

対話システムにおけるペルソナの自動生成による更新

川本 稔己^{1,2} 山崎 天² 佐藤 敏紀² 船越 孝太郎¹ 奥村 学¹

¹ 東京工業大学 ² LINE 株式会社

{kawamoto, funakoshi, oku}@lr.pi.titech.ac.jp

{takato.yamazaki, toshinori.sato}@linecorp.com

概要

話者のプロフィールを集めたテキストであるペルソナを考慮した対話システムでは、ペルソナの変化を捉え適切に更新する必要がある。先行研究では、ペルソナを対話履歴から抽出した上で、既存のペルソナと抽出したペルソナを比較し最新のペルソナのみを残す。しかし、過去のペルソナを削除することで情報の欠落を引き起こす。そこで本稿では、既存のペルソナと抽出したペルソナの両方を考慮したペルソナを、大規模汎用言語モデルを用いて生成的に更新する手法を提案する。その結果、正解のペルソナとの一致度と応答生成性能の向上を確認した。

1 はじめに

ペルソナは好みや性格、行動といった話者の情報を集めたテキストであり、ペルソナを考慮することで、より親和的な応答生成が可能になる [1]。本稿ではペルソナを構成する情報の単位（多くの場合、文に相当）をペルソナ要素と呼ぶ。ペルソナを使った対話データセットの1つに PERSONA-CHAT [2] があるが、ペルソナが変化することは考慮されていない。しかし、ペルソナは常に同一ではなく、時間の経過で変化・増加する。Xu ら [3] は、対話履歴から新しいペルソナを抽出する機構を提案し、時間経過を考慮した Multi-Session Chat データセット (以降, MSC) を用いてペルソナ要素を追加することの有効性を示した。しかし、ペルソナ要素を追加するだけでは、含意や言い換えとなるペルソナ要素が溜まっていくだけでなく、ペルソナ要素の変化に対応できない。そこで Bae ら [4] は既存のペルソナと抽出したペルソナを比較し、最新のペルソナのみを残す手法を提案した。しかし、最新のペルソナのみを残し、過去のペルソナを削除することは情報の欠落を引き起こす。本稿では、既存のペルソナと抽出したペルソナの両方を考慮し、大規模汎用言語モデルを用いてペ

```
Old:I am 33 years old.
New:I am 34 years old.
→ I am 34 years old.
===
Old:I am signing up the gym later today.
New:I finished signing up.
→ I signed up for the gym.
===
Old:<mi>
New:<sj>
→
```

図1 一般的な形式のプロンプト (抜粋)

ルソナを生成的に更新する手法を提案する。

大規模汎用言語モデルは、GPT-3 [5] に代表される、大量のコーパスで自己教師あり学習された言語モデルであり、ショットと呼ばれる少数のサンプルを記述したプロンプトを与えることで、さまざまな言語タスクを解けることが知られている。最近のプロンプトの記述方法の1つとして、Chain of Thought (以降, CoT) がある [6]。一般的なプロンプトではショットとして問題と解答のペアを記述し、言語モデルは与えられた問題の解答を生成する。CoT は、解答を得るための推論過程も記述し、言語モデルは推論過程と解答の両方を生成する手法であり、推論を伴う複数の言語処理タスクで一般的なプロンプトを上回る性能が確認されている [7, 8]。

ペルソナの更新は、ペルソナ要素の意味・関係性を考慮した推論を必要とする。そこで、大規模汎用言語モデルのプロンプトに CoT を利用する手法を提案する。加えて、含意・中立・矛盾に分類する自然言語推論 (以後, NLI) の結果に応じてプロンプトを使い分けることで性能の向上を図る。その結果、情報の欠落を緩和し、自動評価において正解のペルソナとの一致度と応答生成性能の向上を確認した。

2 提案手法

複数セッションからなる、ペルソナを考慮した対話は、以下のフローで行われる [9]。

1. 事前設定のペルソナで1セッション対話する。

表 1 (m_i, s_j) に対するペルソナ要素の更新の種類とその例. $m_i \in$ 既存のペルソナ, $s_j \in$ 抽出したペルソナ, z は m_i と s_j ではないペルソナ要素, m'_i, s'_j は m_i, s_j の少なくともどちらかを書き換えたペルソナ要素を表す.

