

固有名詞に注目した Transformer による雑談対話モデルの構築

郭 恩孚¹ 南 泰浩¹

¹ 電気通信大学大学院

enfu.guo@uec.ac.jp minami.yasuhiro@is.uec.ac.jp

概要

ユーザーの提示する固有名詞にうまく注目するため、本研究では、対話モデルの事前学習において、固有名詞をマスクし、マスクされた部分を予測する固有名詞マスク学習を用いる対話モデルを提案する。モデルを用いて、発話の自然性と応答性を評価する実験を行った結果、固有名詞マスク学習を用いる提案手法のモデルがベースラインモデルと比べ、自然性スコアで0.45ポイント、固有名詞応答スコアで0.54ポイントの向上を示し、提案手法の有用性を確認した。

1 はじめに

日常会話において、人は相手が提示した固有名詞について応答することで、対話がより円滑に行える。近年、雑談対話モデルの研究が進み、生成される発話を一文としてみると、自然な発話を生成できつつある。しかし、対話の接続性に関して「固有名詞について応答する」という観点からみると、無関係の固有名詞での応答や、固有名詞を無視する応答など、ユーザーの発話に含まれる固有名詞にうまく注目できていないことが問題点として挙げられている。

本研究の関連研究としては、ランダムマスク学習 [1] とシングル応答生成学習 [2] を事前学習として用いる雑談対話モデルを構築することが挙げられる。ランダムマスク学習とは、トークンレベルでの学習を行うため、トークンをランダムに置き換え、元の文を復元する学習である。シングル応答生成学習とは、句レベルでの学習を行うため、入力された発話文に対する応答文を生成する学習である。

ランダムマスクでの学習では、トークン分割方法によって、一つの固有名詞が複数のトークンに分割され、固有名詞の復元の学習がうまくいかない場合がある。そこで、本研究では、固有名詞により注目するため、事前学習において、固有名詞をマスクし、

元の文を復元する学習を提案する。実験により、その有用性を検証した。ここでの事前学習時の概念図を図1に示す。図に示すように、事前学習では、固有名詞マスクだけではなく、ランダムマスク、シングル応答生成も実施する。

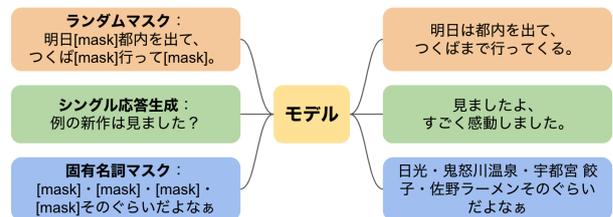


図1 事前学習時の概念図

2 対話学習データ

本研究では Wikipedia[3]、CC-100[4]、ツイートの3つのソースを用いて、学習コーパスを構築した。

各データソースに対して、以下の条件に該当するデータを除外するクリーニング処理を行った。また、単語の分割には MeCab[5] を用いた。

- URL・絵文字などを含む文
- 日本語・英語以外の言語のシンボルを含む文
- 日本語が含まれない文
- 同じ表現が連続で出現する文
- 不適切表現が出現する文

クリーニング処理後の各データに基づいて、下記の学習コーパスを構築した。

- Wiki コーパス：Wikipedia データ
- CC-100 コーパス：CC-100 データ
- ツイートリプライコーパス：ツイートとそのリプライのペアデータ
- ツイート疑似対話コーパス：ツイートとそのリプライからなるリプライチェーン

構築したコーパスのうち、Wiki コーパス、CC-100 コーパス、ツイートリプライコーパスを用いて事前学習を行う。また、ツイート疑似対話コーパスを用

いて Fine-tune を行う。各コーパスのデータ数を表 1 に示す。

表 1 各コーパスのデータ数

コーパス名	データ数
Wiki コーパス	10^8
CC-100 コーパス	10^8
ツイートリプライコーパス	6×10^7
ツイート疑似対話コーパス	10^6

3 学習および評価設定の詳細

本研究は3つの事前学習：ランダムマスク学習、シングル応答生成学習、固有名詞マスク学習、で事前学習を行った。事前学習後、ツイート疑似対話コーパスを用いて Fine-tune を行った。事前学習と Fine-tune の学習時、入力文脈からトークンへの分割は SentencePiece[6] を用いた。この時、クリーニング処理後の 2,000 万のツイートから、語彙数が 32,000、文字カバー率が 99.995% となるように学習を行った。

