

$$category_i = \begin{cases} 1 & \text{if consist category} \\ 0 & \text{unless consist category} \end{cases} \quad (3)$$

#### 【テンプレートデータの参照重み】

4 つ目の指標である「テンプレートデータの参照重み」について述べる．3.1 節で述べたように，メールデータはテンプレートデータは参考に回答しているものがあり，参考にした場合はメールデータに新たにラベルという属性が付与されている．本指標はテンプレートデータをスコアリングの際に当該テンプレートデータがどのくらいメールデータに参照されているかを評価する指標なのである．つまり，3.3 章で述べた検索方法 3 によって検索結果となったテンプレートデータが存在した際に，検索結果となったテンプレートデータを参考に回答を作成したメールデータが 1 つ以上存在するので，参考にしたメールデータのスコアを足しあわせ，合計スコアを参照数でわるということである．本指標は参照されているテンプレートデータにのみ使用することができる．本指標を関数  $label$  で定義すると，テンプレートデータを参照しているメールデータのスコアを  $MailScore$  とし，テンプレートデータを参照しているメールデータの数を  $RefN$  とすると，式 (4) のように表すことができる．この時， $i$  はドキュメントデータの id を表し， $MailScore_j$  は参照しているメールデータを表すものとする

$$label_i = \frac{\sum_{j=1}^{RefN} MailScore_j}{RefN} \quad (4)$$

以上の 4 つの指標を使い，テンプレートデータ，メールデータの評価式をそれぞれ評価する以下にそれぞれの評価式を示す．

#### 【テンプレートデータの評価式】

テンプレートデータは質問と回答が対になっているデータである．したがって，テンプレートデータの合計スコアを  $template$  と定義すると，質問のスコアを  $question_t$  とし，回答スコアを  $answer_t$  とすると，式 (5) のようにあらわすことができる．この時， $i$  はドキュメントデータの id を表し， $w_a$  は回答の重み， $w_q$  は質問の重みとした．また式 (5) 中の  $t$  はテンプレートデータに依存することを表す．

$$template_i = w_{a,t} answer_{i,t} + w_{q,t} question_{i,t} \quad (5)$$

次に式 (5) の  $answer_t$ ， $question_t$  について詳述する．テンプレートデータでは単語の頻度 (式 (1))，文書中での単語の出現位置 (式 (2))，カテゴリとの一致度 (式 (3))，およびテンプレートでデータの参照重み (式 (4)) を利用してス

コアリングしている．したがってテンプレートデータの回答のスコア  $answer_t$ ，と質問のスコア  $question_t$  はそれぞれは式 (6)，式 (7) と定義することができる．この時， $i$  はドキュメントデータの id を表し， $j$  はユーザの質問の解析結果の  $j$  番目の単語，解析結果の単語数を  $k$  とし， $w_a$  はテンプレートデータでの回答の際それぞれの関数に対する重み， $w_q$  質問の際のそれぞれの関数に対する重みを表すものとする．また式 (6) および式 (7) 中の  $t$  はテンプレートデータに依存することを表す．

$$answer_{t,i} = \sum_{j=1}^k (w_{t,a_0} frequency_i + w_{t,a_1} location_i + w_{t,a_2} category_i + w_{t,a_3} label_i) \quad (6)$$

$$question_{t,i} = \sum_{j=1}^k (w_{t,q_0} frequency_i + w_{t,q_1} location_i + w_{t,q_2} category_i + w_{t,q_3} label_i) \quad (7)$$

#### 【メールデータの評価式】

メールデータとは質問と回答が対になっているデータである．したがって，テンプレートデータの合計スコアを  $mail$  と定義すると，質問のスコアを  $question_m$  とし，回答スコアを  $answer_m$  とすると，式 (5) のようにあらわすことができる．この時， $i$  はドキュメントデータの id を表し， $w_a$  は回答の重み， $w_q$  は質問の重みとした．

$$mail_i = w_{m,a} answer_i + w_{m,q} question_i \quad (8)$$

メールデータでは単語の頻度 (式 (1))，文書中での単語の出現位置 (式 (2)) を利用してスコアリングしている．したがって  $answer$ ， $question$  はそれぞれは式 (9) および式 (10) と定義することができる．この時， $i$  はドキュメントデータの id を表し， $j$  はユーザの質問の解析結果の  $j$  番目の単語，解析結果の単語数を  $k$  とし， $w_a$  はテンプレートデータでの回答の際それぞれの関数に対する重み， $w_q$  質問の際のそれぞれの関数に対する重みを表すものとする．また式 (9) および式 (10) 中の  $m$  はテンプレートデータに依存することを表す．

$$answer_{i,m} = \sum_{j=1}^k (w_{m,a_0} frequency_i + w_{m,a_1} location_i) \quad (9)$$

$$question_{i,m} = \sum_{j=1}^k (w_{m,q_0} frequency_i + w_{m,q_1} location_i) \quad (10)$$

