対話の齟齬と介入による解消：LLM を用いた検討

清水周一郎　Yin Jou Huang　村脇有吾　Chenhui Chu
京都大学大学院情報学研究科
{sshimizu,huang,murawaki,chu}@nlp.ist.i.kyoto-u.ac.jp

概要

対話における齟齬は重要な現象であるが，どのような現象であるかは明確でない．本研究では，対話の齟齬について，Clark の言語使用に関する理論をもとにして整理し，また齟齬の解消には第三者による介入が役立つことを指摘する．LLM は高い対話能力と広範囲な知識を持ち，コントロールされた対話の実験を容易にする．本研究では LLM エージェントに齟齬を含む対話とそれに対する介入を行わせ，介入を行うことによる対話の変化を分析する．

1　はじめに

対話における齟齬は重要な現象である．2024 年 1 月 2 日に発生した航空機の衝突事故 [1, 2] においても，管制官と海上保安庁機の搭乗員との間の対話の齟齬が原因の 1 つであると見ることができる．一方で，齟齬がどのような現象であるかを定義するのは難しく，著者らの知る限り齟齬を対象とした自然言語処理の研究は存在しない．齟齬は自然な対話では稀な現象であるため，データを大規模に収集して実験することは容易でない．

大規模言語モデル（LLM）[1)] は，高い対話能力を示し，広範囲な知識を持つ [5, 6]．LLM を対話研究に利用する利点の一つとして，人間の場合では不可能な，条件の同じ対話を複数回行わせることが可能になることが挙げられる．そこで，本研究では，LLM を利用して齟齬を含む対話を生成し，それをもとに齟齬に関する分析を行う．

齟齬は，Clark の提唱した共通基盤（話者と対話相手の間で共有された知識や信念）の解釈の不一致であると捉えられる．これを LLM でモデル化するには，LLM の持つ知識を制限し，異なる知識を持つ LLM エージェントを用意する必要があるが，容易には実現できない．そこで，LLM にその場限りでの知識を持たせ，話者と対話相手に相当する LLM エージェントの間で情報量の差を持たせるための設定として，言葉当てゲームをベースとした設定を提案する．

齟齬の解消においては，話者と対話相手双方の共通基盤をよく理解する第三者の介入が役立つ．LLM は広範囲な知識を持つため，この介入者の役割を果たすのに適している．ゲームの設定においても，話者と対話相手の持つ知識の差を認識させた LLM エージェントにこの役割を行わせることができる．以上を踏まえ，本研究では言葉当てゲームの設定において，齟齬の発生・解消と介入による影響を定量的に評価することを試みる．

2　関連研究

対話システムとしての LLM の登場以前には，対話システムの改善に関する研究が多く行われてきた．対話破綻の研究 [7] では，対話システムの不十分な性能に起因する対話破綻の分類がなされた．また，対話システムの応答の際に共通基盤を考慮することによって応答性能を向上させる研究がなされた [8]．本研究は人間と対話システムの対話ではなく，人間同士の対話で起こりうる齟齬を対象とする点が異なる．

LLM エージェント同士の対話に関する研究が行われるようになってきている．Park ら [9] はゲーム上の環境で 25 の異なる人格を持つエージェントを用意し，各エージェントに記憶，計画，反応，思考の機能を持たせて行動させる実験を行い，LLM エージェントによる社会的行動の発現を示した．本研究では Park らのエージェント設計を参考にしつつ，齟齬を含む対話に焦点を当てている．

1) 本稿では LLM を，言語モデルの目的関数を用い大規模なテキストデータで訓練されたモデルとしてではなく，instruction finetuning [3] や reinforcement learning from human feedback [4]，あるいはそれに相当する処理が行われ，多様なベンチマークで人間に匹敵する，または人間を超える性能を持つモデルを指すのに用いる．（この条件が満たされている場合，使われているデータやパラメータ数が大規模であるかどうかは問わない．）本研究では GPT-4 を用いる．

