

Wizard of Oz 法を用いた学習対話型エージェント

岡本 昌之* 山中 信敏

京都大学大学院情報学研究科 社会情報学専攻

〒 606-8501 京都市左京区吉田本町

{okamoto,nobutosi}@kuis.kyoto-u.ac.jp

概要

本研究の目的は、案内タスクにおける人間同士の実際のテキスト対話から事例を収集しながら徐々に対話モデルを獲得し、人間によるガイドの負荷を軽減する対話型エージェントの構築である。

対話型エージェントを構築するための対話事例の収集や対話モデルの実装にはコストがかかるが、本稿では Wizard of Oz (WOZ) 法と呼ばれる対話事例の収集手法に機械学習を組み合わせることで、徐々に事例収集の負荷を軽減する対話型エージェントの構築手法を提案する。この手法には (a) 事例に基づく開発、(b) 人間と機械との協調によるシステム開発、の 2 つの特徴がある。最初は、システムほとんど適切な応答ができないが、事例が増えるに従って wizard の役割はシステムによって置き換えられる。

我々は、本手法を用いて事例からの対話型エージェント構築ツールを実装した。対話モデルには有限状態機械を用いた対話モデルを用い、漸次的学習のために拡張した確率決定性有限オートマトンの学習アルゴリズムを用いた。また、実際に京都観光案内タスクでエージェントを構築し、ガイド役の人間の負荷軽減、および対話モデル学習にかかるコストの減少を確認した。

1 はじめに

本研究の目的は、Web を介した案内タスクにおいて、人間同士の実際のテキスト対話から事例を収集しながら徐々に対話モデルを獲得し、人間によるガイドの負荷を軽減する対話型エージェントの構築である。

近年、対話を通じて Web サイトや、サイト内のコンテンツを紹介する様々なソフトウェアエージェント、なかでも対話型の Web エージェントが増えつつある。例えば、Extempo¹にはパーテンドー、車のセールスパーソン、掃除に関するアドバイザー等のエージェントがあり、デジタルシティ京都の二条城バスツアーガイド²は、仮想的なバスツアーグループの参加者に対して 3 次元モデルで構築された二条城の紹介を行う。

これらの対話型エージェントはガイドの役割を果たし、ユーザが全ての文章を読んだり、リンクを辿って目的のページを調べる手間を省く。多くのエージェントはインタフェースキャラクタやグラフィカルなインタフェースを用い、単に文章を読み上げるよりも効果的に情報を提示する。

これらのエージェントを設計、開発する場合、シナリオやルール、キーワードを記述する手法と、統計的モデルを用いる手法がある。前者では、シナリオ記述ツールや、PPP Persona で用いられるプレゼンテーションのプランニング技術 [2] を用いることができる。また、後者では、マルコフ決定過程を用いたモデルの研究が数多く行われている [7, 13]。いずれの手法も対話モデルを構成するシナリオ、ルールの記述や、統計モデルを学習するための事例自体はあらかじめ開発者が記述、収集する必要がある。

統計的手法においては学習用事例の収集が困難であり、ルールベースの場合、徐々にルールを追加・修正するのはルールが複雑になるに従って困難になる。そこで、事例の増加に応じて良くなるエージェントの応答精度を次の事例収集に利用すれば、事例収集にかかる負荷や、途中で当初想定されていない対話のパターンが加わった場合の修正にかかる負荷を軽減できると考えられる。しかし、実際の人間同士の対話事例を集めながら対話モデルを獲得し、徐々にガイド役の人間の代わりに案内を行うことで、事例収集の負荷を軽減する学習対話型エージェントに関する研究は行われていない。

本稿では、Wizard of Oz (WOZ) 法と呼ばれる対話事例の収集手法に機械学習を組み合わせた対話型エージェントの構築手法 [10, 11] を提案し、事例からの対話型エージェント構築ツールを紹介する。また、実際に京都観光案内タスクでエージェントを構築し、徐々にガイド役の人間の負荷が軽減することを確認する。

2 Wizard of Oz 法への学習機構の適用

Wizard of Oz (WOZ) 法 [4] は、開発中の対話システムの効果的なシミュレーション手法である。これは、マイク、合成音声、ネットワーク越しのテキスト入出力等を用い、システムのふりをした人間 (wizard と呼ぶ) が、ユーザと対話するというものである。ユーザは実際にシステムを相手にしていると思いながら対話を行うため、得られ

¹<http://www.extempo.com/>

²<http://www.digitalcity.gr.jp/openlab/agentpage-j.html>

るデータは人間同士の対話によって得られる場合とは異なり [3]，より実際のシステムに近い状況での対話データになる．WOZ 法はこれまで，主に音声対話システムの領域で用いられてきた．ヒューマンコンピュータインタラクションの領域においても，教示エージェントのシミュレーション等にこの手法が用いられている [8]．