## 4. 評価

### 4.1 評価実験設定

評価の実験の設定を述べる．実際のメールの中から無作

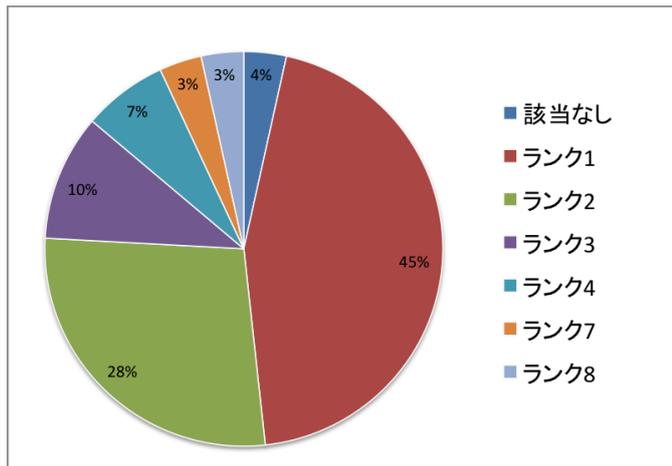


図 8 本システムのランクの割合

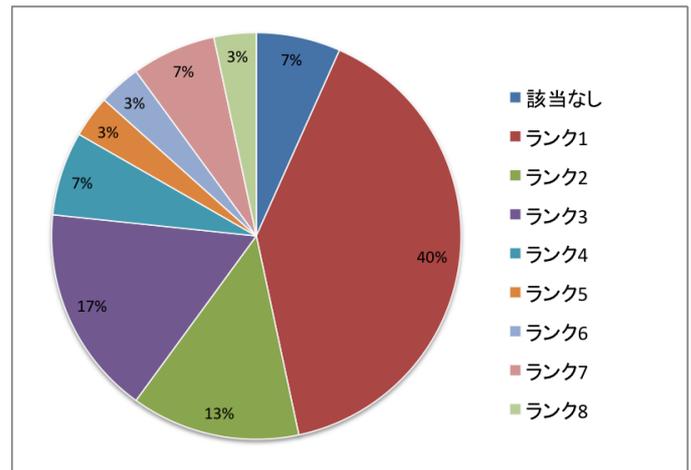


図 9 既存のランクの割合

為に 30 個の質問を選んだ。選んだ際に、おなじ意味の問題は取り除き重複のない問題を選んだ。質問を本システムと既存のシステムに入力しそれぞれの評価を行った。既存システムとは共同研究先の企業の web サイトで使われている単純な単語一致の検索システムである。

本研究ではシステムの評価指標として次の 3 つの指標を用いる。1 つ目はランクの逆数の平均である。2 つ目は 5 位正解率である。3 つ目は上位 5 位における正解データの占有率である。

ランクの逆数の平均はシステムが検索結果を出力した時に、検索結果中にある正解の回答のランクの逆数の平均のことである。

5 位正解率はシステムが上位 5 個の検索結果をを出力したときに、検索結果中に正解がある問題の割合である。

上位 5 位における正解データの占有率とはシステムが検索結果が出力した時に、検索結果の上位 5 位の中に正解の文書がある割合である。以下にランクの逆数の平均、5 位正解率、および上位 5 位における正解データの占有率について詳述し、既存のシステムと本システムで比較をした後に比較結果について考察をする。

#### 【ランクの逆数の平均】

ランクの逆数の平均について述べる。まず、本指標で述べるランクとは検索結果データの順位である。データが検索結果の 1 番目にあれば、ランクは 1 となり、検索結果の 2 番目にあればランクは 2 となる。ランクの逆数の平均とは、システムが検索結果を出力した時に、検索結果中の正解データのランクの逆数の平均を出すものである。図 8 と図 9 にそれぞれ本システムと既存のシステムのランクの割合を示す。それぞれの図のランクの割合からシステムのランクの逆数の平均を求める。本システムでは 0.625 となり、既存のシステムでは 0.531 となり既存のシステムよりもよい値を得ていることがわかる。これは、既存のシステムがランク 1 以外のランクもおおよそランク 4 以内に収まっていることがスコアを上げた原因だと考えられる。ま

