

## 発話を意識した文推薦システムの構築と評価

富坂 亮太<sup>†1</sup> 鈴木 崇史<sup>†2</sup> 相澤 彰子<sup>†3</sup>

会話システムは、ユーザにシステムのフレンドリーさをアピールしたり、ユーザの潜在的な要求を引き出したりするために有効な手段である。ここで、ユーザの発言に対して適切な応答文を提示する会話応答の研究が数多く行われているのに対し、ユーザに対して自分から発言するシステムの研究は十分になされていない。それは、応答文では直前の会話が有効な手がかりとなるのに対して、発話文には過去の会話によらない話題の転換が求められるからである。そのため本研究では、情報推薦の枠組みを適用した発話文選択の手法を提案する。具体的には、ニュース記事に焦点をあてて、ユーザの興味と現在の流行の両方を考慮したニュース記事推薦を実現し、ユーザ実験の結果により提案システムがうまくユーザの興味を取得できていることを示す。

### Development and evaluation of sentence recommendation system for utterance generation

RYOTA TOMISAKA,<sup>†1</sup> TAKAFUMI SUZUKI<sup>†2</sup>  
and AKIKO AIZAWA<sup>†3</sup>

Conversation systems can be used for realizing user-friendly interfaces or identifying latent needs of the users. Here, majority of existing studies on conversation systems focused on utterance generation in response to user's utterance, and not many researches exist concerning utterance generation for initiating conversations. That is because, in the former case, the past utterances can be used as a major source of information, but in the latter case, finding a new topic independently of the past utterances becomes crucial. Based on the background, this study aims at developing a sentence selection system for initial utterance generation. In the paper, we propose a method for new topic selection based on both user's interest and latest topic trend on the Web together with some preliminary experimental results.

### 1. はじめに

近年、インターネットの急速な普及により、人々の生活様態が劇的に変化しつつある。その中で、ソーシャル・ネットワーク・サービス (SNS) と呼ばれるサービスが普及し、いわゆる、インターネット上でのコミュニケーション (オンラインコミュニケーション) が一般的になりつつある。特に、2006年に始まった、Twitter<sup>\*1</sup>のサービスはユーザ数が1億人を超え、多くの企業や組織が利用するまで成長し、社会において無視できないほどの一般的なサービスとして定着しつつある。

このようなオンラインコミュニケーションの様態の中で、ボットと呼ばれる自動的に発言を生成するシステムが、人間の対話相手として至る所で稼働している。例えば、有名なTVキャラクターをキャラクターとした、Twitter Botが存在する。このようなBotにはBlogにかかれた記事から、自動的に文を生成して、Twitter上に流すというものが存在し、81万人という非常に多くの人にフォローされている。

コンピュータと人間との会話を実現するシステムの研究は数多くなされている。例えば、会話システムの研究分野はユーザに適切な応答を返したり、ユーザを適切な答えに導いたりするタスクを研究している1)-3)。また、人工無脳という、会話のコーパスからユーザの発言に対する応答文として適切な答えを探索し返答するというシステムも存在する4)。しかし、これらはユーザに適切な応答を返すということに焦点を当てて研究をされているものであり、Twitter Botのような自分からユーザに対して話題を提供するといったことを目的とした研究は少ない。

そこで、本研究では、このような、ユーザに話題を提供する、つまり、会話の出だしを自動で生成するためのユーザの興味モデルの自動構築手法を提案する。これにより、従来の会話ボットは自分から会話を切り出すことができるようになり、Twitter Botなどは、ユーザの好みに対応した情報を提供できると考える。

†1 東京大学大学院 情報理工学系研究科 コンピュータ科学専攻  
Department of Computer Science, Graduate School of Information Science and Technology, The University of Tokyo

†2 東洋大学 社会学部 メディアコミュニケーション学科  
Department of Media and Communications, Faculty of Sociology, Toyo University

