# 長期対話における人間関係の関係照会・関係更新タスクの設計

安田大朗 [1]　安藤一秋 [2]
[1] 香川大学大学院 創発科学研究科 [2] 香川大学 創造工学部
{s25g366,ando.kazuaki@kagawa-u.ac.jp}

## 概要

長期雑談対話では，会話履歴に基づいてユーザの人間関係を一貫して参照・更新できることが重要である．しかし，人間関係は複数人物を同時に扱い，多段推論を要する関係連鎖（例：A の友達の上司）と，会話に基づく関係変化（例：交際・離婚）を同時に処理する必要があため，人物同定の誤りが生じやすい．既存の長期記憶ベンチマークでは，人間関係の特有な関係照会，多段推論，更新対象の人物ペアの同定および関係更新を切り分けて評価する指標が十分整理されていない．本稿では，人間関係知識を知識グラフとして与える関係照会と会話に基づく関係更新の 2 タスクを設計し，分離評価可能な枠組みを検討する．

## 1　はじめに

大規模言語モデル（LLM）の発展により，ユーザと複数セッションにわたって会話を継続する長期雑談対話が現実的になってきている．長期対話では，過去の会話履歴に基づく応答の一貫性や，ユーザの嗜好・状況への配慮が重要となる [1]．一方で，LLM が扱えるコンテキスト長には制約があるため，外部メモリを用いて必要な情報を蓄積し，検索・参照・更新する枠組みが提案されてきた [2, 3]．また，長期記憶の取り扱い能力を評価するベンチマークも近年整備されつつある [4]．

しかし，長期記憶のうち，人間関係は「人物-関係-人物」で構成される構造的知識であり，単体の属性（例：年齢，職業）やイベント（例：旅行に行った）とは本質的に異なる [5]．人間関係の記憶では，複数の人物を同時に扱う必要があるため，質問や対話の文脈において，どの人物を指しているのか誤って解釈してしまう可能性がある．また，「A の友達の上司」のような関係の連鎖は，推論過程における人物の予測集合が複雑化し，出力が過剰になる場合がある．さらに，会話の進行に伴って人間関係は変化するため，関係更新の判定，更新対象となる人物ペアの特定，矛盾しない関係更新が要求される．特に，実際の対話では，人名が明示されず「アルバイトの同僚」のように間接的に言及されることも多く，間接表現を知識グラフ上の具体的な人物に紐づける処理が誤りを生じさせる要因となり得る．

本研究の目的は，長期対話における人間関係の扱いを，固定知識に対する関係照会と，会話に基づく関係更新に分離し，内在する誤りの要因を切り分けて分析できる評価の枠組みを提示することである．既存の長期記憶ベンチマークは，多様な記憶を包括的に評価する一方で，人間関係特有の課題である人物の取り違えや，更新対象人物ペアの特定，関係更新の妥当性を評価する枠組みが十分に整理されていない．そこで本稿では，人間関係に特化した評価設定の下，関係照会と関係更新を分離して評価できる枠組みを検討し，LLM を用いた人間関係知識グラフにおける誤り要因を分析する．

## 2　関連研究

**長期対話における外部メモリ**　LLM のコンテキスト長の制約を補うため，外部メモリを導入し，必要な情報を選択的に参照・更新する枠組みが提案されている [2, 3]．これらの研究は，長期対話システムにおけるメモリ管理の実装例を示す一方，人間関係のような構造的知識について，更新対象となる人物の特定や関係更新の過程を分解し，誤り要因を分析することは想定していない．

**長期記憶ベンチマーク**　長期対話を対象に，長期記憶の保持・活用能力を評価するベンチマークが整備されつつある [4]．これは，長期記憶を包括的に評価する上で有効であるが，人間関係に焦点を当てた場合，人物の取り違え，関係の段階的推論における誤り，会話に基づく関係更新の誤りといった要因を切り分けて評価することは困難である．

