小規模データにおける文脈情報復元に基づいたマルチターン対話システム

賴 展韜
株式会社バンダイナムコ研究所
z-lai@bandainamco-mirai.com

髙橋 誠史
株式会社バンダイナムコ研究所
m3-takahashi@bandainamco-mirai.com

1 はじめに

シングルターン対話システムが、直前のユーザの発話文だけから適切な応答を生成するシステムであるのに対し、マルチターン対話システムとは、それに加え対話履歴の文脈情報も考慮に入れ適切な応答を生成するシステムである．近年では、シングルターン対話システムにおいて、発話文を基に End-to-End に応答文を生成する Seq2Seq[1] モデルなどが提案されている [2][3]．大量の人間による発話・応答ペアの学習データを用いれば、発話文に対して比較的に高品質な応答文を出力することができる．しかし、シングルターン対話システムがめざましい進歩を遂げている一方、マルチターン対話においてはまだ人間が満足のいくものは少ない．その大きな課題には、日常会話、特に日本語や中国語のようなプロドロップ言語で頻繁に発生する共参照や情報の省略である．Su ら [4] の研究では、2000 人による中国語のマルチターン対話に対し調査を行ったところ、70%以上の対話には、共参照や情報の省略が存在していた．表 1 は、マルチターン対話における 2 つの典型的な例を示している．Context 1 の発話 3 の「あれ」は「昨日の番組」の共参照関係にあり、Context 2 の発話 3 の「なぜ」には「映画が好き」という情報が省略されている．このように、文脈に隠された共参照や省略された情報を与えない限り、正確な応答文を提示することが難しい．

この問題を解決するため、Su ら [4] は発話文を書き換えることでマルチターン対話をシングルターン対話に単純化することを提案している．発話文の書き換えによって、共参照とされた内容と省略された内容をすべて復元するための情報補完を行う．表 1 の実施例では、発話 3’ は発話 3 を書き換え後の発話文である．対話システムは、文脈である発話 1 と発話 2 を考慮せずに、発話 3’ の情報のみで応答文を出力する．このような単純化を行うことにより、対話に必要な情報を維持しつつ、入力情報を圧縮することでマルチターン対話システムの実現の難しさを緩和できる．また、複数行の文脈情報は単一の発話に反映されるため、他の手法よりメモリ効率が高い．

表 1 マルチターン対話の一例．発話 3’ は発話 3 を書き換え後の発話．太字は共参照や省略された部分を表す．

Context 1: 共参照	
発話 1	Human: **昨日の番組**見た？
発話 2	ChatBot: 見た.
発話 3	Human: **あれ**面白かったね.
発話 3’	Human: **昨日の番組**面白かったね.
Context 2：情報省略	
発話 1	Human: **映画**とドラマどっち**が好き**？
発話 2	ChatBot: **映画**.
発話 3	Human: なぜ？
発話 3’	Human: なぜ**映画が好き**？

発話文書き換えの手法について、これまでの研究では、Pointer Network[5] や Copy Mechanism[6] を用いて、復元された発話文をゼロから生成するものがほとんどである．これら生成型手法には二つの課題が存在している [7]．一つ目は、データ準備にかかるコストの高さである．出力文は訓練データに対する依存度が高いため、高品質な発話文を生成するために数十万件以上の大規模データセットが必要である．さらに、書き換え前後の発話文のようなペアデータに人間によるアノテーションが必要となる．二つ目は、推論速度の遅さである．生成型手法はアルゴリズムの性質上、単語の出力は一つ前の出力結果に依存するため、推論速度が遅い．

本研究では、生成型手法の課題を解決するため、発話文復元のタスクを、文脈情報に基づいた発話文に対する編集操作に単純化することを提案する．復元された後の発話文は、文脈や復元前の発話文の単語のみから再構成することで解決する可能性が高く、復元後の発話文と復元前は文法的に近い構造を持っている．例

えば、表 1 の Context 1 では、発話 3 の「あれ」を発話 1 から抽出した参照情報「昨日の番組」に置換することで発話文 3' が得られる．同様に、Context 2 では、発話 3 の「なぜ」の後に発話 1 から抽出した省略情報「映画」、「が好き」を挿入することで発話文 3' が得られる．

