F-033

発話交替の競合修復のためのシステム行動選択に向けた多人数対話データの収集

Collecting Multi-Party Dialogue Data Toward System Action Selection to Resolve Turn-Taking Conflicts

堀口 勇輝 † 山本 賢太 † 武田 龍 † 駒谷 和範 †
Yuki Horiguchi Kenta Yamamoto Ryu Takeda Kazunori Komatani

1. はじめに

音声対話システムにおいては発話内容だけでなく発話タイミングの制御も重要であり，特にターンテイキングを円滑に行う必要がある．通常，相手の発話の終了を検知した後に発話を行うことでターンテイキングが成立する．しかし，意図的な割り込みや，発話間の沈黙を発話交替の合図と勘違いして聞き手が話し出した場合，発話交替の競合が起こり得る．競合が発生した場合，ターンテイキングが正常に行えず，沈黙などが発生し円滑な対話を阻害する．また，多人数対話では聞き手が複数人いるため，ターンテイキングはより複雑になり，競合のパターンも増える．システムは，このような対話中の競合を解消し，元の対話の流れに戻す修復行動を取る必要がある．

先行研究では，ユーザの音声入力を検知した際，それがユーザの割り込みかどうかを判断し，発話を継続したり停止したり，中断した後再開する手法が提案されている [1]．この研究で想定されているシステムの修復行動は単純な再開や停止である．また，システムとユーザの二者対話が想定されており，複数のユーザと対話する多人数の状況は想定されていない．

本研究では多人数対話を含む対話における発話交替の競合に対処するため，適切な修復行動を行うシステムの実現を目指す．そのために本稿では，システムが取るべき修復行動の設計を行った．また，多人数対話における発話交替の競合の発生タイミングとその原因を明らかにするための対話データを収集し，修復行動のアノテーションの方針を定めた．

2. システムの修復行動

本研究ではシステムの発話中にユーザの音声を検知した場合に修復行動を取り，競合を解消するシステムの構築を目指す．そのために，システムが取るべき修復行動を図 1 で示す 7 種類の行動―(1)Continue，(2)Stop，(3)Yield，(4)Resume，(5)Restart，(6)Revise，(7)React にまず分類した．これらの修復行動の設計方針について述べる．まず，そもそも相槌や環境音など，競合がそもそも起こっていない場合はシステム発話を停止せずそのまま話し続けるほうがよい．その場合の行動として (1)Continue を設計する．競合が起こった場合の最も単純な対処として，システムの発話を停止しユーザに発話権を譲る (2)Stop が考えられる．しかし，ユーザが発話を再開するタイミングを測りかね，長い沈黙が発生する可能性がある．その場合を考慮し,「どうぞ」などの発話をすることでユーザに発話権を積極的に譲る (3)Yield の修復行動を設計する．また，ユーザ側が発話を止めた場合，システムも発話を停止していると沈黙が続いてしまう．

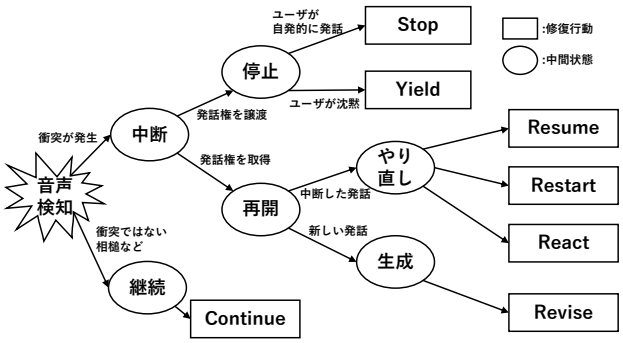

図 1: 発話交替の競合解消のためのシステムの修復行動

そのため，システム側が発話権を持つ場合の行動も設計する．競合が起きた際のシステム発話を再度発話させる行動を，中断地点から話し始める (4)Resume と，始めから話し直す (5)Restart の 2 種類に分け設計する．さらに，競合時のユーザ発話が「すごい」「面白い」のような評価的応答 [2] だった場合，それに反応を示してから発話を続けたほうがよい．そのため，評価的応答に対する定型文的な応答のあとに Resume を行う修復行動 (6)React を追加する．加えて，競合時のユーザ発話がシステム発話への質問あるいは訂正の指摘の場合，その発話をもう一度行うのは不自然である．この場合を想定し，新たな発話を生成する (7)Revise を設計する．

3. 対話データの収録

発話交替の競合を含む多人数対話データを収集するため，システムとユーザ役の被験者 2 名による三者対話の音声を多人数対話システムを用いて収録した．対話中に音声の重複が発生した場合，システムに修復行動をランダムに取らせて競合の様子を含む対話データを収集した．

3.1 対話状況

収録時には，被験者同士が隣り合って座り，2 名の正面のモニターに MMD エージェント ‡を投影させた．被験者にはピンマイクを装着させ，部屋全体の音を収録するマイクを三者の間に設置した．加えて，システムの発話音声も録音した．

システムが旅行代理店の案内役，ユーザ二人は友人同士であり，店を訪れた客という設定のもとで三者の対話を行わせた．ユーザには旅行先を決めるという目的を持たせたが，必ずしも時間内に目的の達成を目指さずともよく，話題の転換も許容した．対話は 1 組あたり 5 分程度の長さで，合計 10 組の対話を収録した．

