

対話サービスプラットフォームの開発

Development of Dialog Service Platform

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

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

We have been developing Dialog Service Platform which can provide dialog user interfaces for various tasks. On the platform, we have implemented a dialog application that recommends cooking recipes considering user's preference. The recipe contents come from our WEB site for recipe selection service. This paper describes overall view of the platform and recipe recommendation application, and reports about trial service of the application for public users.

1. はじめに

我々は様々な用途に対応する対話インターフェースを構築し提供できる対話サービスプラットフォームの開発を進めている。我々が対話技術の開発、および、対話サービスプラットフォームの開発に取り組む背景として次のような課題意識がある。

1. GUIの限界

機器操作の複雑化によりメニュー選択を中心とした従来のGUIでは、要求される操作数の増大や目的機能到達の困難さなどの課題が深刻であり、むしろGUIが操作性の障壁になって機能が十分に利用できない状況が生じている。

2. 開発ノウハウの断片化

通常、商品・サービスの開発案件毎に、既存の対話SDKを利用したり、独自の対話エンジンを開発することが多い。その結果、対話システム開発のノウハウが断片化し蓄積・共有されず、同じ課題解決を繰り返すことに終始し、技術レベルが向上しない状況が生まれている。

3. データ活用視点の不足

従来の商品開発では商品の機能実現が最優先であり、対話は単に機器の操作性向上の手段としか捉えられず、対話から得られるユーザの発話データを蓄積・分析し、ユーザの意図を把握、商品・サービスの価値向上へのフィードバック、対話技術の技術課題の抽出などにつなげることまでは行われていなかった。

すでに大学を含む研究機関や技術系企業から対話システムを構築するためのSDKが多数(例えば、MMDAgent [Lee 2013], netpeople [INAGO] など)リリースされているが、上記の2, 3に挙げた課題に対して、自社において、開発ノウハウの共有化を図り開発効率・技術レベルの向上させ、また、蓄積データの共有化を図り様々な商品・サービス間で新たなユーザ価値の発掘につなげるという視点に立ち、独自の対話サービスプラットフォームを開発している。

連絡先: 山上 勝義, パナソニック株式会社, 東京都江東区有明 3-4-10 東京ファッションタウンビル 西館 5 階
yamagami.katsuyoshi@jp.panasonic.com

2. 対話サービスプラットフォームの概要

本節では開発中の対話サービスプラットフォームの開発の目的と構成について述べる。

2.1 対話サービスプラットフォームの開発の目的

対話サービスプラットフォームを開発する目的は、1. 節で述べた課題を解決するための機能と性能を備えた対話処理の枠組みを実現することである。具体的には

1. 様々な機器・システムへの対話インターフェースの提供
2. 対話処理実装のための共通の枠組みの提供
3. 対話処理にまつわるデータの蓄積・分析の枠組みの提供である。

2.2 対話サービスプラットフォームの構成

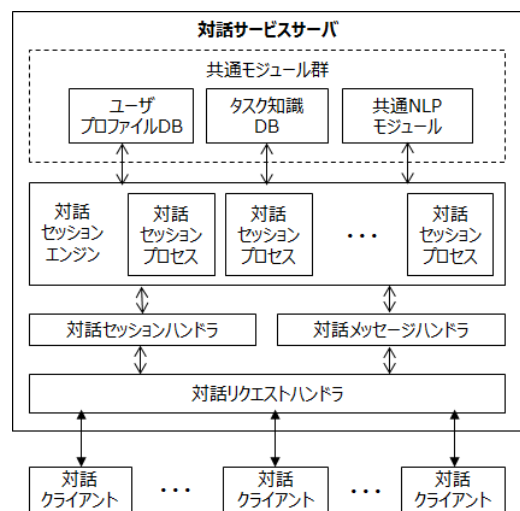


図 1: 対話サービスプラットフォームの構成

対話サービスプラットフォームの構成を図 1 に示す。大きく分けて、対話サービスを利用する複数の対話クライアント群と対話サービスサーバの 2 つから構成される。対話クライア

トの機能はユーザからの発話入力を受付け対話サービスサーバに送信し、対話サービスサーバからの応答出力を受信し、ユーザに提示することである。対話サービスサーバの内部は下記の機能ブロックから構成される。

対話リクエストハンドラ

対話クライアントからの対話処理リクエストを受け付ける

対話セッションハンドラ

対話クライアントからの対話処理リクエストに応じて、対話処理を行う対話セッションエンジンに対して対話セッションプロセスの起動・終了の指示、対話クライアントと対話セッションプロセスとの対応の管理を行う

