タスク指向音声対話における大規模言語モデルを活用した柔軟な発話終了検知の検討

大竹真太 [1]　東佑樹 [1]　杉山雅和 [1]
[1] 株式会社 AI Shift
{otake_shinta, azuma_yuki, sugiyama_masakazu}@cyberagent.co.jp

概要

音声対話システムにおいて発話終了検知は, ユーザ意図の正確な理解とスムーズな対話の実現に不可欠である. 従来の沈黙時間ベースの手法は頑健だが, 発話末尾での不必要な待機による応答遅延がユーザ体験を損なう. 発話の意味的な内容を考慮するアプローチは, よりスムーズな発話終了検知を実現できる可能性がある一方で, 特定のドメインや発話スタイルに依存してしまい, 汎用性に乏しいという課題がある. そこで, 本稿ではタスク指向対話において, 大規模言語モデル（LLM）の文脈理解能力を活用することで, 柔軟かつ高速な発話終了検知を実現する新しい手法を提案する. 社内で収集した電話音声データを用いて検知の遅延時間を評価し, ベースラインよりも約 37.8%短縮できた.

1　はじめに

タスク指向型対話システムは, 解釈性と制御性の観点からパイプライン型のアーキテクチャが一般的に用いられている [1, 2]. このアーキテクチャでは, 音声認識 (ASR), 自然言語理解 (NLU), 対話管理 (DST), 言語生成 (NLG), 音声合成 (TTS) といったモジュールが直列に配置される. 発話終了検知機能は通常 ASR モジュールに統合されており, システム全体の処理フローにおいて最前段に位置することから, その精度は後続の意図理解の正確性を大きく左右する. 発話終了の検知が過度に早い場合, 音声対話システムにおける一種のエラーである発話区間の誤認識 [3] が生じ, ユーザの発話内容を十分に捉えられず, 結果としてユーザの意図を正確に把握することが困難となる. このように, ユーザとの自然な対話を実現する上で, 適切なタイミングでの発話終了検知は重要である.

最も基本的な発話終了検知手法である沈黙時間ベースの手法は, 音量やスペクトル特徴から無音区間を検出し, その長さが一定の閾値を超えた場合に発話終了と判断する. しかし, この閾値設定は難しく, 短すぎると言い淀みで誤判定が生じ, 長すぎるとユーザーの応答待ち時間が長くなり対話が不自然になるという課題がある [4, 5].

この課題に対し，近年では音声活動の時間的パターンを直接モデル化する Voice Activity Projection [6] が提案され，より自然な発話終了検知を実現している [7, 8]. この手法は言語に依存しない汎用的なモデル化を可能にする一方で，タスク指向対話において重要となる発話の意味的なまとまりや完結性といった言語的特徴を考慮し，制御することが難しいという課題がある.

また，発話の意味的な内容を考慮するアプローチとして，音声認識結果から発話の意味的なまとまりや完結性を判断する手法が研究されている [9, 10]. これらの手法は，発話の意味的な内容を考慮することでよりスムーズな発話終了検知を実現できる一方で，特定のドメインや発話スタイルに依存してしまうという課題がある.

本稿では，これらの課題を解決するため，LLM の高度な文脈理解能力と柔軟な推論能力を活用した新しい発話終了検知手法を提案する.

- タスク指向対話における大規模言語モデル（LLM）の文脈理解能力を活用した, 新しい発話終了検知手法の提案
- 対話の文脈と発話内容の意味的関係性を考慮した, 柔軟かつ高速で汎用的な検知手法の実現
- 実際の電話音声対話データを用いた定量的評価による, 提案手法の有効性の実証

本稿の構成は次の通りである. 第 2 章では提案手法について説明し, 第 3 章では実験について述べる. 第 4 章では実験結果とその考察を行い, 第 5 章でまとめと今後の課題について述べる.