	Outputs	m_i	s_j	\rightarrow	Outputs
APPEND	m_i, s_j	I want to be a doctor.	I have a pet fish.		I want to be a doctor. & I have a pet fish.
PASS	m_i	Cheeseburgers are my favorite food.	I like cheeseburgers.		Cheeseburgers are my favorite food.
REPLACE	s_j	I will go to Italy next month.	I went to Italy.		I went to Italy.
DELETE	\emptyset	Having a cold and taking medicine.	Cold is all better now.		\emptyset
FUSION	z	I live in Toronto.	I live in Canada.		I live in Toronto, Canada.
REWRITE	m'_i, s'_j	I have pink and blue hair.	I have purple hair.		I had pink and blue hair. & I have purple hair.

```

Q:You know that he said, "I am 33 years old." Then you
hear that the new information about him is that "I am
34 years old." As a result, how should you store his
information?
A:Let's think step by step. Considering "I am 33 years
old." and "I am 34 years old.", the two pieces of
information are about his age, and the information
suggests that he has aged one year, from 33 to 34. So,
we don't need information about his previous age.
Thus, by keeping only new information, the answer is
"I am 34 years old."
===
Q:You know that he said, "I am signing up the gym later
today." Then you hear that the new information about
him is that "I finished signing up." As a result, how
should you store his information?
A:Let's think step by step. Considering "I am signing up
the gym later today." and "I finished signing up.",
the two pieces of information are about signing up. As
of the previous information, he was going to sign up
for the gym, but he finished signing up for the gym as
of the new information. So, by describing the latest
information, the answer is "I signed up for the gym."
===
Q:You know that he said, < $m_i$ > Then you hear that the new
information about him is that < $s_j$ > As a result, how
should you store his information?
A:Let's think step by step.

```

図 2 Chain of Thought を用いたプロンプト (抜粋)

- 直前の対話履歴からペルソナを抽出する。
- 抽出したペルソナと既存のペルソナから、保持するペルソナを決定する。
- 保持したペルソナで次のセッションを行う。
- 最終セッションの対話でないなら 2 に戻る。

上のフローにおけるタスクは、ペルソナを考慮した応答生成 (1,4), ペルソナの抽出 (2), ペルソナの更新 (3) の 3 つに分けられ、本稿はペルソナ更新タスクに着目する。既存のペルソナを $M = \{m_1, \dots, m_T\}$, 抽出したペルソナを $S = \{s_1, \dots, s_K\}$ とする。本稿では、後述するペルソナ更新アルゴリズムに従いながら、ペルソナ要素の組 (m_i, s_j) を生成的に更新することで、新たなペルソナ要素の集合 M' を得る。

ペルソナ要素の更新 Bae ら [4] は、ペルソナ要素の更新操作として、APPEND, PASS, REPLACE, DELETE の 4 種類を定義した。この 4 種類は、 m_i と s_j をそれぞれ残すか残さないかの二値の組み合わせ

に対応する。しかしこの方法では、過去のペルソナが削除されてしまい情報の欠落を引き起こす可能性がある。そこで本稿では、既存のペルソナと抽出したペルソナの両方を考慮した m_i と s_j 以外の新しいペルソナ要素を得る操作を新たに 2 つ導入する。1 つ目は、Bae らも言及していたものの未導入であった、 m_i と s_j をまとめた z を保持する FUSION, 2 つ目は、 m_i と s_j の少なくともどちらかを書き換えた m'_i と s'_j を保持する REWRITE である。6 種類の操作の例を表 1 に示す。Bae らの 4 分法では、FUSION の事例が PASS または APPEND に分類されるため情報が欠落した、または冗長な状態で保持される。REWRITE の事例は REPLACE に分類されるため情報の欠落が発生する場合がある。FUSION と REWRITE はそれらの課題を解消している。本稿は、FUSION と REWRITE を初めてペルソナ要素の更新に導入し、新しいペルソナ要素を大規模汎用言語モデルを用いて生成する手法を提案する。

