

入力質問と知識表現の柔軟なマッチングによる 対話的ヘルプシステムの構築

日笠 亘 藤井 綱貴 黒橋 禎夫

京都大学大学院 情報学研究科

〒606-8501 京都市左京区吉田本町

{higasa,fujii,kuro}@pine.kuee.kyoto-u.ac.jp

あらまし

本稿では、京都大学の約1200台の計算機教育環境において、アプリケーションなどの初歩的な質問を受け付ける対話的ヘルプシステムの構築と、その試験運用による対話データの収集・評価について述べる。本システムは、自然言語記述型の知識ベースを採用し、ユーザの入力質問と知識ベースとの柔軟なマッチングを行うことを処理の基本とする。さらに、未知語の聞き返し、ユーザ入力の文脈による補完などの処理を行う。1999年7月から試験運用を開始し、これまでに約1000件の利用があり、約30%の成功率である。

Dialogue Helpsystem based on Flexible Matching of User Query with Knowledge-Base

Wataru HIGASA Tsunaki FUJII Sadao KUROHASHI

Graduate School of Informatics, Kyoto University

Yoshidahonmachi, Sakyo, Kyoto 606-8501 JAPAN

{higasa,fujii,kuro}@pine.kuee.kyoto-u.ac.jp

Abstract

This paper describes a project constructing a dialog helpsystem which advises users in using computer facilities and application programs provided by the Center for Information and Multimedia Studies, Kyoto University. The system employs a knowledge-base written in natural language and retrieves a proper knowledge unit by flexible matching of user query with knowledge-base. The system is running since July 1999, received about 1,000 queries so far, and answered 30% of them satisfactory.

1 はじめに

人間と自然に対話できる計算機システムを実現することは自然言語処理研究の大きな目標の一つである。しかし、これまでのところドメインを限ったとしても自然な対話を行うシステムは実現されていない [Wilensky 84, Uehara 89]。これは、人間と計算機との対話データが十分に収集されておらず、対話の現象の調査が十分でないことが大きな要因と考えられる (調査例としては [熊本 99])。

本稿では、このような状況を克服するために、まず試験的な対話システムを構築し、その運用によって対話コーパスを蓄積し、さらにそのコーパスの調査によって対話システムを改良していくというプロジェクトについて述べる。

試験的対話システムとしては、ある程度の利用が見込まれ、かつ自動応答の有用性が大きいものとして、京都大学総合情報メディアセンター¹のヘルプシステムを考える。メディアセンターは京都大学内に約 1200 台の計算機を提供しており、日々多くの学生が利用している。この計算機環境において、アプリケーションなどの初歩的な質問を受け付けるシステムを構築する。

2 質問応答の初期調査と発話の種類

まず、京都大学総合情報メディアセンターにおいて、ユーザと TA との間で実際にどのような質問応答が行われているかを、録音・文書化して調査した (作業時間約 20 時間、録音時間約 1.5 時間)。その結果次のことが分かった。

- ユーザの質問のバリエーションはそれほど多くなく、かつある特定の質問が頻出する。これは、初心者の基本的な質問であることが多い。
- センターの FAQ ページなどにあげられている問題であっても、それを調べずに TA に質問する場合も多い。これらのことから、頻出する基本的質問について対応可能なシステムを作成すれば、実際上有用であり、またユーザとの対話もある程度可能ではないかという見通しがえられた。

さらに、上記の調査の結果、ユーザの発話は以下のように分類可能であることがわかった。

What 型: 事実を問うもの。

- 例) mule とはどんなものですか?
mule って何ですか?

How 型: 方法を問うもの。

- 例) mule で日本語入力する方法を教えてください。
mule で日本語入力したいのですが?

Symptom 型: 症状を提示し、その対処法を求めるもの。

- 例) 個人用のフォルダにアクセスできません。
画面が真っ白になってしまいます。

Request 型: センターに対する要求。

- 例) Visual-C をインストールして下さい。
ファイルの使用制限を増やして下さい。

Addition 型: 以前の質問の修正・追加。

- 例) WindowsNT の場合はどうですか?
返信なら?

