

発話とコマンドの系列を制御する対話エージェント

A dialogue agent that controls the sequence of utterances and commands

遠藤 充 牛尾 貴志 山上 勝義 堀井 則彰
Mitsuru Endo Takashi Ushio Katsuyoshi Yamagami Noriaki Horii

パナソニック株式会社 先端研究本部
Advanced Research Division, Panasonic Corporation

We are promoting development of a service platform to archive the tasks on the backend IT systems through a frontend natural language interface with a dialogue agent. The dialogue agent interacts with both frontend and backend by exchanging information in the form of the dialogue acts. In this paper, we report experimental results of our recipe search task on the trial test site.

1. はじめに

対人窓口業務に必要な不可欠のスキルの1つに、対話により顧客の要望を理解し、要望に合う提案をするスキルがある。例えば、不動産仲介業者は、窓口にて顧客の要望を聞き取り、多数の不動産の物件候補から、要望に合う物件を紹介する。また、医師による問診では、患者から自覚症状を聞き取り、医学知識に照らし合わせた上で、症状の出方から推定される病気を特定する。

これらの業務に共通するのは、業務遂行者と顧客が協調して対話により条件付けを行って、非常に多数の候補から最終的に一つに候補を絞り込むというタスクである。ここで、候補絞り込みのプロセスは対話を通じて行われる。すなわち、両者間の情報のやりとりが逐次的に行われ、情報のやり取りにより両者の内部状態が更新され、状態に応じて動的に次の行動が決定される。したがって、ほとんどの場合、対話による絞り込みには複数回のターンが必要である。

我々は、このような対話によりユーザの意図を理解可能なシステムの実現を目指している。本稿では、データベース検索を含むタスク管理のコマンド系列を、ユーザとシステムのやりとりを表す対話行為の系列と同列に扱う。ドメイン情報を含めないことによりコンパクト化した対話状態を介して対話を制御しても、系列内でドメイン情報の受け渡しを行うことで、コンテンツの絞り込みタスクが達成できるという仮説を検証する。

2. 提案システム

提案システムの構成を図1に示す。提案システムは、フロントエンド、対話エージェント、バックエンドシステムからなる。

2.1 フロントエンド

フロントエンドは、ユーザからの入力テキストを言語解析し、対話行為を対話エージェントに出力する言語理解部と、対話エージェントから対話行為を受け取り、その対話行為に応じてユーザに出力テキストを提示する言語生成部からなる。言語理解部は、[Ushio 15]をベースに、LSTM によって実装した。言語生成部は、対話行為の種類と対応する既定の文フォーマットにキーワードを当てはめてシステムの応答文を生成する。

2.2 バックエンドシステム

バックエンドシステムは、DB 検索部とタスク管理部を備える。タスク管理部の主な機能は、絞り込みの条件となる複数のキー

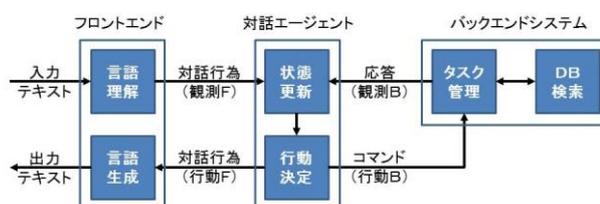


図1: 提案システムの構成

ワードにより DB 検索を行い、絞り込みの結果のデータ候補数が所定の件数以下であれば絞り込み結果とし、そうでなければ次の絞り込みにも有効なキーワードの提案を行う。DBは、中規模 (~数万コンテンツ) サイズを想定する。対話のターン毎にユーザの入力テキストから得られるキーワードをスタックで管理し、対話の履歴自体は管理しない。

2.3 対話エージェント

対話エージェントは、POMDP の枠組みで動作する対話管理のモジュールである。前回の出力(行動)と入力(観測)に基づいて対話状態を更新する状態更新部と対話状態に基づいて行動を決定する行動決定部を有する。POMDP ソルバとして、LUKE[Makino 14]を利用した。ただし、本稿で報告する実験において POMDP ソルバの入力として与えている確率モデルは、信念状態のうち1つの状態だけが確率が 1.0 となるように設定されている。つまり信念状態に曖昧性はなく POMDP の枠組みを用いて、MDP で動作させる。

