大規模言語モデルを利用した音声対話システムのメタ制御

宿里晃太郎 [1]　石垣龍馬 [1]　鈴木順大 [1]　永沼翔翼 [1]
藤本拓真 [1]　河窪大介 [1]　酒造正樹 [1]　前田英作 [1]
[1] 東京電機大学
{20aj076, 20aj012, 20aj078, 20aj098, 20aj112, 22amj10}@ms.dendai.ac.jp
{shuzo, maeda.e}@mail.dendai.ac.jp

概要

大規模言語モデル (LLM) を利用することにより，従来のルールベースによる音声対話システムでは困難であった自然な対話が可能である．しかし，LLM は明示的な出力制御が困難であり，音声対話システム設計者の意図と異なる発話が生起する危険性がある．本稿では，より安定で且つ柔軟な音声対話システムを実現するための LLM を利用したメタ制御手法を提案する．提案手法は，対話シナリオに沿った対話を実現するための対話フロー制御と，自然な対話を実現するためのターンテイキング制御とからなる．それら制御の最適化にも LLM を利用するというメタ制御を音声対話システム上に実装し，その効果を検証した．

1　はじめに

対話システムの構築において，その目的にあわせたシステム主導型の対話を実現するには，ルールベースのシステム設計をすることが一般的である．しかし，このルールベースによる対話システムは，雑談対話のような柔軟な対話において，破綻せずに継続することが困難であった．そこで近年，柔軟で且つ多様な発話生成を可能にする大規模言語モデル（LLM）の利用が進みつつある．しかし LLM は，その出力を明示的に制御することが難しく，安定した対話システムを実現することは必ずしも容易ではない．対話ロボットコンペティション 2023（DRC2023）[1] において課題とされたのは，旅行代理店のカウンターパーソン役としてのアンドロイドロボットのためのマルチモーダル対話システムを構築することであった．この対話の目的は，ユーザであるお客様の希望を聞きながら，ユーザの満足する旅行プランを策定して提示することである．明確に既定されたゴールに向かうような対話の安定性が求められるとともに，ユーザの様々な発話に対して適切に対応する柔軟な発話生成と対話制御が必要となる．

以前，我々の研究グループ [2], [3] ではルールベースを基本とする対話システムを構築し，一部のロボット発話に言語モデルの出力を利用していた．昨年の DRC で優勝したチーム LINE [4] は言語モデルである HyperCLOVA を用いた対話システムを構築したが，対話の流れはルールベースで記述していた．当時の技術水準では，言語モデルによる安定した対話制御までは難しかったと言える．DRC2023 は OpenAI が開発した GPT-4 をはじめとした LLM が登場して以来，初のアンドロイドロボットを用いたコンペティションである．

本稿では我々が DRC2023 において構築したシステム DSML-TDU[5] における実装例を取り上げて本技術の詳細を紹介する．

2　音声対話システム DSML-TDU

DRC2023 のために構築した対話システム DSML-TDU の概要を図 1 に示す[1)]．対話の Introduction と Closing，及び制限時間（10 分）の管理には，ルールベースによる対話制御を行ったが，それ以外の部分における発話生成と対話制御には LLM を利用した．LLM には GPT-4[2)] を，音声認識には Google Speech Recognition (GSR) を用いた GPT-4 を駆動するためのプロンプトには，Dialogue Flow Control Prompt (DFCP) と Turn-Take Control Prompt (TTCP) の 2 種類を用意した．これらプロンプトの特徴は，発話生成に加えて，シナリオ分岐点における分岐判断，円滑なターンテイキングを実現するための制御を行っている点であり，ここではそれをメタ制御と呼ぶこと

