知識グラフと Wikipedia を用いた雑談対話モデルの構築

郭 恩孚 [1]　南 泰浩 [1]
[1] 電気通信大学情報理工学研究科
enfu.guo@uec.ac.jp　minami.yasuhiro@is.uec.ac.jp

概要

本研究では，対話モデルがユーザーの提示するエンティティを元に対話をより広げられるように、対話履歴から次の話題となるエンティティとその要約を外部から取得し、モデルの入力文脈に追加する Fine-tune 手法を提案する。学習したモデルを用いて、発話の自然性、エンティティ応答の妥当性、話題提供の適切性を評価する実験を行った結果、提案手法のモデルがベースラインと比べ、自然性スコアが 0.66 ポイント、エンティティ応答の妥当性スコアが 0.52 ポイント、話題提供の適切性スコアが 0.6 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

近年、Meta の BlenderBot[1] や Google が発表した Meena[2] など、生成ベースの対話システムが盛んに研究されている。対話システムの研究においては、対話履歴のみを入力とするモデルが多く提案されている。しかし、これらのモデルはやや無難な応答を多く生成することが報告されている。[3]。その原因として、学習データ内に、「わからない」や「そうですね」、「いいえ」などの文が多く含まれることが挙げられる。

この問題を解決するため、対話に外部知識トリプルを導入することで、生成ベースモデルが学習した対話データ以外の情報を取り込むことができることが報告されている [4][5]。知識トリプルとは、自然言語処理において、人間の認識や知識を表現するために使用される三つの要素（主語、述語、目的語）からなるデータの構造である。例えば「猫は動物である」という文章を（猫、分類、動物）で表す。

外部知識トリプルの導入により、知識をモデルのパラメータとして保持する代わりに、知識を利用した応答生成というプロセスそのものをモデルのパラメータに保持することができる。その結果、モデルを再学習する必要がなくなり、学習データに出現したことがない知識を用いた応答生成が可能になる。しかし、これらのモデルは、タスク指向型対話や、一問一答のような対話では有効であるが、複数ターンの雑談対話では、過去の対話履歴を考慮できない応答の生成や、新たな話題を持ち出せない応答の生成など、知識トリプルの情報をうまく使っていないことが問題点として挙げられている。

そこで、本研究では、対話履歴から、次の話題となるエンティティとその要約を知識グラフと Wikipedia から取得し、モデルの入力文脈に追加する Fine-tune 手法を提案する。実験では、外部知識として知識トリプルのみ使って Fine-tune したベースラインモデルと外部知識として知識トリプルとエンティティの要約を使って Fine-tune した提案モデルの比較を行った。人手による評価を行った結果、提案手法で学習したモデルは知識トリプルのみ使うモデルよりスコアの向上を確認でき、提案手法の有用性を確認できた。

2　データ

2.1　対話データ

モデルの学習用対話データとして、[6] の 4 つのコーパスを用いた。各コーパスの構築ソースとデータ数を表 1 に示す。モデルの学習では、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い、ツイート疑似対話コーパスを用いて Fine-tune を行う。

コーパス名	概要
Wiki コーパス	構築ソース：Wikipedia データ データ数：10^8
CC-100 コーパス	構築ソース：CC-100 データ データ数：10^8
ツイート リプライコーパス	構築ソース：ツイートとそのリプライのペア データ数：6×10^7
ツイート 疑似対話コーパス	構築ソース：ツイートとそのリプライからなるリプライチェーン データ数：10^6

図 1　各コーパスの構築ソースとデータ数

2.2 知識グラフとエンティティ要約

知識グラフは Wikidata[7] のダンプデータを用いて構築した。Wikidata は共同編集型の知識ベースであり、Wikipedia と同じく Wikimedia Project 群の一つである。Wikidata から（エンティティ 1、プロパティ、エンティティ 2）のような知識トリプルを抽出することができる。

しかし、Wikidata のデータは膨大であり、扱いが困難であると考えられる。そこで、本研究では、計算量と雑談対話の性質から、知識グラフの構築に以下の制限を付けた。