WOZ 法は完全には開発されていないシステムの挙動をシミュレートするのに適した手法である．つまり，開発前にシステムの挙動を確認するような場合（全て人間が行う場合）や，システム全体が完成していない状態で，未開発の箇所を補完しながら既に完成している箇所の動作を確認するような場合に有用である．例えば，ユーザ発話の認識だけ wizard が行い，推論はシステムが行うという方法で対話コーパスを収集する試み [6] が行われている．

もともとの WOZ 法はあくまでシミュレーションの手法であり，設計，開発者はシミュレーションの後，実験結果を踏まえて実際のシステムを開発する必要がある．また，プロトタイピングのために用いられた対話それ自体がシステムに直接反映されるというわけでもない．しかし，学習機構を適用することで，WOZ 法で収集される対話事例を全て吸収し，次の対話に活かすことができれば，そのタスクやシナリオに応じたタスク指向の対話型エージェントを漸次的に構築可能である．このようなエージェントの学習過程を図 1 に示す．

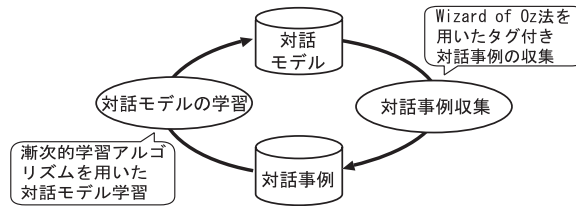


図 1: エージェントの学習過程

また，その手順は以下のようになる．

1. WOZ 法を用いて最初の対話事例を収集する．最初だけは通常の WOZ 法であり，実際にはユーザと wizard によるチャットが行われる．
2. 対話事例から，対話モデルを学習する．ここで学習されたモデルは小さく，十分な推論はできない．
3. 再び WOZ 法による事例収集を行う．ここで，システムは学習された対話モデルから推論を行い，wizard に対して発話候補を提示する．wizard は推論結果が正しければそれを利用し，正しくなければ（先のステップと同様に）自分で発話を入力する．
4. 収集された発話から，再び対話モデルを学習する．このモデルは先のステップよりも大きなものとなる．
5. 上記のステップを繰り返すうちに，徐々に対話型エージェントが構築される．

この枠組みには次の 2 つの特徴がある．

事例に基づく開発 開発者が（ユーザはどのように発話するだろうと）推測する発話ではなく，ユーザと wizard が実際に行う発話がシステムに使われる．ただし，事例に基づく場合には，状況や目的が明確でないと対話も発散してしまう．逆に，状況が設定されればユーザの行為は状況的なものとなり [14]，また人々の行う行動の範囲は自ずと制限される [12]．

人間と機械との協調によるシステム開発 wizard は学習を手助けする役割を持つ．最初は，少しだけの事例から作られるシステムはほとんど適切な応答ができないため，wizard が応答を入力する．これは wizard が教師となって事例を与えていることに相当する．事例が増えてシステムが拡充されると，エージェントの役割は wizard からシステムに置き換えられる．このとき，システムは完全なインタフェースエージェントとして機能する．したがって，我々は WOZ 法を学習するインタフェースエージェントのための人間と機械との協調による開発手法であると考える．

従来のコーパスを用いた対話モデル構築の手法と比べると，本手法ではコーパス収集におけるコストを軽減できる．つまり，システムの推論結果を利用する頻度が増えるに従い，wizard が応答を入力あるいは選択するコストが減少する．

また，実際のユーザと対話する WOZ 法を用いてコーパスを収集する場合，コーパスの品質，特にユーザ発話の品質は wizard の応答時間による影響が大きくなる．例えば，ある検索要求に対して wizard が応答する場合，wizard の応答が遅いとその後エージェントに対するユーザの発話もぞんざいになる，といったことが起こる．学習を組み合わせた方式では，上述のように wizard のコストが下がる分だけ応答時間も短縮されるため，コーパスの品質低下を防ぐ効果があると言える．

上述の手順によるエージェント構築には，以下の要素が必要となる．

- 良い品質の対話事例を収集する環境
- エージェントを構成する内部モデルと学習メカニズム
- 案内に必要となる知識

これらの実現例を、次章で述べる。

3 WOZ 法に基づくエージェントシステム

我々は前章で述べた機構を備えた、WOZ 法に基づく Web 上の対話型エージェント構築支援システムを実装した。このシステムは、比較的単純なタスクにおいて、容易に対話エージェントを構築するためのものである。

本章では、支援システムの概要と、知識・話題の管理、及び対話モデルとその学習アルゴリズムについてそれぞれ説明する。

3.1 システム構成

システムの構成を図 2 に示す。システムは Java 言語で実装されており、Microsoft Agent³を用いたインタフェースキャラクタを備えている。また、形態素解析には奈良先端科学技術大学院大学の茶筌⁴を用いている。