2.4 対話戦略

対話戦略の設計には、医療面接(問診)の教科書的書籍 [Henderson 12]を参考にした。Henderson 12 では、面接の進行過程を「患者中心の面接」と「医師中心の面接」に区別している。この過程を簡略化して対話システムの動作に当てはめると、一貫してシステム主導の対話であり、オープン質問の繰り返しから、クローズド質問の繰り返しに移行する動作に相当する。

また、Henderson 12 では、特に「患者中心の面接」の重要性を説いており、患者の満足度にプラスに影響するとしている。システム主導のオープン質問は、対話の流れを制御するのはシステムであるが、回答内容についてはユーザが任意に決められるので、ユーザ主導と似た側面がある。オープン質問から対話を始めることで、ユーザの満足度に対してプラスの効果を図る。

表1 動作検討用の対話例(問診タスク)

| | テキスト入出力 | フロントエンド向け対話行為 | バックエンド向け対話行為 | 制御の種類 |
|----|-------------------------|--------------------|----------------------------------|-------------------|
| 1 | | | clear() pass() | システム主導 オープン質問 |
| 2 | | | | |
| 3 | System: 今日はどうされましたか? | QST_OPENING_WHAT() | | |
| 4 | User: 頭がくらくらします。 | RES_INFO(頭痛) | | |
| 5 | System: 頭痛ですか? | QST_CONFIRM(頭痛) | | |
| 6 | User: めまいです。 | RES_CORRECT(めまい) | push_key(めまい) pass() | |
| 7 | | | | |
| 8 | | | | |
| 9 | System: 他に気になることはありますか? | QST_WHAT_ELSE() | | |
| 10 | User: 動悸もします。 | RES_INFO(動悸) | push_key(動悸) pass() | |
| 11 | | | | |
| 12 | | | | |
| 13 | System: 他に気になることはありますか? | QST_WHAT_ELSE() | | |
| 14 | User: ないです。 | RES_NO() | | |
| 15 | | | solve() suggest(発熱) | システム主導 クローズド質問 |
| 16 | | | | |
| 17 | System: 発熱はありますか? | QST_YES_NO(発熱) | | |
| 18 | User: ありません。 | RES_NO() | push_negative() suggest(咳) | |
| 19 | | | | |
| 20 | | | | |
| 21 | System: 咳は出ますか? | QST_YES_NO(咳) | | |
| 22 | User: 出ます。 | RES_YES() | push_positive() conclude(病気3) | |
| 23 | | | | |
| 24 | | | | |
| 25 | System: 病気3のようです。 | RES_INFO(病気3) | | システム主導 情報提示 |
| 26 | User: わかりました。 | RES_ACK() | | |

2.5 対話行為

本提案システムにおける対話行為は、例えば、RES_INFO(頭痛)のように表現される。ここで、RES_INFOを対話行為タイプと呼び、「頭痛」を行為項目と呼ぶ。対話行為タイプは、DSTC5[Kim 16]で定義されている対話行為88種類をベースに絞り込みタスクに必要なものを抽出し、足りないものを追加して設定した。

対話行為タイプは、対話における発話間の関係性(例えば、質問されたら回答するなど)を表現するための情報である。行為項目は、発話の内容に関する情報である。通常、対話行為はフロントエンドと対話エージェントの間のやりとりに対して定義されるが、本提案システムでは、バックエンドシステムと対話エージェントのやりとりであるコマンドおよび応答も、対話行為タイプと行為項目からなる対話行為の形式で表す。対話エージェントは、フロントエンド向けのやりとりとバックエンド向けのやりとりを行動決定ポリシーに基づき切り替える。

2.6 対話状態

POMDP モデルでは、対話エージェントからの出力は行動、対話エージェントへの入力は観測と呼ぶ。POMDP モデルによる対話の進行は、行動から始まり、行動と観測が交互に繰り返される。対話状態は、行動と観測に含まれる情報のうち、行為項目を含まない対話行為タイプの系列のみに依存して決定する。