3.1 ランダムマスク

トークンレベルで学習するため、ランダムマスク学習を設定した。ランダムマスク学習では、入力トークンの 15% を [MASK] トークンに置き換え、元の文を復元する。

3.2 シングル応答生成

Suchin ら [7] の研究により、Fine-tune 時のデータと同じドメインのデータを用いて事前学習することで、モデルの性能が向上することが報告されている。本研究では、その結果を踏まえて、ツイートとリプライツイートのペアで、ツイートを入力とし、リプライツイートをターゲット発話とするシングル応答生成学習を設定した。シングル応答生成学習では、入力からターゲット発話を生成する。

3.3 提案手法：固有名詞マスク

SentencePiece は Byte Pair Encoding (BPE) を用いて、トークン分割を行っている。この分割では、辞書サイズが圧縮される一方、一つの固有名詞が複数のトークンに分割され、ランダムマスク学習だけでは、固有名詞を生成する学習がうまくいかない場合がある。この問題に対処するため、本研究では、対話モデルが文脈の中にある固有名詞を陽に捉えられるよう、固有名詞マスク学習を提案する。固有名詞のマスク学習は、入力文を事前に Mecab で形態素解

析を行い、入力文にあるすべての固有名詞を事前に [MASK] トークンに置き換えた上で、トークン分割し、元の文を復元する。

3.4 Fine-tune

一般に、生成型雑談対話モデルでは、過去の発話履歴を考慮する、応答を生成する必要がある。連続する対話を含む対話コーパスで事前学習済みモデルを Fine-tune することで、発話履歴を考慮して応答生成できると考えられる。そこで、本研究では、収集したツイートデータから、一連のリプライ関係にあるツイートを抽出した。ツイートとその一連のリプライに該当するツイートをリプライがなくなるまで抽出し、リプライチェーンを作成し、ツイート疑似対話コーパスを作成した。例えば、一つのリプライチェーンに n 個のツイートである場合、入力とターゲット発話のペアは $[(t_1, t_2), (t_1 - t_2, t_3), (t_1 - t_2 - t_3, t_4), \dots, (t_1 - \dots - t_{n-1}, t_n)]$ となる。

3.5 評価時の応答生成設定

モデルが生成する発話は自然性と多様性の両方を満たすことが望ましい。本研究では Beam Search により提案モデルから発話を生成した、そのパラメータを表 2 に示す。繰り返し同じ表現が出現する発話の生成を防ぐため、生成された系列中にすでに出現した N-gram を出現させないようにした。さらに、「そうですね」といった無難な発話を繰り返し応答しないように、difflib の SequenceMatcher を用いて、現在の生成候補と過去のモデル発話と比べ、類似度が 0.7 以上の発話は候補から除外した。

以上の設定により、多様性を満たすことができる。加えて、候補中、perplexity が比較的小さな発話を選択することで、自然性を満たすこともできる。

表 2 応答生成設定

デコード方法	Beam Search
ビームサイズ	80
最小出力系列長	5
繰り返し出現防止 N-gram	3
発話候補	80

4 実験

4.1 実験設定

モデルアーキテクチャについて、先行研究 [8] では、シンプルな Transformer Encoder-Decoder モデル

が Transformer Encoder のみのモデルよりよい性能を示すことが報告されている。その結果を踏まえて、本研究でもシンプルな Transformer Encoder-Decoder モデルを使用した。モデル設定の詳細を表 3 に示す。

表 3 モデル設定

アーキテクチャ	Transformer
エンコーダ層数	2
デコーダ層数	12
Attention Head 数	12
隠れ層の次元数	1920
Feed-Forward Network の次元数	3840

4.1.1 事前学習

事前学習では、2. で構築した Wiki コーパス、CC100 コーパス、ツイトリプライコーパスを利用した。学習毎の Encoder への入力形式、Decoder からの出力の形式の例を表 4 に示す。学習の Optimizer には Adafactor[9] を用い、学習率を $1e-4$ 、warmup step を 1000 にそれぞれ設定し、目的関数には label smoothed cross entropy を使用した。

