用例ベース対話システムにおける表現集約手法の影響調査

鈴木脩右 [1]　狩野芳伸 [2]　山本和英 [1]

長岡技術科学大学 [1]　静岡大学 [2]

{suzuki,yamamoto}@jnlp.org, kano@inf.shizuoka.ac.jp

1 はじめに

ニューラルベースの対話システムは，大きく分けて生成ベース手法と用例ベース手法がある．生成ベース手法ではEncoder-Decoderで応答を生成し，用例ベース手法では応答候補からユーザ発話に最も適した応答を選択する．用例ベース手法は流暢で意味のある応答が可能となる利点がある．

応答選択に関する初期の研究では、応答を一致させるために対話コンテキスト内の最後の発話のみを使用していた [1]．これをシングルターンの応答選択と呼ぶ．最近の研究では、対話コンテキスト内の全ての発話文を考慮することで，次の発話の選択が容易になることが示されている [2, 3]．これをマルチターンの応答選択と呼ぶ．

マルチターンの応答選択では，対話コンテキストと応答候補の関連性を捉えるための表現集約 (Aggergation) が重要である．Wu らは 2 層ずつの Gated Recurrent Unit(GRU) と Linear 層を用いる手法を提案した [4]．Zhou らは 3 次元の Convolutional Neural Network(CNN) を用いる手法を提案した [5]．Mao らは文情報の差分など様々な観点の情報を Linear 層で集約する Matching Aggregation と，これを Wu らの手法と組み合わせる Hierarchical Aggregation を提案した [6]．

このように様々な手法が提案されているが，集約手法に着目した体系的な比較検討はされていない．そのため，どの手法がより有効な集約を行っているかが不明瞭である．そこで本研究では，より有効な集約手法の明確化を目的とし，同一のフレームワークを用いた各集約手法の比較実験から，その影響を調査する．

2 モデル

2.1 モデル概要

本研究では，Zhou らが提案した Deep Attention Matching Network(DAM) をベースに用いる [5]．モデルの全体図を図 1 に示す．モデルは，Embedding 層，Encoding 層，Attention 層，Aggregation 層，Scoring 層で構成されている．

モデルは対話データ $D = \{(c, r, y)Z\}_{Z=1}^{N}$ を与えられる．$c = \{u_0, ..., u_{n-1}\}$ は対話コンテキスト，$u_i{}_{i=0}^{n-1}$ は各発話文，r は応答文を表す．また，$y \in \{0, 1\}$ はバイナリラベルであり，r が c の適切な応答であるかを示す．

モデルは D とのマッチングスコア $g(c, r)$ を学習する．これにより，コンテキスト c と応答文 r の関連性をスコア付けできる．

2.2 Embedding 層

発話文 $u_i = [w_{u_i,k}]_{k=0}^{n_{u_i}-1}$ と応答文 $r = [w_{r,t}]_{t=0}^{n_r-1}$ を共有された Emedding 層で埋め込みを行う．それぞれ，埋め込まれたベクトルは $\mathbf{U}_i^0 = [e^0_{u_i,0}, ..., e^0_{u_i,n_{u_i}-1}]$ と $\mathbf{R}^0 = [e^0_{r,0}, ..., e^0_{r,n_r-1}]$ で表す．ここで，$e \in \mathbb{R}^d$ は d 次元の単語埋め込みを表している．

2.3 Encoding 層

L 層に積み重ねられた Encoder を用い文表現を獲得する．各 l 番目の層は $l-1$ 番目の出力を受け取り，より洗練された文表現を出力する．獲得した文表現は，Embedding 層の出力も併せて，$[\mathbf{U}_i^l]_{l=0}^{L}$，$[\mathbf{R}_i^l]_{l=0}^{L}$ と表す．l は文表現の粒度を意味する．

Encoder には Transformer [7] を利用した．また，Self Attention のヘッド数は 1，Encoder の層数 L は 5 とした．

