

研究指導 中澤 真 助教授

# ユーザの要求に柔軟な対応をするための対話型情報検索

古川 佳奈

## 1 はじめに

インターネットが広く普及した現在、その用途は情報収集であるという割合が最も高い[1]。また、Yahoo! JAPANが実施したアンケート調査によると、情報収集手段として検索システムを利用したという回答が大半を占めている[2]。このことから、検索システムとは情報収集において大きな役割を果たしていることが分かる。

しかしながら、検索システムを使用する際、どのような検索キーワードを入力すれば求めている情報が手に入るか、ということを考えるのは難しい。必要な情報を得るために何度もキーワードを変更したり、新しくキーワードを付け加えたりした経験が誰にでも少なからずあるだろう。「こんな情報が欲しい」という漠然としたイメージはあっても、情報を手に入れるためには明確なキーワードを入力しなければならない。このように、既存の検索システムは必ずしも使いやすいつとはいえない状況にある。

ここで、人間同士の対話を思い浮かべて欲しい。対話は人間の最も基本的なコミュニケーション手段だ。人間同士の対話では漠然性を含んだ言葉でも相手に伝わる。このことから、ユーザにとって最も慣れ親しんでいる対話という形で要求を入力できれば、前述したようなキーワード入力に対する負担は軽減されると考えられる。

このような考えに基づき、キーワードを入力するのではなく、システムと対話することでユーザが求めている情報を得られる対話型情報検索システムを取り上げる。だが、既存のシステムは漠然とした言葉に十分な対応ができず、漠然性を含む発話をした場合、ユーザの求める情報を適切に絞り込むことができない。

よって、欲しい情報に対してユーザが漠然としたイメージしか持っていない場合でも、対話を通じて求めている情報を入手できるような対話型システムを提案する。本研究では検索時に漠然とした言葉が多く利用される可能性のある飲食店に検索対象を絞ったが、これは一般性を失うものではなく、他分野への適応も可能なシステムになっている。

## 2 検索のキーワードと発話に含まれる漠然性

飲食店を検索するには料理に関する条件や、店についての条件が必要となる。本研究では、自分の目的に合う飲食店を絞り込む際に入力されるキーワードを、

- a. 何料理か(和食, 麺類, ケーキ等)
- b. 料理の特徴(甘い, 辛い等)
- c. 店の場所
- d. 店の名前
- e. 店の条件(駐車場が広い, ランチがある等)
- f. ユーザの状況(二日酔い, ダイエット中等)

といった六つのタイプに分類した。一般的にコンピュータには漠然性を含んでいるキーワードが判断できない。そ

のため、漠然性を含む語とはどんなものかを示す必要がある。そこで、キーワードを分類し、その中でどれが漠然性を含んでいる語にあたるかということコンピュータに教える。これにより、どんなキーワードが入力された場合に漠然性解消を図るべきかが把握できるようになる。ここで、漠然性とは対話の中で「どんなものが食べたいか、どんな店に行きたいか」ということが不明確であることと定義し、分類した中ではb, fにあたる。

既存のシステムの漠然性への対応について、グルメ情報サイトとして代表的な「ぐるなび」を考えてみる。このサイトでは、エリア、料理名はもちろん、目的や予算の条件を指定することで飲食店を検索できる。しかし、このインターフェイスでは漠然性を含んだb, fについてのキーワードには対応できない。本研究ではシステムが具体的な意味を持つ言葉(具体語)をユーザへ提示し、その中からキーワードとなる語を選択してもらおう。そして、b, fのような語をユーザの選択した語と入れ替えることで漠然性の解消を実現させる。

## 3 対話事例を利用した対話システム

### 3.1 キーワードと検索式

入力されたキーワードは検索実行のための検索式に使用されるが、漠然性を含むキーワードが入力された場合、検索式にどのような影響を与えるかについて述べる。検索式はキーワード列 $K_j(j=1, 2, \dots, n)$ とし、この時 $j$ はキーワードの指標である。文献[4]では、ユーザの発話文から直接キーワードを抽出して検索式を生成しているため、b, fのような漠然性を含むキーワードをそのままキーワード列 $K_j$ に追加してしまう可能性がある。このような具体化されていないキーワードからユーザの求める情報を的確に探し出すことは難しい。しかし本システムでは、漠然性を含んだ語ではなく、具体語が検索式に追加されるキーワードとなる。その結果、ユーザの求める情報を的確に探し出すことができると考えられる。