†3 国立情報学研究所/東京大学  
National Institute of Informatics/The University of Tokyo

\*1 <http://twitter.com/>

応答文の自動生成タスクにおいては、応答文を生成するためのヒントとして、相手の発言内容が利用できる。しかし、会話の切り出しにおいては、このようなヒントは無く、コンピュータが自ら相手の求める話題を推測し提案しなくてはならない。ここで、人間の発話行動を振り返ってみて考えると、この会話の切り出しというのは、いわゆるレコメンデーションのようなものであるということに気づく。なぜなら、会話の話題提供は、相手が興味を持つであろう話題を選択して行うものであり、その話題に興味を持って応答してくれるかどうかは、相手の判断に委ねられるからである。

我々は、この会話の切り出しのための話題提供を(文の)レコメンデーションであると考え、ニュース記事のレコメンデーションシステムの文献5)を参考に、ユーザとのやりとりの中から、ユーザの興味モデルを自動獲得し、ユーザが興味を持つ文を推薦できるシステムを構築する。ここで、文献5)では、推薦するニュース記事に新規性を持たせるために、ユーザの興味モデルは「政治」「国際」など粒度の大きなニュースカテゴリに基づき行われている。しかし、本稿は、会話の話題提供が目的であり、ユーザに既知であっても身近で興味度の強い情報を提供することが重要であると考え。そのため、ユーザの興味のある文を解析し、より詳細なユーザモデルを獲得することを目指した。また、同論文は、大規模なクリックログを利用してユーザモデルの獲得を行っていたが、ユーザモデルを獲得するために、ユーザにあらかじめ大規模なクリックログを作成してもらうのは、非現実的であり、会話というタスクにおいては、ユーザとのやりとりのなかで、自動的に獲得できる手法を目指した。

これらの考えに基づき、本研究では、ユーザが興味を示した文から、自動的にユーザモデルを獲得し、そのモデルを用いて、ユーザに提供すべき文を選択するシステムの構築を行った。

## 2. 関連研究

会話の切り出しをテーマとした研究は少ないが、情報推薦に関する研究は数多く存在する(6)–(8)。情報推薦には、協調フィルタリングに基づく方法とコンテンツに基づくフィルタリング方法が存在する。このうち協調フィルタリング(9)は、同じような嗜好を持つ他者の行動を利用して、ユーザに情報を推薦する手法であるが、会話というリソースを用いて、同じ嗜好を持つユーザを特定する事は難しいため、この手法を自然に会話のタスクに取り入れるのは難しいと考える。

一方、コンテンツフィルタリングでは、ユーザプロファイリングという手法が一般に用

いられる(10)。これは、ユーザの行動履歴から、ユーザの興味のある分野を何らかのモデルで表現し、そのモデルに基づき情報を推薦する手法である。たとえば杉山の研究(10)では、ユーザの過去の検索から、ユーザの興味モデルのベクトルを作成し、それを用いて、ユーザが興味を持つであろうページをフィルタリングして提示するという手法を用いている。本研究も、これに習い、ユーザの興味のある文書群の用語から、ユーザの興味モデルを作成し、情報推薦する枠組みを提案する。

ここで、ユーザの興味は刻一刻と変化するものであり、ただ単純にユーザモデルを適用するのは問題が生じることが指摘されている(5)(10)。5)の論文では、このユーザの興味の変化は、地域の流行に作用されて起こるものであり、ユーザの現在の嗜好は、ユーザ本来の興味(長期的なスパンで作成したユーザプロファイル)とユーザの所属する地域のコミュニティ全体の興味の重ね合わせであると結論づけている。特定の地域の流行を利用してユーザに情報を推薦することは、広い意味で、協調フィルタリングを行っていると言えなくもない。また、杉山の研究(10)では、ユーザの興味の減衰モデルを作成することで、ユーザの興味の時間変化の問題を解決することを試みている。