† 大阪大学産業科学研究所 SANKEN, The University of Osaka

‡MMDAgent-EX https://mmdagent-ex.dev/ja/

3.2 多人数対話システム

本研究で用いた対話収録システムの構成について述べる．入力にはモノラル入力のピンマイクを 2 本用い，各マイク入力に対して発話区間検出を実行する．この処理には，DNN-HMM ベースの VAD ツールである pyadintools[§] [3] を用いている．ユーザ発話区間の音声を音声認識モデルに送信する．音声認識モデルには Kotoba-Whisper-v2.2[¶]を使用し，各マイクチャンネルごとに話者の音声を認識している．

音声認識結果は対話管理モジュールに送られ，そのテキストを GPT-4o-mini に入力し，システムの応答文を生成する．生成された応答は音声合成エンジン VoiceText によって wav ファイルに変換され，Python ライブラリの pygame を用いて再生される．pygame 上で wav ファイルの再生，中断，再開といった発話の制御を実現している．また，音声再生時には MMDAgent-EX に発話の内容に応じたリップシンク動作命令をリアルタイムで送信し，音声とエージェントの動作を連動させている．

システム発話中にユーザの発話開始を検知した場合には，発話重複用の処理を行う．重複が発生するたびに，同一の対応が連続しないよう考慮し，5 種類の行動を順に実行する．Continue の場合，ユーザの発話を無視してシステム発話を継続する．その他の場合にはシステム発話を一時停止し，たとえば Yield では 2 秒後に「どうぞお話しください」と案内し，Restart では中断された発話内容を最初から再生，Resume では中断された箇所から再生する．

4. 収録データに対するアノテーション

収録したデータに対し，第一著者と第二著者の 2 名によりアノテーションを行った．アノテーション対象箇所は，対話中にシステムが修復行動をランダムに行った箇所全てで，合計で 102 か所存在した．それぞれの箇所について，システムがどの修復行動を取るべきであったかのラベルを付与した．ラベルは Continue，Stop，Yield，Resume，Restart，React，Revise の 7 つである．

今回行ったアノテーションの基準について述べる，図 1 にあるように，まずシステム発話中にユーザ音声を検知した段階で，相槌やエラーによるものだと判定した場合 Continue とする．そうでない場合，競合後システムとユーザが発話を停止するまでの内容を聞き，ユーザが発話権を持って次話者となるべきだと判定した場合は Stop か Yield を付与する．ユーザがすぐに話を再開していれば Stop，そうでなければ Yield となる．システムが発話権を持つべきと判定した場合，ユーザ発話が訂正あるいは質問の場合 Revise を付与する．そうでない場合，システムが競合時の発話をもう一度発話することを想定し，Resume か Restart を付与する．競合前に発話した内容に重要な情報が含まれていれば Restart を付与し，そうでなければ Resume を付与する．また，Restart とした場合冗長になると判断した場合も Resume を付与する．ただし，ユーザが評価的応答をした場合には React を付与する．

2 名のアノテータによるアノテーション結果を表 1 に示す．Cohen の κ 値は 0.523 であった．アノテーションが一致しなかった事例として多かったのが Stop と Revise で割れたパターンである．一致しなかった事例を以下に示す．S はシステム，U はユーザ役の被験者を指す．また．[の時点で競合が発生している．

S: 特に行きたい都市や観光スポットはありますか？
例えば，札幌や [函館など
U: [おすすめとかあったりしますか？

この競合に対し，ユーザ発話の最後までの内容を考慮に入れたアノテータは Revise を選択したが，ユーザ発話の直後までを考慮したアノテータはまだユーザ発話の内容が確定していないと考え，ユーザの発話を待機する Stop を選択した．このように，競合時からユーザが発話を終了するまでの内容のうちどこまで考慮するかの基準がアノテータ間で異なっていたことが不一致の原因ではないかと考察できる．今後，基準について再考し，明確な基準のもと再度アノテーションを行う必要がある．また，現在の修復行動の分類は二者対話を想定したものであるが，今後は多人数対話特有の現象を考慮した修復行動の追加についても検討する必要がある．

表 1: アノテータ A,B によるアノテーション結果

A\B	Co	St	Yi	Rsm	Rst	Rea	Rev	計
Co	31	0	2	1	0	6	1	41
St	1	19	1	1	0	0	5	27
Yi	2	1	4	0	0	0	1	8
Rsm	4	0	2	1	2	0	1	10
Rst	0	0	0	0	1	0	0	1
Rea	0	1	0	0	0	4	0	5
Rev	1	1	1	0	0	2	5	10
計	39	22	10	3	3	12	13	102

Co: Continue，St: Stop，Yi: Yield，Rsm: Resume
Rst: Restart，Rea: React，Rev: Revise

5. おわりに

本稿では，発話交替の競合を解消するためシステムが取る修復行動の設計について述べた．また，多人数対話データを収録し，本稿内で設計した修復行動を元に対話データ中の競合事例にアノテーションを試み，その一致率を算出するとともにアノテーション不一致の原因について考察した．今後はアノテーション基準の精密化や修復行動分類の再考を行っていく．

参考文献

[1] Ethan Selfridge, Iker Arizmendi, Peter Heeman, and Jason Williams. Continuously predicting and processing barge-in during a live spoken dialogue task. In *SIGDIAL Conference*, pp. 384–393, 2013.

[2] Yasuharu Den, Nao Yoshida, Katsuya Takanashi, and Hanae Koiso. Annotation of Japanese response tokens and preliminary analysis on their distribution in three-party conversations. In *Oriental COCOSDA*, pp. 168–173, 2011.

[3] Ryu Takeda and Kazunori Komatani. Scale-invariant online voice activity detection under various environments. In *APSIPA ASC*, pp. 1–6, 2024.

[§]pyadintools https://github.com/ouktlab/pyadintoo
[¶]Kotoba-Whisper-v2.2 https://huggingface.co/kotoba-tech/kotoba-whisper-v2.2