1)　本システムの DFCP は [6] のプロンプトに改良を加えたものである．
2)　2023 年 10 月時点の gpt-4-0613 を使用．

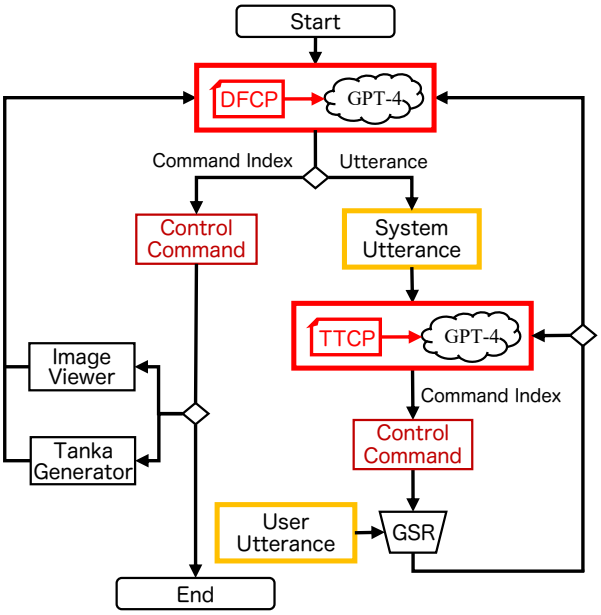

図 1 Overview of dialogue flow in DSML-TDU system

とした．

3 プロンプトによるメタ制御

3.1 Dialogue Flow Control Prompt (DFCP)

DFCP は，発話生成を行いつつ同時に対話フローの全体を制御するためのもので，146 行，4905 文字からなる．その一部を，図 2 と図 3 に示す．図 2 は発話内容や対話における流れを定めるものであり，対話履歴の内容によってシステムの発話内容や会話の流れが変化する．本システムでは，GPT-4 による発話生成は DFCP と対話履歴のみによって決定される．

図 3 は，DFCP 上であらかじめ定義したコマンドを実行するためのものであり，コマンド番号とそれに対応するコマンドがリストとして記述されている．DFCP による GPT-4 への指示は，コマンドのいずれかを実行する必要があるかどうかを対話の流れから判断し，必要とされた場合には，対応するコマンド番号を出力として生成することである．

コマンド一覧にはその名前や概要を記述するが，それだけでは GPT-4 による積極的なコマンドの実行は期待できない．2 番のように客から要望があった際にはコマンドを実行するが，会話の進行度を監視して，こちら側から提案することは少ない．そのため，「あれが達成されたため，これをする」のように記述することで積極的なコマンドの実行を要求することができる．

```
# 命令書
貴方は旅行代理店店員のプロフェッショナル店員です．
以下の条件に従ってください．

# 条件
あなたの最終的な目標は，京都市内で２つの観光地をめぐる旅行プランを決めることです．
基本的に以下の「タスク」の流れに沿って１つずつ実行してください．
ただし，お客様の要望次第では順番を入れ替えたり，タスクを飛ばしても構いません．

# タスク
1: まずはお客様のことを知りたいと言い，，お客様の趣味や最近やっている事を聞いてください．
...
2: お客さんの趣味を聞いてうれしかった事を伝えた後に「それでは本題に入りましょう」と言ってください，．
3: 「お客様が京都でどのような体験をし ，どんな思い出を作りたいのか，それが伴になります」と言った後京都市内の旅行でどのような体験をしたいのか聞いてください
...
10: 決まった旅行プランの確認をしてください．
...
# あなたのプロフィール
あなたは旅行代理店の店員です．名前は翔子です．
お客さんと話すのが大好きで，少しお茶目な女性です．
今回、お客様に最高の提案ができるように沢山お客様の事を聞こうと頑張っており，お客様に興味津々です．
```

図 2 Part of dialogue flow control prompt (DFCP)

ことができる．

```
# 条件
必ず発話をする前に，以下の「コマンド一覧」を実行するべきか判断し，実行するべき場合にはコマンドを選んで，数字１桁のみを出力してください．

# コマンド一覧
0: タスクの全てのフェーズが完了し，会話もひと段落ついたので終了する
1: １箇所目の観光地と２箇所目の観光地が決まって，食事処も提案し，会話もひと段落ついたので，プランを確認する
2: 観光地の様子や画像を見せてほしいと言われたので，観光地画像の表示を試みる
3: プランを提案する
...
```

