マルチターン対話に対する逐次的 Transformer の検討

丸田要
都城工業高等専門学校
marutak@cc.miyakonojo-nct.ac.jp

概要

雑談を円滑に行うには，単に直前のユーザによる発話である質問に対して返答するだけでなく，これまでの雑談内容を反映した返答が必要なケースが十分考えられる．しかし，非タスク指向型対話システムの現状は，ユーザの会話に対して不自然な応答を行うことが多く対話が破綻する問題がある．そこで，単純に直前のユーザ発話 1 文に対する応答文を学習するのではなく，これまでの対話内容であるマルチターン対話ログの複数文に対して応答文の生成を学習することで，円滑な雑談対話を継続できる応答文生成ができると考えた．そのため，本論文では自然な対話の継続という目標を達成するための第一歩として，逐次的に構築した Transformer の対話モデルを提案し検討する．

1 はじめに

近年，深層学習による人工知能システムが注目を浴びている．その中には対話システムも存在している．その対話システムはタスク指向型対話システムと非タスク指向型対話システムに分けることができる．タスク指向型対話システムはユーザからの特定の要求に対する情報の提供を目的としており，雑談のように継続した会話は考慮されていない．例えば，Apple の Siri や Yahoo Japane の Alexa がタスク指向型対話システムである．それに対して，非タスク指向型システムは特定の要求への情報の提供を目的としておらず，主に雑談形式の対話の事を指す．例えば，日本マイクロソフトのりんなが非タスク指向型対話システムであり雑談を行う．

特に，雑談を円滑に行うには，単に直前のユーザによる発話である質問に対して返答するだけでなく，これまでの雑談内容を反映した返答が必要なケースが十分考えられる．例えば，1 ターン前のシステム応答に対するユーザ発話文が，そのシステム応答文への質問文である場合，直前のユーザ発話文のみから次のシステム応答文を生成する事は不適切である．そのため，円滑で自然に雑談を継続するためには直前のユーザ発話 1 文のみを学習するには不十分であると考えられる．

しかし，非タスク指向型対話システムの現状は，直前のユーザ発話 1 文のみを入力としてシステム応答文を生成するため，ユーザの会話に対して不自然な応答を行う事が多く対話が破綻する問題がある．また，事前に設定した形式的な発話ではユーザの楽しみが薄く対話に対するモチベーションが下がるという問題が存在している．

そこで，単純に直前のユーザ発話 1 文に対する応答文を学習するのではなく，これまでの対話内容であるマルチターン対話ログの複数文に対して応答文の生成を学習することで，円滑な雑談対話を継続できる応答文生成ができると考えた．そのため，本論文では自然な対話の継続という目標を達成するための第一歩として，逐次的に Transformer[1] を構築して，各ターンの対話文を各タイミングの Transformer に入力することで，複数の応答文を一度の学習で利用する対話モデルを提案し検討する．

1.1 マルチターン対話文例

対話を複数回繰り返すマルチターン対話の重要性を示す対話文例を記述する．表 1 では，第 4 発話を生成する事を考えると，マルチターン対話を考慮しない場合は，第 3 発話のみを入力として第 4 発話を生成する必要があるが，生成は困難である．また，第 3 発話と異なり，第 1 発話のみを入力として第 4 発話を生成することは比較的に容易であると考えられる．

表 1 マルチターン対話文例

第 1 発話	発話者 A	あなたの家は何処ですか？
第 2 発話	発話者 B	山の中駅って知ってる？
第 3 発話	発話者 A	知っているよ。
第 4 発話	発話者 B	その駅の近くに私の家はあるよ。

このように，生成する応答文の直前の発話文のみ

を入力とせず，マルチターン対話を考慮する必要性は高いと考えられる．

2 関連研究

マルチターン対話に着目した応答文生成の手法として，Serban 等が提案している RNN を対話ベースの階層的に拡張した HRED(Hierarchical Recurrent Encoder-Decoder)[2] がある．HRED では，話者間の対話のやり取りを 1 ターンと定義しており，対話ログを 1 ターン毎に分割して階層化された RNN の 1 階層分の Encoder に入力している．