本研究は、前述のように、Liuらによるニュース記事のレコメンデーション手法を参考にしている(5)。この論文では大規模なGoogle Newsのクリックログを解析し、ユーザの興味には、ユーザ本来の興味に加え、ユーザが所属している地域の話題性も影響することを示している。また、Google Newsの10のカテゴリ分類を用いて、ユーザの興味モデルとともにユーザの所属する地域の話題性のモデルを作成し、ユーザにニュース記事を推薦するシステムを提案し有効性を実証している。本研究ではこの論文の知見に基づき、ユーザの興味モデルの獲得とともに、リアルタイムな話題の獲得を試みる。ここで、Riuらの研究では、ユーザの興味モデルは、ニュース記事の大カテゴリに基づく大雑把なものであったが、本研究では、ユーザの選択した文からの詳細なユーザモデルの作成を試みる。また、Riuらの研究では大量のアクセスログを用いて、モデルの獲得を行っているが、本研究では、ユーザの選択による数文の興味文からユーザモデルを獲得し、さらに、広く普及している情報発信ツールであるTwitterから流行している用語を判別する。

## 3. 提案手法

### 3.1 提案手法の流れ

提案手法の簡単な流れを図1に示す。本研究では、ユーザの興味や話題性を語の重みベクトルで表現し、このような語の重みベクトルを「モデル」と呼ぶことにする。「モデル」と

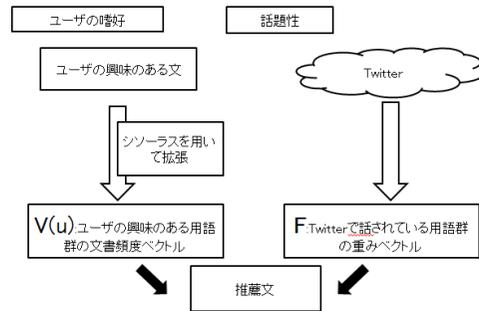


図 1 提案手法の流れ  
Fig.1 flow of my approach

推薦候補文の間の類似度に基づきスコアを定め、そのスコアに基づき文章を推薦する。

具体的には、図 1 のように、ユーザーの興味モデル、話題性のモデルをそれぞれ作成し、その両方を考慮したモデルに基づき、最終的にユーザーに提示する文を選択する。ユーザーの興味モデルは、ユーザーがあらかじめ選択した興味文に基づき作成する。その際に、ユーザーが選択した文内の用語を、そのまま用いるのではなく、シソーラスを用いて文内の用語を拡張することにより、ユーザーのモデルがスパースにならないように配慮する。話題性のモデルは、Twitter のつぶやきを用いて、各用語の出現頻度の時間変化を計算することによって作成する。最終的にこの二つのモデルを用い、ユーザーへ推薦する文章の候補文のリストから最もスコアが高い文を計算し、ユーザーに提示する。候補文として今回は最新の Yahoo! ニュースの記事を用いて実験を行った。

### 3.2 文の評価のモデル

本稿では、候補文からユーザーに推薦する文を選ぶための評価指標として、ユーザー本来の興味モデルと話題性のモデルの重ね合わせを用いる。このモデルを式で表すと以下のようになる。

$$R(u, S, t) = \sum_{w \in W(S)} R(u, w, t)$$

$$R(u, w, t) = I(u, w) \times F(w, t)$$

上式で  $R(u, S, t)$  はユーザー  $u$  の文章  $S$  に対する時点  $t$  での興味の度合いを表している。 $I(u, w)$  はユーザー  $u$  の用語  $w$  に対するユーザーの興味のスコアであり、 $F(w, t)$  は用語  $w$  の話題性のモデルのスコアである。 $R(u, w, t)$  は時点  $t$  におけるユーザー  $u$  の用語  $w$  に対する興味を示し、その文書  $S$  に含まれる用語についての重ね合わせが  $R(u, S, t)$  となる。 $W(S)$  は、 $S$  に含まれる用語集合である。以下では、これらのモデル  $I(u, w)$  と  $F(w, t)$  の具体的な獲得の仕方について論じてゆく。

### 3.3 ユーザーの興味モデルの獲得

本節では上述の  $I(u, w)$  のモデルの獲得方法について述べる。本研究では、あらかじめユーザーに興味のある文を選択してもらい、それを学習データとして、ユーザーの興味ベクトルを獲得する。なお、ここではあらかじめユーザーに文を選択しておいてもらったが、ユーザーとのインタラクションによって、順次ユーザーの興味モデルを構築していくことも可能である。