Answer 型: システムの質問に対する返答。

- 例) WINDOWS です。
圧縮ファイルは zip ファイルです。

3 対話的ヘルプシステムの概要

本稿で述べる対話的ヘルプシステムは、前節の What 型、How 型、Symptom 型の質問に回答し、Addition 型、Answer 型などによるやり取りにも対応するものとする。なお、Request 型については対象外とする。

システムの概略を図 1 に示す。システムの各要素は以下のような働きをする。

ユーザインターフェース: ユーザとシステムの対話は、WWW ブラウザの CGI のフォーム送受信機能を紹介して行う。

入力解析部: ユーザの自然言語入力を構文解析し (KNP [黒橋 94])、さらに文末に発話タイプ (2 節) の分類規則を適用することによって、入力発話内容構文木と発話タイプを抽出する。

知識データベース: 知識ベースは、図 1 に示すように、項目とその説明文という、辞典のような形式で与える。

対話管理部: 質問処理の基本は、入力発話内容構文木と知識ベースとのマッチングを行い、最も類似度の高い部分を見つけて、それに対応する回答を表示することである。

メール送信部: ユーザが自分の入力した質問内容をセンターにメールできる機能。この機能によってユーザの質問入力の労力が無駄にはならず、システムの利用価値・利用率が増すと期待される。

¹ <http://www.media.kyoto-u.ac.jp/>

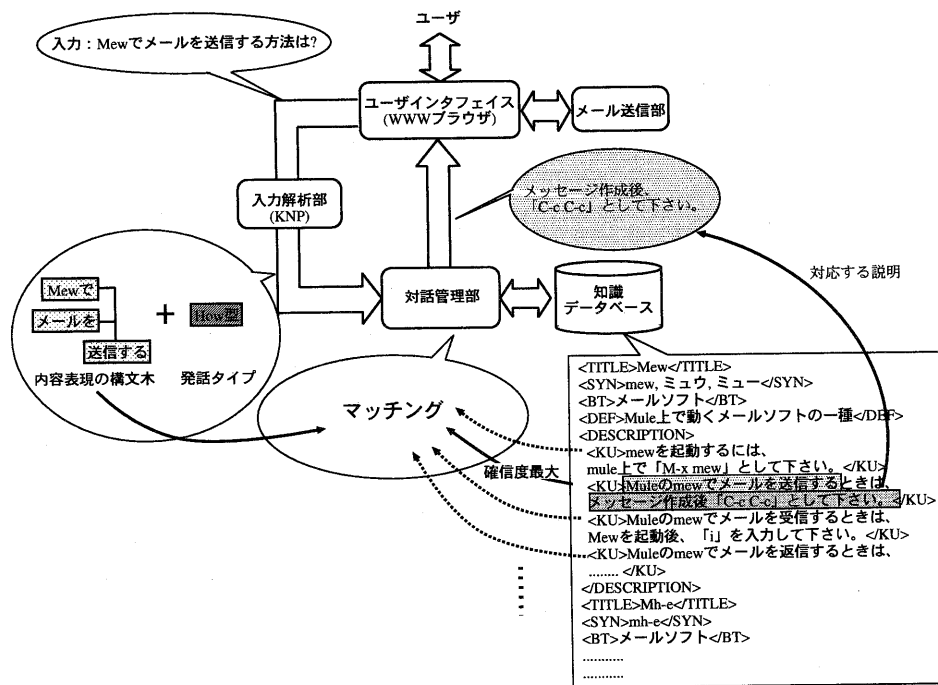


図 1: 対話的ヘルプシステムの概要

4 知識ベース構築と入力質問との柔軟なマッチング

4.1 知識ベースの概要

タスク指向型の対話システムを構築するためには、タスク (ドメイン) についての知識を、十分な表現力で、保守・拡張がしやすいように作成することが重要な課題である。

初期段階の対話的ヘルプシステムでは、知識ベースとしてスロット名・スロット値ペアからなるフレーム型ものを構築していた [日笠 99]。しかし、この方式では、一旦設計したスロットによって表現力が制限されるため、保守・拡張に非常にコストがかかるという問題があった。

そこで、現在のシステムでは図 1 のようなフレーム型と自然言語型の記述を併用する方式を採用した。知識ベースの一項目はドメイン内の種々の概念に関する説明をまとめたものである。見出し語 (<TITLE> タグ)、同義語 (<SYN> タグ)、上位語 (<BT> タグ) などの明確な部分はフレーム型で記述し、見出し語の定義 (<DEF> タグ)、種々の説明 (<DESCRIPTION> タグ) は自然言語型の記述とした。このように、自然言語型の記述を採用したことで、知識ベースの表現能力が飛躍的に向上し、知識ベースの構築作業が格段に容易になった。