● 着眼点

表1の動作検討用の対話例を示す。上から順に、フロントエンド向けとバックエンド向けの区別なしに対話行為を時系列順にたどり、偶数番目と奇数番目の項目の組(つまり、観測と行動の組)に注目すると、例えば、10番目のRES_INFO(動悸)と11番目のpush_key(動悸)に含まれる「動悸」のように、行為項目が一致している。他のすべての組み合わせについても同様である。したがって、観測である対話行為の行為項目を、次の行動の対

話行為の行為項目としてそのままコピーすることで、フロントエンドとバックエンドシステムの間ドメイン情報の流れは制御できる。つまり、対話エージェントは、ユーザの発話に含まれるキーワードをそのままバックエンドシステムに渡したり、バックエンドシステムから指定されたキーワードをそのままユーザに投げかけたりすればよく、行動決定としては次の対話行為タイプのみ決定すればよい。

行為項目を含んだ対話行為全体については種類数が多くなるが、行為項目を含まない対話行為タイプは種類数が少ないため、管理すべき対話状態の数を少なく抑えることができる。

2.7 データベース

バックエンドシステムが参照するデータベースは、当社が運営する献立提案サイト「ウィーククナビ¹」のレシピコンテンツの一部である。

2.8 対話エージェントの学習

対話エージェントの動作は、対話データからの学習により得られる対話戦略(対話モデルと行動決定ポリシー)により制御される。学習用の対話データはルールに基づいて生成している。

生成の過程は、以下のとおり;

- フロントエンド向けの対話行為列の断片(行動、観測、行動)をリストで与え、その組み合わせで生成される対話を全探索して出力する。断片には、2.4で述べた対話戦略が含まれるようにする。ただし、ループにより無限長の対話が含まれるので、ループを回る回数を制限する。
- バックエンド向けの対話行為列を挿入する。挿入のルールは、後ろ向きに設定されるものがある。例えば、フロントエンド向けのRES_INFO()という行動に先行して、バックエンドからconclude()という観測が挿入される。

¹ <https://weekcook.jp>

- 行動と観測の種類と履歴に基づき、状態番号を決定し、行動と観測の間に挿入する。

生成された対話データに基づき、対話モデル(状態遷移確率表と観測確率表)を学習する。次に対話モデルを、報酬表とともに POMDP ソルバに入力して、行動決定ポリシーを学習する。

報酬表では、学習用の対話データに沿った対話行為系列には正の値、そうでない対話行為系列には負の値を設定している。これにより、対話モデルの学習に用いられた対話系列が全てトレースされる。学習用の対話データの生成時に、バックエンドシステム向けの対話行為列が後ろ向きに挿入されることがあるが、学習した対話戦略に従うと、前向きに逐次処理で動作することとなる。なお、本稿の範囲では、瞬時報酬のみで全ての行動が決定できるが、言語処理部の誤りを含む入力に対して、明示的な確認を行うかどうかの行動決定に関しては、後の観測によってその必要性が判断されるため、累積報酬を考慮することで最適化される。

図 2 に、評価に用いる対話モデルの対話状態と、0 より大きい状態遷移確率を可視化したグラフの例を示す。この例での状態数は 90 であり、POMDP ソルバは問題なく動作する。

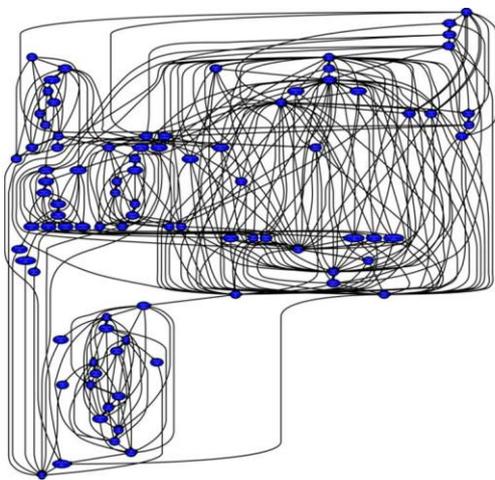


図2: 対話状態と状態遷移

3. 評価実験

提案システムに基いて実装した対話によるレシピ提案サービス (CookChat¹) を一般公開して評価を行った。レシピ選びを目的とするユーザと、CookChat の能力に興味があるユーザが混在することを想定し、クラウドソーシングを通じて、絞込みタスク(気に入ったレシピの選択)の達成の動機付けを行ったユーザで評価を行った。