また、本研究では、データに対する学習効率を向上するため、共参照解析と省略補完を独立した 2 つのサブタスクに分離した複数タスク学習手法を提案する．結果として、自動評価スコアと推論速度において既存手法を上回った．特に数千件以下の小規模データに対して、データ数減少による性能の低下を抑えることができた．

2 関連研究

不完全な発話文に対する書き換えはさまざま自然言語処理タスクで広く採用されている．機械翻訳では、Seq2Seq モデルから出力された文章の品質向上に利用されている [8]．テキスト要約では、See ら [9] が抽出された要約文を書き換えることでより正確な要約文を生成する手法を提案している．対話生成では、Pan ら [10] は、まず文脈から BERT[11] を介して単語を選択し、これらの単語を組み合わせて書き換えられた発話を生成するカスケードモデルを提示している．Liu ら [12] はセマンティックセグメンテーションの手法を用いて単語レベルでの編集行列を予測する手法を提案している．

しかし、以上の手法はいずれも共参照解析と省略補完などのタスクを同時にモデルを学習させた、いわゆる複数タスク学習のアプローチである．Lee ら [13] はこの複数タスク学習の小規模データの場合においての性能低下への懸念を指摘している．これに対し、Houlsby ら [14] は Adapter を用いて、タスクごとに Fine-tuning を行う際に、ごく一部のパラメータのみ学習する手法を提案している．Pfeiffer ら [15] は Adapter 構造を改善した上、さらに各タスクで学習した情報を共有する AdapterFusion 構造を提案し、Houlsby らに対する優位性を示している．

また当然ながら、本研究は共参照解析と関連がある．Joshi ら [16] は SpanBERT を用いて共参照解析におけるスパン表現を強化している．Wu ら [17] は共参照解析をクエリベースのスパン予測として定式化をし、SpanBERT を用いて機械読解タスクとして解決を行っている．

3 提案手法

本研究の提案手法のモデル構造を図 1 に示す．左側はモデルが表 1 の Context 1 を入力情報とし、共参照解析タスクにおいて、発話文の省略情報位置および置換位置のタグを抽出する過程を示す．右側は使用する Transformer[18] モデルの内部構造の中で、複数タスク学習に関わる部分を表す．

3.1 入力表現

入力情報に対して文区切りと形態素分割をした後、入力情報の先頭に [CLS]、各文章の末尾に [SEP] トークンを挿入する (Word Embeddings)．また、モデルに入力する際、文脈部分と発話文部分を区別するための特殊トークンを付与する (Dialog State Embeddings)．さらに各トークンの位置に応じた埋め込みを付与する．(Positional Embeddings)．

3.2 出力層

出力層では選択モデルに基づき最終層のベクトルを全結合層に入力する．入力情報の全トークンに対してそれぞれの位置タグスコアを計算し、各位置タグはスコア最大となるトークンの位置を採用する．

$$h = \text{Transformer}(x) \in \mathbb{R}^{r_h * |x|} \tag{1}$$

$$l_i = \text{softmax}(W_i \cdot h + b_i) \tag{2}$$

ここで、h はモデルの隠れ状態、$|x|$ は文脈と発話文をつないだ入力情報の長さ、r_h は隠れ状態の次元数を表す．1 訓練データに対する損失関数 L は各位置タグのクロスエントロピーの平均とする:

$$L = -\frac{1}{k}\sum_{i=1}^{k}\sum log l_i \mathbb{I}(T_i) \tag{3}$$

ここで k は抽出するタグ T_i の数、$\mathbb{I}(T_i)$ は正解データの復元された発話文における位置タグ (GroundTruth) の one-hot 表現を表す．

最後に、位置タグ $T_i = \text{argmax}(l_i)$ として計算する．共参照解析 ($k = 4$) において、T_1 と T_2 はそれぞれ先行詞の開始位置、終了位置、T_3 と T_4 はそれぞれ照応詞の開始位置、終了位置を表す．$[w_{T_3},...,w_{T_4}]$ のトークンが $[w_{T_1},...,w_{T_2}]$ のトークンに置換される．省略補完 ($k = 3$) において、T_1 と T_2 はそれぞれ省略情報の開始位置、終了位置、T_3 は省略情報の挿入位置を表し、

