タスク指向対話システムの外部表知識の参照能力向上

深谷 竜暉　三輪 誠　佐々木 裕
豊田工業大学
{sd18071,makoto-miwa,yutaka.sasaki}@toyota-ti.ac.jp

概要

既存の深層学習を用いた End-to-End のタスク指向対話システムは，応答の生成と外部知識の参照を同時に行うため，応答全体を通じて外部知識を適切に参照できない場合がある．本稿では，外部知識として表を用いるタスク指向対話システムを対象に，応答全体に適した表知識の参照が可能な End-to-End の対話システムを目指して，応答の概形を生成した上で，必要な表知識を参照する手法を提案する．タスク指向対話データセットにおける評価の結果，提案手法は応答全体を利用しないベースラインよりも BLEU スコア・Entity F_1 を向上でき，より正確に表知識を参照できることを明らかにした．

1　はじめに

タスク指向対話システムは，レストランの予約・天気予報の照会・飛行機のフライトの予約など様々な場面での活用が期待されており，産業的に注目を集めている．対話システムの伝統的なパイプラインモデルは言語理解・対話状態追跡・言語生成のモジュールによって構築される [1] が，各モジュールの構築には追加のアノテーションが必要となる場合が多い．この問題に対して，深層学習を用いて，外部知識を学習フレームワークに組み込み，明示的な対話状態や行動などのアノテーションを必要とせずに応答を生成できる End-to-End の対話システム [2] が提案されている．外部知識は表形式で表現できることが多いため，外部知識が表形式で表現された表知識を参照する対話システムは多く提案されており，本研究でも表知識を参照した対話システムを対象とする．

これまでに提案された End-to-End の対話システムの出力では表知識の参照を誤る場合が多くみられる．図 1 に示す例は，本研究でベースラインとする最先端の手法の 1 つである DF-Net が表知識を誤って参照した例である．この例では，応答の文脈からすると，距離を表す 5 miles を参照するのが適切だが，実際には誤って住所の情報を参照して応答してしまっている．

Poi	Distance	Traffic	Poi type	Address
Toms house	3 miles	heavy	friend's house	580 Van Ness Ave
Valero	5 miles	heavy	gas station	200 Alester Ave
Chef Chu's	3 miles	no traffic	chinese restaurant	394 Van Ness Ave
...	...	...	...	...

ユーザ ： Find me the nearest gas station .
システム： The nearest gas station is Valero, 200 Alester Ave away

図 1　表知識を誤って参照した例：正しい応答は “The nearest gas station is Valero, 5miles away.”

本稿では，このような応答の文脈と参照する表知識の内容の不整合を減らすことで，表知識の参照をより正確に行う End-to-End の対話システムの構築に取り組む．

2　関連研究

これまで，表知識を End-to-End の対話システムに取り込むための様々な手法が提案されている．Yang ら [3] は表知識をグラフ形式で取り込むモデルを提案したが，解析の結果，表知識を適切に参照できていないことを報告している．Qin ら [4] は表知識をトリプル形式で取り込んだが，生成した応答の文脈に対して参照している表知識が間違っている問題があったことを報告している．

3　DF-Net

ここでは，ベースラインとして用いるモデルである DF-Net [4] の応答生成と表知識の参照について説明を行う．

3.1　応答生成

注意機構を用いた Seq2Seq モデル [5] を基にして入力の対話 X と表知識 B から応答 Y を生成する．BiLSTM [6] によりエンコードを行い，長さ I の入力に対して中間表現 $\boldsymbol{H} = (\boldsymbol{h}_{\mathrm{enc},1}, \boldsymbol{h}_{\mathrm{enc},2}, \ldots, \boldsymbol{h}_{\mathrm{enc},I})$ を求

Poi	Distance	Traffic	Poi type	Address
Toms house	3 miles	heavy	friend's house	580 Van Ness Ave
Valero	4 miles	heavy	gas station	200 Alester Ave
Coupa	2 miles	moderate	tea place	394 Van Ness Ave
...	...	...	...	...

user : I need gas .
system : Valero is 4miles away .
user : What is the address ?
system : Valero is at 200 Alester Ave .