3 齟齬の発生と解消

3.1 齟齬の定義

齟齬について，Clark の共同作業に関する理論（[10] pp. 148-153, 234-235）をもとにして整理する．[2]Clark は，対話は**話者**と**対話相手**とが織り成す**共同作業**の一種であるとし，共同作業は**共同行動**から成るとした．共同行動には i) 行動と注目，ii) 提示と同定，iii) 合図と認識，iv) 提案と考慮の 4 つの段階がある．例えば，A が椅子を指差しながら B に「ここに座って」と言った場合，これは以下のようになる．

i. A は B の知覚の範囲内で椅子を指差して発声し，B はそれに注意を向ける．
ii. A は指差しと発声によって信号を作り出し，B はその信号を認識する．
iii. A は B が椅子に座るよう頼み，B はその頼みを認識する．
iv. A は B が A のために椅子に座ることを提案し，B はその提案を受け入れるかどうか考える．

対話は共同作業の一種であり，この n 段階目までが達成されるが $n+1$ 段階目が達成されない（$n = 0, 1, 2, 3$）ことによって，異なる種類のミスコミュニケーションが起こりうる．例を挙げると，

$n = 0$　相手がイヤホンをしていて話者の声が聞こえない．
$n = 1$　話者がアメリカ人に日本語で話しかけ，相手に理解されない．
$n = 2$　話者が「どの人が X さんですか？」と聞き，相手が「X さんを知りません」と答える．
$n = 3$　話者が「ここに座って」と言うが，相手が意図的に座らない．

我々は，日本語の「**齟齬**」に相当するのは，この $n = 2$ の場合であるとみなし，本研究はこうした齟齬を対象とする．[3]

3.2 齟齬の発生

齟齬の発生する原因について，Clark の共通基盤の分類を参考にして整理する（[10] pp. 92-121）．共通基盤は，まず，共同体共通基盤と個人的共通基盤に分類できる．共同体共通基盤は，いわゆる文化的な共同体のもつ，その共同体の中でのみ共有されている情報の総体である．Clark は共同体を国籍，職業，趣味など 13 種類に分類している（[10] p. 103）．ここで，共同体とは，他の共同体が持たない専門的な情報を持つ集団であるとしている．個人的共通基盤は，ある個人と別の個人の関わりに基づく情報である．

これを踏まえて，齟齬の発生する原因について考える．まず，個人的共通基盤については，話者と対話相手のそれぞれが同じように蓄積させるため，齟齬の原因となるとは考えにくい．一方で，共同体共通基盤については，話者と対話相手が持っている情報が異なるため，齟齬の原因となりうる．例えば，transformer という単語を聞いたときに，NLP の研究者であればモデルを，電気技師であれば変圧器を，映画の好きな人であれば映画を思い浮かべるだろう．このように，対話者間の持っている情報量に差があることは，齟齬が生じるための条件の一つであると考えられる．

齟齬は，このような原因から，話者の発話の意図を，対話相手が誤解することに始まる．誤解した対話相手は，その後の発話で，話者が想定していなかった内容の発話をする．本稿では，これを齟齬が発生した時点であると考え，以後，齟齬が解消されるまでの間，対話に齟齬が生じていると考える．

3.3 齟齬の解消

次に，齟齬の解消について検討する．Clark は，共同作業の参加者は，共同行動が成功したという証拠を得る（closure）ことによって，その共同行動を共通基盤の一部として確立する（grounding）と述べている（[10], pp. 221-252）．例えば，話者の「X を知っていますか？」という発話に対し対話相手が「はい」と答えた場合，話者は相手が X を知っているという証拠を得たことになる．

これを齟齬に当てはめて考えてみる．齟齬は，話者の発話の意図を，相手が誤解[4]することに始まるので，話者が，相手が理解したという証拠を得たとき，齟齬が解消されたと見ることができる．

2） 本稿で用いる専門用語は Clark が用いたものの拙訳である．付録 A に訳の対応を示す．
3） 聞き間違いは $n = 1$ の場合に相当し，聞き間違いしたまま対話を続けて齟齬が生じることは考えられるが，本研究の対象範囲外とする．
4） 誤解はある参加者の状態を指し，齟齬は対話全体の状態を指すものとする．