### 3.2 システム構成

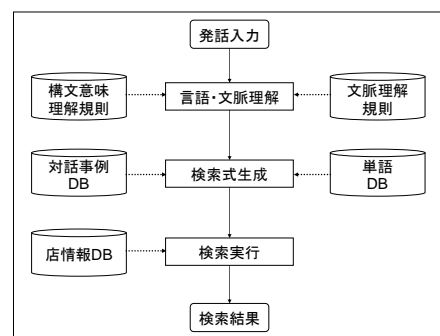


図1 対話システム構成図

<sup>1</sup> <http://www.gnavi.co.jp/>

本システムを実現させるにあたり、対話事例DB<sup>2</sup>を利用する。対話事例DBとは、システムとユーザによる過去の対話事例を蓄積したものであり、図1に示すように、検索式生成の際にこれを利用する[3][4]。単語DBは発話文から抽出された、意味素を持つキーワードが入っているものとし、ユーザ発話中の漠然とした語を、具体化させた上で検索式に追加させるために使用される。なお、意味素については後述する。

情報検索実行に必要な不可欠となる検索式にはキーワードが使用されることから、具体化されているか否かにかかわらず、キーワードとして抽出された語はすべて検索式に追加されてしまう。検索式に具体化されていないキーワードが含まれていれば、情報を的確に絞り込むことは不可能だ。このことから情報の絞り込みができるかどうかは、検索式の構成次第であるといえる。よって、情報検索において大きな役目を果たす検索式生成は漠然性の解消において特に重要となる。

#### 4 共起関係強度の算出方法

ユーザの漠然とした要求を具体化させるために、共起という概念を用いて実現させる方法をここでは述べる。コンセプトとしては、ユーザの発話文からキーワードを抽出し、そのキーワードと共起関係の強い具体語を対話事例DBから探し出す。そしてその具体語を、漠然性を含む語と入れ替えることによって漠然性を解消する。具体的手法を次に述べる。

##### 4.1 t-score

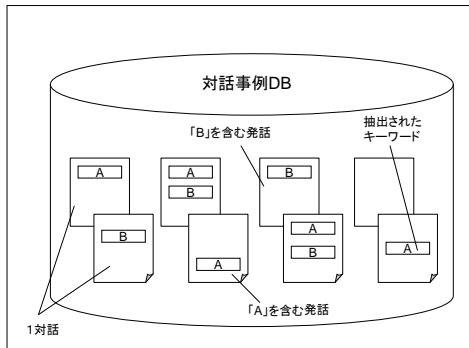


図2 対話事例DB

一般的に、共起とは一つの文に二つの単語が同時に出現することであるが、ここでは1対話中に複数の単語が同時に出現することを共起と定義する。1対話とは、あるユーザが発話文を入力してから検索結果に辿り着くまでの、システムとの対話のことである。共起関係の強さを計る方法は複数存在するが、本研究ではt-scoreを使用する。一般的にt-scoreは式(1)のように定義される[5]。

$$t\text{-score} = \frac{\text{共起頻度} - \frac{\text{個別出現頻度の積}}{\text{総数}}}{\sqrt{\text{共起頻度}}} \quad (1)$$

本研究において、式(1)の共起頻度は共起が発生している対話数となる。例えば、1対話の中で、ある単語「A」

とある単語「B」が両方出現した場合、それらは共起関係にあると見なし、単語「A」と単語「B」の共起頻度は1と数えられることになる。総数とは対話事例DB内の全対話数であり、個別出現頻度とは単語「A」が出現している対話数と、単語「B」が出現している対話数をそれぞれ示している。図2ならば総数=8、共起頻度=2となり、個別出現頻度の積は「Aの含まれる対話数」=5、「Bの含まれる発話数」=4より、 $5 \times 4 = 20$ となる。