対話メッセージハンドラ

対話クライアントからの対話メッセージ(ユーザの発話テキストなど)の対応する対話セッションプロセスとの送受信を受け持つ

対話セッションエンジン

対話クライアントからの対話メッセージを処理する(対話処理を行う対話セッションプロセスは複数同時実行可能である)

共通モジュール群

対話セッションプロセスが行う対話処理実行に必要な共通モジュール群であり対話クライアントを利用するユーザのプロファイルデータベース、対話処理で実行される対話タスクのタスク知識データベース、対話処理実行時に呼び出される自然言語処理(NLP)モジュールからなる

上記の構成から分かる通り、対話サービスサーバは複数の対話クライアントからの対話リクエスト(対話メッセージ)を複数同時に処理することができる。つまり、同一対話サービスの対話処理、ならびに、異種の対話サービスの対話処理を複数同時に実行することができる。

上記の構成要素のうち実装する対話タスクに依存する部分は、対話セッションエンジンであり、他の構成要素は非依存である*1。

対話サービスサーバはクラウド上の仮想サーバ(Linux, CentOS, Ver.7.3)上に実装している。対話リクエストハンドラはTomcat(Ver.7.0.69)で実装しており現在はREST型の対話リクエストに対応している。対話セッションハンドラ、対話セッションエンジン、共通モジュール群はPython(Ver.2.7.11)で実装している。対話メッセージハンドラはメッセージ・キューライブラリRabbitMQ(Ver.3.6.5)で実装している。

対話セッションエンジンを複数配置することもでき、同時に処理可能な対話セッションプロセスをスケールすることができる。複数配置により対話サービスサーバを停止することなく、対話セッションエンジンの更新を行うこともできる。

図1の構成には記載していないが、対話セッションプロセスで実行される対話処理のログを蓄積する機能も有しており、対話クライアントとの間でやりとりされるユーザ発話・システム応答のデータを含む対話データをすべて蓄積することができる。

3. 実証実験トライアルサービスの概要

2.節で述べた対話サービスプラットフォーム上に、対話によりレシピ提案ができるチャットシステム”CookChat”[CookChat 2017]を実装した。さらにCookChatを実証実験を目的としてトライアルサービスとして一般公開(2017/1/24公開開始)した*2。CookChatで提案するレシピは、当社が運営するレシピ提案サイト”ウィークックナビ”[WeekCook 2014]が保有するレシピから選別したものである。

CookChatの対話の具体的な実装については遠藤らの報告[Endo 2017]を参照のこと。

3.1 レシピ提案対話CookChatの概要



図2: CookChatの操作画面

CookChatは、テキスト入力、あるいは、チャット画面上に提示される選択ボタンのクリックによる対話を通じて、好みのレシピを選ぶことができるチャットシステムである。図2にCookChatの操作画面の例を示す。CookChatにおけるシステムの応答テキストは左側の白い吹き出しに表示され、ユーザが入力した発話テキストは右側の薄青の吹き出しに表示される。システム応答の吹き出しの左側には、ウィークックナビのマスケットキャラクター”クックちゃん”のアイコンが表示され、システムの応答内容に応じて、アイコンの表情が切り替わる。

CookChatの対話シナリオは、医療面接(問診)の教科書的書籍[Henderson 2012]を参考に、レシピ選択に対するユーザの満足度を考慮しつつ、なるべく効率的にレシピを提案する観点で、ユーザ主導でユーザの要望を尋ねるシナリオとシステム主導でユーザの要望を尋ねるシナリオの2つを軸に検討し構

*1 ユーザプロファイルデータベース、タスク知識データベース、共通NLPモジュールが対話タスクに依存する可能性もある

*2 アクセス先は <https://www.weekcook.jp/trial/cookchat/1p.html>

成した。具体的には CookChat のレシピ提案の対話シナリオは次の順序に沿って展開する。

オープン質問

ユーザがレシピに関する好みを表したテキストを回答として入力する

クローズド質問

ユーザはレシピの好みに関する質問に「はい/いいえ/どちらでもない」の3択で回答する

レシピ一覧選択

ユーザは一覧表示されたレシピの情報を参照しながら気に入ったレシピを選択する

オープン質問の対話シナリオでは、例えば「豆腐を使った中華」、「おすすめは?」などの表現を回答として入力することができ、ユーザが回答を入力する度に、提案するレシピの候補が絞り込まれていく。オープン質問の対話シナリオでユーザが要望する回答を言い尽くした場合は「条件入力終了」というボタンをクリックすることにより、クローズド質問の対話シナリオに移る。