Chain of Thought プロンプト ペルソナ要素を適切に更新するためにはペルソナ要素の意味を理解する必要がある。例えば、 $m_i =$ “マンションに住んでいる。”, $s_j =$ “一軒家に住んでいる。” の場合、マンションから一軒家に引っ越したと考えられるので、更新したペルソナ要素は $m'_i =$ “以前マンションに住んでいた。” と $s'_j =$ “一軒家に住んでいる。” となる。 $s_j =$ “福岡に住んでいる。” の場合は、マンションが福岡にあると考えられるので、更新したペルソナ要素は $z =$ “福岡でマンションに住んでいる。” となる。このようにペルソナ要素の更新には推論を必要とするため、図 1 の形式で記述する一般的なプロンプトではなく、段階的に推論を行い解答を得ることで性能の向上が確認されている CoT を利用し、図 2 の形式でプロンプトを記述する。

NLI を利用したプロンプトの使い分け ペルソナ要素ペア (m_i, s_j) は含意・中立・矛盾のいずれかの論理関係を持ちうるが、論理関係の違いによって、

同じ種類のペルソナ要素の更新を行う場合であっても CoT の推論過程が異なる。異なる推論過程を記述したプロンプトを使用することは生成時のノイズになると考えられるので、 (m_i, s_j) と同じ論理関係のペルソナ要素ペアを利用したショットのみで構成するプロンプトを使用する。 (m_i, s_j) とショットに記述されたペルソナ要素ペアの性質や関係が類似していることで性能の向上が期待される。具体的には、図 2 の形式で記述した各ショットを、ペルソナ要素ペアの NLI [10] の結果に応じて分け 3 種類のプロンプトを用意する。更新を行うときは、 (m_i, s_j) の NLI 結果に応じてプロンプトを使い分け生成を行う。

ペルソナ更新アルゴリズム 最終的に保持するペルソナ M' を得る処理は、ペルソナ更新アルゴリズムによって行う。更新アルゴリズムの詳細は付録 A に示す。Bae らの更新アルゴリズムでは、要素ペア (m_i, s_j) に対して分類器を適用し、その分類結果に基づいてペルソナの更新処理を行う。これに対し我々の更新アルゴリズムでは、元の要素ペアと生成されたペルソナ要素を照合することで、どの更新操作が生成モデルで選択されたのかを特定し、それに基づいて実際のペルソナ更新処理を行う。

3 実験

3.1 実験設定

ペルソナを考慮した応答生成は、MSC で fine-tuning した Transformer [11] ベースのモデル BST2.7B [3] を利用した。ペルソナの抽出には MSC で fine-tuning した BART [12] を利用した。ペルソナ要素を更新するための大規模汎用言語モデルは、主に英語で事前学習された 660 億個のパラメータを持つ公開済みの OPT [13]、日本語と英語で事前学習された 390 億個のパラメータを持つ HyperCLOVA [14] の 2 種類で比較を行う。プロンプトには、図 2 と同様の形式で 5-ショットを記述した。NLI 分類器には、Dialogue NLI [10] で fine-tuning した DeBERTa [15] を利用した。Dialogue NLI のテストデータでの正解率は 91.3%であった。実装は ParlAI [16] を利用し、特に記載のない設定は ParlAI のデフォルトに従う。応答生成モデルに入力するペルソナ要素の順序は、結果に影響を与える可能性があるため、付録 B に示す実験を行った結果、最も Perplexity が低いランダム順で統一する。

ペルソナの更新を評価するために、MSC のテス

トデータで以下の 2 つの実験を行う。

1. MSC の正解のペルソナと手法ごとのペルソナの一致度を ROUGE-1, 2 (F1) [17] で評価。
2. 応答生成の性能を評価するため、手法ごとのペルソナを使い、応答生成モデルの Perplexity をセッションごとに評価。

数値差の評価の補助とするため、Paired Bootstrap Resampling [18] による有意差検定を行う。多重比較補正は行わない。

3.2 比較手法

以下の手法の間で比較を行う。

Accumulation 抽出したペルソナを既存のペルソナに単に追加する手法。

Classification ペルソナ要素ペアを分類してペルソナを更新する手法。Bae らの手法 [4] を再現することを目的としている。データセットがないため 4 クラス分類器を新たに訓練する NLI transfer (fine-tune) ではなく、既存の NLI モデルを用いて含意を PASS, 中立を APPEND, 矛盾を REPLACE に対応づける 3 クラス分類手法 (NLI zero-shot) を再現する。ペルソナ要素ペアが DELETE に分類されることはないが、Bae らによれば DELETE の割合は 5% 程度であり、最大で 95% の精度を持ち得ると考えられるため、十分比較手法となりうる。