4.1.2 Fine-tune

Fine-tune では、2. で構築したツイート疑似対話コーパスを利用した。Encoder への入力、話者情報を表す [SPK] トークンを導入し、過去の全発話を入力する。ユーザーの発話に該当する部分の先頭に [SPK1] トークン、モデルの発話に該当する部分の先頭に [SPK2] トークンを用いた。入力のテンプレートを表 5 に、入力の例を表 6 に示す。また、Feed-Forward Network および Attention の dropout 率を 0.1 に設定し、学習の Optimizer には事前学習時と同様に Adafactor を使用した。学習率を $1e-4$ 、warmup step を 2000 にそれぞれ設定し、目的関数には label smoothed cross entropy を用いた。

4.2 実験結果および分析

固有名詞マスク学習を実施していないモデルをベースラインモデルとし、固有名詞マスク学習を実施した提案モデルと比較するため、実験参加者 7 名でそれぞれ 5 会話について評価した。評価者と異なる実験参加者とモデルを使った対話システムの会話では、実験参加者に 2 つの固有名詞を提示し、10 ターン以上継続した対話となるよう指示した。ま

た、評価では、下記の軸について 5 段階 (1: とてもそう思わない ~ 5: とてもそう思う) 評価とした。

- 自然性: 対話全体が自然か
- 固有名詞応答 (1): 実験参加者が提示した 1 つ目の固有名詞について応答できているか
- 固有名詞応答 (2): 実験参加者が提示した 2 つ目の固有名詞について応答できているか

自然性の評価結果を図 2 に、応答の評価結果を図 3 と図 4 に示す。また、各軸のスコアの平均を表 7 に示す。さらに、各軸の平均スコアに有意な差があるかどうかを検定した。有意水準を 0.05、帰無仮説を「各軸の平均スコアに差がない」とした。その検定結果を表 8 に示す。

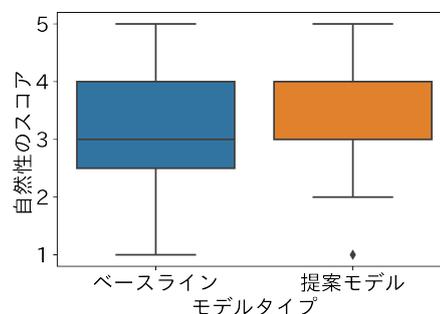


図 2 自然性の評価結果

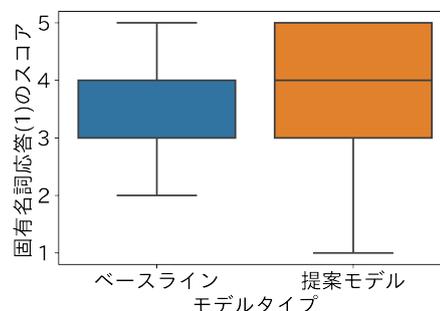


図 3 固有名詞応答 (1) の評価結果

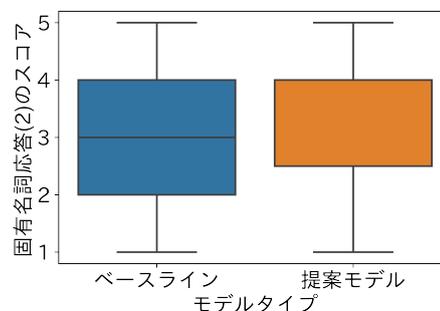


図 4 固有名詞応答 (2) の評価結果

表 8 により、各軸の P 値は有意水準の 0.05 より小さいため、結果は有意であり、「各軸の平均スコア

表4 各事前学習学習の入出力形式の例 (トークナイズ後)

事前学習学習	入力形式	出力形式
ランダムマスク	明日は 都内 [MASK] つくば [MASK] くる。	明日は 都内 を出て つくば まで行ってくる。
シングル応答生成	珍しい よね 笑 よろしくです。	よろしくです。 笑
固有名詞マスク	明日は [MASK] を出て [MASK] まで行ってくる。	明日は 都内 を出て つくば まで行ってくる。

表5 Fine-tune 時の入力形式のテンプレート

雑談:[SPK1] ユーザーの発話:U₁[SEP] [SPK2] モデルの発話 S₁[SEP][SPK1]...[SEP][SPK2]

表6 Fine-tune 時の入力例 (トークナイズ後)