2 提案手法

2.1 概要

提案手法は，ASR の出力テキストまたは音声を直接 LLM に入力し，言語情報を考慮した発話終了の判定を行う．これにより，従来手法における不要な待機時間の削減と，柔軟な発話終了検知の両立を目指す．提案手法は，図 1 のように大きく 2 つの主要な手順から構成される．まず，音声入力から適切なタイミングで LLM に渡す入力チャンクの作成を行う．次に，作成されたチャンクに対して LLM による発話終了検知を実施する．なお, LLM による発話終了検知が行えなかった場合に，従来の沈黙時間ベースの手法で発話終了を検知する．

2.2 入力チャンクの作成

入力チャンクは発話開始時刻から音声をバッファリングしたものとする．Google Speech-to-Text API[1)]（以下, Google STT）のストリーミングレスポンスである `stability` に基づき入力チャンクを作成する．`stability` は音声認識結果の安定性を表す指標であり，この値が高いほど音声認識結果が変化しにくいことを意味する．`stability` がある閾値を超えた時点までにバッファリングされたテキストないし音声を入力チャンクとして LLM による処理を実行する．

2.3 LLM による発話終了検知

LLM による判定には，入力モダリティおよび実行タスクによって以下のような構成を提案する．

2.3.1 入力モダリティ

入力モダリティとして, ASR モジュールの出力テキストであるテキストチャンク入力と音声信号を直接入力する音声チャンク入力の 2 つを LLM への入力形式として検討する．

2.3.2 実行タスク

LLM による発話終了検知では，以下の 2 つのタスク構成を提案する．

1. **発話終了検知のみ**：発話の継続可能性を 7 段階のリッカート尺度（1: 確実に発話が終了している 〜 7: 確実に発話が続く）で評価する．判定の際は，音声認識特有の誤認識や途切れの可能性，言い淀みや言い直しなどの口語表現を考慮する．音声入力の場合は，音量やスピードの変化，イントネーション，無音区間の長さや位置なども考慮要素として加える．
2. **発話終了検知と意図理解のマルチタスク**：発話終了検知に加え，対話システムのドメインに応じて事前定義された意図カテゴリへの分類と，対話タスクの遂行に必要なスロット抽出を行う．発話終了検知と意図理解タスクを同時に行うことで，言い淀みによる誤った終了検知を防ぎつつ，発話内容に基づいた意図理解を早期に行い，対話システムの応答遅延を短縮し，よりスムーズな対話が実現できると考えられる．

なお, 発話終了検知はリッカート尺度の評価値が 3 以下の場合に発話終了と判定する．また, 入力が音声の場合は音声認識タスクも同時に実行する．

3 実験

3.1 比較手法

3.1.1 ベースライン手法

入力音声チャンクの音量を計算して無音かどうかを判定し, 音量が閾値を超えた場合に発話開始を検知する. 発話開始後, 一定時間以上の無音区間が継続した場合に発話終了と判定する. また, 誤検知を防ぐため発話開始直後の短い無音を無視する猶予期間を設けた. 以下, この手法を Silence-based と記載する. なお, 無音区間の継続長の閾値は評価データセットにおける発話終了検知の誤検知率が 0 となり, 最小の遅延時間になるように 2.0 秒に設定した.

3.1.2 提案手法のバリエーション

提案手法は以下の 2 つの要素の組み合わせにより構成される．

1. **入力モダリティ**
   - テキストチャンク入力 (T-)
   - 音声チャンク入力 (S-)
2. **実行タスク**
   - 発話終了検知のみ (-EOS)
   - マルチタスク (-MT)

例えば，T-EOS は「テキスト入力」，「発話終了検知のみ」を組み合わせた手法を表す．これらの組み合わせにより，計 4 種類の提案手法のバリエーションを評価した．なお, 本稿では LLM として, Gemini