知識ベースの内容は、2 節で述べた質問応答の実地調査、センターの FAQ (約 100 項目)、センターへの質問メール (約 150 通) を元に人手で作成した。現在の項目数は約 240 項目である。

4.2 自然言語による知識記述の形式

知識ベースの <DESCRIPTION> 部分には、その見出し語に関連する問題について「ある場合に、どうするか」という知識を以下の構造で記述する。

```
<KU>
  条件部 + 条件を示す表現 + 説明部
  (ある場合) (どうするか)
</KU>
```

この 1 つのユニットを知識ユニットと呼ぶことにする。

ユーザの質問は、基本的に「ある場合にどうすればよいか」という構造なので、質問の「ある場合に」の部分と各知識ユニットの条件部とのマッチングを行い、類似するものを見つけ、その説明部を表示することで基本的な対話を成立させることができる (図 1 参照)。

条件を示す表現は、人間が知識ベースを書くときの不自然さを少なくし、かつ条件部と説明部が自動認識できるよ

未知語	既知語	発話タイプ	システムの発話
1	0	What	“未知語”については分かりません
1	0	What 以外	“未知語”とは何ですか？
1	1	*	“未知語”とは何ですか？
1	2以上	*	(通常の処理)
2以上	0/1	*	質問に回答できません。
2以上	2以上	*	(通常の処理)

表 1: 未知語の扱い

5.1 未知語の処理

対話処理では、まずはじめに入力文中の各自立語がシステムのデータベース(4.3節)に登録されているかどうかを調べる。ここでデータベースに登録されていない単語はシステムにとっての未知語となる。未知語があらわれる状況は次のように分類することができる。

1. 知識ベースで全く扱っていない専門語
例) シェルスクリプト, クローンなど
2. 類義語, 関連語などは知識ベースで扱っているもの
例) モザイク, Mozilla など
3. 知識ベースで扱っている専門語のタイプミス
例) Internetmai, メーリなど
4. 一般語 例) 名前, 夏休みなど

このうち2,3の場合には、入力された語が未知語であることをユーザに知らせることによって、既知語を入力し直してくれる可能性がある。一方、4の一般語の場合は、聞き返すことに意味はないが、それを無視しても質問中の他の既知語を手掛かりに正しく回答できる可能性がある。

たとえば「プロキシサーバーの名前は何かですか？」という質問では「プロキシサーバー」が既知語、「名前」が未知語(一般語)となるが、この場合「プロキシサーバー」だけで知識ベースを検索してもよいと考えられる。

そこで、まず未知語が一般語であるかどうかを区別し、一般語である場合には特別な処理は行わない。一般語であるかどうかは、小学生用の国語辞典(三省堂例解小学国語辞典)の見出し語であるかどうかで判断する。

未知語が一般語でない場合は、未知語の数、既知語の数、発話タイプに応じて表1のように対応する。

未知語が1語の場合、What型以外であれば「“未知語”とは何ですか？」という聞き返しを行う。一方、「“未知語”って何？」のようなWhat型の質問に対してこの聞き返しを行うのは不適切なので「“未知語”については分かりません」と発話する。

最大確 信度(%)	候補ユニット	
	単一	複数
100 ~ 60	説明部 を表示	(差異が1カ所) 「“差異の列挙”から“上位概念” を選んで下さい」 (差異が2カ所以上) 「少し曖昧な質問ですので“条件 部の列挙”から選んで下さい」
60 ~ 30	説明部 を表示	「少し曖昧な質問ですので“条件 部の列挙”から選んで下さい」
30 ~ 0	「質問に回答できません」	

表 2: 候補ユニットの扱い

未知語があったとしても既知語が2語以上ある場合には、それらを基に知識ベースの探索がある程度可能であるので、聞き返しは行わない。

以下に、未知語の聞き返しを行う対話例を示す。

U1: Internetmai は使えますか？
S1: Internetmai とは何ですか？
U2: Internetmail でした。
S2: メディアセンターでは新たに OutlookExpress4 をインストールした為利用できません。