HRED の RNN 部分を Transformer に拡張した手法として，Santra 等が提案している Hierarchical Transformer[3] や，岩間等が提案しているマルチターン対話の応答文生成を行う階層型 Transformer[4] がある．これらの階層型 Transformer は，2 つの Encoder と 1 つの Decoder で構成されており，トークンレベルの Encode と文脈レベルの Encode の 2 階層に分けて処理している．

また，その他にマルチターン対話に着目した手法として，Li 等が提案している手法 [5] もある．この手法は発話したユーザの特徴を捉えている埋め込みベクトルを応答文の生成時に反映させることで，ユーザ毎に発話の傾向に一貫性を持たせるモデルである．

2.1 Transformer

Vaswani 等が提案した Transformer[1] は RNN を使わずに Attention 機構のみを使用した Encoder-Decoder モデルの手法である．

対話システムに Transformer を適用すると，図 1 のように，Encoder 側に発話文を入力して，Self-Attention を行うことで発話文での単語間の関連度を計算する．

Decoder 側では，エンコードされた単語ベクトルと応答文を入力とした Multi-Head Attention を行い発話文と応答文の関連度を算出する．その情報から応答文を生成していくモデルが Transformer である．

3 提案手法

雑談において複数回の会話のやり取りを学習して応答文を生成する際には，単に直前のユーザによる発話である質問等に対して返答するにはユーザの発話とシステム応答の関係が 1 対 1 である．その場合は，直前の発話文のみを学習対象とすれば良い．しかし，これまでの雑談内容から影響を受けた応答文を生成するには直前の発話文のみを学習対象にしては不十分である．そこで，3 ターン前の対話文から学習対象とする手法を提案し検討する．

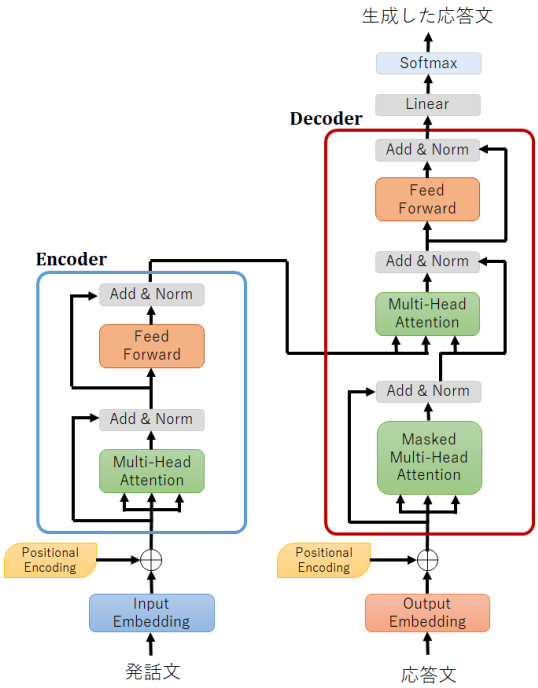

図 1 Transformer の構成

具体的には，図 2 のように 3 個の Transformer を逐次的に構成する．そして，1 ターン目の Transformer-1 の Encoder には，第 1 発話文を入力する．2 ターン目の Transformer-2 の Encoder には，乱数 r が閾値 s よりも大きい場合は，トレーニングデータの第 2 発話文を入力する．しかし，乱数 r が閾値 s よりも小さい場合は，Transformer-1 の Decoder の出力から生成した第 2 発話文を Transformer-2 の Encoder に入力する．乱数 r は $0 \leq r \leq 1$，閾値 s は $0 \leq s \leq 1$ とする．同様に，3 ターン目の Transformer-3 の Encoder には，乱数 r が閾値 s よりも大きい場合は，トレーニングデータの第 3 発話文を入力する．しかし，乱数 r が閾値 s よりも小さい場合は，Transformer-2 の Decoder の出力から生成した第 3 発話文を Transformer-3 の Encoder に入力する．