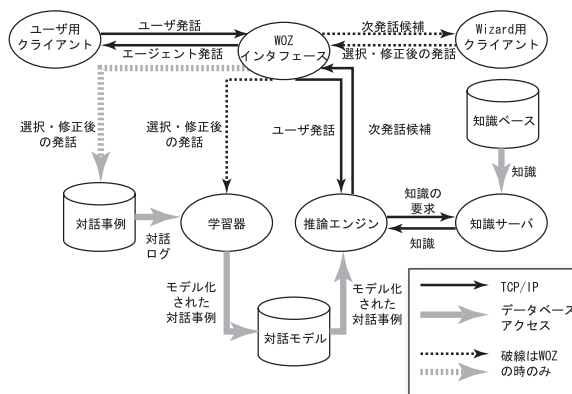


図 2: エージェントシステムの構成

クライアントは、ユーザ用 (図 3(a)) と wizard 用 (図 3(b)) の 2 種類があり、ともにテキストチャットのインタフェースを備えているが、wizard 用クライアントについては WOZ 法のために以下の拡張機能が利用される。

- システムからの応答候補，想定発話リストの選択メニュー
wizard はシステムの推論した応答の候補から選択するためのメニューと，タスクに応じてあらかじめ決められた発話の候補から選択するためのメニューを利用して発話を行う。
- 発話へのアノテーション
システムの処理を単純にするために，wizard は発話へのアノテーション (タグ付け) を行う。
wizard はユーザ，エージェントの各発話に対し，タスクに応じてあらかじめ決められた発話タグ (発話の分類) を付加する。
また，発話内部には紹介に関するタグと URL に関するタグの 2 種類を付加する機能がある。紹介に関するタグは，知識ベースから文章を呼び出して発話に埋め込むために使われる。URL に関するタグは，エージェントの発話と Web ページの表示を連動させるために用いられる。

3.2 有限状態機械を用いた対話モデル

対話モデルには，マルコフモデル，隠れマルコフモデル，n-gram モデル等の統計モデルに基づくモデルや，有限状態機械 (Finite-State Machine : FSM) に基づくモデルなど，様々なアプローチが考えられるが，本稿で述べるシステムでは，対話を発話の系列とみなし，また開発者が経過を解釈可能なモデルとするために，FSM を用いた対話モデルを用いる。

³<http://www.microsoft.com/msagent/>

⁴<http://chasen.aist-nara.ac.jp/>

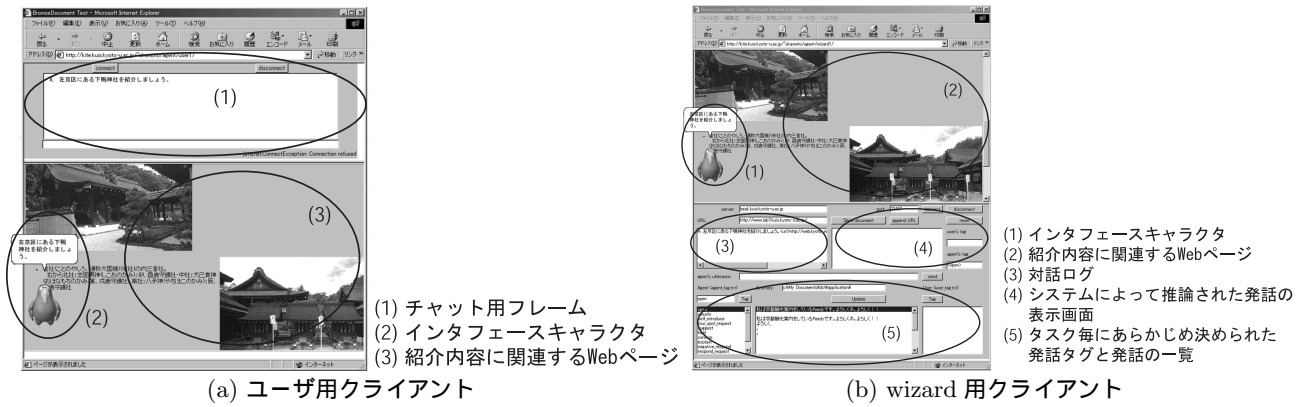


図 3: クライアント

FSM を用いた対話型のシステムは数多く提案されている (例えば, [1]) が, 我々の手法では, キーワードを用いた簡単なテキスト処理を用いて, タスクに応じた対話を行うことを試みる.

このシステムでは, 対話を発話の列であると考え, それぞれの発話は (a) 話者 (ユーザまたはエージェント), (b) 発話の内容, (c) タスクに依存する発話タグ, で構成される. つまり, 各対話は発話タグの系列であり, それぞれの発話タグに対して単語列が付随している, と捉える. これらの系列から構成される FSM 上の遷移は, 発話タグの粒度で対話の流れを表していると言える.