ユーザの発話 U1 は How 型、「Internetmai」が未知語、「使える」が既知語であるので、システムは未知語の聞き返しを行う。このときシステムは既知語1語の Answer 型を期待する内部状態となる。

ここでユーザが期待通りに U2 のような言い直しを行えば、この「Internetmail」を、保持している U1 の「Internetmai」に上書きし、「Internetmail は使えますか？」という質問と解釈して知識ベースを探索する。この結果、S2 のような回答が可能となる。

5.2 複数候補の処理

これまでに述べたように、本システムでは入力とのマッチングによって知識ユニットの選択を行う。マッチングによる確信度計算の結果、確信度最大の知識ユニットと、その90%以上の確信度をもつ知識ユニットを以降の処理対象とする。これを候補ユニットと呼ぶことにする。

次に問題となるのは、候補ユニットが複数ある場合にどうするか、また、確信度の大きさによってどのように処理をかえるかという問題である。これらについて現在のシステムでは表2のように対処している。

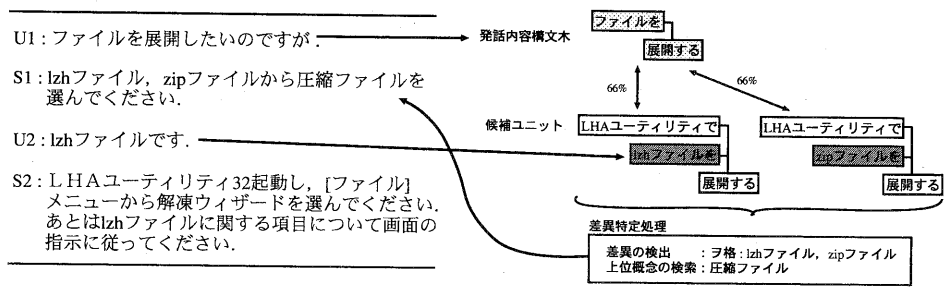


図 4: 差異特定処理

最大確信度が 30% 以下の場合には、適切な知識ユニットである場合はほとんどないので「質問に回答できません」と発話する。

最大確信度が 60% 以上で候補ユニット複数の場合は、候補ユニットの条件部の差異を検出し、それをユーザに提示して、指定を依頼する。この処理を**差異特定処理**と呼ぶことにする。

図 4 に差異特定処理の例を示す。ユーザの発話 U1 によって知識ベースを検索した結果、図 4 に示すように 2 つの候補ユニット (確信度 66%) が得られる。ここで、候補ユニットの条件部の差異を調べると、「フォ格: lzh ファイル, zip ファイル」という差異が検出される。

差異特定処理では、このように差異が 1ヶ所であれば、その共通の上位概念をドメイン・オントロジーから検索し、「差異の列挙」から“上位概念”を選んでください」と発話する。図 4 の例では上位概念が「圧縮ファイル」であるので S1 のような発話となる。ここでユーザが U2 の発話を行えば、それに対応する候補ユニットを選択し、S2 のようにその説明部を表示する。

候補ユニット間に差異が 2 カ所以上ある場合は、上記のように差異の列挙を行ってもユーザにとって理解しやすいかたちにはならないので、各候補ユニットの条件部をそのまま列挙する発話を行う。候補ユニットの確信度が 50% 以下 (30% 以上) の場合にも、同様に条件部を列挙する。

5.3 文脈に依存した入力の処理

ユーザの発話は、それまでの文脈に関係がない、まったく新しい発話である場合と、それまでの文脈と関係を持ち、一部を修正したり、新たな条件を追加したりする場合がある。

たとえば、文脈との関係がある典型例として次のような対話が考えられる。

U1: Mew でメールを送信する方法は？

S1: メッセージ作成後「C-c C-c」として下さい。

U2: 返信する場合は

S2: Mule を起動し、返信したいメッセージにカーソルを移動して下さい。本文の引用をする場合は“A”を、引用無しの場合は“a”を押して下さい。

この場合のユーザ発話 U2 は U1 の一部修正であり、「Mew でメールを返信する方法は？」ということを用意している。

このような文脈依存の発話を扱うために、本システムでは**文脈補完処理**を行う。文脈補完処理は、ユーザの発話中の各語を、直前の発話内容構文木の各語と比較し、同一カテゴリーのものがあれば入れ替えを行い、同一カテゴリーのものがなければ追加を行う。ここで、カテゴリーの同一性はドメイン・オントロジーの最上位カテゴリー (4.3節) が同じかどうかで判断する。そして、修正・追加された発話内容構文木によって通常の知識ベースの探索を行う。