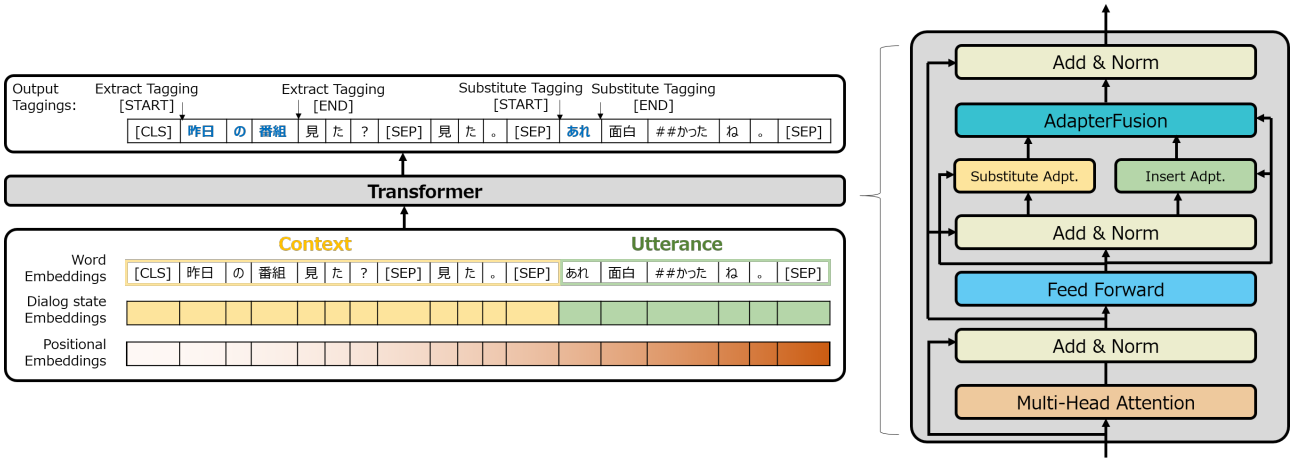

図 1　提案手法のモデル構造

w_{T_3} のトークンの後に $[w_{T_1},...,w_{T_2}]$ のトークンを挿入する．T_i が [CLS] トークンを指す場合や $T_{i-1} \geq T_i$ の場合、書き換えは行わない．なお、本手法の書き換えの対象は、1 つの発話文につき、共参照と情報省略に対しそれぞれ 1 件以内に限定する．

3.3　複数タスク学習

本研究では共参照解析と省略情報補完、この二つのタスクに対し二段階の学習手法を提案する．第一段階では、Adapter を用いて各タスクの情報をカプセル化したタスク固有のパラメータを学習する．ここで Adapter は Pfeiffer らが提案した構造 [15] を用いる．共参照解析を学習した Adapter と省略情報補完を学習した Adapter は図 1 の右側でそれぞれ「Substitute Adpt.」と「Insert Adpt.」で表す．第二段階では AdapterFusion[15] でこれらの学習した表現を結合させる．第一段階の知識抽出と第二段階の知識結合を分離することで、第一段階においてデータの利用効率を向上させるとともに、複数タスクの同時学習で発生しやすい致命的な忘却やタスク間のアンバランスを回避しつつ、両方のタスクで学習した知識を効果的に共有することが可能である．

4　実験

4.1　データセット

本実験では Restoration-200K[10] データセットを用いた．Restoration-200K は SNS 上で収集した 20 万件の中国語のマルチターン対話データセットである．各会話は 6 つの発話と 1 つのラベルで構成されている．最初の 4 つの発話は対話の文脈情報、5 番目の発話は現在の発話文である．ラベルは、5 番目の発話文において共参照もしくは情報省略が発生したかどうかを示すラベルである．ラベルが 1 の場合、6 つ目の発話文が欠落した情報が復元された発話文となるが、ラベルが 0 の場合、6 つ目の発話文は 5 番目の発話文と同じものとなる．なお、これらの対話データから 2,000 件をランダムにサンプリングしたところ、共参照が発生したのが 33.5%、省略が発生したのが 52.4%、共参照も省略も発生しない発話文は 29.7%であった．