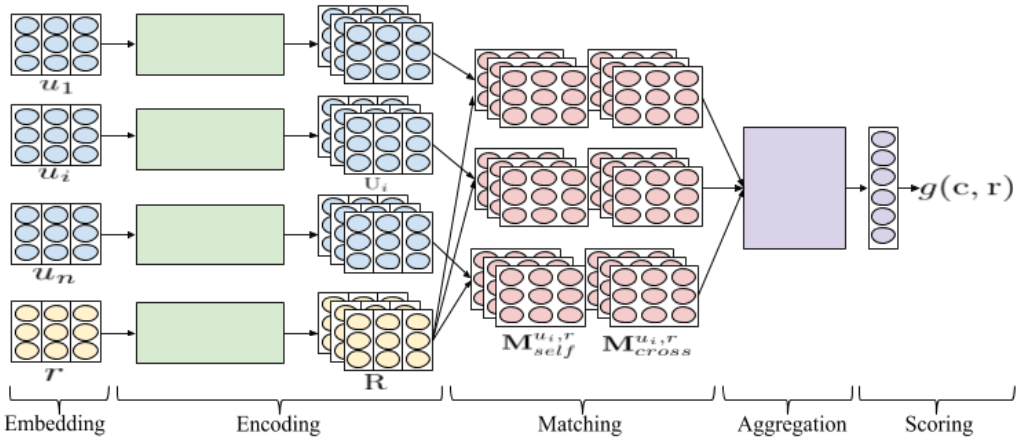

図 1: モデル概要

2.4 Matching 層

$[\mathbf{U}_i^l]_{l=0}^L$ と $[\mathbf{R}^l]_{l=0}^L$ から，各粒度 l 毎に 2 種類の複合表現 $\mathbf{M}_{self}^{u_i,r,l}$，$\mathbf{M}_{cross}^{u_i,r,l}$ を得る．

$\mathbf{M}_{self}^{u_i,r,l}$ は次式で表す．

$$\mathbf{M}_{self}^{u_i,r,l} = \{\mathbf{U}_i^l[k]^T \cdot \mathbf{R}^l[t]\}_{n_{v_i} \times n_r} \quad (1)$$

行列の各要素は $\mathbf{U}_i^l[k]$ と $\mathbf{R}^l[t]$ の内積であり，各粒度毎のテキストの関連度合いを表している．

$\mathbf{M}_{cross}^{u_i,r,l}$ は次式で表す．

$$\tilde{\mathbf{U}}_i^l = Attention(\mathbf{U}_i^l, \mathbf{R}^l, \mathbf{R}^l) \quad (2)$$

$$\tilde{\mathbf{R}}^l = Attention(\mathbf{R}^l, \mathbf{U}_i^l, \mathbf{U}_i^l) \quad (3)$$

$$\mathbf{M}_{cross}^{u_i,r,l} = \{\tilde{\mathbf{U}}_i^l[k]^T \cdot \tilde{\mathbf{R}}^l[t]\}_{n_{v_i} \times n_r} \quad (4)$$

$\tilde{\mathbf{U}}_i^l$ と $\tilde{\mathbf{R}}^l$ はそれぞれ，$\mathbf{U}_i^l$ と $\mathbf{R}^l$ の Attention 重みを掛けたものである．Attention には，Encoding 層と同様の Self Attention を利用した．

2.5 Aggregation 層

Matching 層で獲得した複合表現 $\mathbf{M}_{self}^{u_i,r,l}$，$\mathbf{M}_{cross}^{u_i,r,l}$ を集約する．本研究では 4 つの集約手法を比較する．

- Encoder Aggregation
- CNN Aggregation
- Matching Aggregation
- Hierarchical Aggregation

2.5.1 Encoder Aggregation

これは Wu らが提案した 2 層ずつの GRU と Linear 層を用いる手法である．1 層の Linear 層で複合表現 $\mathbf{M}_{self}^{u_i,r,l}$，$\mathbf{M}_{cross}^{u_i,r,l}$ を集約し，1 層の GRU で集約ベクトルを得る．獲得した集約ベクトルを 2 層目の GRU で Encode することで情報を強化し，2 層目の Linear 層で最終的な集約を行う．式で表すと次の通りである．

$$\mathbb{A}_{i,l} = \mathrm{ReLU}(W \left[\mathbf{M}_{self}^{u_i,r,l} \odot \mathbf{M}_{cross}^{u_i,r,l} \right] + b) \quad (5)$$