1） https://cloud.google.com/speech-to-text?hl=ja

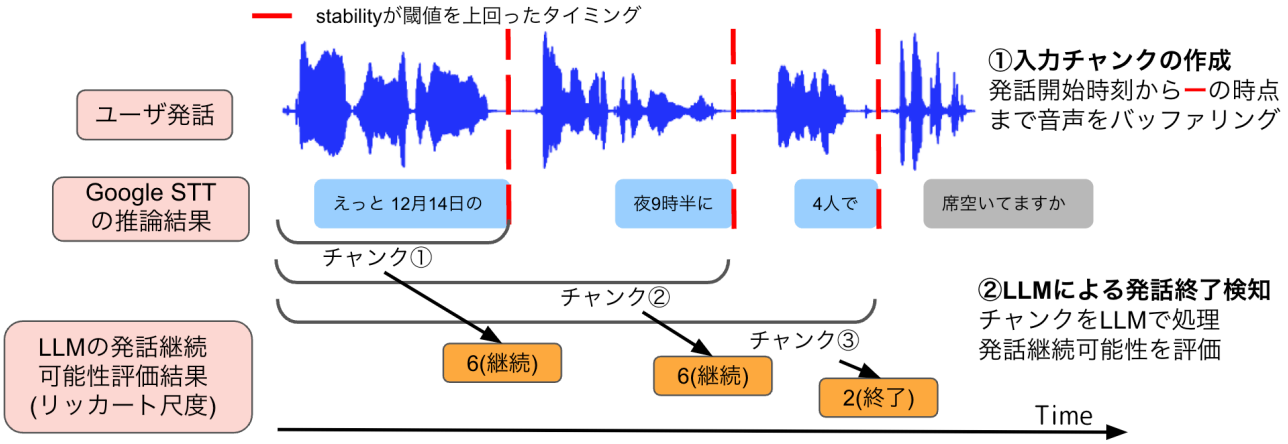

図 1: **提案手法の概略図.**

1.5 Flash [2] を使用した．なお, LLM により発話終了を検知できなかった場合はベースラインと同様の Silence-based の手法で検知する．

3.2 評価データセット

本稿では，付録 B に示す飲食店予約対話デモシステムを用いて収集した電話音声データを評価に使用した．データ収集には 59 名の弊社社員が参加し，合計 294 サンプルの電話音声セグメントを取得した．各音声セグメントの発話区間の特定には，pyannote-audio の VAD モデル[3] を使用し，バックグラウンドノイズ等による誤検出については手動で修正を行った．

3.3 評価指標

以下の 2 つの指標を評価に用いた．

- **平均遅延時間**：発話終了検知の速度を評価する指標として，実際の発話終了時刻に対する検知時刻の遅延を測定した．具体的には，遅延時間 $L = t_d - t_e$ [s] を算出し，$L \geq 0$ となるサンプルの平均値を評価に用いた．ただし，t_d は発話終了検知時刻，t_e は実際の発話終了時刻を表す．なお，LLM を用いた発話終了検知の場合は，LLM による処理時間を足した値を用いる．
- **誤検知率**：発話終了検知の頑健性を評価する指標として，実際の発話終了時刻より前に終了を検知してしまうケースの割合を算出した．本実験では誤検知率 E を $E = N_e/N_{total}$ で算出し, 0.15 秒の誤差を許容した．ただし，N_e は $L < -0.15$ となるサンプル数，N_{total} は全サンプル数を表す．

これらの指標により，提案手法の応答性と頑健性の両面から評価を行った．

4 実験結果と考察

表 1: **各手法における平均遅延時間と誤検知率.**

手法	平均遅延時間 [秒]	誤検知率 [%]
Silence-based	1.75	0.00
T-EOS	1.64	3.06
S-EOS	1.40	9.52
T-MT	1.38	7.48
S-MT	**1.09**	21.77

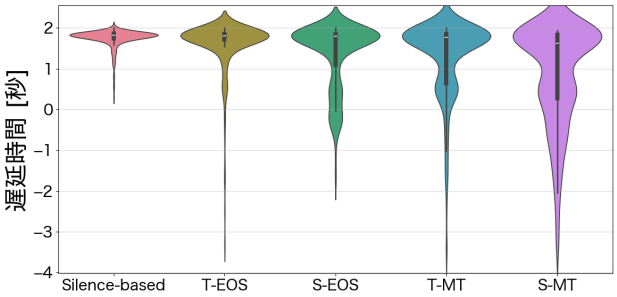

図 2: **各手法における遅延時間の分布.**

4.1 提案手法による遅延時間の短縮

表 1 に示すように，全ての提案手法バリエーションにおいて，ベースライン（Silence-based）と比較して平均遅延時間が短縮された．特に音声入力かつマルチタスクの S-MT では，平均遅延時間が 1.09 秒となり，ベースラインの 1.75 秒に短縮された．しかし