また，逐次的に構成した各 Transformer ではニューラルネットワークにおける各パラメータを共有するように構成している．

推論時には，生成する応答文となる第 4 発話文は各第 n 発話のみを使用して生成する．これにより，1 つのモデルで多様性のある 3 つの応答文を生成することが期待出来る．また，直前の発話ではない第

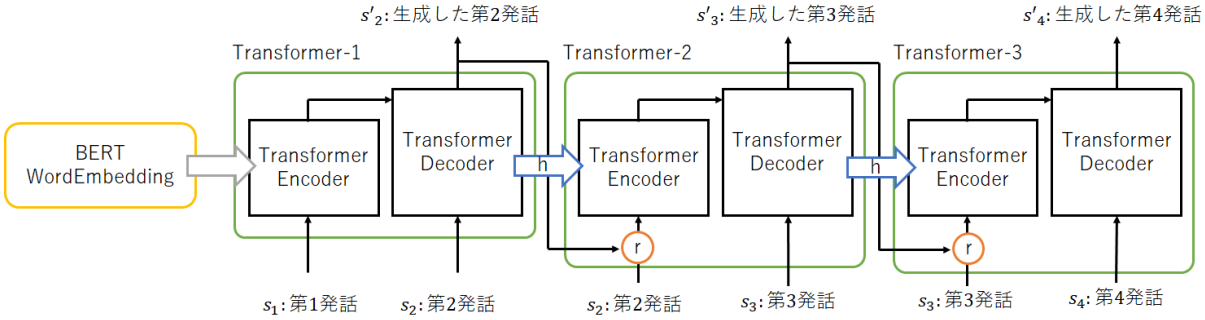

図 2　提案手法のモデル

1 発話や第 2 発話に依存した応答文の生成も可能となると予測する．

また，提案手法の Transformer における Word Embedding には事前に BERT[6] で学習した単語分散表現を使用する．事前学習の BERT のトレーニングデータには Wikipedia を利用した．

4　評価実験

3 ターンの対話データに対する評価実験を行う．推論時に，第 3 発話のみから生成した第 4 発話と，第 2 発話と生成した第 3 発話から生成した第 4 発話と，第 1 発話と生成した第 2・3 発話から生成した第 4 発話の比較を行う．Baseline の手法は逐次的ではない Transformer を用いる．

4.1　データセット

AAAI のワークショップである DSTC9 の trac3 で公開されている対話コーパスを使用している．使用したデータの詳細は表 2 である．

表 2　DSTC9-trac3 の対話コーパス

単語数	ボキャブラリー数	3 ターン対話文セット
335580	6635	5000

4.2　評価方法

生成した第 4 発話と教師データの第 4 発話を比較するために式 (1) と式 (2) の BLEU[7] を使用した．

$$BLEU = BP \times \exp(\sum_{n=1}^{N} \omega_n \log p_n) \tag{1}$$

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \tag{2}$$

式 (1) と式 (2) において，c は提案したモデルで生成した文の長さ，r は教師データにおける文の長さであり，ω_n は重み，p_n は単語の一致率を表す modified n-gram precision である．

4.3　実験結果

表 3 は第 3 発話のみから第 4 発話を生成した結果であり，表 4 は第 2 発話から第 4 発話を生成した結果である．そして，表 5 は第 1 発話から第 4 発話を生成した結果である．表 3，4，5 における N は式 (1) の N である．

表 3　第 3 発話のみでの応答文生成

	N=1	N=2	N=3	N=4
Baseline	0.09117	0.02245	0.01459	0.01206
提案手法 (s=0.1)	0.09069	0.02239	0.01452	0.012
提案手法 (s=0.5)	0.09077	0.02244	0.01457	0.01205
提案手法 (s=1.0)	0.08929	0.02215	0.01439	0.01189