**LLM の知識グラフ理解**　LLM にグラフ構造を入力する方法として，ノードとエッジを隣接リストや

**表 1**　関係照会タスクと関係更新タスクのカテゴリ定義と例

| タスク | カテゴリ | 説明 | 例 | 正解 |
|---|---|---|---|---|
| 関係照会 | c1 | 1-hop 照会（隣接人物に限定） | 土屋の同僚は誰？ | [近藤] |
| | c2 | 2-hop 照会 | 石井の友達の後輩は誰？ | [成田] |
| | c3 | 属性条件付き 1-hop 照会 | 私の研究室の後輩の友達は誰？ | [秋山] |
| 関係更新 | 置換 | 会話により関係が置換される | 土屋と付き合った | [私, 土屋, 恋人] |
| | 削除 | 会話により関係が取消される | 木村と別れた | [私, 木村,NONE] |
| | 維持 | 関係変化がない | 武田と斎藤はこの前も話してた | [] |

トリプルに変換してテキスト化する手法が提案されており，入力形式の違いが性能に与える影響も報告されている [6, 7, 8]．これらの研究は，主にグラフの照会や推論を対象としており，長期対話における関係更新や，人名が省略された表現を含む人間関係の継続的な管理などは議論の中核に含まれていない．

# 3　評価設計

本稿では，人間関係を知識グラフとして与えたうえで，固定された人間関係に対する関係照会と，会話に基づく関係更新のタスクの二つを設定し，それぞれ評価する．

## 3.1　人間関係知識グラフの定義

本稿では，人間関係を「人物-関係-人物」の組で表現し，関係を知識グラフ上の有向エッジに対応付ける．各人物ノードは，属性（年齢，職業，グループ，サブグループ）を持つ．知識グラフは，ユーザを中心に構成し，ユーザとそれ以外の人物ノードは，すべて 1-hop で接続されるものとする．グループは，大学生の人間関係が主に大学，家族・親戚，アルバイトに集約されるという調査結果 [9] に基づき，この 3 グループを採用する．調査結果に基づき，本稿の実験では，総ノード数を 38（大学 15，家族・親戚 12，アルバイト 11）とする．サブグループは，各グループ内の集団を表し，その数は調査結果を参考に定義した．年齢，職業，サブグループ名（例：研究室，授業，親戚）については，評価タスクの設計のため著者が付与した．

エッジは，あらかじめ定義した関係語彙（付録 4）に基づいて生成する．対称関係（例：友達，同僚）は，双方向の関係が一致するものとし，非対称関係（例：先輩/後輩，上司/部下）は，対応する逆関係を事前に定義して用いる．サブグループ内では，対称関係と非対称関係の 2 種類のエッジを生成する．対称関係は，ノードの生成順に基づいて隣接ペアを形成する．非対称関係は，人物を年齢順にソートし，隣接ペアとしてエッジを付与する．また，アルバイトは，最年長者から年少者に向かってスター型にエッジを付与する．この設計により，同一の人物ペア間に複数の有向関係が存在し得る（例：同僚かつ上司/部下）．本稿では，LLM へ知識グラフを入力する形式として，人物一覧と関係一覧（A-関係-B），サブグループの所属一覧（例：サークル: A, B）をテキストとして与える．

## 3.2　関係照会タスク

関係照会タスクでは，ある時点の知識グラフを入力として与え，質問に対して該当する人物名の集合を正しく出力できるかを評価する．入力は，人物一覧，関係一覧，サブグループの所属一覧，質問文であり，出力は該当人物名とする．