2） https://ai.google.dev/gemini-api/docs/models/gemini?hl=ja#gemini-1.5-flash
3） https://huggingface.co/pyannote/segmentation-3.0

表 2: **発話終了検知の評価値のクロス集計表.** 行と列の値は LLM によるリッカート尺度の発話終了検知の評価値である。枠線に囲まれた領域の値は各評価値のカウント数である. なお, 4* は 4〜7 を表す.

(a) T-EOS vs S-EOS

	1	2	3	4*
1	0	0	0	0
2	0	3	0	0
3	1	7	8	4
4*	0	10	39	222

(b) T-MT vs S-MT

	1	2	3	4*
1	1	1	0	1
2	11	6	0	4
3	3	18	26	7
4*	1	15	46	154

(c) T-EOS vs T-MT

	1	2	3	4*
1	0	0	0	0
2	0	3	0	0
3	2	3	15	0
4*	1	15	39	216

(d) S-EOS vs S-MT

	1	2	3	4*
1	0	1	0	0
2	11	7	0	2
3	5	20	19	3
4*	0	12	53	161

ながら, 誤検知率はベースラインよりも高くなってしまった.

4.2 入力モダリティの影響

表 1 から, 音声入力（S-）を用いた手法は，テキスト入力（T-）と比較して，遅延時間に関して優れた性能を示していることがわかる．また, 表 2a および表 2b を見ると, S-EOS と S-MT は T-EOS と T-MT が 4 を付与したサンプルに対して，1〜3 の値を付与する傾向があり, 音声入力を用いた方が積極的に発話終了と判断していることがわかる. これは音声に関する何らかの情報が発話終了の検知に活用されたためと考えられる.

4.3 タスク構成の影響

発話終了検知のみ（-EOS）とマルチタスク（-MT）の比較において，後者は平均遅延時間が短縮される一方で，誤検知率が増加する傾向が確認された．表 2c および表 2d の結果から，MT はより積極的に発話終了を判断する特性を示している．これは，意図理解の予測が完了したと判断されるタイミングで発話終了を検知するというモデルの振る舞いに起因すると考えられる．定性的分析において，音声的な途切れが存在しない状況でも，発話の意図が明確になった時点で発話終了を検知するケースが多く観察された．例えば,「すみません 明日の予約をキャンセルしたいんですけど」という発話に対し,「すみません 明日の予約をキャンセル」の時点で発話終了を検知するケースが確認された．このような場合，意図理解タスクの観点からは適切な検知が行われているものの，ターンテイキングの観点からは最適なタイミングとは言えない．一方で発話終了検知の誤りが意図理解タスクの性能に悪影響を及ぼすケースも多く見られた. 例えば,「今日 7 時の予約なんですけど ちょっと遅く変更できますか」という発話において，途中で発話終了を検知し，結果として意図理解タスクで誤った予測を行うケースがあった. これらの誤り種別を区別するために, 意図理解の精度を考慮し, 発話終了検知の評価指標に意図理解タスクの評価指標を含めることが有効であると考えられる.

4.4 遅延時間の分布

図 2 に示すように，ベースラインの分布は 1.75 秒付近に集中している一方，提案手法では 2 つのピークを持つ分布が観察された．発話終了検知のみを行う T-EOS と S-EOS では，ベースラインと比較して遅延時間が短縮されたサンプルが存在し, それらが小さな裾野の広いピークを形成している. マルチタスクを導入した T-MT と S-MT では，はっきりとした分布のピークが 0.5 秒付近に形成され，提案手法による遅延時間の短縮効果がマルチタスク推論によってさらに増強されたことが確認された．この結果は，タスク指向対話システムにおける発話終了検知において，言語的特徴の考慮が有効であることを示唆している.