3.4 第三者による介入

最後に，第三者による介入について考える．共同行動に関係する人の分類として，話者，対話相手，副参加者，傍観者，盗聴者がある（[10], p. 14）．第三者はこのうちの副参加者に相当し，話者が発話時点で直接の対話相手としては認識していないが，話を聞いている人として認識しており，対話に参加している者である．

経験的に，齟齬の解消においては，対話全体を理解する第三者の介入が鍵となることが多い．例えば，研究室での研究のミーティングにおいて，発表者と質問者の間で齟齬が起きたとき，両者の意図をよく理解する先生が対話に介入することで話が進むことがよくある．共通基盤の文脈で捉えれば，第三者が話者と対話相手の共通基盤を双方の対話者よりもよく理解している場合に，第三者による介入が有効であると考えられる．

第三者が対話に介入する意義は，対話の**効率**の向上であると言える．対話においては，話者が対話相手に伝えたい内容（メッセージ）を持っており，対話相手が話者のメッセージを理解することによって対話が成立する．この際，メッセージがなかなか聞き手に伝わらないと，話し手はメッセージを表現を言い換えて繰り返すことになる．これを対話の効率が悪い状態であるとし，一方でメッセージがスムーズに対話相手に伝わる状態を対話の効率が良い状態であるとする．第三者が話者と対話相手双方の意図を理解し，適切に対話に介入することができれば，対話の効率を向上させることができる．

4 LLM を用いた検討

4.1 状況設定

齟齬の発生を適切にモデル化するには，3.2 節で述べたように，LLM がある一定の範囲の情報のみを持つ必要がある．しかし，LLM は膨大な量のテキストデータで訓練されており，ある範囲の知識のみを持つように設定することはできない．[5)] そこで，人工的に齟齬が必ず起きるような対話の設定を考え，その上で介入による齟齬の解消を試みる．

具体的には，言葉当てゲームに手を加えたものを LLM エージェントに行わせる．エージェントとして，通常のエージェント A，B，C，介入機構 D，オラクル O を設定する．A，B は質問者である．何らかの単語をあらかじめ与えておき，その単語に関するヒントを C に与える．ここで，A と B に異なる単語を与えることで，A と B の持つ情報量に差をつけ，A や B 自身には共通の単語を想定していると認識させることで，齟齬の発生を人工的にモデル化する．C は回答者であり，A や B のヒントをもとに，単語を推測する．D は対話に介入する第三者であり，必要に応じて対話に介入し，齟齬を指摘する．ここで，3.4 節で述べたように，D は A や B よりもよく共通基盤を理解している必要があるため，A と B が異なる単語を想定している可能性があることをあらかじめ認識させる．O は，LLM エージェントによる対話を自然なものに近づけるために導入した機構である．エージェント同士が対話をする中で，あるエージェントが発話を他の特定のエージェントに向けたものとして発する場合がある．こうした状況を考慮するため，O は対話全体を俯瞰し，どのエージェントが発話するべきかを対話の各時点で判断する．

4.2 エージェント設計

エージェント A，B，C は以下の機能をもつ．

- `memory`: エージェントの初期状態および全エージェントによる話者と発話を時系列順に並べたもの．
- `speak()`: 現在の発話履歴に続けて，適切な発話を返す．

介入機構 D は，通常のエージェントの機能に加えて，以下の機能をもつ．

- `should_speak()`: 現在の発話履歴を受け，対話に介入するかどうかを判断する．

オラクル O は，通常のエージェント同様の `memory`，および以下の機能をもつ．

- `decide_speaker()`: 現在の発話履歴を受け，A，B，C のうち誰が次に発話すべきか判断する．

具体的なプロンプトは付録 B に示す．各エージェントの発話順を示すアルゴリズムは以下の通りである．

5) 著者らが行った事前実験では，プロンプトの調整によって LLM の持つ知識の範囲を調整することは困難であった．例えば，NLP の知識を持たない電気技師を模倣させようとしても，NLP の知識を持っているかのような振る舞いをした．LLM の持つ特定の知識を忘却させる研究等は現在行われているところであり，そうしたモデルの活用は今後の課題としたい．