##### 【仮想データ1】

- 「疲れた」と「甘い」の両方:22
- 「疲れた」の数:164
- 「甘い」の数:532
- 総数:100000

「疲れた」と「甘い」のt-score=4.50

##### 【仮想データ2】

- 「疲れた」と「さっぱりした」の両方:5
- 「疲れた」の数:164
- 「さっぱりした」の数:369
- 総数:100000

「疲れた」と「さっぱりした」のt-score=1.97

実際に「疲れた」と「甘い」のt-scoreを、仮想データ1を使用して算出し、同時に「疲れた」と「さっぱりした」のt-scoreも仮想データ2を使用して算出した。t-score計算で算出された値の大きい方が共起関係が強いとされることから、「疲れた」と「さっぱりした」のt-score値よりも「疲れた」と「甘い」のt-score値の方が共起関係は強いという結果となる。

##### 4.2 t-scoreとMI-scoreの比較

前述した通り、共起関係の強さを計る計算方法は一つではないが、その中でもt-scoreを使用することにした理由について述べる。ここで実際にt-scoreとMI-scoreを同一の仮想データを使用して算出し、比較する。MI-scoreの式は以下の通りである。

$$MI\text{-score} = \log_2 \frac{\text{共起頻度} \times \text{総数}}{\text{個別出現頻度の積}} \quad (2)$$

共起頻度が同じでも個別出現頻度が異なる場合、どの程度の差が生じるかを知るために、仮想データ2と仮想データ3を使用し、比較する。「疲れた」と「さっぱりした」の共起関係強度をMI-scoreで算出すると、MI-score=3.05となった。次に仮想データ3についてもt-scoreとMI-scoreをそれぞれ算出する。

##### 【仮想データ3】

- 「疲れた」と「しょっぱいもの」の両方:5
- 「疲れた」の数:164
- 「しょっぱいもの」の数:93
- 総数:100000

t-score=2.17

MI-score=5.03

MI-scoreに注目すると、「疲れた」と「さっぱりした」の共起回数と、「疲れた」と「しょっぱいもの」の共起回数が共に5回であるのにも関わらず、仮想データ2ではMI-score=3.05、仮想データ3ではMI-score=5.03と2つの値の差が大きい。この差は「さっぱりした」を含む対話

2 データベース。

数と「しょっぱいもの」を含む対話数の違いから生じており、MI-scoreは単語の頻度そのものが少ない場合、共起頻度に関係なく値が高くなってしまおうという特徴を持っているためである。このことについて、仮想データ4を使用し詳しく検証する。

【仮想データ4】

- 共起頻度:2
- ある単語「X」の出現頻度:2
- ある単語「Y」の出現頻度:2
- 総数:100000

このデータの場合、100%の確率で共起しているが、共起回数=2となっており、100000という総数の中で「X」と「Y」の共起は滅多に起こらないといえる。それに反してMI-scoreの値を計算してみると、MI-score=15.61とかなり高い値が算出されてしまう。これでは不合理であり、MI-scoreでは的確な評価付けができないといえる。同様にt-scoreを計算した結果、t-score=1.41とそれほど高くない値が出た。これにより、目的の単語が含まれている対話数に左右されずに正しい評価付けを行うことができるt-scoreを採用することにした。

5 意味素の概要

5.1 意味素の役割

t-scoreの計算は共起関係にある語に対して行うと上述したが、共起関係にある語同士に、常に関連性があるとはいえない。そこで、t-scoreはその語と関連性のある語に対してのみ算出するとし、単語同士の関連付けは意味素を用いるものとする。意味素とは語を分類するためにその語の特徴を表すもの、簡単にいえば単語の特徴を記した付箋のようなものであり、これをユーザの発話から抽出されたキーワードに持たせる。ここで、ある単語とある意味素を関連付けることを、「持たせる」という。既存の研究では、コンピュータの文法理解を支援するため、単語の文法的な位置付けに使用されているものが多い[8][9]。この考えを利用して、本システムでは漠然性を含む語を意味素として具体語に持たせることで、二つの語の関連性を表現する。例えば、「杏仁豆腐」「パフェ」「ケーキ」といった単語には「デザート」「甘い」という語を意味素として持たせる。そしてユーザが「甘いものが食べたい」というような発話をした場合に「甘い」という意味素を持つ「杏仁豆腐」や「パフェ」を提示するのである。

5.2 意味素を使用する利点

単純に共起関係の強い語をユーザへ提示するのではなく、入力される各キーワードに意味素を持たせ、目的の意味素を持つ語の中で共起関係の強い語を提示することの利点を述べる。