3.1 実験条件

CookChat では、絞り込まれたレシピの候補が 9 件以下の場合に一覧表示となる。一覧からレシピを 1 つ選択し「このレシピに決定！」のボタンを押すことでタスク達成とする。また、対話の中で CookChat でのレシピ選び体験に関する主観評価のアンケートを行う。クラウドソーシングで「日常的に献立を考える人」を評価ユーザとして集めた。ユーザ 1 人につき 3 回以上の対話を行う。

¹ <https://www.weekcook.jp/trial/cookchat/lp.html>
「ウィークックナビ」にて、一般利用者を対象として期間限定 (2017 年 1 月 24 日～2017 年 3 月 29 日) で公開されたトライアルサービス

3.2 実験結果

168 名のユーザで合計 580 対話中、535 対話が絞り込みタスクを達成し、タスク達成率は 92.2%となった。主観評価アンケートによると提案されたレシピへの満足度は 69.4%であった。

絞込みタスクの実行過程の一例を図3に示す。横軸は、入力のインデックス番号であり、縦軸はレシピの候補数である。縦軸は 1 を加えた後に対数スケール化した。候補数は、入力を受理する前の候補数である。同じ例の詳細な情報を表2に示す。

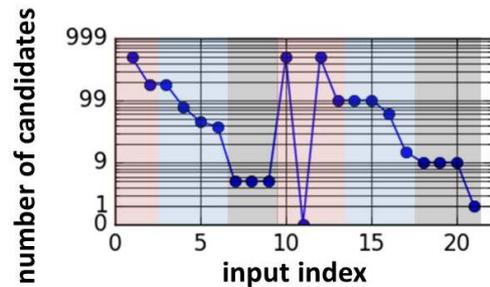


図3: 絞込み過程の様子

この例ではオープン質問、クローズド質問、リスト選択の流れが二回繰り返されている。9 番目の入力で、チャットの最初に戻るという指示を受けている。10 番目の入力に対して、システムは知識獲得を試みるが、11 番目の入力で無回答となる。順調なケースでは、一回目のレシピ選択ですぐに決定されるが、このような種々のリカバリーを通じて、タスクが達成されている。

4. 関連研究

先行研究においても、行為項目の削除により信念空間を縮小する提案 [Young 10] や、DB を環境に含めた強化学習の枠組みの提案 [Zhao 16] がある。本提案システムには、それらと同様のアイデアを含むが、対話行為タイプの選択により、フロントエンドとバックエンドとの間のドメイン情報の流れを制御することでタスクを達成するという構成に重きがある。

5. おわりに

対話のやり取り上の役割を表す対話行為タイプと、発話の内容を表す行為項目を別々に制御することで、対話モデルのコンパクト性とタスクの高い達成度が両立できることを確認した。

参考文献

- [Kim 16] Kim, Seokhwan, et al. "The Fifth Dialog State Tracking Challenge." *Proceedings of the 2016 IEEE Workshop on Spoken Language Technology (SLT)*. 2016.
- [Makino 14] Makino, T., and M. Shiro. "LUKE-learning underlying knowledge of experts- User Manual." (2014).
- [Henderson 12] M. Henderson, et al. "The Patient History: An Evidence-Based Approach to Differential Diagnosis, second edition" McGraw-Hill Medical (2012) (山内豊明監訳『聞く技術 答えは患者の中にある 第2版』日経 BP 社, 2013)
- [Ushio 16] Ushio, Takashi, et al. "Recurrent convolutional neural networks for structured speech act tagging." *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016.

[Young 10] Young, Steve, et al. "The hidden information state model: A practical framework for POMDP-based spoken dialogue management." Computer Speech & Language 24.2 (2010): 150-174.