O Generation OPT を用いた提案手法。

H Generation HyperCLOVA を用いた提案手法。

提案手法の切除実験として、図 1 の形式のプロンプトを使った **w/o CoT** と、NLI を行わず 1 種類のプロンプトを使用する **w/o NLI** の実験を行う。また、応答生成については、性能の上限を確かめるため、正解のペルソナを利用した **Gold Persona** とも比較する。

3.3 実験結果

正解のペルソナとの一致度を測った実験結果を表 2 に示す。H Generation は、Accumulation よりも ROUGE-1, 2 ともに高く正解のペルソナとの一致度が向上した。それ以外の手法は、Accumulation と比較して、ROUGE-1, 2 の少なくともどちらかが低い値となり、一致度の向上は確認できなかった。平均ペルソナ要素数に関しては、更新を行う手法は Accumulation と比較して全て減少しており、含意や言い換えとなるペルソナ要素をまとめたり削除して

A: Let's think step by step. Considering "I'm now looking for a new job." and "I got a job.", the two pieces of information are about his current job. As of the previous information, he was looking for a new job, but he got a job as of the new information. So, by describing the latest information, the answer is "I got a job."

A: Let's think step by step. Considering "I am an artist." and "I am a vet.", the two pieces of information are about occupation. As of the previous information, he was an artist, but he is a vet now. It is assumed that he changed his occupation. Therefore, by rewriting the previous information in the past tense and combining it with the new information, the answer is "I was an artist, but now I am a vet."

図3 提案手法によりペルソナ要素の更新を行ったペルソナの生成例。"Let's think step by step."以降を生成している。

表2 正解のペルソナとの一致度を測った実験結果と平均ペルソナ要素数。†はAccumulationと比較して検定で有意($p < 0.05$)であることを示す。

	ROUGE-1	ROUGE-2	ペルソナ要素数
Accumulation	61.58	38.99	34.91
Classification	55.38	34.08	15.43
O Generation	62.79 [†]	38.65	21.44
w/o CoT	62.06	37.74	18.93
w/o NLI	61.92	37.21	20.43
H Generation	63.52[†]	39.81[†]	28.37
w/o CoT	61.90	37.92	25.64
w/o NLI	61.60	37.59	26.41
Gold Persona	-	-	19.26

表3 各手法のペルソナを使った応答生成の実験結果。各セッションごとのPerplexityで評価。

Session	2	3	4	5
Accumulation	8.830	9.010	9.123	9.202
Classification	8.952	9.148	9.257	9.321
O Generation	8.914	9.058	9.145	9.191 [†]
w/o CoT	8.917	9.054	9.149	9.214
w/o NLI	9.075	9.151	9.248	9.317
H Generation	8.947	8.990[†]	9.088[†]	9.153[†]
w/o CoT	8.956	9.009	9.101	9.161
w/o NLI	8.998	9.065	9.120	9.186
Gold Persona	8.837	8.967	9.058	9.141

いると考えられる。しかし、Classificationは、Gold Personaよりもペルソナ要素数が少なく、過去のペルソナを削除した影響でペルソナ要素が減りすぎている可能性がある。

次に、応答生成の実験結果を表3に示す。全てのセッションにおいて、ClassificationはAccumulationより高いPerplexityとなり、ペルソナ要素ペアを分類して更新する手法の効果は確認できなかった。提案手法ではH Generationが最も低いPerplexityであり、Accumulationと比較すると、セッション2以外ではH Generationのほうが低かった。Gold Personaともセッション5でのPerplexityの差が0.012であり、対話が長期なら提案手法が有効であることを確