まず、上述したようにt-scoreの計算は共起関係にあるすべての語を対象とするのではなく、共起関係にある語の意味素を持つ語に限定してt-scoreを算出する。これによりt-scoreの計算対象が少なくなり、t-score算出の効率化が図れるということが利点の一つである。

もう一つの利点は、意味素を持たせることによって、「漠然とした語にはこの語を提示すれば具体化させられる」

という対応関係を定義付けることができるため、入力した語と関連性がないような語は提示されないようになる。これについて、具体的な例を示す。

【対話例1】

USR1:最近ちょっと疲れ気味なんだけど・・・

SYS1:甘いものはいかがですか？

USR2:甘いもの・・・パフェとか食べたい。友達と二人で行く予定なんだ。

SYS2:喫茶店はいかがですか？

対話例1を見ると、「疲れ気味」という語と「喫茶店」という語が共起として発生しており、意味素を用いなければ「疲れ気味なんだけど」とユーザが入力した際、突然「喫茶店はどうですか？」と提示してしまうような事態が起こり得る。意味素を用いることで、そのような事態を回避できる。

5.3 具体語と意味素の関連性

入力されるキーワード(2章参照)を具体語と意味素に分けると、その対応関係は表1の通りである。具体語は意味素の語よりも、どんな食べものを示しているかが明確になっている。つまり「甘いものが食べたい」とユーザが入力し、「パフェはいかがですか？」とシステムが応答したとすると、意味素に相当する語は「甘い」であり、具体語に相当する語は「パフェ」である。この時、「甘いものが食べたい」と言うよりも「パフェが食べたい」と言った方が、食べたいものに対する要求は明確であるといえる。

表1 具体語と意味素の関係

具体語	意味素
a	b
b	f
d	c, e

表1を見ると、bが具体語にも意味素にも出現していることが分かるが、これは、bが入力される語によって具体語にも意味素にも成り得るものであることを示している。例えばユーザが「疲れ気味なんだけど」といった発話をし、それに対して「甘いものはいかがですか？」とシステムが応答したとする。システムの提示した「甘い」という語がbに相当するが、この時はbが具体語として役割を果たしている。また、ユーザが「甘いものが食べたい」と発話したとすると、「甘い」という語が先程と同様にbにあたるが、この場合のbは意味素となる。ここで問題となるのは、fについてのキーワードが入力された場合であるが、この時はbに分類された語を提示し、それに対してユーザが肯定的な反応をすれば、次にaに分類された語を提示するというように、逐次的に具体化させる。

5.4 意味素を持つ条件

具体語wと意味素wが共起関係にある場合、すなわち1対話中にWとwが同時に出現した場合、Wにwを持たせる。意味素は共起が発生したと見なされれば持たせるとするが、持たせる意味素は共起回数の多い順に3つまでとする。これは、意味素によってt-score値の計算対象を制限しているのに、一つの語が意味素を多く持っていれば、結局t-score値の計算対象はさして減らないため

ある。

### 6 一連の処理動作

これまでは、提示するキーワードをどのように決定するかについて述べてきた。これは図3の具体語提示部分に相当する。次に、本システムにおける対話例の流れの全体像について説明する。

表2 具体語の提示形式

具体語	提示形式
a, d	(具体語)はいかがですか？
b	(具体語)ものはいかがですか？

システムがユーザの発話に、人間同士の対話のような自然な形で応答することが最も理想的だが、そのための課題は多く、実現は困難である。そこで、提示する具体語の分類によって、具体語提示は表2のような出力形式とした。具体語を提示するのは、ユーザ発話から抽出したキーワードが具体語に分類されなかった場合であり、入力されたキーワードと共起関係にある具体語のうち t-score 値が大きい語3つを、値が大きい順に一つずつユーザへ提示する。ただし、t-score を算出し、入力された語と意味素の語の共起関係が一番高いと見なされたとしても、t-score < 2 の場合はユーザに提示しないものとする。なぜならば、そもそも t-score とは「共起が発生したのは偶然ではない」ということを示すものであり、値が0に近いほど偶然である確率が高く、2以上の値ならば意味があるとされているためである[10]。