$$h_{i,l} = \mathrm{GRU}(\mathbb{A}_{i,l}, h_{i,l-1}) \quad (6)$$

$$h'_l = \mathrm{GRU}(h_0, h'_{i-1}) \quad (7)$$

$$\mathbf{A} = W' [h'_0, ..., h'_n] + b' \quad (8)$$

ここで，$\odot$ はアダマール積である．

2.5.2 CNN Aggregation

これは Zhou らが提案した 3 次元の CNN を用いる手法である．CNN は 2 層ずつの convolution 層と max pooling 層から構成されており，CNN の優れた情報抽出を活かした集約を行う．式で表すと次の通りである．

$$\mathbb{A}_{i,k,t} = [\mathbf{M}_{self}^{u_i,r,l}[k,t]]_{l=0}^L \oplus [\mathbf{M}_{cross}^{u_i,r,l}[k,t]]_{l=0}^L \quad (9)$$

$$\mathbf{A} = \{\mathbb{A}_{i,k,t}\}_{n \times n_{v_i} \times n_r} \quad (10)$$

ここで，$\oplus$ は連結操作である．

2.5.3 Matching Aggregation

Mao らが提案した Linear 層で複数情報を集約する手法である．複合表現 $\mathbf{M}_{self}^{u_i,r,l}$，$\mathbf{M}_{cross}^{u_i,r,l}$，複合表現の

差分，アダマール積，差分同士のアダマール積を用いることで様々な観点からの情報を集約に用いることが可能になる．集約表現から更に密な集約をする必要があるため，本論文では 1 層の Linear 層を用いた．式で表すと次の通りである．

$$\mathbb{A}_{i,l} = \mathrm{ReLU}(W \begin{bmatrix} \mathbf{M}_{self}^{u_i,r,l} \\ \mathbf{M}_{cross}^{u_i,r,l} \\ \mathbf{M}_{self}^{u_i,r,l} - \mathbf{M}_{cross}^{u_i,r,l} \\ (\mathbf{M}_{self}^{u_i,r,l} - \mathbf{M}_{cross}^{u_i,r,l}) \odot (\mathbf{M}_{self}^{u_i,r,l} - \mathbf{M}_{cross}^{u_i,r,l}) \\ \mathbf{M}_{self}^{u_i,r,l} \odot \mathbf{M}_{cross}^{u_i,r,l} \end{bmatrix} + b) \tag{11}$$

$$\mathbf{A} = W' \mathbb{A}_{i,l} + b' \tag{12}$$

2.5.4 Hierarchical Aggregation

Mao らが提案した Mathcing Aggregation と Encoder Aggregation を組み合わせた手法である．Matching Aggregation で集約し，1 層の GRU で集約ベクトルを得る．獲得した集約ベクトルを 2 層目の GRU で Encode し，Linear 層で最終的な集約を行う．式は，式 6-8,11 で表した通りである．

2.6 Scoring 層

集約表現 $\mathbf{A}$ から Linear 層を用いて最終的なスコア $g(c,r)$ を算出する．また，算出したスコアから損失を計算する．本研究では損失関数には Cross Entropy Loss を使用した．これを次式で表す．

$$g(c,r) = \mathrm{sigmoid}(w^* \mathbf{A} + b^*) \tag{13}$$

$$L(\cdot) = -\sum_{(c,r,y)\in D} [y log(g(c,r)) + (1-y) log(1-g(c,r))] \tag{14}$$

3 実験

データセットには Ubuntu Corpus V2.0[8] を使用した．これは Ubuntu フォーラムから収集されたマルチターンの対話データセットである．学習データは，対話コンテキストと応答文，応答の正例・負例を表すバイナリラベルで 1 セットが構成されており，100 万セットが含まれている．正例は実際の人間の応答であり，負例はデータからランダムにサンプリングされたものである．検証データとテストデータは，対話コンテキストと 10 個の応答候補文で 1 セットが構成されており，それぞれ約 19 万セットが含まれている．10 個の応答候補文には正例が 1 個あり，他の応答候補文は全て負例である．

モデルは学習時に対話コンテキストと応答文のマッチングスコアを学習し，検証・テスト時に 10 個の応答候補文をスコア付けし，上位の応答を出力する．

評価指標には Recall@$k = \frac{\sum_{i=1}^{k} y_i}{\sum_{i=1}^{10} y_i}$ を用いる．これは選択された上位 k 個の候補の中に正例が含まれていることを意味し，先行研究でも重要な指標とされている．k の値は先行研究と同様に $k = 1, 2, 5$ とした．