上記の例では、U2 の「返信」が U1 の「送信」と同一カテゴリー (動作語) であるので置き換えられ、その結果 S2 の回答がえられる。

上記のユーザ発話 U2 は Addition 型であるため、それ自身が文脈に依存しない新たな質問である可能性はなく、文脈補完処理だけを行う。一方、発話タイプが他の型の場合には、その発話だけでは文脈依存であるかどうかは判断できない。そのため、文脈補完処理と、その発話自身による通常の処理の両方を行い、知識ベースとのマッチングの結果大きい確信度が得られる方の解釈を優先する。このような、文脈補完処理と通常の処理の両方を行うのは、発話タイプが次のように遷移する場合とする。

How 型 → How 型

Symptom 型 → Symptom 型

What 型→ What 型

What 型→ How 型

これ以外のタイプ遷移の場合には、文脈依存であることはないと考え、通常の処理のみを行う。

たとえば、What 型から How 型の連鎖で、文脈との関係がある例として次のような対話が考えられる。

U1: Mewって何ですか？
S1: Mule 上で動作するメールソフトの一種です。
U2: 起動方法は？
S2: Mule を起動してから mule 上で “M-x mew” としてください。

この場合はユーザ発話 U2 は U1 の追加情報であり、「Mew の起動方法は？」と解釈する。

また、How 型から How 型への連鎖の場合でも、文脈との関係がなく、まったく新しい質問が行われる例として次のような対話が考えられる。

U1: Outlook でメールを送りたい。
S1: まず OutlookExpress を起動して “メールの作成” を行ってから、送受信ボタンを押してください。パスワードを要求された場合はワークステーションにログインするときのパスワードを入力してください。
U2: unix で zip ファイルを展開する方法は？
S2: コマンドラインで “unzip [zip ファイル名]” と入力してください。

この場合は、「unix で Outlook で zip ファイルを展開する方法は？」という文脈依存の解釈と、U2 単独の解釈の両方の可能性を調べるが、前者の方がより確信度の高い知識ユニットをえられるので、この解釈を優先する。

5.4 対話を円滑に行うための処理

以上の対話管理の処理に加えて、対話を円滑に行うために次のような工夫をしている。

- 挨拶 (こんにちは、ありがとう) 等に対しても例外処理によって返事を行う。
- どのような場合でも、2 回続けて同一の発話は行わない。たとえば、未知語の聞き返しを 2 回続けて行う場合も、「“未知語”って何ですか？」「やはり“未知語”が分かりません。表現をかえてお願いします。」と表現をかえる。

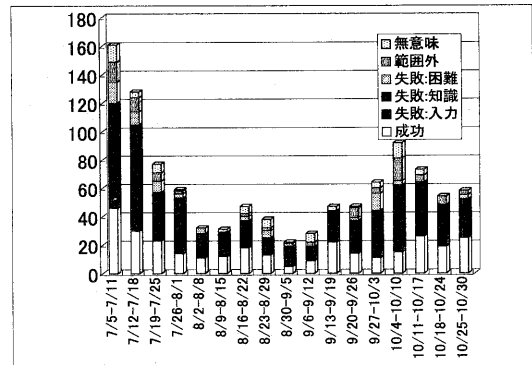


図 5: 対話データの評価 (週当たり)

- システムの回答発話に含まれる単語に対してユーザが What 型の質問をして、知識ベース中にその定義記述がない場合、「“未知語”については分かりません」と発話するのではなく「それについては詳しい説明ができません」と発話する。

6 対話データの評価

本システムは、1999年7月5日より京都市総合情報メディアセンターのホームページで公開し、試験運用している。この間の対話ログはすべて保存し、対話データベースとして整理している。

対話データベースでは、タスク指向対話としての意味的まとまりを人手でタグ付けしている。このまとまりをセッションと呼ぶことにする。1セッションは、質問・回答の1ペア、質問・聞き返し・再質問・回答の2ペアなど、様々な場合がある。