次に、クローズド質問の対話シナリオでは、オープン質問で絞り込まれたレシピの候補をさらに絞り込む。各レシピにあらかじめ付与された検索タグに関するクローズド質問 (例えば「炒め物でよろしいですか?」など) を行い、ユーザが選択肢で回答する度にレシピの候補がさらに絞り込まれる。絞り込まれたレシピの候補数があらかじめ設定した数を下回ると、レシピ一覧選択の対話シナリオに移行し、ユーザは一覧形式で表示されるレシピの中から気に入ったレシピを選択する。

我々は、知識グラフデータ ConceptNet5 [Speer 2012] にレシピ関連の用語に関する概念関係を追加し、レシピ関連用語のオントロジーを作成した。これにより CookChat が理解可能なレシピに関連の表現の幅を拡大するとともに、ユーザの発話の要望に沿うレシピが見つからない場合に代替となるレシピの提案ができる。具体的には、オープン質問に対してユーザが回答した発話テキストから抽出したレシピ関連のキーワードが、レシピに付与された検索タグにヒットしない場合、作成したオントロジーを用いて上位概念を辿り、検索タグにヒットするキーワードを導出する。これにより、ユーザが要望した好みに直接相当するレシピが無い場合に、レシピ提案の範囲を上位概念まで広げて、代替となるレシピを提案することができる。

3.2 CookChat の運用状況

CookChat を公開してから4週間 (2017/1/24 - 2017/2/20) の運用状況について説明する。

図3に週ごとのアクセス数の推移、および、各週のアクセス数の累積アクセス数を示す。4週間の合計アクセス数は1493であり、週平均のアクセス数は373.2である。

CookChat の対話シナリオに、ユーザへのアンケートを組み込んで、次の質問に対するユーザの回答が得られるようにした。

Q1 あなたの好みにあったレシピが提案できましたか?
(そう思う/どちらでもない/思わない)

Q2 また使おうと思いますか?
(そう思う/どちらでもない/思わない)

Q3 感想や要望など意見をお願いします
(自由記述)

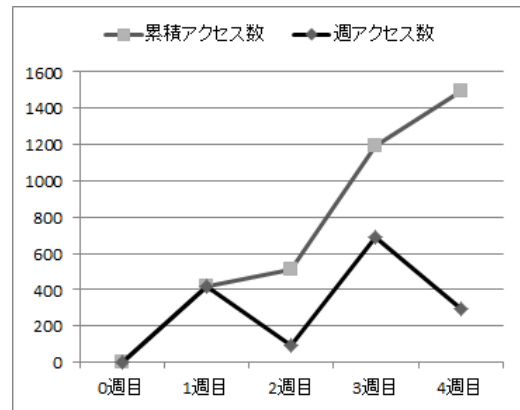


図3: CookChat のアクセス数推移

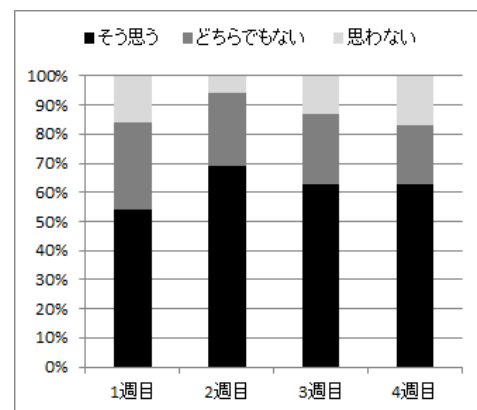


図4: Q1 に対する回答の割合

図4、図5に、それぞれ、各週毎の Q1, Q2 の3択の回答の割合を示す。4週間全体での Q1 に対する「そう思う」と答えた割合は60.3%、Q2 に対して「そう思う」と答えた割合は70.1%であった。

表1に Q3 に対する回答の一部を示す*3。今回 CookChat で実装したレシピ提案の対話シナリオは、ユーザの発話テキストからレシピの好みを表すキーワードを抽出し、それを検索キーにして、レシピに付与された検索タグとのマッチングにより、レシピ候補を絞り込むというものであった。

チャットでのレシピ選びと WEB での検索によるレシピ選びとの違いがあまり見いだせないというネガティブな評価は、WEB ブラウザ等でのキーワード検索に馴染みのあるユーザからであると思われる。Q3 の自由回答にあるように、レシピを直接特定するキーワードだけではなく「こってり」「あっさり」のような表現さらに「温まる料理」「パーティ用の料理」「お酒に合う料理」などの、レシピに関する人間の解釈を含んだ表現を起点とした対話を実現することが、チャットという形態を活かしたキーワード検索との差別化の1つの方向性であると考えられる。

一方で、キーワードによる絞り込み型のチャットでも、便利に楽しく使えるといったポジティブな評価が得られた。コンテンツ選びの手段に関する先入観が無いユーザにとっては、チャット (対話) を通じて、レシピ選びをする体験も違和感なく受け