た、ランクの逆数の平均は大きいということは、回答の候補を提案する時の順序が適切であると言える。本結果が意味していることは、本システムは既存のシステムに比べ回答の候補を提案する順番を決定するスコアリング関数が優れているということである。

#### 【5 位正解率】

5 位正解率について述べる。本指標はランクの逆数の平均で使ったランクを使用する。質問に対してランク 5 までに正解を出力することのできたものを正解とし、正解率をはかる指標である。よって、図 8 と図 9 からそれぞれのシステムの 5 位正解率を求める事ができる。実査に 5 位正解率を求めると本システムでは 86.6% となり、既存のシステムでは 80% となり既存のシステムと比べ本システムの 5 位正解率が高いことがわかる。本結果より、質問の回答の候補となりえる回答が検索結果のより上位になるようにスコアリング関数が作用していると考えられる。したがって、本システムは既存のシステムに比べ質問の回答の候補になりえるデータを上位に集約して表示できていると言える。

#### 【上位 5 位における正解データの占有率】

上位 5 位における正解データの占有率について述べる。本指標はシステムから検索結果が出力されたときに、検索結果の上位 5 位の中に正解の文書がある割合である。つまり、上位 5 位のうちユーザからの質問に対して正解となりえる回答が 2 つあった場合は本指標は 40% となる図 10 と図 11 に本システムと既存のシステムの 5 位正解率を示す。本指標が高ければ、ユーザからの質問に対して様々な角度からの回答が回答候補として出力されていることになる。したがって、本指標が高ければ、ユーザからの質問に対する解決案がランク 1 以外でも正解になる可能性が高くなると考えることができる。それぞれの図の上位 5 位における正解データの数からそれぞれのシステムの上位 5 位における正解データの占有率を求めると、本システムでは 38.6% となり、既存のシステムでは 36.6% となり既存システムと

比べ本システムの上位 5 位における正解データの占有率は高いといえる。図 10 と図 11 から、既存システムでは上位 5 位における正解データの数は 2 問が最多となっている。しかし、本システムでは上位 5 位における正解データの数は 3 問が最多となっている。上記の結果も本システムのスコアリング関数が適切であることを表している。