図 2 表知識を参照する対話例

める．

$$\boldsymbol{h}_{\mathrm{enc},i} = \mathrm{BiLSTM}_{\mathrm{enc}}\left(x_i, \boldsymbol{h}_{i-1}\right) \tag{1}$$

デコードは GRU にて行い，デコーダの中間表現 $(\boldsymbol{h}_{\mathrm{dec},1}, \boldsymbol{h}_{\mathrm{dec},2}, \ldots, \boldsymbol{h}_{\mathrm{dec},T})$ により出力 $(y_1, y_2, \ldots, y_{T-1})$ を生成する．エンコーダの最終時点の中間表現をデコーダのはじめの時点の中間表現 $\boldsymbol{h}_{\mathrm{dec},0}$ に利用する．

$$\boldsymbol{h}_{\mathrm{dec},t} = \mathrm{GRU}_{\mathrm{dec}}\left(y_{t-1}, \boldsymbol{h}_{\mathrm{dec},t-1}\right) \tag{2}$$

モデルは時点 t の応答 y_t を生成するために，注意表現 $\boldsymbol{h}'_{\mathrm{dec},t}$ を計算する．$\boldsymbol{h}'_{\mathrm{dec},t}$ にはエンコード時の中間表現 $\boldsymbol{H}$ の注意を用いる．$\boldsymbol{h}_{\mathrm{dec},t}$ と $\boldsymbol{h}'_{\mathrm{dec},t}$ から学習される重み U を用いて次の語の生成確率の分布 $\boldsymbol{y}_t$ を計算する．ここで生成する語は，語の属性名を表現するスケッチタグ（@を用いた語）を含む．図 2 に示す対話の場合，システムはスケッチタグを用いて，「@Poi is at @Address.」といった応答の概形を生成するように学習する．スケッチタグは同じ時点 t でデコーダにて表知識の要素に置き換えられる．

$$\boldsymbol{o}_t = \boldsymbol{U}\left[\boldsymbol{h}_{\mathrm{dec},t}, \boldsymbol{h}'_{\mathrm{dec},t}\right] \tag{3}$$

$$p\left(y_t \mid y_1, \ldots, y_{t-1}, X, B\right) = \mathrm{Softmax}\left(\boldsymbol{o}_t\right) \tag{4}$$

3.2 エンコーダでの表知識の参照

表知識を参照する際の精度は，表知識を参照するためのクエリに大きく依存する．Memory Networks [7] を適用して表知識 B と入力の対話 X に関する専用メモリ M を用意する．ここでは k ホップの Memory Networks を適用する．Memory Networks では，表知識と入力の対話を表現する訓練される埋め込みベクトル $\boldsymbol{C} = (\boldsymbol{C}^1, ..., \boldsymbol{C}^{k+1})$ を各ホップで使用する．これにより情報を持つためのメモリが拡張され，適切な情報を保持したクエリを作成することができる．M は $M = [B; X] = m_1, ..., m_{b+T}$ で表され，m_i が語に関するメモリを表し，b と T はそれぞれ表知識に含まれる語の数と入力の対話に含まれる語の数である．エンコーダでは最後の中間表現をクエリとする．

$$\boldsymbol{q}^1_{enc} = \boldsymbol{h}_T \tag{5}$$

ホップ k における注意の重みは以下のように計算される．

$$\boldsymbol{p}^k_i = \mathrm{Softmax}\left(\left(\boldsymbol{q}^k_{\mathrm{enc}}\right)^\top \boldsymbol{c}^k_i\right) \tag{6}$$

$\boldsymbol{c}^k_i$ はベクトル $\boldsymbol{C}^k$ を用いた i 番目のメモリ位置の埋め込み表現である．ここでデコード時に用いるグローバルメモリポインタ $G = (g_1, \ldots, g_{b+T})$ を式 (7) によって作成する．

$$\boldsymbol{g}^k_i = \mathrm{Sigmoid}\left(\left(\boldsymbol{q}^k_{\mathrm{enc}}\right)^\top \boldsymbol{c}^k_i\right) \tag{7}$$