認した。OPTを使った手法は、HyperCLOVAを使った手法と比較して性能が低下した。モデルサイズはHyperCLOVAよりOPTのほうが大きく、OPTは英語に特化しているが、事前学習に使われたデータの質や量、ハイパーパラメータ等が性能に影響を与えていると考えられる。切除実験の結果、提案手法はw/o CoT, w/o NLIよりも優れた結果となり、CoTと、NLIでプロンプトを使い分ける手法の効果を確認した。

3.4 ペルソナ要素の更新例

H Generationでペルソナ要素の更新を行った生成例を図3に示す。違和感のない推論過程が生成でき、入力に与えたペルソナ要素をそのまま出力するだけではなく、FUSIONのようにペルソナ要素をまとめ、過去のペルソナ要素の情報も保持している例も確認した。しかし、"I am an artist."と"I am a vet."を比較した場合に、言語モデルの出力ではアーティストから獣医に転職したと推論しているが、アーティストと獣医を兼ねている可能性もあり、ペルソナ要素ペアだけではどのように更新すべきか判断することが難しい例もあった。

4 おわりに

本稿では、ペルソナを考慮した対話システムにおいて、FUSIONとREWRITEの2つの更新手法を新たに導入し、過去の情報も考慮したペルソナ要素を生成してペルソナを更新する手法を提案した。情報の欠落の緩和を目的とした提案手法により先行研究と比較して正解のペルソナとの一致度、応答生成の性能の両方を向上することを確認した。情報の欠落が防がれていることは事例分析からも確認された。今後人手評価を行い、応答生成がどのような観点で改善されているか確かめたい。また、ペルソナの更新性能は、ペルソナの抽出や応答生成といった他の機構の性能に依存するため、これらの機構も含めて総合的に向上させていきたいと考えている。

参考文献

- [1] Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. A survey of available corpora for building data-driven dialogue systems. **arXiv preprint arXiv:1512.05742**, 2015.
- [2] 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 **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.
- [3] Jing Xu, Arthur Szlam, and Jason Weston. Beyond goldfish memory: Long-term open-domain conversation. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 5180–5197, Dublin, Ireland, May 2022. Association for Computational Linguistics.
- [4] Bae Sanghwan, Kwak Donghyun, Kang Soyoung, Young Lee Min, Kim Sungdong, Jeong Yuin, Kim Hyeri, Lee Sang-Woo, Park Woomyoung, and Sung Nako. Keep me updated! memory management in long-term conversations. In **Findings of the Association for Computational Linguistics: EMNLP 2022**, p. 3769–3787, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics.
- [5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Nee-lakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. **Advances in neural information processing systems**, Vol. 33, pp. 1877–1901, 2020.
- [6] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In **Advances in Neural Information Processing Systems**, Vol. 35. Curran Associates, Inc., 2022.
- [7] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. **arXiv preprint arXiv:2210.09261**, 2022.
- [8] Jason Wei, Yi Tay, and Quoc V Le. Inverse scaling can become u-shaped. **arXiv preprint arXiv:2211.02011**, 2022.
- [9] 川本稔己, 山崎天, 佐藤敏紀, 奥村学. 大規模汎用言語モデルによるペルソナを考慮した応答生成. 言語処理学会 第 28 回年次大会, 2022.
- [10] Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. Dialogue natural language inference. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 3731–3741, Florence, Italy, July 2019. Association for Computational Linguistics.
- [11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 30. Curran Associates, Inc., 2017.
- [12] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics.
- [13] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. **arXiv preprint arXiv:2205.01068**, 2022.
- [14] Boseop Kim, HyoungSeok Kim, Sang-Woo Lee, Gichang Lee, Donghyun Kwak, Jeon Dong Hyeon, Sunghyun Park, Sungju Kim, Seonhoon Kim, Dongpil Seo, Heungsub Lee, and et al. What changes can large-scale language models bring? intensive study on HyperCLOVA: Billions-scale Korean generative pretrained transformers. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 3405–3424, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.
- [15] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In **International Conference on Learning Representations**, 2020.
- [16] Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. ParlAI: A dialog research software platform. In **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 79–84, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
- [17] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In **Text Summarization Branches Out**, pp. 74–81, Barcelona, Spain, 2004. Association for Computational Linguistics.
- [18] Philipp Koehn. Statistical significance tests for machine translation evaluation. In **Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing**, pp. 388–395, Barcelona, Spain, July 2004. Association for Computational Linguistics.
- [19] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 6894–6910, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