- エンティティに日本語名が存在するデータ
- 雑談対話において、有意義なプロパティを含むデータ

ここでは「雑談対話において、有意義なプロパティ」とは、以下の項目とする。

- 人物関連プロパティ
- 人物・組織項目用プロパティ
- 人物項目用プロパティ
- 組織項目用プロパティ
- 私生活関連プロパティ

これらのプロパティの探索には [8] を用いた。結果として、1,991,103 個のエンティティ、224 種類のプロパティ、56,552,434 個のトリプルを用いて知識グラフを構築した。構築した知識グラフの例を図 2 に示す。

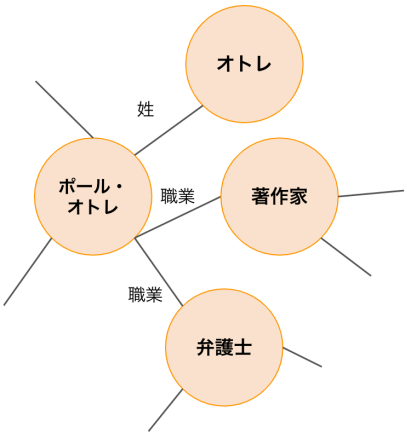

図 2 構築した知識グラフの例

また、各エンティティの要約情報は、Wikipedia を用いて抽出した。その例を表 1 に示す。

表 1 エンティティとその要約例

エンティティ名	要約
ポール・オトレ	ポール・マリー・ギスラン・オトレは、作家、起業家、空想家、法律家、平和活動家である。...
ポルトガル	ポルトガルは、南ヨーロッパのイベリア半島に位置する共和制国家。ユーラシア大陸最西端の国である。...

3 提案手法

対話履歴から、直近のユーザー発話やシステム発話の中に含まれるエンティティを抽出する。エンティティの抽出には Ginza[9] を用いた。抽出したエンティティを元に、2.2 で構築した知識グラフを用いて、次の話題となるエンティティが存在するかどうかを確認する。次の話題となるエンティティが複数存在する場合、ランダムに一つを選び、そのエンティティの要約を取得する。次の話題となるエンティティが存在しない場合、対話履歴から抽出したエンティティの要約を取得する。取得したエンティティの要約を知識情報とし、対話履歴と共にモデルに入力し、知識応答生成を行う。また、対話履歴からエンティティ抽出できなかった場合、知識情報の部分を空欄にし、対話履歴のみモデルに入力し、一般応答生成を行う。その一連の処理のフローチャートを図 3 に示す。

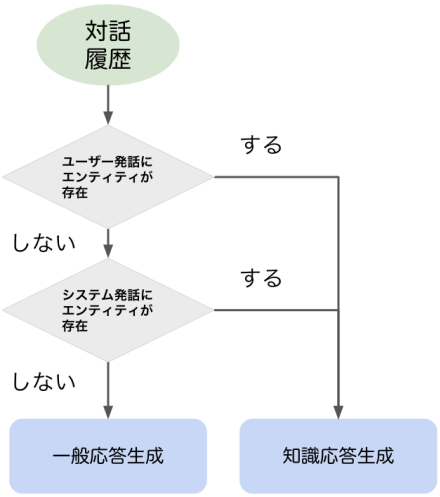

図 3 処理のフローチャート

4 実験

4.1 実験設定

モデルアーキテクチャには、Transformer Encoder-Decoder を用いた。モデルの詳細な設定を表 2 に示す。

表 2 モデル設定

アーキテクチャ	Transformer
エンコーダ層数	12
デコーダ層数	12
Attention Head 数	12
隠れ層の次元数	768
Feed-Forward Network の次元数	3072

4.2 事前学習

事前学習では、2.1 の Wiki コーパス、CC-100 コーパス、ツイートリプライコーパスを利用した。また、[6] と同じく 3 つの学習：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を行った。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。シングル応答生成学習では、入力からターゲット応答を生成する。固有名詞マスク学習では、入力文を事前に Sudachi[10] で形態素解析を行い、入力文にある固有名詞を事前に [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。ただし、[6] では固有名詞を全部置き換えるに対して、本研究では 1 つの固有名詞のみ置き換えるようにした。各事前学習の入出力形式を表 3 に示す。学習の Optimizer には Adafactor[11] 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。