グローバルメモリポインタの正解ラベル $G^{\mathrm{label}} = (\hat{g}_1, \ldots, \hat{g}_{b+T})$ は以下のように定義する．

$$\hat{g}_i = \begin{cases} 1 & \text{if Object}\left(m_i\right) \in Y \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

これによりグローバルメモリポインタ G は，正解の応答に出現した語の生成確率が高くなるように訓練される．最後に $\boldsymbol{c}^{k+1}$ に対する注意の重みの和によってメモリ $\boldsymbol{o}^k$ を読み出し，クエリベクトル $\boldsymbol{q}^{k+1}_{enc}$ を更新する．

$$\boldsymbol{o}^k_{\mathrm{enc}} = \sum_i \boldsymbol{p}^k_i \boldsymbol{c}^{k+1}_i, \quad \boldsymbol{q}^{k+1}_{\mathrm{enc}} = \boldsymbol{q}^k_{\mathrm{enc}} + \boldsymbol{o}^k_{\mathrm{enc}} \tag{9}$$

$\boldsymbol{q}^{k+1}_{enc}$ はエンコードされた表知識であり，デコーダで用いられる．

3.3 デコーダでの表知識の参照

各時点では Copy Net にならいデータセット内の語彙と表知識の語彙のどちらから語を生成するかを決めるためのモードを選択する．スケッチタグが生成された時点は複製モードで，そうでなければ生成モードとなる．各時点で中間表現 $\boldsymbol{h}_{dec,t}$ と注意表現 $\boldsymbol{h}'_{dec,t}$ を連結してクエリ $\boldsymbol{q}^1_{\mathrm{dec}}$ を作り表知識を参照する．

$$\boldsymbol{q}^1_{\mathrm{dec}} = [\boldsymbol{h}_{\mathrm{dec},t}, \boldsymbol{h}'_{\mathrm{dec},t}] \tag{10}$$

$$p^k_i = \mathrm{Softmax}\left(\left(\boldsymbol{q}^k_{\mathrm{dec}}\right)^\top \boldsymbol{c}^k_i g^k_i\right) \tag{11}$$

$P_t = \left(p^k_1, \ldots, p^k_{b+T}\right)$ は表知識の各語の生成確率であり，式 (4) によってスケッチタグが生成された時点 t では，最も高い確率を持つ語を選択してスケッチタグと置き換える．

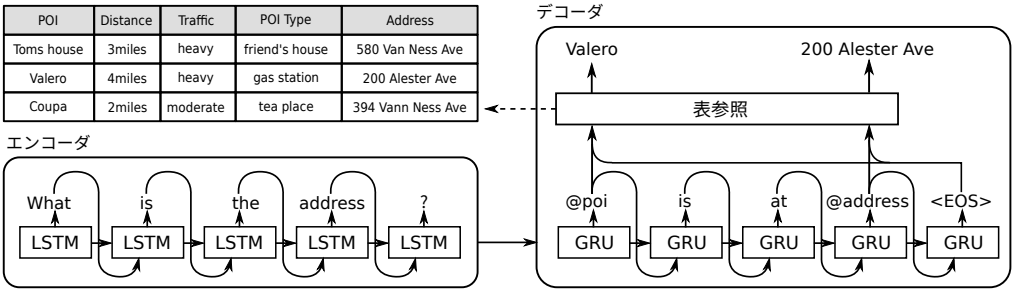

図 3　提案手法の概要

4　提案手法

応答の文脈と参照する知識の内容の不整合を減らすために，生成する応答の概形の文脈表現を利用して知識の参照を行う手法を提案する．提案手法の概要を図 3 に示す．既存手法の中でも比較的高い応答性能を持つ DF-Net [4] をベースラインとして用い，応答生成とエンコーダでの表知識の参照についてはそれぞれ 3.1 節, 3.2 節と同様に行う．提案手法では式 (4) でスケッチタグが生成された場合，同じ時点で表知識の要素に置き換えることはせず，応答の概形を最後まで生成し終えた後に，デコーダにて表知識の参照を行う．提案手法のデコーダでの表知識の参照について以下で述べる．