図5は公開日(7月5日)から10月29日までの週当たりのセッション数とその評価を示したものである。週平均60件程度のセッションがある。1ユーザは1回のアクセスで平均2.6セッションの対話を行っており、1セッションの平均入力数は1.9回であった。

セッションの評価としては、以下のいずれかのタグを与えている。

- 成功: ユーザに適切な回答が提示できた。
- 失敗-入力: 入力解析の失敗が原因で適切な回答ができなかった。
- 失敗-知識: 知識の不足が原因で適切な回答ができなかった。
- 失敗-困難: 現在の機能では適切な回答が不可能。
- 範囲外: システムが想定するドメインを逸脱するもの。

無意味：質問として意味がない。

これまでの全期間の成功率（範囲外、無意味のものを除いた上での成功の割合）は約3割である。ただし、対話管理機能の拡張、知識ベースの拡充などにより成功率は徐々に上昇している（最近3週間の成功率は39%）。

表3に実際の対話例を示す。

7 おわりに

本稿では、対話的ヘルプシステムの構築について述べた。当初の目標どおり、公開しているシステムはかなり利用されており、対話の成功率も試験システムとしては満足できるものであると考えている。

本システムの運用によって対話のデータベースを構築することができるので、これをもとに対話現象の調査を行い、ユーザモデルの扱い、ユーザ意図の推論などの本格的な対話の研究へのあしがかりとしたい。

また、本システムはドメイン独立であるように設計しているのので、知識ベースを記述するだけで、他のドメインの対話システムを構築することができる。この点をいかし、さらに多様な対話現象のデータを収集するために、京都大学附属図書館の参考調査業務の対話システムの構築を行う予定である。

参考文献

- [Wilensky 84] R.Wilensky, Y.Arens and D.Chin, Talking to UNIX in English: An overview of UC, Comm. of the ACM, vol. 27 (1984)574-593.
- [Uehara 89] Kuniaki UEHARA: An Intelligent On-Line Help System: ASSIST, Future Generation Computer System 5 (1989) 11-20
- [黒橋 94] 黒橋 禎夫, 長尾 眞: 並列構造の検出に基づく長い日本語文の構文解析, 自然言語処理, Vol.1 No.1 (1994)
- [熊本 99] 熊本 忠彦, 伊藤 昭: WWWを介した対話システムとの対話におけるユーザの振舞いについて, 情報処理学会論文誌 Vol.40 No.2 (1999)
- [日笠 99] 日笠 亘, 黒橋 禎夫: 大学における計算機教育環境での対話的ヘルプシステムの作成, 言語処理学会第5回年次大会 (1999)

<Thu Jul 08 19:31:49 1999>

U1: UNIXでスクリーンセーバを起動する方法は?

S1: 「スクリーンセーバ」というキーワードが分かりません。「スクリーンセーバ」とは何ですか?

U2: スクリーンセーバー。

S2: コマンドラインで“xlock”としてください。

<Thu Aug 19 21:20:52 1999>

U1: muleでメールを読むにはどうすればいいですか。

S1: “mh-e” “mew” からメールソフトを選択してください。

U2: mh-eを使います。

S2: まだ mh-e を起動していない場合は、mule 上で “M-x mh-rmail” として下さい。新しくメールを取り込みたい場合は “i” を押してください。メールにカーソルを合わせて “.” を押すと内容が表示されます。

<Thu Sep 16 22:32:20 1999>

U1: UNIXで、Mewを使うにはどうすればいいのか。

S1: Mule を起動してから mule 上で “M-x mew” としてください。具体的な使い方は “http://www.mew.org/” を参照してください。

U2: メールを消去するにはどうすればいいのか。

S2: Summary モードで “d” を押して ‘D’ マークを付けた後に、実行の “x” を押してください。

<Mon Oct 25 12:42:28 1999>

U1: Netscapeの使い方を教えてください

S1: 左下の [スタート] ボタンをクリックし、その後、[プログラム] → [Netscape Navigator] の順でクリックしてください。

U2: スタートボタンって何ですか

S2: 「スタートメニュー」は Windows の機能をまとめて一覧にしたものことで、デスクトップ画面左下に表示される [スタート] アイコンをクリックして出てくるプルダウンメニューのことです。

U3: クリックとは何ですか?

S3: クリックについては詳しく説明できません。

表 3: 実際の対話例