しかし、これだけだと、ユーザーの興味ベクトルがあまりにもスパースになりすぎてしまい、大規模な学習セットを作成しないと、まともに機能しなくなってしまう。(たとえば、コンピュータについての文が選択された場合コンピュータという用語を含む文しか推薦できない。) しかし大規模な、データセットをユーザー毎に作る事は、ユーザーの負担が大きい。そのため、本研究では、文書リソースをシソーラスのように用いて、用語を拡張し、その拡張された用語を  $I(u, w)$  に追加することにより、ユーザーモデルを作成する。これにより、少ない学習データセットから、より詳細なユーザーモデルの獲得を目指す。

この操作は、クエリ拡張のタスクと似ている、クエリ拡張は、ユーザーから与えられたクエリのスパースネスを解消するため、ユーザーのクエリに適切な情報を加えたり、ユーザーのクエリをより適切なものへと置き換えることを目的として行われる。クエリ拡張の研究として WordNet をクエリ拡張のソースとして用いた Yokoyama らの研究 11) や、Wikipedia をソースとして用いた Peetz らの研究 12) などがある。また、ユーザーから与えられた少数のコアとなる情報を元にオントロジー構造を構築するタスクだとみることでもできる。クエリ拡張と同様に、Wikipedia をソースとし、オントロジー構造を作ろうとする研究 13) や、Wikipedia と WordNet の両方を用いてオントロジーを作ろうとする研究 14) などが存在する。

本研究では Wikipedia<sup>\*1</sup>、日本語 WordNet<sup>\*2</sup>の他に、Yahoo!検索エンジン<sup>\*3</sup>を用語拡張

\*1 <http://ja.wikipedia.org/wiki/>

\*2 <http://nlpwww.nict.go.jp/wn-ja/>

\*3 <http://www.yahoo.co.jp/>

のためのリソースとして用いた。これは、Wikipedia と WordNet は、増え続ける新語や、複雑な略語などに対応できない可能性があると考えたためである。後述の実験で、それぞれのリソースを別々に用いた場合と同時に用いた場合について比較を行う。それぞれのリソースについての基本的な情報を表 1 にまとめた。また、具体的にそれぞれのリソースを用いて用語を拡張する方法については後述する。

ユーザが選んだ文から用語を切り出す方法としては、Wikipedia、日本語 WordNet については、それぞれが所持している用語すべてを用いて、ユーザの選んだ文からとの最長一致を行い用語を切り出した。Yahoo!検索エンジンについては、Twitter をリソースとした、可変長 N グラム構造を用いて、用語の切れ目の判別を行った。この手法については我々の研究 15) を参照されたい。

### 3.3.1 Wikipedia による用語拡張

Wikipedia とはウィキペディア財団が運営する、オンラインの百科事典である。それぞれの項目は、Wiki でかかれており、誰でも編集する事ができる。そのため、数多くの人の記事の保守変更を行っているため、最新の言葉がいち早く更新され、また、厳密なルールのおかげで、それなりに信用がおける内容となっている。

本研究では、この百科事典のリンク構造を利用し、ある用語に関連の深い用語を抜き出す。具体的には、用語  $i$  から又は用語  $i$  へリダイレクトしている用語を全て同じ用語とみなし、また、それらの説明ページのアブスト部分から、リンクが貼られている用語を全て同義語として抜き出し、 $I(u, w)$  に追加した。

### 3.3.2 日本語 WordNet による用語拡張

WordNet とは、英単語を synset と呼ばれる同義語のグループにわけ、synset 間の関係を、synlink と呼ばれるリンクで結ぶことにより用語間の関係を定義したソーラスである。本研究では、対象の用語  $w$  を含む synset に含まれる用語を  $I(u, w)$  に追加した。

### 3.3.3 Yahoo!検索エンジンによる用語拡張

さらに、本研究では、Yahoo!検索エンジンを拡張したい用語で検索し、各ページの summary 部分から、用語を切り出し、文に含まれる用語を拡張した。用語の切り出し方としては、Twitter をコーパスとした可変長 N グラムを用いて抽出した。