FSM の学習には, 確率決定性有限オートマトン (PDFA) の学習アルゴリズムの 1 つである MDI アルゴリズム [15] を漸次的な学習に拡張したアルゴリズム [9] を用いる. アルゴリズムの動作を以下に述べる.

- (1) 事例の系列から樹状の PDFA (PTA : Prefix Tree Acceptor) が作られる (図 4(a)).
- (2) 入力シンボル系列の辞書式幅優先の順序⁵で, 2 状態の組について, 状態をマージする場合としない場合での同値性を調べる. マージする際に, 遷移が非決定的になるような状態が生じれば (図 4(b)) 再帰的にマージされる (図 4(c)).

同値性の判定はマージ前の PDFA と PTA との Kullback-Leibler 距離 (相対エントロピー) と, マージ後の PDFA と PTA との Kullback-Leibler 距離の, 縮退した状態数あたりの差分によって行う. つまり, 元の PTA を P , マージ前, マージ後の PDFA をそれぞれ A, A' とおき, A, A' の状態数をそれぞれ $|A|, |A'|$, 2 つの PDFA A_1 と A_2 に対する Kullback-Leibler 距離を $D(A_1||A_2)$ と表すと, パラメータ α_M に対して

$$\frac{D(P||A') - D(P||A)}{|A| - |A'|} < \alpha_M$$

が成り立つときに同値とみなされる. これは, 2 状態を 1 つにまとめたときに, 遷移可能なシンボル系列の集合がパラメータ α_M の基準において近いことを表している.

- (3) 同値である場合, その 2 状態は 1 つの状態にマージされる. 元の MDI アルゴリズムでは, 徐々に学習データが増えることを考慮していないため, 次の学習では同値性について再度計算する必要があるが, 我々の手法では, マージされた 2 状態の組を記録することで, 次の学習で再度同じ計算を繰り返すことを防ぐ.
- (4) 最終的に, 汎化された PDFA が構成される.

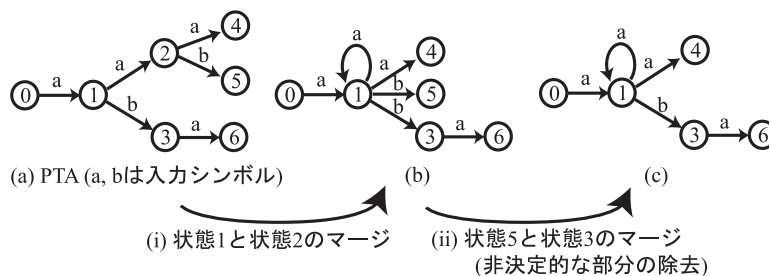


図 4: 状態のマージング

本システムでは, それぞれの発話タグは FSM の入力シンボルを表し, 学習された FSM は対話モデルを表す, また, 元の対話事例は尤もらしいユーザからの入力とエージェントからの出力の集合を意味する. 図 5 は対話モデルとしての FSM と対話事例の対応を表す.

⁵ 入力シンボルが a, b の 2 種類の場合, λ (空入力), $a, b, aa, ab, ba, bb, \dots$ の順序となる. 図 4(a) の状態番号はこの順に付けられている.

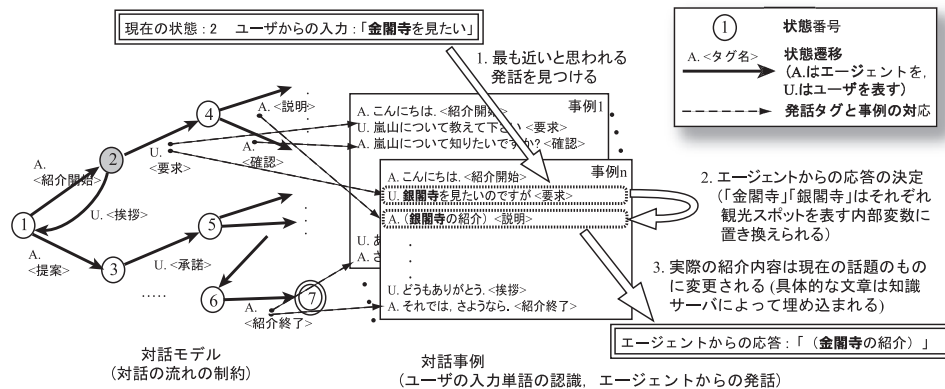


図 5: 対話モデルと対話事例

```
<tourspot>
<name> 金閣寺 </name>
<alias> 鹿苑寺 </alias>
<category> 寺 </category>
<url>http://www. ...</url>
<content> 金閣寺は室町時代に...</content>
<way> 市バス 系統で...</way>
<fee> 大人は 円 </fee>
</tourspot>
```