デコーダでの各時点で中間表現 $\boldsymbol{h}_{\mathrm{dec},t}$ と注意表現 $\boldsymbol{h}'_{\mathrm{dec},t}$ を連結して $q0$ を作る．

$$\boldsymbol{q0}_{\mathrm{dec}} = [\boldsymbol{h}_{\mathrm{dec},t}, \boldsymbol{h}'_{\mathrm{dec},t}] \tag{12}$$

表知識参照用のクエリ $\boldsymbol{q}^1_{\mathrm{dec}}$ は $q0$ とスケッチ応答を生成し終えた時点 T での中間表現 $h_{\mathrm{dec},T}$ を用いて作成する．

$$\boldsymbol{q}^1_{\mathrm{dec}} = \boldsymbol{W}[\boldsymbol{q0}_{\mathrm{dec}}, \boldsymbol{h}_{\mathrm{dec},T}] \tag{13}$$

$\boldsymbol{W}$ は重みである．表知識の参照は式 (14) にて行う．

$$p^k_i = \mathrm{Softmax}\left(\left(\boldsymbol{q}^k_{\mathrm{dec}}\right)^\top \boldsymbol{c}^k_i g^k_i\right) \tag{14}$$

$P_t = \left(p^k_1, \ldots, p^k_{b+T}\right)$ は表知識の各語の生成確率であり，式 (4) によってスケッチタグが生成された時点では，スケッチタグを最も生成確率が高い語と置き換える．

5　実験と考察

5.1　実験に用いたデータセット

データセットは表知識を参照するタスク指向対話を集録した KVRET [2] を使用した．KVRET はカーアシスタントシステムとユーザとの複数回に渡る対話を扱い，対話の内容は道案内と予定管理，天気予報の三つのドメインに渡る．図 2 に示した対話は KVRET に収録されている対話の例である．

表 1　データセット（KVRET）の統計

対話数	3,031
1 対話あたりの発話数	5.25
発話あたりの単語数	9
表知識に用いられる単語数	284
表知識の列属性名の種類の総数	15

表 2　BLEU スコアおよび Entity F_1

モデル	BLEU($\pm\ \sigma$)	Entity F_1($\pm\ \sigma$)
DF-Net	14.0($\pm$ 0.8)	60.7($\pm$ 1.1)
提案手法	14.4($\pm$ 0.6)	61.1($\pm$ 1.4)

データセットの詳細を表 1 に記す．本実験では，全 3,031 対話のうち 2,425 対話を訓練用データに，302 対話を検証用データ，残りの 304 対話をテスト用データとした．

5.2　評価手法

提案モデルの出力として得られる応答がどれだけ正解の応答と関連しているかを BLEU [8] によって評価する．そして出力される応答がどれだけ対話の内容に沿って適切に表知識を参照できているかを Entity F_1 で評価する．Entity F_1 は，規定の語を応答で使用した割合を全対話で平均することで算出する．本実験では正解の応答で使用している表知識の要素を規定の語とする．

5.3　結果

提案モデルの応答性能を KVRET を用いて BLEU スコアと Entity F_1 で評価する．テストデータにおける各スコアを表 2 に示す．比較のため，基準となる DF-Net の各スコアも示す．Entity F_1 は 0.4% ポイント向上し，上がり幅は 0.7% となった．このこ

表 3 提案手法と DF-Net の応答例:[] 内は表知識から抜き出した要素であり，$_{o}$ と $_{x}$ はそれぞれ表知識を適切に参照できたかどうかを表す.