### 3.4 話題語の獲得

本研究で、話題語の学習源として、大規模な SNS サービスである Twitter に着目した。Twitter 上ではリアルタイムに多くのユーザがつぶやいており、また API を経由して常に最新の情報を取得することができる。Twitter はコーパスとしてみると、常に更新され続

け、さまざまな人間による文体を含むという価値の高い言語資源であるといえる。我々は、この Twitter でのつぶやきの総数に着目し、ある用語  $w$  の流行の度合いとして用語  $w$  の Twitter でのつぶやき数の変化を調べ、つぶやき数の増加の割合(減少するような場合は 1 とした。なぜなら、流行している用語はユーザの興味を引き付けるが、逆に廃れつつある語がユーザの興味に負のバイアスをかけるとは考えにくいと考えるためである。)を、用語  $w$  の流行のスコア  $F(w, t)$  とした。具体的には、用語  $w$  の今月 ( $t$ ) のつぶやき総数と、半年前のつぶやきの総数を比較し、その比をスコアとした。これを、式で表すと、

$$F(w, t) = \frac{freq(w, t)}{freq(w, t - \delta t) + 1} (if F(w, t) > 1), 1 (otherwise)$$

となる。本稿では予備実験の結果に基づき  $t = 6$  ヶ月とした。つぶやきの総数は Google リアルタイム検索<sup>\*4</sup>を用いて調べた。本稿では数ヶ月に渡る Twitter コーパスの作成ができなかったため、Google リアルタイム検索を代わりに用いて、呟きの総数を調べている。Google リアルタイム検索とは、Google が最近開始した、Twitter でのつぶやきを検索できるシステムで、ここ数ヶ月のつぶやきの推移を取得することができるサービスである。現在のところ 2010 年 2 月からのデータしか取得できないが、いずれ、2006 年 3 月まで遡って閲覧できるとされている。

## 4. ユーザ実験

以上の、我々のシステムの有効性を確かめるためにユーザ実験を実際に行った。

### 4.1 実験の概要

まず、被験者に 4 つの記事を提示し、面白そうだと思う記事を選択してもらうためのユーザインターフェイス(図 2)を作成し、実際に被験者 4 名に選択してもらった。被験者に、興味を持った順に任意の数の記事を選んでもらうという作業を 92 個のセットについて行ってもらった。

表 1 リソース  
Table 1 Resource

	WikiPedia	WordNet	Yahoo!検索エンジン
単語数	1,129,927	92,241	
重なり単語数		32,335	-

\*4 <http://www.google.com/realtime?esrch=RealtimeLaunch::Experiment>

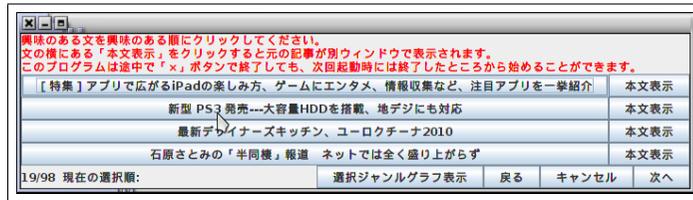


図 2 実験に用いたプログラム  
Fig. 2 program for user experiment

この、92 個のデータセットのうち 91 個を被験者の興味モデル獲得のための学習データとして学習し、残り 1 個の記事 4 つをこのモデルを用いてランクづけし、被験者がこの 4 つの記事から選択した個数を  $X$  とするとランク  $X$  番めまでに、実際に被験者が選択した記事とマッチしている個数を  $Y$  とし、 $Y/X$  を評価値とした。これを 92 個のデータセットと被験者 4 名それぞれについて行い、それらすべての平均値を最終的な評価値とした。