表 1 の上段 3 カテゴリに質問文の分類を示す．c1 は，指定された関係に基づく 1-hop 照会であり，主語として与えられた人物から，明示された関係で到達する隣接人物集合を返す．c2 は，2-hop 照会であり，質問文に明示された順序に従って 2 つの関係を段階的に推論する．具体的には，「{人名}の{関係 1}の{関係 2}は誰？」という形式の質問文に対し，人名の関係 1 について 1-hop で人物集合を得る．そして，集合内の各人物に対して関係 2 を推論し，得られる人物集合を回答とする．c3 は，属性条件付きの 1-hop 照会であり，人名を明記せず，「サブグループ＋関係語」で明示する．具体的には，「私の{サブグループ}の{関係 1}の{関係 2}は誰？」という形式の質問文を用いる．まず，知識グラフ上でサブグループに所属し，かつ関係 1 を満たす人物を同定する．その後，同定された人物に対する関係 2 を回答とする．なお，正解集合が過度に大きくなることを防ぐため，同定結果が一意となる人物のみ採用する．

評価は，集合ベースで実施し，予測集合と正解集合の間の各 F1-score を算出し，知識グラフと質問を LLM に入力した際の平均性能を求める．また，完全一致では，厳密に集合が一致した割合を評価する．また，出力形式が崩壊した事例については，出力不

可として，その数をカウントする．

### 3.3 関係更新タスク

関係更新タスクでは，会話内容に基づいて人間関係が変化したかを判定し，更新対象となる人物ペアと更新後の関係を特定できるか評価する．入力は，人物一覧，関係一覧，所属一覧，会話文（3 発話）である．会話文には，「日本語日常対話コーパス」[10] の Dailylife トピックからランダムに抽出した連続する 2 発話の後に関係変化または関係の言及を含む 1 発話を付与する．出力は，関係の更新が不要な場合には空配列 `[]` とし，必要な場合は `[人物 1, 人物 2, 関係]` の形式で，更新対象人物ペアと新しい関係を順不同で返す．発話者はすべて「私」とし，判定対象となる関係は，関係一覧に含まれる関係，または関係削除を表す `NONE` に限定する．本タスクで用いる関係語彙は {恋人，配偶者，NONE} とする．

更新操作は，表 1 の置換・削除・維持の 3 カテゴリである．置換は，人物ペア間の関係が別の関係に置換される場合（例：友達/知人/同僚 → 恋人，恋人 → 配偶者）を指す．削除は，既存の関係が取り消され，`NONE` が正解となる場合（例：別れた，離婚した）である．維持は，関係の変化がなく，空配列が正解となる場合（例：よく話していた，一緒にいた）である．さらに，誤り要因を切り分けて分析するため，評価データを 2 軸で構成する．1 つ目は，関係変化の主語がユーザ自身であるか（Self），第三者同士であるか（Third）である．2 つ目は，人名を明示した言及（Direct）か，「サークルの同僚」のように一方の人物が間接的に省略された言及（Indirect）かである．Indirect は「関係＋所属」によって相手が一意に定まる例のみ採用する．

評価データの生成は，更新が生じる場合に変化する人物ペアを 1 つに限定する．評価では，関係更新の有無を二値分類として扱い，F1-score を算出する．また，更新が必要なデータに限定し，更新対象の人物ペアと関係が完全一致した割合を Accuracy として算出する．加えて，JSON 形式の出力が崩壊した事例を出力失敗として集計する．

## 4 LLM による評価実験

### 4.1 実験設定

**評価データ**　関係照会タスクでは，10 個の知識グラフそれぞれから 30 件の評価データを生成し，合計 300 件とする．カテゴリは c1，c2，c3 を各 100 件とし，知識グラフごとに各カテゴリ 10 件で揃える．

関係更新タスクでは，3 つの更新操作（置換・削除・維持）において各 100 件とし，知識グラフごとにそれぞれ 10 件で揃える．各知識グラフから 30 件を生成し，合計 300 件とする．また，300 件の評価データは，誤り要因を詳細に分析するため，2 つの軸で構成されている．1 つ目は，関係変化の主語がユーザ自身である場合（Self）と第三者同士である場合（Third）であり，それぞれ 150 件ずつ含む．2 つ目は，人名を明示した言及（Direct）と一方の人物が間接的に省略された言及（Indirect）であり，Direct を 220 件，Indirect を 80 件ずつ含む．