図 3 Example of command control in DFCP

例えば，対話目的が達成されたと判断すれば，GPT-4 は「0」を出力し，Closing へ移行する．また，どのコマンドも実行不要と判断すれば，GPT-4 は発話文をテキストとして出力する．

DFCP において特に着目するべき点は，発話生成とコマンド実行のためのプロンプトを 1 つに統合し，文章生成の代わりにコマンドを実行するという選択肢を与えた点である．これにより，システムの発話前後で柔軟にコマンドを実行できる．

類似する機能として OpenAI が提供する function calling[3] というものがあるが，会話の中での実行タ

3) https://platform.openai.com/docs/guides/function-calling

イミングを GPT-4 自身に決めさせるためにこれを実装した．現状のシステムでは発話生成とコマンド実行が同時に行われないため function calling でも代用できるが，プロンプト (図 3) をチューニングする事で，発話生成の中にコマンドを埋め込むことができる．

3.2 Turn-Take Control Prompt (TTCP)

TTCP は，対話におけるターンテイキングを制御するためのプロンプトであり，27 行，534 文字からなる．その一部を，図 4，5 に示す．このプロンプトによって，GPT-4 は，実行すべきと判断したコマンド番号を出力する．自然な対話を継続するためには，ターンテイキングの的確な判断，制御が必要である．この制御が適切に行われないと，不用意な発話の割り込みや意図しない沈黙が発生し，対話破綻の原因となる．沈黙時間が話者間に発生すると，ユーザは行動判断に迷い，不安になる．

沈黙時間の発生には様々な要因があり，システム側の音声認識や発話生成によって発生する待ち時間が代表的なものである．また，ユーザがアンドロイドロボットに対して持つ認知モデルは人間に対する場合とは異なるため，対話の非対称性が生じる [7]．そのためユーザ側の発話過程は人間に相対する場合と異なる可能性があり，それによって，意図しない発話割り込みや沈黙を引き起こす．

こうした，ターンテイキングの不安定性を解消するためには，人間の認知状態を推定することが有効である．本システムでは，ユーザ, あるいは，システムの発話からターンテイキングに関するユーザの認知状態を GPT-4 に推定させることを試みた．プロンプトでは，推定結果に応じて，音声認識 GSR の音声認識区間を制御し，意図しない発話の割り込みをし，円滑で自然な対話が実現することを目指した．こうした制御は，ユーザが持つ対話参与感の向上にも寄与すると期待できる [8]．

直前の発話テキストと文脈からターンテイキングを推定する手法である TurnGPT [9] を Ekstedt らが提案している．ここでは，Ford and Thompson [10] が提唱した「質問が構文的には完全であっても，文脈によっては完全であるとは限らない」ことによる実用的な完全性 (pragmatic completeness) について言及し，それを前提としてシステムを構築している．

我々もそれに倣い，プロンプトには直前の発話だけではなく，数ターン分の対話履歴も含めた．それによって直前の発話と文脈から推定し，ユーザに会話のペースを合わせることができる．

```
以下の条件に必ず従ってください．

# 条件
これまでの対話履歴から，店員の発話に対して，客に返答を求めるかどうかを判断してください．
...

# 出力条件
「店員発話タイプ一覧」の中にある，数字１桁のみを出力してください．

# 対話履歴
...

# 店員発話タイプ一覧
0: 発話が質問であり、返答が必須
1: 発話が質問であり、返答が任意
2: 発話が質問ではなく、返答が任意
3: 発話が質問ではなく、返答が不要
...
```