5 おわりに

本稿では，LLM の文脈理解能力を活用したタスク指向音声対話における発話終了検知を提案し，ベースラインと比較して遅延時間を大幅に短縮した．特に音声入力とマルチタスク推論を組み合わせた手法は，遅延時間に関して最も優れた性能を示した.

しかしながら, 実用化に向けて実際の運用シーンに即した誤検知率の評価と改善が不可欠である. この課題解決のため, 今後はプロンプトの洗練や入力チャンクの最適化といったモデル側の改善に加え, 実用的な音声バッファリング制御などシステム側の工夫を検討する必要がある. 評価においては, ターンテイキングの円滑化と後続の意図理解精度への影響という 2 つの側面から誤検知を捉え, 詳細な分析を行うことが重要である.

参考文献

[1] Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, Vol. 63, No. 10, pp. 2011–2027, 2020.

[2] Jianfeng Gao, Michel Galley, and Lihong Li. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, p. 1371–1374, New York, NY, USA, 2018. Association for Computing Machinery.

[3] 駒谷和範, 福林雄一朗, 池田智志, 尾形哲也, 奥乃博. 音声対話システムにおける誤り原因の階層的分類とその推定に基づく発話誘導. 全国大会講演論文集, 第 70 回, 「情報爆発」時代に向けた新しい IT 技術基盤, pp. 97–98, 03 2008.

[4] S. McGlashan, D. C. Burnett, J. Carter, P. Danielsen, J. Ferrans, A. Hunt, B. Lucas, B. Porter, K. Rehor, and S. Tryphonas. Voice Extensible Markup Language (VoiceXML): Version 2.0. W3C Recommendation, Mar 2004. W3C Recommendation.

[5] S. Witt. Modeling user response timings in spoken dialog systems. *International Journal of Speech Technology*, Vol. 18, No. 2, pp. 231–243, 2015.

[6] Erik Ekstedt and Gabriel Skantze. Voice activity projection: Self-supervised learning of turn-taking events. In *Interspeech 2022*, pp. 5190–5194, 2022.

[7] Koji Inoue, Bing'er Jiang, Erik Ekstedt, Tatsuya Kawahara, and Gabriel Skantze. Real-time and continuous turn-taking prediction using voice activity projection, 2024.

[8] Kazuyo ONISHI, Hiroki TANAKA, and Satoshi NAKAMURA. Multimodal voice activity projection for turn-taking and effects on speaker adaptation. *IEICE Transactions on Information and Systems*, Vol. advpub, p. 2024HCP0002, 2024.

[9] Raveesh Meena, Gabriel Skantze, and Joakim Gustafson. Data-driven models for timing feedback responses in a map task dialogue system. *Computer Speech & Language*, Vol. 28, No. 4, pp. 903–922, 2014.

[10] Linda Bell, Johan Boye, and Joakim Gustafson. Real-time handling of fragmented utterances. In *Proc. NAACL workshop on adaptation in dialogue systems*, pp. 2–8, 2001.

A　実験で用いたプロンプト

図 3 に S-EOS で用いたプロンプトを示す. このプロンプトをベースとして, 各々の提案手法のプロンプトを作成した. 入力モダリティがテキストの場合, 入力が音声認識結果であることを明記し, 無音区間や音声の途切れ方などの音声特有の情報を考慮するような指示を削除した. また, 出力形式から `transcription` を削除した. タスクがマルチタスクの場合は, 意図分類とスロットフィリングを行うように指示を追記し, 出力形式に `intent` と `slot` を含めた.