この実験を、以下のそれぞれのパターンにより行い、結果を比較した

- 記事をランダムに選んだ場合の期待値 (ベースライン)(R0)
- ユーザモデル  $I(u, w)$  を 1 とした場合 (流行のみを考慮した場合)(R1)
- $F(w, t)$  を 1 とし (流行を考慮しない), ユーザモデル  $I(u, w)$  の獲得に WordNet を用いた場合 (R2)
- $F(w, t)$  を 1 とし (流行を考慮しない), ユーザモデル  $I(u, w)$  の獲得に Wikipedia を用いた場合 (R3)
- $F(w, t)$  を 1 とし (流行を考慮しない), ユーザモデル  $I(u, w)$  の獲得に Yahoo!検索エンジンを用いた場合 (R4)
- ユーザモデル  $I(u, w)$  の獲得に WordNet を用いた場合 (R5)
- ユーザモデル  $I(u, w)$  の獲得に Wikipedia を用いた場合 (R6)
- ユーザモデル  $I(u, w)$  の獲得に Yahoo!検索エンジンを用いた場合 (R7)

#### 4.2 結果

前節で説明した実験の結果を表 2 に示す。この結果を見ると、まず流行性のみを考慮した場合 (R1), baseline(R0) より高い数値を得られていることがわかる。この結果から、Twitter で話題になっているような用語に、ユーザの興味が集まっているということがわかり、本手法により、流行性から導き出される、ユーザの現興味を引き出せたと考えられる。次にユー

ザモデルを考慮した場合 (R2~4) に、ベースライン (R0) よりよい結果を得られている事が分かる。これにより、それぞれのリソースを用いて用語を拡張しユーザモデルを作成した本手法により、ユーザの興味モデルを作成することに成功していることが示されているといえる。この、三つのリソースの中では、Wikipedia を用いた場合 (R3) が最も良い数値を得ている。これは、Yahoo!検索エンジンでは余計な用語をモデルに追加しすぎてしまい、WordNet では絶対的に用語の量が足りていないというのが原因だった。

さらに、流行性のモデルとユーザの興味モデルの両方を用いた場合 (R5~R7) のとき、それぞれ、流行性を考慮しない場合や、ユーザの興味モデルを考慮しない場合より高い数値を得ることに成功している。よって、現在のユーザの興味は流行性とユーザの本来の興味の両方に影響されており、これは、本手法で提案する、両者を考慮した文推薦の有効性を示す結果である。

結果を詳しく観察すると、さらに結果を向上させるための示唆がいくつか得られた。この要因として、一つにユーザの興味モデルの拡張の問題があげられる。Wikipedia, WordNet による拡張では、文中のすべての用語を拡張できていなく、ユーザモデルがスパースになっているように見受けられた。特に、学習、推薦に用いた文章がニュース記事だったため、文中に複雑な新語、略語が多く見受けられ、拡張が起こらなかつたり作成されたユーザモデルとマッチしなかつた用語などが多く見受けられた。逆に、Yahoo!検索エンジンによる拡張では、余計な用語が入りすぎてしまい、どのような文にも一様のスコアを与えてしまう問題が見受けられた。この、問題を解決するため、それぞれのリソースの拡張の仕方の改善及びそれぞれの長所を生かしたモデルの重ね合わせ手法の探索を今後の課題としたい。もう一つ、の要因として、Twitter からの流行性の獲得時の計算の問題があげられる。現在、流行性のモデルの計算に、今現在と過去の一点でのつばやき数を用いてしまっており、どのようにつばやき数が変動したかを考慮できていない。具体的には、過去の一点の時点では、つばやき数が 0 で現在で流行っているわけではないが少しつばやかれているような用語や、流行が過ぎて現在つばやき数が減っているような用語などは高い値が出てしまう。また、定期的に流行が来る用語などもうまく取れない可能性があり、過去の一点の時点から流行っており、今も流行っている用語などもうまくとることができていない。今後は、こういったつばやき数の変動もうまくとらえた評価尺度を考えていく必要があると考えている。

#### 5. おわりに

本研究では、会話での話題の切り出しに焦点をあて、話題の切り出しとレコメンデーション

ンの類似性から，ユーザが求める文を自動的に選択し，ユーザに提示するシステムの提案を行った．そして，レコメンデーションの手法を文選択というタスクに当てはめ，実際にユーザの興味を文からモデル化する手法を提案し，実際にその手法の有効性をユーザ実験によって示した．