Algorithm 1 LLM エージェントによる対話と介入

```
Initialize A, B, C, D, O, max_n_utts
A.speak()
B.speak()
C.speak()
n_utts ← 3
while n_utts < max_n_utts do
  if D.should_speak() then
    D.speak()
  end if
  speaker ← O.decide_speaker()
  speaker.speak()
  n_utts ← n_utts + 1
end while
```

4.3 評価

LLM エージェントによる対話に対し，1) 齟齬が発生した時点，2) 介入が行われた時点，3) 齟齬が解消された時点の 3 点について著者らが評価を行う．まず，各発話に 1 から番号をつける．齟齬が発生した時点は，3.2 節で述べたように，齟齬が表面化した発話の番号，すなわちある発話がそれまでの発話の内容に沿っていない場合，その発話の番号とする．介入が行われた時点は，介入機構が介入が必要であると判断し発話したときの発話の番号である．齟齬が解消された時点は，3.3 節の議論をもとに，全ての参加者が齟齬があったことを理解した時点とする．すなわち，A，B，C のそれぞれについて，齟齬があったことを認識した旨の発話をした発話番号を記録し，その最大値を齟齬が解消された時点とする．これらをもとに，効率指数を，齟齬が解消された時点と齟齬が発生した時点の差として定義する．

4.4 実験設定

実験は全て英語で行った．LLM としては OpenAI の `gpt-4-0613` を用いた．temperature は 1.0 とした．最大発話数は 20 発話とした．但し，対話に発展性がないと判断した場合（終わりの挨拶の繰り返しになる場合や次のゲームに移る場合）はそれ以前でも対話を打ち切った．A の単語と B の単語のペアが同一のものについて 5 回対話を行い，3 ペアについて試した．ここでは，`gpt-4-0613` に生成させた類義語のペアから選び，(A の単語, B の単語) = (beautiful, gorgeous), (wisdom, knowledge), (end, finish) とした．介入機構ありの場合となしの場合の計 30 対話を行わせ，評価を行った．

表 1 介入なしの場合とありの場合の評価結果．齟齬発生時点，介入時点，効率指数は（平均）±（標準偏差）の形で示している．†: 齟齬が発生したものについてのみ計算した．‡: 齟齬が解消したものについてのみ計算した．介入なしの場合は 1 件のみのため省略した．

	介入なし	介入あり
齟齬発生件数	13/15	13/15
齟齬発生時点 †	5.46 ± 0.93	5.62 ± 1.15
介入時点 †	-	6.62 ± 1.15
齟齬解消件数 †	1/13	9/13
効率指数 ‡	-	6.33 ± 3.37

4.5 結果

結果を表 1 に示す．まず，齟齬が必ず発生すると想定されるシナリオで実験したものの，実際に齟齬が表面化しなかったものが各 2 件あった．これは LLM がプロンプトの指示に従わない hallucination の問題が原因である．次に，齟齬が発生した時点については，介入なしの場合とありの場合で大きな差は見られなかった．齟齬が発生するまで介入機構は介入しないため，予想された結果であると言える．介入時点について見ると，齟齬発生時点の 1 発話後に介入が行われている．このことから，介入機構の介入を判断する部分が適切に機能していることが分かる．齟齬解消件数を見ると，介入なしの場合に比べ，介入ありの場合の方が齟齬を解消できている件数が多く，介入機構によって齟齬が解消できていることが分かる．但し，介入なしの場合でも，プロンプトの調整により齟齬を解消できるようになる可能性もあるため，今後の課題である．効率についても，介入なしの場合には件数が少なく定量的に評価できなかったため，今後の課題とする．

5 おわりに

本研究では，対話の齟齬について Clark の理論をもとに整理した上で，齟齬の解消には第三者による介入が役立つことを指摘した．LLM エージェントに齟齬を含む対話とそれに対する介入を行わせ，齟齬の発生・解消と介入による影響の定量的評価を試みた．今後は，聞き間違いに起因する齟齬や，異文化コミュニケーションにおける齟齬などについても研究したい．

謝辞

本研究に関して有益な議論をいただいた黒橋禎夫教授，河原達也教授，Rafik Hadfi 准教授，Devish Lala 研究員に感謝します．本研究は JSPS 科研費 JP23KJ1356 の助成を受けたものです．

参考文献

[1] NHK NEWS WEB. 羽田空港事故交信記録やり取り詳細 “18 分の避難”機内で何が. https://www3.nhk.or.jp/news/html/20240103/k10014308031000.html, 2023.