A ペルソナの更新アルゴリズム

提案手法のペルソナの更新アルゴリズムを図 4 に示す。

Input:
 $M = \{m_1, m_2, \dots, m_T\}$
 $S = \{s_1, s_2, \dots, s_K\}$
 $NLI(m, s) \rightarrow \{\text{“含意”}, \text{“中立”}, \text{“矛盾”}\}$
 $p \rightarrow \{p_{ent}, p_{neu}, p_{con}\}$

Output: $M' = \{m'_1, m'_2, \dots, m'_{|M'|}\}$

```

1: function UPDATE_PERSONA(X, Y)
2:   if Y = ∅ then
3:     return X
4:   end if
5:   for all y_j ∈ Y do
6:     X' ← ∅
7:     U ← ∅
8:     f_del ← false
9:     for all x_i ∈ X do
10:      if NLI(x_i, y_j) = “含意” then
11:        p ← p_ent
12:      else if NLI(x_i, y_j) = “中立” then
13:        p ← p_neu
14:      else if NLI(x_i, y_j) = “矛盾” then
15:        p ← p_con
16:      end if
17:      if G(x_i, y_j, p) = ∅ then
18:        f_del ← true
19:      else if G(x_i, y_j, p) = x_i then
20:        f_del ← true
21:        U ← U ∪ x_i
22:      else if the number of G(x_i, y_j, p) is 1 &
          G(x_i, y_j, p) ≠ y_j then
23:        f_del ← true
24:        U ← U ∪ G(x_i, y_j, p)
25:      else
26:        X' ← X' ∪ G(x_i, y_j, p)
27:      end if
28:    end for
29:    if f_del = true then
30:      X' ← X' - y_j
31:      X' ← UPDATE_PERSONA(X', U)
32:    end if
33:    X ← X'
34:  end for
35:  return X'
36: end function
37: M' ← UPDATE_PERSONA(M, S)
38: return M'

```

図 4 提案手法のペルソナの更新アルゴリズム。 $G(x_i, y_j, p)$ は大規模言語モデルでプロンプト p を利用し、 x_i と y_j のペルソナ要素ペアでペルソナ要素の更新を行うことを表す。 $\{p_{ent}, p_{neu}, p_{con}\}$ はそれぞれ含意・中立・矛盾に対応したプロンプトである。

B ペルソナ要素の順序による影響調査

ペルソナを考慮した対話の実験を行うにあたって、応答生成モデルに与えるペルソナ要素の順序が対話性能に影響があるのか不明であるため調査を行う。ペルソナ要素の順序は古い順 (Old), 新しい

表 4 応答生成モデルに与えるペルソナ要素の順序を変化させた実験結果。† は Sort Old と比較して検定で有意 ($p < 0.05$) であることを示す。

	Session 2-5
Sort Old	8.927
Sort New	8.914
Sort Related	8.916
Sort Random	8.904†

順 (New), 直近の発話との関連度順 (Related), ランダム (Random) の 4 種類で比較する。直近の発話との関連度順とは、ユーザの直前の発話とペルソナ要素の各埋め込み表現のコサイン類似度が高い順とした。埋め込み表現を得るためには SimCSE [19]¹⁾ を使い、[CLS] トークンの埋め込み表現を利用した。それ以外の実験設定は、3.1 節の設定に従い、応答生成性能をセッション 2 からセッション 5 の平均の Perplexity で評価する。

結果を表 4 に示す。ペルソナ要素の順序により、Perplexity が変化することから、ペルソナ要素を入力に与える順序は応答生成に影響を与えることが判明した。その上で、ペルソナ要素をランダムに並べた Sort Random の Perplexity が最も低いことを確認した。

1) <https://huggingface.co/princeton-nlp/sup-simcse-roberta-large>