図 4 Example of command control in TTCP（直前の発話がシステムの場合に使用）

```
以下の条件に必ず従ってください．

# 条件
客が発話しましたが，さらに話そうとしている場合があります．
これまでの対話履歴から，客が続きを話しそうか考え，客の発話を待つべきかどうかを判断してください．
...

# 出力条件
「客発話タイプ一覧」の中にある，数字１桁のみを出力してください．

# 対話履歴
...

# 客発話タイプ一覧
0: 客が続きを話しそうなため店員は話し始めてはいけない，
1: 客が続きを話すかもしれないため店員は話し始めない方がよい，
2: 客が続きを話さないかもしれないため店員が話し始めてもよい，
3: 客が続きを話さなそうなため店員が話始めたほうがよい，
...
```

図 5 Example of command control in TTCP（直前の発話がユーザの場合に使用）

4 おわりに

本論文では，対話制御に関わるすべてを LLM によって動作させるメタ制御によって音声対話システムを構築するための手法を提案した．DRC2023 の課題に対して，提案システムが想定通りに動作することを検証した．2 種類のプロンプト DFCP と TTCP も統合し，一つのプロンプトによってシステムを動かすことも原理的には可能であり，今後検討していく予定である．

謝辞

本研究は JSPS 科研費 JP19H05693 の助成を受けたものです.

参考文献

[1] Takashi Minato, Ryuichiro Higashinaka, Kurima Sakai, Tomo Funayama, Hiromitsu Nishizaki, and Takayuki Nagai. Overview of Dialogue Robot Competition 2023. In **Proceedings of the Dialogue Robot Competition 2023**, 2023.

[2] Daisuke Kawakubo, Hitoshi Ishii, Riku Okazawa, Shunta Nishizawa, Haruki Hatakeyama, Hiroaki Sugiyama, Masaki Shuzo, and Eisaku Maeda. Spoken dialogue strategy focusing on asymmetric communication with android robots. In **Proceedings of the Dialogue Robot Competition 2022**, 2022.

[3] Makoto Kawamoto, Masaki Shuzo, and Eisaku Maeda. Improving user's sense of participation in robot-driven dialogue. In **Proceedings of the Dialogue Robot Competition 2022**, 2022.

[4] Takato Yamazaki, Katsumasa Yoshikawa, Toshiki Kawamoto, Masaya Ohagi, Tomoya Mizumoto, Shuta Ichimura, Yusuke Kida, and Toshinori Sato. Tourist guidance robot based on HyperCLOVA. In **Proceedings of the Dialogue Robot Competition 2022**, 2022.

[5] Kotaro Shukuri, Ryoma Ishigaki, Jundai Suzuki, Tsubasa Naganuma, Takuma Fujimoto, Daisuke Kawakubo, Masaki Shuzo, and Eisaku Maeda. Meta-control of dialogue systems using large language models. In **Proceedings of the Dialogue Robot Competition 2023**, 2023.

[6] 鈴木順大, 石垣龍馬, 宿里晃太郎, 河窪大介, 酒造正樹, 前田英作. ただ 1 つのプロンプトによるタスク指向型対話システムの実現. **言語処理学会第 30 回年次大会 (NLP2024)**, 2024. (発表予定).

[7] Daisuke Kawakubo, Masaki Shuzo, Hiroaki Sugiyama, and Eisaku Maeda. Asymmetric communication: cognitive models of humans toward an android robot. **Frontiers in Robotics and AI**, Vol. 10, No. 1267560, 2023.

[8] Makoto Kawamoto, Masaki Shuzo, and Eisaku Maeda. Improving user's sense of participation in robot-driven dialogue. **Advanced Robotics**, 2023.

[9] Erik Ekstedt and Gabriel Skantze. TurnGPT: a transformer-based language model for predicting turn-taking in spoken dialog. In **Findings of the Association for Computational Linguistics: EMNLP 2020**, pp. 2981–2990, 2020.

[10] Cecilia Ford and Sandra A. Thompson. Interactional units in conversation: syntactic, intonational, and pragmatic resources for the management of turns. **Studies in Interactional Sociolinguistics**, Vol. 13, No. 3, pp. 134–184, 1996.