**利用するモデル**　本稿では，評価タスクの妥当性を確認するため，日本語理解に強い `LLM-jp-13B`[1)] [11] による few-shot（付録 B）を実施する．

**表 2**　関係照会の性能 (LLM-JP)

| カテゴリ | 出力失敗 | Precision | Recall | F1 |
|---|---|---|---|---|
| c1: 1-hop | 1 | **0.298** | **0.523** | **0.349** |
| c2: 2-hop | 9 | 0.100 | 0.312 | 0.133 |
| c3: 属性条件 | 17 | 0.124 | 0.330 | 0.168 |
| Overall | 27 | 0.174 | 0.388 | 0.217 |

### 4.2 評価結果

**関係照会**　表 2 に関係照会タスクの結果を示す．Overall では，平均 F1-score が 0.2217，完全一致は 13 件であり，人物集合を過不足なく列挙することは困難であった．カテゴリ別にみると，1-hop 照会（c1）の平均 F1-score は 0.349 であった．2-hop 照会（c2）は，平均 F1-score が 0.133 と大きく低下した．これは，2-hop 照会が段階的な推論を必要とするため，人物集合の管理が複雑になり，関係のない人物が集合に混入した可能性が考えられる．このことから，関係照会によって得られる人物集合の規模そのものが性能に大きく影響していることが示唆される．属性条件付き 1-hop 照会（c3）は，平均 F1-score が 0.168 であった．属性条件（例：「研究室の友達」「店舗 A の同僚」）を手掛かりとして，知識グラフ上の人物を一意に特定した上で照会に答えることが困難であると考えられる．

以上より，関係照会の難しさは，ホップ数の増加と属性条件付きの人名が明示されない参照において顕著であり，人物集合の制御と人物の同定が要求される条件で性能が低下する．

1) https://huggingface.co/llm-jp/llm-jp-3.1-13b-instruct4

**表 3**　関係更新の全体性能と詳細分析 (LLM-JP)

| カテゴリ | 出力失敗 | F1 | Accuracy |
|---|---|---|---|
| 置換 | 13 | **0.758** | **0.410** |
| 削除 | 11 | 0.675 | 0.360 |
| 維持 | 0 | — | — |
| Overall | 24 | 0.702 | 0.385 |
| Self | 9 | 0.641 | 0.320 |
| Third | 15 | **0.759** | **0.450** |
| Direct | 1 | **0.730** | **0.600** |
| Indirect | 23 | 0.655 | 0.063 |

**関係更新**　表 3 に，関係更新タスクの結果を示す．Overall の F1-score は 0.702 と比較的高く，関係更新の判定は一定程度可能である．一方，Accuracy は 0.385 に留まり，更新を検知できた場合でも，正しい人物ペアと関係を同時に特定することは困難といえる．操作別にみると，置換は F1-score が 0.78，Accuracy が 0.410 と，削除と比較して一定の性能が確認された．また，維持については更新が必要な正例が存在しないが，誤更新のケース（FP）は 7 件であった．詳細分析では，人名が明示される `Direct` の Accuracy が 0.600 であるのに対し，属性条件を含む `Indirect` では 0.063 と低い．これは，関係更新の難しさが，更新操作の分類よりも，属性条件を踏まえて知識グラフ上の人物を一意に同定できるかに依存していることを示す．また，`Third` は `Self` より F1-score と Accuracy が高く，主語が明記された第三者同士の方が関係変化の検知・人物同定の性能が高いことが示された．