さらに、上記割合に沿って改めて抽出した 2,000 件の対話データに対し人間の翻訳者が和訳し (1 名の作業者の約 1 週間の作業量として想定)、一部中国語独自の表現を日本語に適した表現に差し替えた対話データを「Restoration-2K-ja」で表す．中国語データセットと日本語データセットそれぞれ 10%をバリデーション、10%をテストに用い、その他を訓練データに用いた．

4.2　ベースライン

ベースラインを Su ら [4] の提案手法の中で、最も性能が良い「T-Ptr-λ」とし、BeamSearch の BeamSize は 4 を用いた．だたし元論文で「T-Ptr-λ」は PyTorch での著者実装が公開されていなかったため、公平性を考慮して推論速度の比較では PyTorch 実装が公開されており、かつ元論文の比較手法の中で最もシンプルな「L-Gen」を基準とした．なお、本研究の提案手法の中から、共参照と省略補完を単一の Adapter で学習したモデルと、図 1 のような複数の Adapter で学習したモデルで同時に実験を行った．結果をそれぞれ「Single Adpt.」、「Multi Adpt.」で表す．

4.3　訓練設定

事前学習済みモデルには、中国語と日本語ではそれぞれ Joint Laboratory of HIT and iFLYTEK

表 2　各モデル ROUGE-N スコア、コサイン類似度スコアおよび推論速度の比較結果

Dataset	Model	ROUGE			Cosine Similarity		Speedup
		ROUGE-1	ROUGE-2	ROUGE-L	spaCy	Sentence-BERT	
Restoration-200K	Baseline	86.19	75.54	83.18	0.871	0.923	1.00 ×
	Single Adpt.	87.11	75.70	83.66	0.873	0.924	14.9 ×
	Ours (Multi Adpt.)	**88.34**	**76.31**	**87.35**	**0.875**	**0.930**	**14.8** ×
Restoration-2K-ja	Baseline	3.13	2.23	2.91	0.337	0.311	1.00 ×
	Single Adpt.	68.48	60.61	68.31	0.701	0.715	14.7 ×
	Ours (Multi Adpt.)	**70.97**	**60.78**	**70.02**	**0.729**	**0.737**	**14.7** ×

Research が公開した BERT-wwm-Chinese、東北大学が公開した BERT-base-mecab-ipadic-bpe-32k whole-word-mask を使用した．入力系列長の上限を 128 とし、上限を超える文章は学習・評価から除外した．バッチサイズを 32 に設定した．訓練と推論には NVIDIA Tesla V100 を 1 枚用い、実装は PyTorch で行った．

4.4　評価指標

自動評価の指標として、ベースライン [4] に倣い、ROUGE-1、ROUGE-2、ROUGE-L を用いる．また、単語の一致率だけでなく、復元前後の発話文の意味的近似度を比較する観点から、spaCy および Sentence-BERT[19] で計算する復元前後の文章ベクトルのコサイン類似度を用いた．spaCy の事前学習済みベクトルには、中国語と日本語でそれぞれ「zh-core-web-lg」、「ja-core-news-lg」を使用した．推論速度の比較においては 10 件推論した際の平均値を指標とした．

4.5　実験結果

自動評価の指標を用いて評価した実験の結果を表 2 に示す．Restoration-200K において、ベースラインと比べ、ROUGE スコアでは提案手法が最も高い結果となった．文章ベクトルのコサイン類似度においても、spaCy と Sentence-BERT で計算したスコアの上昇を確認した．また、複数 Adapter を用いた手法と単一 Adapter と比較した場合において、複数 Adapter を用いた手法の方が性能が向上した．

推論速度の比較において、ベースラインの速度を 1.00 とした場合、提案手法が 14 倍以上速く、ベースラインを大きく上回った．

一方、Restoration-2K-ja において、ベースラインの手法では GroundTruth と近い構造を持つ発話文を出力することができなかったが、提案手法ではベースラインを大きく上回るスコアとなった．