4.3 Fine-tune

Fine-tune では、2.1 のツイート疑似対話コーパスを利用した。Encoder への入力は、話者情報を表す [USER][SYS] トークン、知識情報を表す [KL] トークンを導入し、先頭に「[SYS] の発話を生成する：」を付け、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 [USER]、末尾に [/USER] トークンをつけた。システムの発話に該当する部分の先頭に [SYS] トークン、末尾に [/SYS] トークンをつけた。知識情報に該当する部分の先頭に [KL] トークン、末尾に [/KL] トークンをつけた。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 Adafactor を使用した。学習率を 1e-4、warmupstep を 2000 にそれぞれ設定し、目的関数には label smoothed cross entropy を用いた。

4.3.1 ベースライン

知識情報の部分に知識トリプルを入れ、Fine-tune し応答文を生成するものを、ベースラインとして用いる。その入力例を表 4 に示す。

表 4 ベースラインモデルの Fine-tune 時の入力形式のテンプレート

[SYS] の発話を生成する:[USER] ユーザーの発話:U_1[/USER] [SYS] モデルの発話 S_1[/SYS]...[/USER][KL] 知識トリプル:T[/KL][SYS]

4.3.2 提案モデル

知識情報の部分にエンティティの要約を入れ、Fine-tune し、応答文を生成するものを提案モデルとして用いる。その入力例を表 5 に示す。

表 5 ベースラインモデルの Fine-tune 時の入力形式のテンプレート

[SYS] の発話を生成する:[USER] ユーザーの発話:U_1[/USER] [SYS] モデルの発話 S_1[/SYS]...[/USER][KL] エンティティ要約:E[/KL][SYS]

4.4 評価時の応答生成設定

モデルの応答は自然性と多様性の両方を満たすことが望ましい。そこで、本研究では Beam Search によりベースラインモデルと提案モデルから応答を生成し、以下の処理を行う。同じ表現が繰り返し出現する応答の生成を防ぐため、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応答と比べ、類似度が 0.7 以上の応答は候補から除外した。残りの候補の中、perplexity が比較的小さな応答を選択することで、自然性と多様性を同時に満たす応答を選んだ。応答生成設定のパラメータを表 6 に示す。

表 6 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

表 3　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	入力形式	出力形式
ランダムマスク	最近 は 猫 が [MASK] を引く ようになった ので [MASK] か 。	最近 は 猫 が ソリ を引く ようになった ので しょう か 。
シングル応答生成	機会 が あれば 、 行 こう かなでも 、 東京 住み だ よ 。	なん か 勢い で 誘っちゃ って ごめん ね
固有名詞マスク	[MASK] ビュー で 表示 する フォルダ を 識別 する スキーム 。	ニコニコ動画 ビュー で 表示 する フォルダ を 識別 する スキーム 。

4.5　実験結果及び分析

ベースラインモデルと提案モデルを比較するため、評価者 5 名でそれぞれ 10 対話について評価した。評価者と異なる実験参加者に 2 つのエンティティを提示し、モデルを使った対話システムと、10 ターン以上継続した対話となるよう指示した。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) 評価をしてもらった。

- 自然性: 対話全体が自然かどうか
- エンティティ応答の妥当性: モデルは実験参加者が提示したエンティティをうまく応答できたか
- 話題提供の適切性: モデルは実験参加者が提示したエンティティから他の話題を提供できたか

その評価結果を図 4 に示す。図 4 より、提案モデルはベースラインモデルより自然性スコアが 0.66 ポイント、エンティティ応答スコアが 0.52 ポイント、話題提供スコアが 0.6 ポイント向上している。

モデル	自然性	エンティティ応答の妥当性	話題提供の適切性
ベースライン	3.25	3.31	3.15
提案モデル	**3.91**	**3.83**	**3.75**

図 4　モデルの評価結果

ベースラインモデルと提案モデルの生成例を表 5 に示す。ユーザーが提示した「ユニバーサル・スタジオ・ジャパン」というエンティティについて、ベースラインモデルでは、「ユニバーサル・スタジオ・ジャパンはいいですね」といった知識トリプルを使わずにやや無難な応答を生成した。一方、提案モデルでは知識要約の情報を含んだ「私はスーパー・ニンテンド・ワールドに行ったことがあります」という応答を生成しており、より踏み込んだ対話をしている。以上の結果により、実験参加者が提示したエンティティの情報を入力文脈に挿入することで、よりそのエンティティに近い文が生成され、対話全体の自然性も上がったと考えられる。