図 6: 観光スポットを表す XML 文の一部

ユーザの発話認識には、形態素解析された単語についてキーワードマッチングとキーワードベクトルの内積を用いたスコア付けを用いて、過去の履歴中の発話で最も近いものにマッピングするという手法を用いる。候補の計算は全ての発話に対して行うのではなく、あらかじめ状態遷移図上で遷移可能な発話タグを持つ履歴についてのみ行う。ユーザ発話に対して応答の候補を推論する例については、3.4 節で述べる。

3.3 知識・話題管理

タスクに必要とされる知識については、2つのアプローチが考えられる。それほど大きくない Web サイトの紹介など、紹介対象のコンテンツがある程度限定されている場合は知識自体も事例として取り扱うのが簡便である。紹介対象のコンテンツが膨大な場合、知識を事例として扱うのは現実的でない。このような場合、テンプレート等を用いて、1つの発話で多数の知識に対応できるようにする必要がある。本稿のシステムでは、後者の手法で、Web ページと関連付けるために XML によるアノテーション情報を置いた知識ベースを用いる。それぞれのコンテンツは案内の際にキーワードベースで呼び出され、発話を構成する。

アノテーションによって、各コンテンツについて、コンテンツ自体の名前、コンテンツの他の呼び方、カテゴリ (観光案内の場合、寺、神社のような観光スポットの種類等の分類)、紹介する本文、付加的な情報 (観光案内の場合、交通手段、利用料等)、の各情報が蓄積される。

例として、観光スポット「金閣寺」を表す XML の一部を図 6 に示す。この例では、name がコンテンツの名前、alias が別名、category がカテゴリ、content が紹介する本文、way、fee が付加的な情報をそれぞれ表す。

これらの構造を話題管理に用いることで、話題の変化に対して矛盾ない応答を返す。

管理される話題は (a) 紹介中のコンテンツ、(b) そのコンテンツの属するカテゴリ、から構成される。

あるコンテンツを紹介する状況は 3 種類に分類される。それぞれの状況における、話題の制約は以下の通りである。

ユーザからの直接の要求 ユーザから「金閣寺について教えて下さい」のように直接要求される場合、システムはそのまま金閣寺の紹介を行う。この時、コンテンツは「金閣寺」に、カテゴリは「寺」にセットされる。その後で「行き方を教えて下さい」のように付加的な情報を要求された場合、現在の話題「金閣寺」について交通手段を説明する。

ユーザからの間接的な要求 ユーザから「おすすめのお寺を教えてください」のように間接的に要求される場合、システムはカテゴリ「寺」に属するコンテンツの紹介を (エージェント毎のコンテンツの優先順位に従って) 行う。あるいは、コンテンツ「金閣寺」を紹介した後で「他のお寺を教えてください」のような要求を受けた場合にも、現在のカテゴリ「寺」に属するコンテンツを選択・紹介する。

エージェントからの提案 エージェントから「お寺を紹介しましょうか?」等の提案を行う場合、発話時にカテゴリ「寺」をセットする。これに対し、ユーザが「はい、お願いします」のように提案を承諾する場合、現在のカテゴリに属するコンテンツ (寺) を紹介する。

3.4 動作の流れ

本節では、システムの利用方法について述べる。

あるタスクにおいて、対話エージェントを構築するためにあらかじめ必要となる要素は以下の通りである。

- タスクの遂行に必要な、発話タグの一覧
- 紹介する Web サイトと、紹介に必要なタグ付けされた付加情報
- wizard が使う発話一覧 (必須ではないが、入力の負荷を減らすためによく使う発話をあらかじめ登録する)

システムは以下のように動作する。ここで、例示する観光スポット「金閣寺」「銀閣寺」は知識ベースに登録されているものとする。

1. ユーザが発話を入力すると、推論エンジンは次発話を推論し、WOZ インタフェースを經由して wizard 用クライアントに送信する。例えば、図 5 で、状態 2 のときにユーザから「金閣寺を見たい」という発話が入力されると、システムは履歴中で状態 2 から遷移できるユーザの発話タグと、それに対するエージェント発話の発話タグの組を探し、ユーザの入力と履歴中のユーザ発話の単語ベクトルの内積を計算する。また、知識ベースに含まれるキーワード「金閣寺」は観光スポットを表す内部変数に置き換えられる。図 5 では、スコアの高い発話として「銀閣寺を見たいのですが」が選択され(「銀閣寺」も同様に観光スポットを表す内部変数に置き換えられている)、それに対するエージェントからの応答として、履歴中のスコアの高い発話への応答として、(銀閣寺の紹介)が選択される。最終的に、現在の話題である金閣寺を用いた(金閣寺の紹介)が応答候補となる。wizard がいない場合には、最もスコアの高い候補がエージェントからの応答として直接ユーザに送られる。
2. wizard が候補から 1 つ選択すると、推論された発話とその発話タグは推論エンジンに送られる。適切な候補がない場合、wizard はシナリオに応じてあらかじめ作られた発話候補のリストから 1 つ選び、発話タグを追加・修正する。同時に、発話時に Web ページを表示する場合は URL を付加する。推論エンジン上では、発話タグに応じて状態遷移が行われる。
3. いずれの場合も、wizard から送信される発話はエージェントの発話としてユーザ用クライアントにも送信される。発話に URL が含まれる場合、対応する Web ページがユーザ用クライアントに表示される。