## 4.3　誤り分析と考察

**関係照会**　誤りの主な要因としては，正解集合の一部のみを出力する欠落や，関係外の人物を混入させる過剰生成が確認された．c1 では複数名からなる人物集合の出力において欠落が生じた．c2 では，2-hop の関係そのものはある程度辿れているものの，1-hop 目で過度に広い人物集合を取り，その結果として，2-hop 目で誤った人物が残る事例が多く見られた．c3 では，サブグループで一意に定まる中間人物の同定が不安定であり，特に属性や関係語の対応づけが曖昧になると，誤った中間人物に切り替わるか，関係語をそのまま人物名として取り扱うような出力が生じた．これらの結果は，LLM が関係語の意味的理解は可能でも，人物集合の管理や人物同定を伴う推論が弱いことを示唆している．

以上より，関係照会タスクの難しさは，関係理解そのものよりも，人物集合の正確な列挙や，中間人物の正確な同定と切り替え，さらに 2-hop における推論の多段階化に起因すると考えられる．

**関係更新**　関係更新タスクでは，更新検出の全体の F1-score が 0.702 であり，更新の有無を判断する能力は一定程度確認された．一方で，更新対象の人物ペアと関係が完全一致した割合（Accuracy）は 0.385 に留まり，更新イベントは検知できても，誰と誰がどの関係に変化したかという対象人物の同定が主要な誤りであることが示された．

以上より，関係更新タスクは，関係照会タスクと同様に，イベントの有無や種類の判断よりも，更新対象となる人物の正確な同定が性能に大きく影響することが確認された．

**知識グラフの入力形式の影響**　本稿では，知識グラフを人物一覧，関係一覧，所属一覧としてテキスト化し，LLM に入力した．しかし，入力が長くなるほど参照すべきエッジの見落としが増加し，特に同一人物ペアに複数関係がある場合，どの関係を優先的に参照すべきかが曖昧となり，判定性能が低下する．今後は，質問の主語ごとに隣接リストを集約する形式（例：`A: {友達:[...], 先輩:[...]}`），JSON などの構造化，あるいは知識グラフ全体を一括入力せず，候補となる近傍のノードのみを抽出して与える方式などを比較し，入力形式が誤りに与える影響を検証する必要がある．

# 5　おわりに

本稿では，長期対話における人間関係の扱いを，固定知識に対する関係照会と，会話に基づく関係更新に分離して評価する枠組みについて検討した．小規模な人物関係知識グラフと日本語会話から評価データを構成し，日本語 LLM で評価した結果，関係照会タスクでは人物集合の管理と多段推論が不安定であること，関係更新タスクでは更新イベント自体の検出よりも更新対象となる人物ペアの同定が主要な誤り要因であることを明らかにした．特に，推論に必要なホップ数の増加や属性条件付きの人物同定が関係照会・更新の双方において困難であり，人物同定能力の不足が長期対話の一貫性を損なう要因となり得ることが示唆された．

今後の課題として，人物同定の改善手法の適用や，知識グラフの入力形式が与える影響の切り分けなどが挙げられる．これらを通して，人間関係に特化した長期記憶の評価基盤を拡張し，実際の長期対話システムの設計に有効な指針を明らかにする．

# 参考文献


[1] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In Iryna Gurevych and Yusuke Miyao, editors, **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 2204–2213, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[2] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems, 2024.

[3] Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory, 2023.

[4] Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 13851–13870, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[5] Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. Dialogue-based relation extraction. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 4927–4940, Online, July 2020. Association for Computational Linguistics.

[6] Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models, 2024.

[7] Elan Markowitz, Krupa Galiya, Greg Ver Steeg, and Aram Galstyan. Kg-llm-bench: A scalable benchmark for evaluating llm reasoning on textualized knowledge graphs, 2025.

[8] Zike Yuan, Ming Liu, Hui Wang, and Bing Qin. GraCoRe: Benchmarking graph comprehension and complex reasoning in large language models. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, **Proceedings of the 31st International Conference on Computational Linguistics**, pp. 7925–7948, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics.

[9] 内田由紀子, 遠藤由美, 柴内康文. 人間関係のスタイルと幸福感：つきあいの数と質からの検討. 実験社会心理学研究, Vol. 52, No. 1, pp. 63–75, 2012.