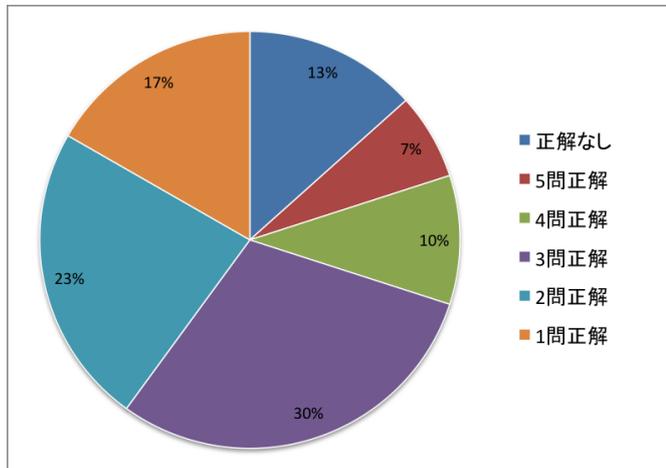


図 10 本システムの上位 5 位における正解データの数

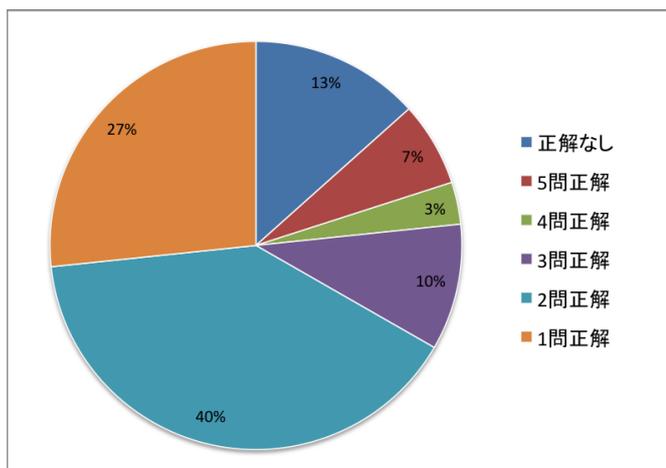


図 11 既存の上位 5 位における正解データの数

## 5. おわりに

本論文では、サポートセンターが抱えているユーザからの質問は増加しているがほとんどの質問に「よくある質問」回答することができる。また、「ユーザが抱えているサービスが多様化している」、「複雑化しヘルプページから目的のデータを探し出すことができない」、といった問題点を解決するためにランキングアルゴリズムを用いた質問応答システムを試作した。また、試作したシステムの性能評価を行い、有用性を確認した。結果として、全ての指標で既存のシステムよりも良い結果を得ることを確認した。

本結果には 2 つの原因があると考えられる。1

つ目は本システムではメールでデータを使用することで、少ないテンプレートデータを補足することができ、より広範囲な質問に応えられたことである。2 つ目は既存のシステムに比べ、本システムのスコアリング関数が優れていたことがあげられる。

今後の課題としては、システムのスコアリング、検索の精度を上げることである。まず、スコアリングを上げるためには、関数にかける重みをチューニングすることがあげられる。チューニングする際にユーザがした質問に対して出力した時に回答の何番目が押されているかといったデータを取り、押されたデータを上位に来るように重みを学習する方法が考えられる。また、検索の精度を上げるには検索の方法に確率的な要素を取り入れ、SVM などの手法を使った検索方法に切り替えるといったことがあげられる。

### 5.1 謝辞

本研究の一部は、株式会社 KDDI ウェブコミュニケーションズとの共同研究である。株式会社 KDDI ウェブコミュニケーションズの助成を受けている。

### 参考文献

- [1] Zhang, D. and Lee, W.S.: A web-based question answering system, *the SMA Annual Symposium 2003, Singapore, 2003*
- [2] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Geri Gay: Accurately Interpreting Click-through Data as Implicit Feedback, *the 28th annual international ACM SIGIR conference on Research and development in information retrieval (2005)*
- [3] D. Sculley: Large scale learning to rank, *In NIPS 2009 Workshop on Advances in Ranking(2009)*
- [4] 那須川 哲也: コールセンターにおけるテキストマイニング, *人工知能学会誌 16(2), 219-225, 2001-03-01*
- [5] 車 智修, 鍋島 啓太, 水野 淳太, 岡崎 直観, 乾 健太郎: 文書構造を用いた Why 型質問応答システム, *第 27 回人工知能学会全国大会 (2013)*
- [6] 石下 円香, 狩野 芳信, 神門 典子: 質問応答システムでの解答に向けた大学入試問題の分析, *第 27 回人工知能学会全国大会 (2013)*
- [7] 片山 亮, 川村 秀憲, 鈴木 恵二: QA サイトにおける質問推薦へ向けた履歴データの分析, *電子情報通信学会技術研究報告. AI, 人工知能と知識処理 109(439), 11-16, 2010-02-22*
- [8] 岩間 雄太, 伊藤 孝行: ヘルプセンターのための問い合わせ応答支援システムの試作, *平成 25 年度電気関連学会東海支部連合大会 (2013)*
- [9] 渋沢 潮, 林 貴宏, 尾内 利紀夫: 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. 54, No. 7(2013)*
- [10] 岡野原 大輔: *高速文字列解析の世界*, 岩波書店 (2012)
- [11] Toby Segaran: *集合知プログラミング*, オライリー・ジャパン
- [12] 西田 圭介: *Google を支える技術 巨大システムの内側の世界 (WEB+DB PRESS プラスシリーズ)*, 技術評論社 (2008/3/28)
- [13] 奥村 学, 磯崎 秀樹, 東中 竜一郎, 永田 昌明, 加藤 恒昭: *質問応答システム*, コロナ社 (2009)

## 【正誤表】

ランキングアルゴリズムを用いた質問応答システムの試作

岩間 雄太, 伊藤 孝行 (名古屋工業大学)

2013/10/25 8 ページ目

### 5.1 謝辞

誤：本研究の一部は，は株式会社KDDIウェブコミュニケーションズとの共同研究である．

株式会社KDDIウェブコミュニケーションズの助成を受けている．

正：本研究の一部は，株式会社KDDIウェブコミュニケーションズとの共同研究である．