対話
目標: [hotel_keen, 578_arbol_dr, rest_stop, car_collision_nearby, 3_miles]
直前の発話: what is the rest_stop with the quickest route nearby ?
正解の応答: hotel_keen is a rest_stop located 3_miles away there is a car_collision_nearby but should delay you to long.
DF-Net: [hotel_keen]$_{o}$ is [5_miles]$_{x}$ way.
提案手法: the closest rest_stop is [hotel_keen]$_{o}$, [3_miles]$_{o}$ away.
目標: [Valero, gas_station, road_bloack_nearby, 5_miles]
直前の発話: find me the nearest gas_station.
正解の応答: the nearest gas_station is located 5_miles away need more info ?
DF-Net: the nearest gas_station is [valero]$_{o}$, [200_alester_ave]$_{x}$ away.
提案手法: the nearest gas_station is [valero]$_{o}$ [5_miles]$_{o}$ away.
目標: [panda_express, 3_miles, pizza_hut, 5_miles]
直前の発話: i need directions to a local fast food restaurant please.
正解の応答: there is a panda _express 3 _miles from you as well as a pizza _hut 5 _miles from you.
DF-Net: [round _table]$_{x}$ is [5_miles]$_{o}$ away.
提案手法: the nearest pizza_restaurant is [pizza_hut]$_{o}$, it is [5_miles]$_{o}$ away.

とから，今回作成したモデルは DF-Net よりも表知識を適切に参照できていることが分かった．また，BLEU も 0.4 ポイント向上し，上がり幅は 2.9% となった．これにより，生成する応答全体に適した表知識を参照することで，より正解に近い応答を生成できることが分かった．

5.4 解析

提案手法と DF-Net が生成した応答の例を表 3 に示す．1 つ目の例では，DF-Net が距離を表す表知識の要素の参照を誤ったが，提案手法では正しい要素を参照できていることが分かる．2 つ目の例では，DF-Net が応答の文脈に適さない表知識の要素を参照しているが，提案手法は文脈に適した要素を参照できていることが分かる．これにより，提案手法は応答全体に適した表知識を参照できており，ベースラインより適切に表知識を参照できていることが分かる．しかし，3 つ目の例では，提案手法は誤った知識の参照はしていないが，応答に使用すべき情報が欠けており，応答そのものの生成能力に不十分な点があることが分かる．この問題を解決するには，応答生成時にも表知識を参照し，モデルが出力する応答に含まれるべき表知識を適切に応答に使用するようモデルに学習させる，といったような改善が必要だと考えられる．

6 おわりに

本稿では，応答全体に適した表知識の参照が可能な End-to-End の対話システムの実現を目指して，応答の概形を生成した上で，必要な表知識を参照する手法を提案した．タスク指向対話データセットにおける評価の結果，提案手法は応答全体を利用しないベースラインよりも BLEU スコア・Entity F_1 を向上でき，より正確に表知識を参照できることが明らかになった．今後はモデルのさらなる改善を目指す，また，生成した応答を解析した結果，提案手法が応答を生成する際に応答に十分な情報が含まれない場合があるという課題が見つかったため，この改善も行う予定である．

参考文献

[1] Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. Pomdp-based statistical spoken dialog systems: A review. **Proceedings of the IEEE**, Vol. 101, No. 5, pp. 1160–1179, 2013.

[2] Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. Key-value retrieval networks for task-oriented dialogue. In **Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue**, pp. 37–49, Saarbrücken, Germany, August 2017. Association for Computational Linguistics.

[3] Shiquan Yang, Rui Zhang, and Sarah Erfani. GraphDialog: Integrating graph knowledge into end-to-end task-oriented dialogue systems. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing**

(EMNLP), pp. 1878–1888, Online, November 2020. Association for Computational Linguistics.

[4] Libo Qin, Xiao Xu, Wanxiang Che, Yue Zhang, and Ting Liu. Dynamic fusion network for multi-domain end-to-end task-oriented dialog. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 6344–6354, Online, July 2020. Association for Computational Linguistics.

[5] Mihail Eric and Christopher Manning. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. In **Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers**, pp. 468–473, Valencia, Spain, April 2017. Association for Computational Linguistics.

[6] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. **IEEE transactions on Signal Processing**, Vol. 45, No. 11, pp. 2673–2681, 1997.

[7] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. **arXiv preprint arXiv:1503.08895**, 2015.

[8] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In **Proceedings of the 40th annual meeting of the Association for Computational Linguistics**, pp. 311–318, 2002.