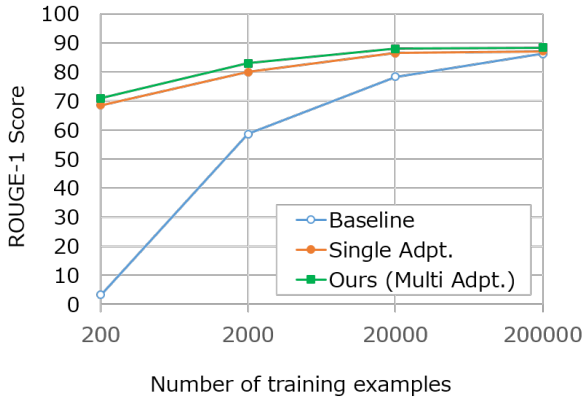

図 2　訓練データサイズごとの ROUGE-1 スコア

データ数による性能への影響を検証するため、Restoration-200K からランダムで抽出した 200 件, 2,000 件と 20,000 件のサブセットで計算した ROUGE-1 スコアを図 2 で示す．特に 200 件や 2,000 件など小規模データにおいて、本研究の提案手法のスコアが既存手法より大幅に上昇し、200,000 件のデータを全部使用したモデルのスコアと比較しても、減少量が最大 20 ポイントに抑えることができた．この結果から、本研究の提案手法は数千件以下の小規模データに対しても、共参照や情報補完タスクの学習にフォーカスすることで、文脈情報を発話文に反映させることができた．

5　おわりに

本研究では、小規模データにおける文脈情報復元に基づいたマルチターン対話システムの改善に取り組んだ．提案手法では、マルチターン対話システムの文脈理解タスクを、編集操作によって共参照と省略情報を発話文に復元するタスクに単純化した．また、共参照と省略補完の学習を複数の Adapter で分離することにより、小規模データにおいて既存手法を大きく上回った．

参考文献

[1] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, Vol. 27, pp. 3104–3112. Curran Associates, Inc., 2014.

[2] Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. Towards a human-like open-domain chatbot, 2020.

[3] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M. Smith, Y-Lan Boureau, and Jason Weston. Recipes for building an open-domain chatbot, 2020.

[4] Hui Su, Xiaoyu Shen, Rongzhi Zhang, Fei Sun, Pengwei Hu, Cheng Niu, and Jie Zhou. Improving multi-turn dialogue modelling with utterance rewriter. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 22–31, Florence, Italy, July 2019. Association for Computational Linguistics.

[5] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Vol. 28, pp. 2692–2700. Curran Associates, Inc., 2015.

[6] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1631–1640, Berlin, Germany, August 2016. Association for Computational Linguistics.

[7] Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. Encode, tag, realize: High-precision text editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5054–5065, Hong Kong, China, November 2019. Association for Computational Linguistics.

[8] Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alex Waibel. Pre-translation for neural machine translation, 2016.

[9] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[10] Zhufeng Pan, Kun Bai, Yan Wang, Lianqiang Zhou, and Xiaojiang Liu. Improving open-domain dialogue systems via multi-turn incomplete utterance restoration. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1824–1833, Hong Kong, China, November 2019. Association for Computational Linguistics.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[12] Qian Liu, Bei Chen, Jian-Guang Lou, Bin Zhou, and Dongmei Zhang. Incomplete utterance rewriting as semantic segmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2846–2857, Online, November 2020. Association for Computational Linguistics.

[13] Jason Lee, Kyunghyun Cho, and Thomas Hofmann. Fully character-level neural machine translation without explicit segmentation, 2017.

[14] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp, 2019.

[15] Jonas Pfeiffer, Aishwarya Kamath, Andreas Ruckle, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning, 2020.

[16] Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5803–5808, Hong Kong, China, November 2019. Association for Computational Linguistics.

[17] Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6953–6963, Online, July 2020. Association for Computational Linguistics.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Vol. 30, pp. 5998–6008. Curran Associates, Inc., 2017.

[19] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.