[2] NHK NEWS WEB. 羽田空港事故管制官 海保機に離陸順番 1 番と伝え許可と認識か. https://www3.nhk.or.jp/news/html/20240105/k10014310951000.html, 2023.

[3] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned Language Models are Zero-Shot Learners. In **International Conference on Learning Representations**, 2022.

[4] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. arXiv:2203.02155, 2022.

[5] OpenAI. GPT-4 Technical Report. arXiv:2303.08774v4, 2023.

[6] Google Gemini Team. Gemini: A Family of Highly Capable Multimodal Models. arXiv:2312.11805, 2023.

[7] 東中竜一郎, 荒木雅弘, 塚原裕史, 水上雅博. 雑談対話システムにおける対話破綻を生じさせる発話の類型化. 自然言語処理, Vol. 29, No. 2, pp. 443–466, 2022.

[8] Pei Zhou, Hyundong Cho, Pegah Jandaghi, Dong-Ho Lee, Bill Yuchen Lin, Jay Pujara, and Xiang Ren. Reflect, Not Reflex: Inference-Based Common Ground Improves Dialogue Response Quality. In **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, 2022.

[9] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative Agents: Interactive Simulacra of Human Behavior. In **Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology**, 2023.

[10] Herbert H. Clark. **Using language**. Cambridge university press, 1996.

A　Clark の用語とその訳の対応表

表 2　訳の対応

speaker	話者
addressee	（対話）相手
participant	参加者
side participant	副参加者
bystander	傍観者
eavesdropper	盗聴者
joint activity	共同作業
joint action	共同行動
execution and attention	行動と注目
presentation and identification	提示と同定
signaling and recognition	合図と認識
proposal and consideration	提案と考慮

B　プロンプト

B.1　各エージェントの初期状態

```
agents:
  agent1:
    name: "Alice"
    initial_memory: "You are Alice. You
        are going to play a game with Bob
        and Claire. You and Bob have the
        same word in mind. Claire needs to
        guess the word. The word is '${
        word1}'. Once Claire makes a guess,
         you can give one hint at a time.
        You cannot use the word '${word1}'
        or '${word2}' in the hint. You
        start the conversation by
        explaining the game and giving a
        hint."
  agent2:
    name: "Bob"
    initial_memory: "You are Bob. You are
        going to play a game with Alice and
         Claire. You and Alice have the
        same word in mind. Claire needs to
        guess the word. The word is '${
        word2}'. Once Claire makes a guess,
         you can give one hint at a time.
        You cannot use the word '${word1}'
        or '${word2}' in the hint."
  agent3:
    name: "Claire"
    initial_memory: "You are Claire. You
        are playing a word guessing game
        with Alice and Bob. Alice and Bob
        have the same word in their minds
        and you need to guess it. You can
        only guess one word at a time."
  meta_agent:
    name: "David"
    initial_memory: "You are David. You
        are listening to a conversation
        between Alice, Bob, and Claire.
        They are playing a word guessing
        game. Alice and Bob have a word in
        their mind and Claire needs to
        guess it. Actually, the words Alice
         and Bob have in mind could be
        different. You usually just listen
        to the conversation, and only when,
         rememeber, only when a
        miscommunication happens, you
        intervene in the conversation and
        give some feedback."
oracle:
  initial_memory: "You are listening to a
      conversation between Alice, Bob, and
      Claire. Your task is to decide who
      among Alice, Bob, and Claire is most
      likely to speak next at a certain
      point in the conversation."
```

B.2　エージェントの機能

```
speak():
    prompt = " ".join(self.memory) + "
        What would you say? You would say:
        "
should_speak():
    prompt = " ".join(self.memory) + "
        Should you speak now? Yes/No: "
decide_speaker():
    prompt = " ".join(self.memory) + " Who
         is most likely to speak now?
        Answer in one word from Alice, Bob,
         and Claire: "
```