4 使用評価

本章では、京都の観光案内を行うエージェントを実際に構築し、事例が増加した時の wizard の負荷の変化について調べる。

このシナリオでは、最初に、エージェントは自己紹介を行う。次に、ユーザの要求を確認するか、あるいは観光スポットを提案する。要求あるいは提案内容が決まると、その観光スポットについて紹介する。説明の後、他の観光スポットへとユーザを紹介するか、要求がなければ対話を終了する。

このシナリオで用いた発話タグの一覧を図 7(a) に示す。

状況	エージェント	ユーザ
1. 導入	紹介開始の挨拶 自己紹介	自己紹介
2. 興味の確認	TS への誘導 TS の確認	TS の要求 曖昧な TS の要求
3. 提案	TS の提案	提案の承諾 提案の拒否
4. 紹介	TS の紹介 TS に関する質問 への回答 紹介不可	TS に関する質問
5. 他の TS への誘導	他の TS への誘導	誘導の拒否
6. 終了	紹介終了への誘導 紹介終了	紹介終了の承諾 紹介終了の拒否
7. その他		相槌 判別不能

(a) タグの一覧

A. 私は京都観光案内をしている Peedy です。よろしくね。<紹介開始の挨拶>
A. 私は去年京都の二条城でガイドをしてそれから今は... <自己紹介>
A. 京都でどこか観光したいと思うところはありませんか? <TS への誘導>
U. 西のほう。<曖昧な TS の要求>
A. もう少し、具体的に教えて下さい。 <TS の確認>
U. 太秦 <曖昧な TS の要求>
A. 散策コースを紹介しましょうか? <TS の提案>
.....
U. 広隆寺の近くで食事したいのですが。 <TS の要求>
A. すみません、それはよくわかりません。 <紹介不可>
U. じゃあいいです。 <相槌>
.....
A. それでは、さようなら。 <紹介終了>

(b) 対話例

図 7: 京都観光案内タスクにおけるタグの一覧と対話例 (A. はエージェントを、U. はユーザを、TS は観光スポットをそれぞれ表す)

また、事例が増えるに従ってどの程度推論の精度が向上するかを調べた。入力対話として、観光案内タスクで WOZ 法により収集された 41 対話 (合計 1147 発話, 平均 28.0 発話) のうち, 31 対話を順に入力して学習し, 数対

話毎に学習された対話モデルの質を残りの 10 対話を用いてそれぞれ評価した．これらの対話は京都付近に済む大学生のユーザ約 10 人との間で行われたものである．対話で用いられる発話は通常の書き言葉だけではなく，インフォーマルなテキストチャットで用いられるような話し言葉も多量に含む．対話例の一部を図 7(b) に示す．

事例を徐々に増やす場合，評価方法として学習データとテストデータを分けずに，学習データの入力を行いながら，同時にその入力に対する推論結果を評価する，という方法も考えられるが，本節で用いる対話データは主に話者の違いによるばらつきを含むため，入力データによるグラフの上下動が大きくなる．対話例のうち，任意に選んだ 10 対話をテストデータとし，毎回同じ入力を用いることで，推論の精度に関するばらつきは防がれる．

上記の手順による，ユーザの入力に対する wizard の介入比率の推移を図 8 に示す．ここで，横軸は学習した対話の数 (1, 10, 20, 31 対話) を，縦軸はユーザ発話への応答に関する wizard の介入割合 (0 はシステムが完璧に推論できていることを，1 は完全に wizard が対話していることをそれぞれ意味する) を表す．グラフはシステムが推論した候補のうち，1 番目の候補が正しい時のみシステムからの応答が正しいとみなす場合と，3 番目の候補まで選択する場合とを表す．また，この時に行われた対話モデル学習 (パラメータは $\alpha_M = 0.1$ とした) において，学習にかかるコストとして同値性の判断回数について元の MDI アルゴリズムと漸次的アルゴリズムとで比較した結果を図 9 に示す．また，最終的に学習された対話モデルを図 10 に示す．