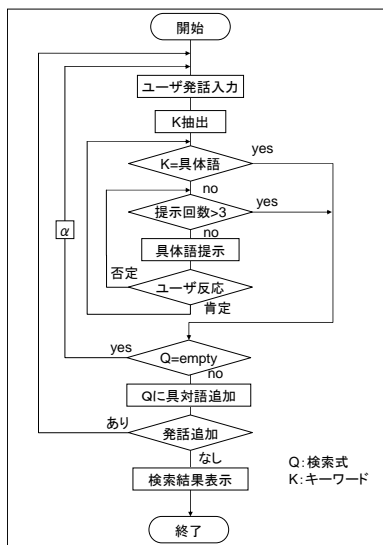


図3 検索式生成における処理の流れ

図3の具体語提示では、順に提示した3つの具体語すべてに対してユーザが否定的な反応を返した場合、図3のαの箇所です。

SYS: 具体的に食べたいものはありますか？

SYS: 他に希望はありますか？

と、ユーザに問いかけ、再度入力をしてもらう。これは、検索式に具体語が追加されるまでユーザへの発話を促すものとする。

ユーザの発話文から抽出された具体語、及び肯定的

な反応を返された提示語を、2章で述べた検索式に追加して検索を実行する。なお、提示した具体語へのユーザの返事は「良い」「悪い」の2つのみとし、「良い」と答えた場合を肯定的な反応と見なす。

この一連の処理は今後のためすべて対話事例DBに保存されるが、検索式の修正を行った場合はその修正回数に限らず、最終的な検索結果が表示されるまでを1対話として対話事例DBに保存する。

### 7 むすび

情報検索の際に入力するキーワード選択の援助を目的とした、キーワードを提示するシステムについて提案した。この提案によって、ユーザの要求が漠然性を含んでいる場合でも、その要求の明確化や検索結果の絞込みが可能となる。これによって、求めている情報に辿り着くまで何度も入力するキーワードを修正するといったような、キーワード選択に対するユーザ負担が軽減すると考えられる。

また、今回はシステムの提示語に対するユーザの返答を「良い」「悪い」の2つに限定したが、今後自然言語処理技術の研究が進めば、ユーザの発話入力へのシステムの応答は人間同士の自然な対話へとより近付くことができるかと期待される。

実証実験を実施することはできなかったが、t-score 値の算出やキーワード提示が正しくされているかについて検証し、必要であれば t-score 計算式の改良を図ることが課題として挙げられる。

### 参考文献

- [1] <http://www.stat.go.jp/>, 家計消費状況調査, 総務省統計局, 2006年.
- [2] <http://docs.yahoo.co.jp/info/research/>, 第20回Yahoo! JAPAN インターネット利用者アンケート結果, Yahoo! JAPAN, 2006年10月.
- [3] 入江友紀, 松原茂樹, 川口信夫, 山口由紀子, 稲垣康善, “意図タグつきコーパスを用いた発話意図推定手法”, 人工知能学会研究資, SIG-SLUD-A301-2, pp. 7-12, 2003年6月.
- [4] 村尾浩也, 川口信夫, 松原茂樹, 稲垣康善, “対話事例を利用した音声対話システム”, 電子情報通信学会, 2000年9月.
- [5] <http://oscar.lang.nagoya-u.ac.jp/>, 杉浦研究室, 杉浦正利.
- [6] 八木恒和, “EDR概念辞書とコーパスを用いた語義曖昧性解消に関する研究”, 東京工業大学博士論文, 2004年.
- [7] 長尾真, 人工知能と人間, 岩波新書, 1992.
- [8] 佐竹正臣, 白井清昭, 奥村学, “照応関係を考慮した新聞記事の固有表現抽出”, 第8回言語処理学会年次大会, pp. 9-12, 2002年3月.
- [9] 小山照夫, “動詞の挿入による日本語複合語の構造解析”, NII Journal, No. 2, pp. 39-44, 2001年.
- [10] 石川慎一郎, “言語コーパスからのコロケーション検出の手法”, 統計数理研究所共同研究レポートNo. 190, pp. 1-14, 2006年.
- [11] 細川健一郎, 藤江真也, 小林哲則, “検索・提案型対話システムのためのユーザとインタラクションによる適応的意図理解”, 人工知能学会研究資, SIG-SLUD-A302-5, pp. 21-28, 2003年11月.
- [12] 松尾豊, 石塚満, “語の共起の統計情報に基づく文書からのキーワード抽出アルゴリズム”, 人工知能学会論文誌, Vol. 17 No. 3, pp. 217-223, 2002年5月.