今後の課題として，前述のとおりユーザモデルの作成方法の見直しが挙げられる．Wikipedia や WordNet などは，用語を拡張できていない事例が多く見受けられ，また，Yahoo!検索エンジンによる手法は，Summary 部分から抜き出してしまっているのが，あまり関係のない用語まで抜き出しているのが，見受けられた．

また，現在 Twitter でのつぶやきの変化を，半年前と今という二値で計算しているが，これは，半年前に偶然，何らかの影響で高くなってしまおうと，正しく取得できないという問題が見受けられた．これも，長いスパンを考慮できるような評価関数を実装する必要がある．

本研究は，長期的には，会話システムの構築を視野にいれたものであり，選択された文からより会話らしい文を作成したり，応答したりできるシステムを目指したい．

謝辞 本研究の一部は，国立情報学研究所公募型共同研究「多種テキストデータからのコミュニケーション・スタイルの抽出ならびにその分析と応用」(代表:鈴木崇史)より支援を受けています．ここに記して感謝します．

## 参 考 文 献

- 1) 清田陽司, 黒橋禎夫, 木戸冬子: 大規模テキスト知識ベースに基づく自動質問応答: ダイアログナビ, 自然言語処理全国大会, Vol.10, No.4, pp.145-176 (2003).
- 2) Shilin Ding, Gao Cong, Chin-yew Lin, Xiaoyan Zhu: Using Conditional Ran-

表 2 実験結果  
Table 2 Result of experiment

	評価値
R0(baseline)	0.631
R1	0.641
R2	0.671
R3	0.657
R4	0.665
R5	0.677
R6	0.680
R7	0.675

- 3) Wen-Yun Yang, Yunbo Cao, Chin-Yew Lin: A Structural Support Vector Method for Extracting Contexts and Answers of Questions from Online Forums, *ACL2008*, pp.710-718 (2008).
- 4) 富坂亮太, 鈴木崇史: <知っておきたいキーワード>人工無脳(会話ボット), 映像情報メディア学会誌 2010年1月号, Vol.64, No.1, pp.64-66 (2010).
- 5) Jiahui Liu, Peter Dolan, Elin Rønby Pedersen: Personalized News Recommendation Based on Click Behavior, *IUI2010*, pp.31-40 (2010).
- 6) 土方嘉徳: 情報推薦・情報フィルタリングのためのユーザプロファイリング技術, 人工知能学会論文誌, Vol.19, No.3, pp.365-372 (2004).
- 7) Jia Wang, Qing Li, Yuanzhu Peter Chen, Zhangxi: Recommendation in Internet Forums and Blogs, *ACL2010*, pp.257-265 (2010).
- 8) Yize Li, Jiazhong Nie, Yi Zhang, Bingqing Wang, Baoshi Yan and Fuliang Weng: Contextual Recommendation based on Text Mining, *Coling2010*, pp.692-700 (2010).
- 9) 福原知宏: 協調フィルタリングに関する研究動向, <http://www.race.u-tokyo.ac.jp/fukuhara/Research/paper/98/cofil.pdf> (1998).
- 10) 杉山一成: ユーザからの負担なく構築したプロファイルに基づく適応的 Web 情報検索, 電子情報通信学会論文誌, Vol.J87-D-I, No.11, pp.975-990 (2004).
- 11) Ai Yokoyama, Vitaly Klyuev: Search Engine Query Expansion Using Japanese WordNet, *IEEE2010*, pp.1-5 (2010).
- 12) Maria-Hendrike Peetz and Martin Lopatka: Query Expansion with Wikipedia, *IIR2006-2007* (2008).
- 13) Fei Wu, Daniel S. Weld: Open Information Extraction Using Wikipedia, *ACL2010* (2010).
- 14) Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum: YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia, *WWW 2007*, pp.697-706 (2007).
- 15) 富坂亮太, 鈴木崇史, 相澤彰子: 自由対話実現のための自動文生成モデルの提案, 情報処理学会創立 50 周年記念全国大会, pp.627-628 (2010).