また、提案モデルの低評価の生成例を図 6 に示す。対話履歴により、猫と猫砂の話をしているため、ユーザーの発話に出現した「ライオン」は「ライオン（企業）」というエンティティのことを指してると考えるのが妥当である。しかし、システムは知識グラフ上の「ライオン（動物）」というエンティティに紐付けた。その結果、誤った知識を使って、唐突に動物のライオンについて発話してしまい、ユーザーを困惑させている。対話履歴からユーザー発話の中に含まれているエンティティを理解し、知識グラフのエンティティに正しく紐付くことが今後の課題である。

対話履歴	…（略） SYS：私もハムスター飼ってみたいですねー USER：ぜひぜひ、そういえば、最近ユニバーサル・スタジオ・ジャパンに行きました。
知識トリプル	[ユニバーサル・スタジオ・ジャパン, 以下を含む, スーパー・ニンテンドー・ワールド]
知識要約	スーパー・ニンテンドー・ワールドは、世界のユニバーサル・テーマパークに建設された、日本のコンピュータゲームシリーズであるマリオシリーズを始めとする任天堂の作品を題材としたエリアである。…(略)
ベースラインモデル応答	ユニバーサル・スタジオ・ジャパンはいいですね。
提案モデル応答	私はスーパー・ニンテンドー・ワールドに行ったことがあります。

図 5　ベースラインモデルと提案モデルの応答文生成例

対話履歴	…(略) SYS：猫を飼うのは大変でしょうか？ USER：猫砂を選ぶのが大変でした。 SYS：どんな猫砂を選びましたか？ USER：ライオンのものにしました。
知識要約	ライオンは、哺乳綱食肉目ネコ科ヒョウ属に分類される食肉類。 別名はシシ（獅子）。オスの体重は250kgを超えることもあり、ネコ科ではトラに次いで2番目に大きな種である。 …
提案モデル応答	獅子ですか、トラに次いで大きいですね。

図 6　低評価の生成例

5　おわりに

本研究では、対話履歴から、次の話題となるエンティティとその情報を知識グラフと Wikipedia から取得し、モデルの入力文脈に追加する手法を提案した。人手評価の結果、知識トリプルで学習したモデルと比較し、自然性、エンティティ応答の妥当性、話題提供の適切性スコアがともに向上していることが確認できた。一方、対話履歴から抽出したエンティティを知識グラフの違うエンティティと紐付け、誤った知識を使った応答が生成されるといったことも生じた。前後の文脈などからより正確にエンティティ抽出することが今後の課題である。

参考文献

[1] 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.

[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] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In **Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 110–119, San Diego, California, June 2016. Association for Computational Linguistics.

[4] Mojtaba Komeili, Kurt Shuster, and Jason Weston. Internet-augmented dialogue generation, 2021.

[5] Debanjan Chaudhuri, Md Rashad Al Hasan Rony, and Jens Lehmann. Grounding dialogue systems via knowledge graph aware decoding with pre-trained transformers, 2021.

[6] 郭恩孚, 南泰浩. 固有名詞に注目した transformer による雑談対話モデルの構築. 言語処理学会 第 28 回年次大会 発表論文集, pp. 1500–1504, 2022.

[7] Wikimedia Foundation. Wikidata downloads. 2022.

[8] Wikidata prop explorer.

[9] Mai Hiroshi and Masayuki. 短単位品詞の用法曖昧性解決と依存関係ラベリングの同時学習. 言語処理学会第 25 回年次大会, 2019.

[10] Kazuma Takaoka, Sorami Hisamoto, Noriko Kawahara, Miho Sakamoto, Yoshitaka Uchida, and Yuji Matsumoto. Sudachi: a japanese tokenizer for business. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, **Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)**, Paris, France, may 2018. European Language Resources Association (ELRA).

[11] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. Vol. abs/1804.04235, 2018.