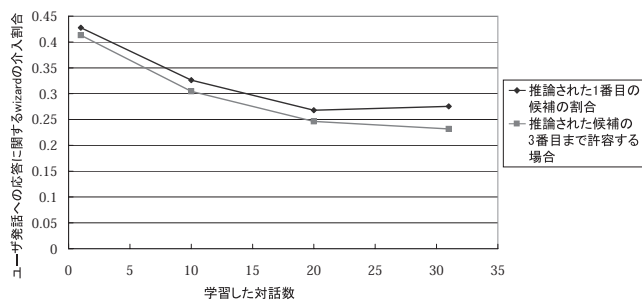


図 8: wizard の負荷の推移

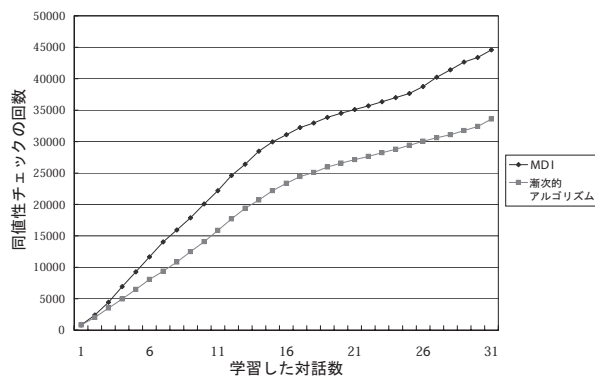


図 9: 同値性の比較回数 (累計, $\alpha_M = 0.1$)

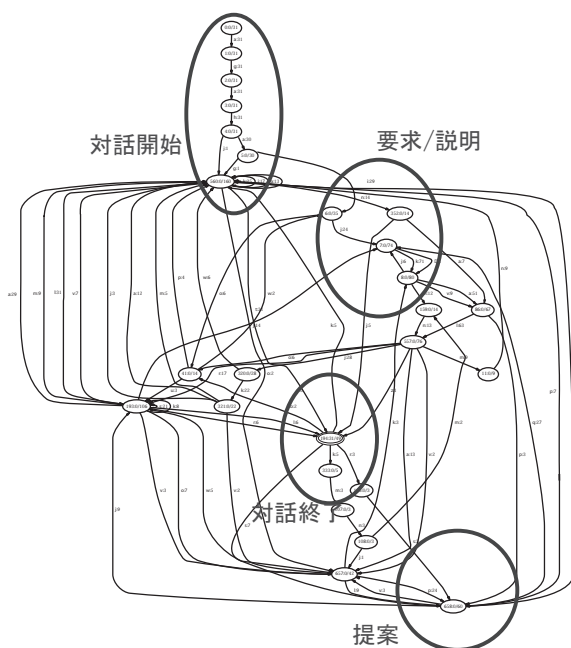


図 10: 学習された対話モデル

(いくつかの状態と遷移は省略されている)

図 8 より，学習が進むにつれて wizard の負荷が減少していることが分かる．ただし，システムが推論した発話の 1 番目だけを選択する場合，31 対話学習した結果の方が 20 対話の場合よりも性能が悪くなっている．これは，様々な単語ややりとりへの対応に関しては状態遷移図の効果が見れ，徐々に正しく対応できる発話が増える一方で，同じ単語が異なる文脈で使われる場合が増えることで，発話候補の絞り込みの時点で誤った応答も候補に含まれてしまうためである．

口語など質問時の言い回しが多岐に渡る対話に対しては，事例の増加に従って wizard の負荷が大きく減少しているが，観光スポットを直接尋ねる発話の割合が大きい場合，多くの場合最初から正しい応答を返すことができるもの，事例の増加によって逆に他の種類の発話と誤って認識されるケースが見られた．

現状の手法をそのまま用いた場合には 20 数%が限界であると考えられるが，候補の 3 番目まで選択肢に含めた場合，依然として性能は上がり続けており，発話認識の際のスコア付けや，候補からの絞り込みを改良することで更に性能を向上させる余地があるといえる．

対話モデルについては，事例が増加するにつれて対話の典型的なパターンが学習されていると言える．つまり，学習が進むにつれてエージェントからの観光スポット提案のように，特にある観光スポットの案内から別の観光スポットの案内へ話題が移る箇所に関して遷移が利用されるようになる．FSM の大きさは，31 例学習した時点で 28 状態 98 遷移となった．図 10 より，対話中の状況に対応できる箇所がある程度まとまっているの確認することができる．

この大きさの FSM は，最初に全て対話のシナリオを決めることができれば人手でも記述可能であるが，当初想定されていなかったやりとりを後から追加するような場合，人手で毎回修正するには手間のかかるものであると考

えられる。

また、図 9 より、今回の評価では全体として、漸次的アルゴリズムを用いることで比較回数が約 25%減少している。提案手法では、このような漸次的学習アルゴリズムを用いることがエージェント構築のコストを下げることに繋がるといえる。