*3 回答の文言はユーザが入力した原文を要約したものである

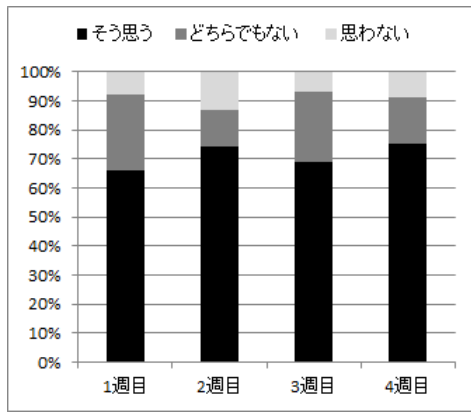


図 5: Q2 に対する回答の割合

表 1: Q3 (自由記述) の回答例

ポジティブな回答
何を作るか相談している感覚で楽しい 検索サイトより身近で便利に探せる 調べなくても材料を言えば候補が出てくる チャット形式なので検索が楽しい やりとりがスムーズで違和感がない 「おすすめレシピ」の提案があるので良い 細かい要望に対しても提案してくれた アイコンが可愛くて楽しい
ネガティブな回答
いちいちチャットをするのは手間 味付けや温度の要望が理解されない 「こってり」「あっさり」などの感性的な要望を理解してほしい キーワードを指定する検索サイトとの違いが感じられない レシピの候補数が少なすぎる 理解する表現が限られている

入れられると考えられる。

このように、レシピ選びに関するユーザの期待は多様である。ユーザがどのようなレシピ提案の流れを望んでいるか、つまりどんなタイプの対話シナリオを望んでいるかをさらに分析し、対話シナリオの展開を柔軟に適應できるメカニズムに発展させることも必要になると思われる。

なお、CookChat の運用開始から原稿執筆時点 (2017/2/28) までの間に、収集した対話ログを分析し、対話セッションエンジンの対話シナリオ実装を改良し、更新を複数回行った。その間、対話サービスサーバが停止するなどの致命的な不具合の発生はなく安定稼働を続けている。

4. おわりに

本稿では我々が開発している対話サービスプラットフォームの開発の背景となる課題について述べた。そして、その課題を解決を可能にする対話サービスプラットフォームの構成と実装の概略について述べた。さらに、対話サービスプラットフォーム上に実装したレシピ提案対話 (CookChat) の内容、CookChat の実証実験トライアルサービスの運用状況、レシピ提案をチャットにより行うことへのユーザ評価の結果とその考察について述べた。

今後の対話サービスプラットフォームの開発の主な課題は以

下の通りである。

音声入出力への対応

音声認識機能、音声合成機能、バージョン機能などの実装により音声対話に対応する (対話クライアント側での実装、および、対話サービスサーバ側での実装)

対話セッションエンジン実装の SDK 化

レシピ提案の対話シナリオの実装を SDK 化 (内部実装モジュールの共通化・実装手順のスキーム化) により様々なタスクでの対話シナリオ実装を容易にする

タスク知識ベースの拡張

タスク知識 (タスクドメインにおける概念の関係性) の規模を拡大することにより、ユーザが期待する対話シナリオ実現の可能性を広げる

謝辞

本対話サービスプラットフォームの開発に関して、ソフトウェア開発プロセスの観点からの有益な助言および指導をいただいた SLAD 株式会社 代表取締役 井上 健司 氏に感謝する。

参考文献

- [Lee 2013] Akinobu Lee, Keiichiro Oura, Keiichi Tokuda.: “MMDAgent — A Fully Open-Source Toolkit for Voice Interaction Systems.”, In Proc. IEEE ICASSP2013, pp. 8382-8385, (2013)
- [INAGO] INAGO Inc.: “netpeople: ASISTANT PLATFORM”, <http://www.netpeople.co/ja/netpeople/>
- [CookChat 2017] Panasonic Corp.: “CookChat”, Panasonic Newsroom Japan, (2017), <http://news.panasonic.com/jp/topics/151614.html>
- [WeekCook 2014] Panasonic Corp.: “ウィーククックナビ”, Panasonic Newsroom Japan, (2014), <http://news.panasonic.com/jp/topics/2014/38411.html>
- [Endo 2017] 遠藤 充, 牛尾 貴志, 山上 勝義, 堀井 則彰.: “発話とコマンドの系列を制御する対話エージェント”, 人工知能学会全国大会 (第 31 回) 論文集, (投稿中), (2017)
- [Henderson 2012] 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)
- [Speer 2012] Robert Speer, Catherine Havasi.: “Representing General Relational Knowledge in ConceptNet 5.”, In: International conference on language resources and evaluation (LREC), pp. 3679-3686, (2012)