モデルの実装には Allennlp [9] を使用した．ハイパーパラメータはバッチサイズは 32，埋め込み次元数及び Encoder の隠れ層数は 200，最適化には Adam を使用した．比較のため，各モデルにおいて共通するパラメータは Scoring 層の入力次元を除き統一した．これは集約手法によって集約表現 $\mathbf{A}$ の次元数が異なるためである．

表 1: モデル概要

	Recall@1	Recall@2	Recall@5
Encoder	0.108	0.213	0.437
CNN	0.197	0.355	0.715
Matching	0.140	0.284	0.529
Hierarchical	0.098	0.189	0.476

4 結果・考察

実験結果を表 1 に示す．得られた結果から CNN Aggregation が最も優れていた．CNN の優れた情報抽出が表現集約において有効であることを示している．次点で Matching Aggregation が優れており，様々な観点からの情報が集約に有効であることが分かる．これより，2 つの手法を組み合わせることでより優れた集約が行えると考える．

また，GRU を用いた 2 種類の集約手法は精度が低いことも分かった．これは Matching Aggregation と Hierarchical Aggregation の結果より，2 層の GRU に原因があると考える．具体的には 2 点の原因を挙げる．

1. 集約不十分による集約ベクトルの不完全
2. 集約ベクトルの強化による情報の過剰な複雑化

1. の場合，1 層目の GRU に原因がある．複合表現を Linear 層で一度集約を行ってから，GRU で更に集約しベクトルを生成する．Linear 層での集約は情報量が違うが Matching Aggregation でも行われているため，精度低下の要因ではない．そのため，GRU による集約が不十分であるために，不完全な集約ベクトルを生成したと考える．

2. の場合，2 層目の GRU に原因がある．この GRU

によって集約ベクトルの情報を強化できるとされている．しかし，この強化によって情報の過剰な複雑化を起こしているために，集約が困難となり精度が低下していると考える．

これらは 2 層の GRU をそれぞれ用いずに実験を行うことで証明ができる．この実験は今後の課題とする．

5 おわりに

本研究では，用例ベースの対話システムにおける表現集約手法の影響調査を行った．集約手法には，先行研究で提案された Encoder Aggregation，CNN Aggregation，Matching Aggregation，Hierarchical Aggregation の 4 種類を用いた．実験結果より，CNN の情報抽出や，様々な観点からの情報が，表現集約において有効であると分かった．また，2 層の GRU は集約の不十分や過剰な複雑化を引き起こし精度が低くなると考える．今後は，CNN Aggregation と Matching Aggregation を組み合わせた手法の提案や，GRU が引き起こす精度低下の原因特定を行っていきたい．

謝辞

本研究は，平成 29-31 年学術研究助成基金助成金 挑戦的研究（萌芽）課題番号 17K18481 の助成を受けています．

参考文献

[1] Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. A dataset for research on short-text conversations. In *Proc. of EMNLP*, pp. 935–945, 2013.

[2] Xiangyang Zhou, Daxiang Dong, Hua Wu, Shiqi Zhao, Dianhai Yu, Hao Tian, Xuan Liu, and Rui Yan. Multi-view response selection for human-computer conversation. In *Proc. of EMNLP*, pp. 372–381, 2016.

[3] Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proc. of ACL*, pp. 496–505, 2017.

[4] Yu Wu, Wei Wu, Chen Xing, Can Xu, Zhoujun Li, and Ming Zhou. A sequential matching framework for multi-turn response selection in retrieval-based chatbots. *arXiv preprint 1710.11344*, 2017.

[5] Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. Multi-turn response selection for chatbots with deep attention matching network. In *Proc. of ACL*, pp. 1118–1127, 2018.

[6] G. Mao, J. Su, S. Yu, and D. Luo. Multi-turn response selection for chatbots with hierarchical aggregation network of multi-representation. *IEEE Access*, Vol. 7, pp. 111736–111745, 2019.

[7] 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 *NIPS*, pp. 5998–6008. 2017.

[8] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proc. of SIGDIAL*, pp. 285–294, 2015.

[9] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. AllenNLP: A deep semantic natural language processing platform. In *Proc. of Workshop for NLP-OSS*, pp. 1–6, 2018.