[10] 赤間怜奈, 磯部順子, 鈴木潤, 乾健太郎. 日本語日常対話コーパスの構築. 言語処理学会 第 29 回年次大会論文集, pp. 108–113. 言語処理学会, 2023.

[11] LLM-jp, Akiko Aizawa, Eiji Aramaki, et al. Llm-jp: A cross-organizational project for the research and development of fully open japanese llms. **arXiv preprint arXiv:2407.03963**, 2024.

# A　関係語彙リスト

実験で用いた関係語彙を表 4 に示す．対称関係は，人物ペアの両方向が同一関係を持ち，非対称関係は人物ペアで逆関係を持つ．

**表 4**　関係語彙

| | |
|---|---|
| **対称関係** | |
| 友達，知人，同僚，兄弟，いとこ，恋人，配偶者，NONE | |
| **非対称関係（逆関係）** | |
| 先輩 → 後輩 | 後輩 → 先輩 |
| 上司 → 部下 | 部下 → 上司 |
| 親 → 子 | 子 → 親 |
| 祖父母 → 孫 | 孫 → 祖父母 |

# B　評価で用いたプロンプト

### 関係照会タスク（3-shot）

関係照会のプロンプト

以下の人物関係グラフを用いて，「私」から見た関係に関する質問に回答してください．質問が指定する関係のみを辿って回答してください．

【関係照会】
- c1: 「A の関係」は 1-hop のみを辿り，該当する人物集合を出力する．
- c2: 「A の関係 1 の関係 2」は 2-hop のみを辿り，A から関係 1 で到達した人物に対して関係 2 を適用して得られる人物集合を出力する．
- c3: 「A の<所属>の関係 1 の関係 2」は，まず【所属一覧】の<所属>に属し，かつ【関係】において A の関係 1 に該当する人物 X を特定する．その後，人物 X の関係 2 に該当する人物集合を出力する．

【出力制約】
- 出力は 1 行の JSON 配列のみとする（例: [田中; 佐藤]）．
- JSON 配列以外の文字（説明文，ラベル，改行，句読点など）は出力しない．
- 【人物】に含まれる名前以外は出力しない．重複は含めない．

【人物】
人物一覧
【所属一覧】
サブグループ一覧
【関係】
人物関係一覧
【例】
few-shot 例
<入力>
質問: ...
<解答>

### 関係更新タスク（3-shot）

関係更新のプロンプト

会話と人物関係グラフを参照し，context3 で生じた関係更新のみを 1 行の JSON 配列で出力してください．
【会話】
- context1: ユーザの発話
- context2: システムの発話
- context3: ユーザの発話（関係変化がある場合，ここで述べられる）

【更新判定】
- 「付き合った」→ 恋人，「結婚した」→ 配偶者，「別れた」「離婚した」→ NONE
- 関係更新がない場合は [] を出力する
- 明示的なイベント語がない限り，推測で更新しない

【人物同定】
- 直接言及: 「A と B は〜した」場合，A と B を更新対象とする
- 主語省略: 「B と付き合った」のように主語省略がされた場合，話者を「私」とする．
- 属性制約: 人名が省略される場合（例：「研究室の先輩」「部活の友達」「いとこの配偶者」など）
  - 【所属一覧】および【関係】を用いて，省略された人物を一意に特定する
  - 人物が一意に定まらない（0 人または複数人）場合は推測せず [] を出力する

【出力制約】
- 出力は 1 行の JSON 配列のみとする
- 形式: [人物 A; 人物 B; 関係] または []
- 使用可能な関係ラベル: 恋人，配偶者，NONE（NONE は関係削除を表す）
- 説明文や追加トークンは出力しない

【人物】
人物一覧
【所属一覧】
サブグループ一覧
【関係】
人物関係一覧
【例】
few-shot 例
<入力>
context1: ...
context2: ...
context3: ...
<解答>