[Zhao 16] Zhao, Tiancheng, and Maxine Eskenazi. "Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning." arXiv preprint arXiv:1606.02560 (2016).

表 2 : CookChat による対話例

図 3 のグラフと同じ例。テキスト入出力の欄で、括弧付きの入力は選択ボタンによる入力である。
また、背景色は、オープン質問、クローズド質問、情報提示、リスト選択からなる大局的な対話戦略を表現している。

| 入力番号 | テキスト入出力 | フロントエンド向け対話行為 |
|------|---|--|
| 1 | System: どんな料理をお探でしょうか? お好きなジャンルや使いたい食材などの条件をどうぞ。 | QST_OPENING_WHAT () |
| | User: お弁当に入れるおかず | RES_INFO (副菜) |
| 2 | System: 副菜ですね。 他にも条件があれば、テキスト入力をお願いします。 | FOL_CONFIRM (副菜) QST_WHAT_ELSE () |
| | User: (条件入力終了) | RES_NO () |
| 3 | System: それでは他の条件について、こちらから質問しますので、はい・いいえを選択して答えてくださいね。 | INI_EXPLAIN () |
| | User: (はい) | QST_YES_NO (サラダ) RES_YES () |
| 4 | System: 野菜料理でよろしいですか? | QST_YES_NO (野菜料理) |
| | User: (はい) | RES_YES () |
| 5 | System: 和風でよろしいですか? | QST_YES_NO (和風) |
| | User: (はい) | RES_YES () |
| 6 | System: 定番料理でよろしいですか? | QST_YES_NO (定番料理) |
| | User: (はい) | RES_YES () |
| 7 | System: 候補レシピが絞り込まれたので提案します。 このレシピはどうでしょうか。気に入ったレシピがあれば、下のボタンでレシピの番号を選択してください。 | RES_INFO (*) |
| | User: (もっと見たい) ... ~~~ 中略 ~~~ | |
| 10 | System: どんな料理をお探でしょうか? お好きなジャンルや使いたい食材などの条件をどうぞ。 | QST_OPENING_WHAT () |
| | User: 鶏胸肉 | RES_INFO (鶏胸肉) |
| 11 | System: 鶏胸肉ですね。 入力いただいた条件は知らない表現でした。 具体例になる料理名やジャンルなどのヒントを教えてください。 | FOL_CONFIRM (鶏胸肉) QST_EXAMPLE () |
| | User: (戻る) | RES_NO () |
| 12 | System: わかりました。それでは前の質問に戻ります。 他にも条件があれば、テキスト入力をお願いします。 | FOL_INFO_CANCEL () QST_WHAT_ELSE () |
| | User: 鶏胸肉を使った主菜 | RES_INFO (主菜・主食, 肉料理) |
| 13 | System: 主菜・主食で肉料理ですね。 他にも条件があれば、テキスト入力をお願いします。 | FOL_CONFIRM (主菜・主食, 肉料理) QST_WHAT_ELSE () |
| | User: (条件入力終了) | RES_NO () |
| 14 | System: それでは他の条件について、こちらから質問しますので、はい・いいえを選択して答えてくださいね。 | INI_EXPLAIN () |
| | User: (どちらでもよい) | QST_YES_NO (焼き物) RES_NEUTRAL () |
| 15 | System: 和風でよろしいですか? | QST_YES_NO (和風) |
| | User: (はい) | RES_YES () |
| 16 | System: 定番料理でよろしいですか? | QST_YES_NO (定番料理) |
| | User: (はい) | RES_YES () |
| 17 | System: 揚げ物でよろしいですか? | QST_YES_NO (揚げ物) |
| | User: (いいえ) | RES_NO () |
| 18 | System: 候補レシピが絞り込まれたので提案します。 このレシピはどうでしょうか。気に入ったレシピがあれば、下のボタンでレシピの番号を選択してください。 | RES_INFO (*) |
| | User: (もっと見たい) ... ~~~ 中略 ~~~ | |
| 21 | System: ①のレシピに決定しますか? | |
| | User: (このレシピに決定) | |
| 22 | System: アンケート: あなたの好みにあったレシピが提案できましたか? | |
| | User: (そう思う) | |