表 4　第 2 発話と生成した第 3 発話

	N=1	N=2	N=3	N=4
Baseline	0.09117	0.02245	0.01459	0.01206
提案手法 (s=0.1)	0.09109	0.02250	0.01460	0.01206
提案手法 (s=0.5)	0.09626	0.02354	0.01523	0.01257
提案手法 (s=1.0)	0.08912	0.02213	0.01438	0.01189

表 5　第 1 発話と生成した第 2・3 発話

	N=1	N=2	N=3	N=4
Baseline	0.09117	0.02245	0.01459	0.01206
提案手法 (s=0.1)	0.08962	0.02222	0.01443	0.01192
提案手法 (s=0.5)	0.10219	0.02467	0.01592	0.01314
提案手法 (s=1.0)	0.08912	0.02213	0.01438	0.01189

5　考察

表 4・5 から，閾値 $s = 0.5$ の時に Baseline よりも BLEU 値が高くなっている．特に第 2 発話と生成した第 3 発話から生成した場合と，第 1 発話と生成した第 2・3 発話から生成した時に，$N = 1$ におい

て BLEU 値が Baseline よりも良い値となっている．しかし，閾値 $s = 0.1$ と $s = 1.0$ の結果は，全体的に Baseline よりも悪化していることが分かる．

この事から，生成対象である第 4 発話の直前発話である第 3 発話のみから生成するよりも，その前の第 1 発話や第 2 発話の影響を大きくして第 4 発話を生成する方が良いケースが存在すると考えられる．

また，閾値 s が低すぎる場合は，第 3 発話の影響が大きく第 1 発話や第 2 発話を重視した第 4 発話の生成が出来ないと予想できる．逆に閾値 $s = 1.0$ だと，第 3 発話の影響が極端に小さくなったことで，第 4 発話が第 3 発話に対して単純な返答の場合の生成精度が低くなったと考察できる．

しかし，全体的に BLEU 値が低いことが分かる．これは，Transformer のパラメーターチューニングを更に行うことで改善することが出来ると考えられる．

6 おわりに

本論文では，マルチターン対話に着目した応答文生成において，複数文を一度に逐次的に構築した Transformer のモデルで学習する手法を提案した．

Baseline である逐次的ではない Transformer と比較する実験を行った．提案手法は，閾値を $s = 0.5$ にした時に，生成する応答文の直前の発話文よりも前の発話文の影響を大きくして生成した際に良い応答文の生成ができるケースもあることが確認できた．

今後は，単純な閾値での制御のみではなく，生成した発話文とトレーニングデータの発話文の合成による制御も検証していきたい．また，Transformer 自体のパラメータチューニングを行う必要もある．

参考文献

[1] A.Vaswani, N.Shazeer, N.Parmar, J.Uszkoreit, L.Jones, A.N.Gomez, L.Kaiser, I.Polosukhin, Attention Is All You Need, In Advances in Neural Information Processing Systems, pp.6000-6010, 2017.

[2] I.V.Serban, A.Sordoni, Y.Bengio, A.Courville, J.Pineau, Building end-To-end dialogue systems using generative hierarchical neural network models, AAAI 2016, pp.3776-3783, 2016.

[3] B.Santra, P.Anusha, P.Goyal, Hierarchical Transformer for Task Oriented Dialog Systems, NAACL 2021, pp.5649-5658, 2021.

[4] 岩間寛悟，狩野芳伸，再帰的にエンコードを行う階層型 Transformer によるマルチターン雑談対話の応答生成，NLP2020，pp.649-652，2020.

[5] J.Li, M.Galley, C.Brockett, G.Spithourakis, J.Gao, B.Dolan, A Persona-Based Neural Conversation Model, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp.994-1003, 2016.

[6] J.Devlin, M.W.Chang, K.Lee, K.Toutanova, BERT:Pre-training of Deep Bidirectional Transformers for Language Understanding, NAACL-HLT 2019, pp.4171-4186, 2019.

[7] K.Papineni, S.Roukos, T.Ward, W.J.Zhu, BLEU: a Method for Automatic Evaluation of Machine Translation, ACL, pp.311-318, 2002.