5 おわりに

本稿では、案内タスクにおいて人間同士の実際のテキスト対話から事例ベースで漸次的に対話モデルを構築し、人間によるガイドの負荷を軽減する対話型エージェントの構築手法を提案した。

この枠組みでは、Wizard of Oz 法を用いた対話収集に学習機構を加えることで事例から漸次的に対話型エージェントが構築される。我々は、対話モデルに FSM を用いた Web 上で動作するエージェント構築支援ツールを実装し、京都の観光案内タスクに適用した。31 対話の学習を通じて、人間によるガイドの負荷が軽減されること、特に口語のような多数の言い回しを含む発話に対する効果が大きいことを確認した。

今後は、このようなガイドエージェントをデジタルシティ京都⁶のようなインターネット上の社会的基盤 [5] において活用することを考えている。

謝辞

本研究を進めるにあたり、数々の有益な助言を頂いた京都大学大学院情報学研究科社会情報学専攻の石田亨教授に感謝致します。また、対話データには新エネルギー・産業技術開発機構 (NEDO) 委託事業「シニア支援システムの開発」プロジェクトにおいて収集された対話ログを用いました。

参考文献

- [1] J. Alexandersson, E. Maier and N. Reithinger, “A Robust and Efficient Three-Layered Dialogue Component for a Speech-to-Speech Translation System,” *Proceedings of European Chapter of the Association for Computational Linguistics (EACL-95)*, pp. 188–193, 1995.
- [2] E. André, T. Rist and J. Müller, “Integrating Reactive and Scripted Behaviors in a Life-Like Presentation Agent,” *Proceedings of International Conference on Autonomous Agents*, pp. 261–268, 1998.
- [3] J. M. Carroll and A. P. Aaronson, “Learning by Doing with Simulated Intelligent Help,” *Communications of the ACM*, Vol. 31, No. 9, pp. 1064–1079, 1988.
- [4] N. M. Fraser and G. N. Gilbert, “Simulating Speech Systems,” *Computer Speech and Language*, Vol. 5, No. 1, pp. 81–99, 1991.
- [5] T. Ishida, J. Akahani, K. Hiramatsu, K. Isbister, S. Lisowski, H. Nakanishi, M. Okamoto, Y. Miyazaki and K. Tsutsuguchi, “Digital City Kyoto: Towards A Social Information Infrastructure.” M. Klusch, O. Shehory and G. Weiß(Eds.), *Cooperative Information Agents III*, Lecture Notes in Artificial Intelligence 1652, pp. 23–35, Springer-Verlag, 1999.
- [6] K. Itou, T. Akiba, O. Hasegawa, S. Hayamizu and K. Tanaka, “A Japanese spontaneous speech corpus collected using automatically inferencing Wizard of Oz system,” *Journal of the Acoustical Society of Japan - Section E*, pp. 207–214, 1999.
- [7] E. Levin, R. Pieraccini and W. Eckert, “Using Markov Decision Process for Learning Dialogue Strategies,” *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP-98)*, 1998.
- [8] D. Maulsby, S. Greenberg and R. Mander, “Prototyping an Intelligent Agent through Wizard of Oz,” *Proceedings of International conference on Human Factors in Computing Systems (INTERCHI-93)*, pp. 277–284, 1993.
- [9] M. Okamoto, “Incremental PDFA Learning for Conversational Agents,” *IEEE International Workshop on Knowledge Media Networking (KMN-02)*, 2002 (to appear).
- [10] 岡本 昌之, 山中 信敏, “Wizard of Oz 法を用いた対話型 Web エージェントの構築,” *人工知能学会論文誌*, Vol. 17, No. 3, pp. 293–300, 2002.
- [11] M. Okamoto, Y. Yang and T. Ishida, “Wizard of Oz Method for Learning Dialog Agents,” M. Klusch and F. Zambonelli (Eds.), *Cooperative Information Agents V*, Lecture Notes in Artificial Intelligence 2182, pp. 20–25, Springer-Verlag, 2001.
- [12] B. Reeves and C. Nass, *The Media Equation : How People Treat Computers, Television, and New Media Like Real People and Places*, Cambridge University Press, 1996.
- [13] S. Singh, M. Kearns, D. J. Litman and M. A. Walker, “Empirical Evaluation of a Reinforcement Learning Spoken Dialogue System,” *Proceedings of National Conference on Artificial Intelligence (AAAI-2000)*, pp. 645–651, 2000.
- [14] L. A. Suchman, *Plans and Situated Actions : The Problem of Human-Machine Communication*, Cambridge University Press, 1987.
- [15] F. Thollard, P. Dupont and C. de la Higuera, “Probabilistic DFA Inference Using Kullback-Leibler Divergence and Minimality,” *Proceedings of International Conference on Machine Learning (ICML-2000)*, pp. 975–982, 2000.

⁶<http://www.digitalcity.gr.jp/>