雑談: [SPK1] みんなで 浅草 に行く のだよ [SEP] [SPK2] めちゃくちゃ 楽しそう やん [SEP] [SPK1] 食べ歩き する [SEP] [SPK2]

に差がない」という帰無仮説は棄却された。

表7により、事前学習時に固有名詞マスク学習を設定した提案モデルがベースラインモデルより自然性、固有名詞応答(1)と(2)のスコアがともに向上することが確認できた。ユーザーが提示した固有名詞により注目し、応答することで、対話全体の自然性も上がったと考えられる。

提案モデルとベースラインモデルによる生成例を図5に示す。ユーザーが提示した「鬼怒川温泉」という固有名詞について、ベースラインモデルでは、「そうですね。」といった無難な応答を生成していることに対して、提案モデルでは「鬼怒川温泉」という固有名詞を含んだ、「鬼怒川温泉、有名ですね」という応答を生成している。

今回の評価では実験参加者に1対話で提示した2つの固有名詞に対する発話の分析を行った。

図3と表7みると、ユーザーが提示した1つ目の固有名詞について、応答スコアが0.45ポイント向上している。図4と表7みると、ユーザーが提示した2つ目の固有名詞について、応答スコアが0.63ポイント向上している。

過去の対話履歴をモデルの入力としているため、過去の発話が直前の発話のノイズになる場合もあることを考えると、上記の結果により、固有名詞マスク学習を事前学習時に設定することで、実験参加者が2つ目の固有名詞を提示する時、提案モデルは1

表7 モデルの評価結果

モデル	自然性	固有名詞 応答(1)	固有名詞 応答(2)
ベースライン	3.20	3.37	2.77
提案モデル	3.65	3.82	3.40

表8 モデルの評価の検定結果

P 自然性	P 固有名詞応答(1)	P 固有名詞応答(2)
0.0303	0.0472	0.0095

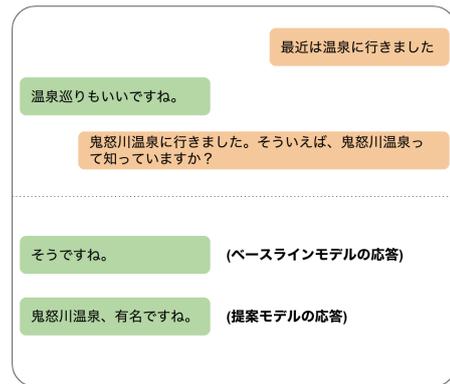


図5 モデルによる生成例

つ目の固有名詞から2つ目の固有名詞に注目を切り替えることができ、固有名詞マスク学習の有用性を確認できた。

5 おわりに

ユーザーが提示した固有名詞に注目する対話モデルを構築した。固有名詞に注目するため、事前学習時に、入力文脈に含む固有名詞を[MASK]トークンに置き換え、元のトークンを予測する固有名詞マスク学習を提案した。実験により、提案手法の人手評価を行い、固有名詞マスク学習を設定したモデルの自然性、固有名詞応答スコアを評価した結果、提案手法の有用性を確認できた。

事前学習時固有名詞マスク学習を実施することにより、対話モデルがより会話文脈中に含まれた固有名詞に注目できることがわかった。一方で固有名詞マスク学習において、学習データに含まれなかった固有名詞については、会話中に注目しておらず、それらに注目するようモデルの構築は今後の課題である。

参考文献

- [1] Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiayang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation, 2021.
- [2] 杉山弘晃, 成松宏美, 水上雅博, 有本庸浩, 千葉祐弥, 目黒豊美, 中嶋秀治. Transformer encoder-decoder モデルによる趣味雑談システムの構築. 人工知能学会研究会資料 言語・音声理解と対話処理研究会, Vol. 90, p. 24, 2020.
- [3] Wikimedia Foundation. Wikimedia downloads. 2021.
- [4] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, Online, July 2020.
- [5] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to Japanese morphological analysis. In **Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing**, pp. 230–237, Barcelona, Spain, July 2004. Association for Computational Linguistics.
- [6] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. Vol. abs/1808.06226, 2018.
- [7] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, Online, July 2020.
- [8] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. Recipes for building an open-domain chatbot. Vol. abs/2004.13637, 2020.
- [9] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. Vol. abs/1804.04235, 2018.