与えられた音声データはタスク指向対話における各ターンのシステムからの質問に対するユーザ発話です。この音声データに対して話者がさらに発話を続ける可能性を 7 段階で評価してください。会話の継続性の判定話者がさらに発話を続ける可能性を 7 段階で評価してください。発話が完全に終了したと明確に判断できる場合を除き、継続の可能性を考慮して評価してください。特に、システムからの質問に対して明確な回答がまだ得られていない場合や、ユーザが追加で情報を伝えようとしている可能性がある場合は、高い継続性を示す評価を選択してください。

1: 確実に発話が終了している (これ以上発話が続くことは考えられない)
2: ほぼ確実に発話が終了している (ごくわずかな可能性はあるが、ほぼ終了)
3: おそらく発話が終了している (継続の可能性は低い)
4: どちらともいえない (発話が終了したとも継続するとも判断できない)
5: おそらく発話が続く (継続の可能性が高い)
6: ほぼ確実に発話が続く (間を置いて発話が継続する可能性が非常に高い)
7: 確実に発話が続く (発話が途中で区切れており、続きの発話が確実にある)

判定の際には以下の要素を考慮してください。その中でも音声的な特徴を重視してください。
- 末尾の品詞：
  - 終助詞、助動詞、あるいは名詞などの単語で区切られて終了する可能性が高い。
  - 「〜に」や「〜の」は発話が継続している可能性が高い。
  - 「〜だ」や「〜です」は発話が終了している可能性が高い。
  - 「すいません」は意図が明確でないので継続している可能性が高い。
- 音声認識特有の誤認識や途切れの可能性：認識エラーによって不自然に発話が途切れている可能性を考慮する。
- 言い淀みや言い直しなどの口語表現：これらは発話の終了を意味しないため、継続の兆候として捉える。
- 無音区間の長さや位置：発話の末尾に不自然に短い無音しかない場合や、発話途中の一呼吸程度の短い無音は継続の可能性がある。
- 音声の途切れ方：発話が自然な終わり方でなく、不自然に途切れている場合は、発話が継続する可能性が高い。
- システムからの質問に対する応答の完結度：質問に対して十分に答えられているか、追加の情報提供が必要ではないかを判断する。

出力形式:

```
{
    "continuationLikelihood": <1-7 の整数>,
    "transcription": <文字起こし結果>,
}
```

図 3: 提案手法 (S-EOS) で用いたプロンプト

B　飲食店予約対話システム

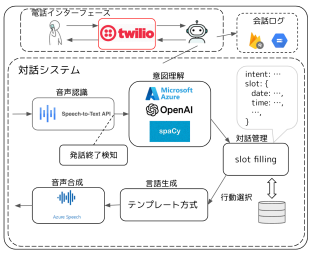

図 4: **システム構成図.**

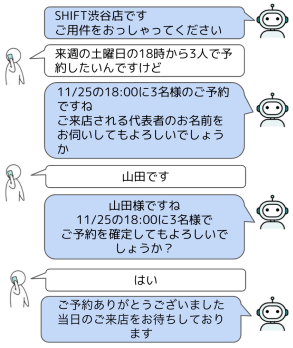

図 5: **飲食店予約対話の一例.**

評価データ収集に使用した飲食店予約対話システムのシステム構成図を図 4 にシステムとの対話例を図 5 に示す. 評価データの収集にあたっては，弊社社員に対して，新規予約，予約変更，予約確認，キャンセル，店舗への問い合わせなど，実際の利用シーンを想定した様々な用件を指示し，電話による対話データを収集した．これにより，実環境に即した自然な対話データの取得を試みた.

C　沈黙時間の閾値を変化させたときの実験結果

表 3: **発話終了検知の遅延時間と誤検知率.**

手法	閾値 (s)	平均遅延時間 (s)	誤検知率 (%)
Silence-based	4.0	3.75	0.00
	3.0	2.75	0.00
	2.0	1.75	0.00
	1.0	0.73	5.44
T-EOS	4.0	3.47	3.06
	3.0	2.56	3.06
	2.0	1.64	3.06
	1.0	0.69	8.50
S-EOS	4.0	2.92	8.50
	3.0	2.16	8.50
	2.0	1.40	9.52
	1.0	0.61	13.27
T-MT	4.0	2.84	7.82
	3.0	2.13	7.82
	2.0	1.38	7.48
	1.0	0.63	12.59
S-MT	4.0	2.14	22.45
	3.0	1.62	22.11
	2.0	1.09	21.77
	1.0	0.50	25.51