# 複数人テキスト会話のためのニューラル応答選択モデル

大内 啓樹
奈良先端科学技術大学院大学
情報科学研究科
ouchi.hiroki.nt6@is.naist.jp

坪井 祐太
日本アイ・ビー・エム株式会社
東京基礎研究所
yutat@jp.ibm.com

## 1 はじめに

テキストでの会話のやりとりを行うソーシャルメディア (Twitter や Reddit など) やインターネットリレーチャット (Ubuntu IRC など) の登場に伴い，テキスト会話のデータが大量に手に入るようになり，データ駆動型の End-to-End 学習によるテキスト対話生成の研究が盛んに行われている [9, 8, 7].

しかし，対話生成は評価が困難であるため，対話生成タスクにつながる前段階的なタスクとして，適切な応答を与えられた選択肢から選択する**応答選択**タスクに関する研究が行われている [4, 2]．図 1 は応答選択タスクの例を示している．各行が，その時刻における発話者 (User) と発話 (Utterance) を示している．最後の行の発話が空欄になっており，この空欄を埋める発話として適切な発話を，選択肢 1 および 2 から選択する．

[4] では，大量のテキスト会話ログから発話者 2 人による会話を抽出し，コーパスを作成して応答選択タスクに取り組んでいるが，実際の会話は 3 人以上で行われる場合も多く，複数人による会話を考慮する必要がある．また，[4] のモデルは，すべての発話者の発話を 1 つの系列として結合し，リカレントニューラルネットワーク (Recurrent Neural Network; RNN) を用いて適切な応答を選択するため，各発話者の発話の一貫性が損なわれてしまう可能性がある．

これらの問題を解決するため，本研究では,「複数人テキスト会話における応答選択タスクのためのデータ作成」と,「発話者を考慮した応答選択モデルの提案」に取り組む．

## 2 複数人テキスト会話コーパス

$M$ 人の発話者から成る会話をコーパスとして収集する．コーパスの作成手順として，前処理をした後，会話部分の抽出を行う．

| User | Utterance |
|---|---|
| davmor2 | daftykins: cycle pfff isn't that what ssh is for |
| daftykins | not to someones house with no Linux systems o0 |
| shauno | windows is getting ssh in 10 ;) |
| daftykins | They also don't run Windows! |
| shauno | It's already in osx? |
| daftykins | [ ] |

1. yeah but i'm not gonna setup SSH on someones MBP
2. they have it on their backlog for next iteration

図 1: 応答選択タスクの例

### 2.1 前処理

Ubuntu IRC Logs[1]から，チャットのログを取得し，⟨ 発話者 ID，発話 ⟩ のペアを作成する．次に，Twitter tokenizer[2][5] を用いて，抽出した発話の単語分割を行う．

### 2.2 会話の抽出

前処理で作成したデータを用いて，$M$ 人による $N$ 発話から成る「文脈 (context)」と 1 発話から成る「応答 (response)」の組を抽出する．

図 2 は抽出アルゴリズムを示している．インプットとして，各行が前処理済みの ⟨ 発話者 ID，発話 ⟩ のペアである配列,「文脈」とする発話数 $N$，発話者数 $M$ を与える．アウトプットとして，⟨ 文脈，応答 ⟩ のペアの集合が得られる．

図 2 の 3，4 行目で，$t-N$ 時刻から $t-1$ 時刻の発話を「文脈」，$t$ 時刻目の発話を「応答」とする．5，6 行目では，関数 Agent(·) を用いて，各発話の発話者 ID の集合を取得している．7，8 行目で，応答の発話者が「文脈」に登場しており，かつ，「文脈」に登場する発話者が $M$ 人だった場合に，⟨ 文脈，応答 ⟩ のペアをサンプルとして抽出する．

[1] http://irclogs.ubuntu.com/
[2] http://www.cs.cmu.edu/ ark/TweetNLP/

**Input:** Texts (Array: Each row is a tuple),
$N$ (The number of previous utterances),
$M$ (The number of agents)
**Output:** Samples

Samples $\leftarrow \emptyset$
**for** $t = N$ to |Texts| **do**
  context $\leftarrow$ Texts$[t-N:t]$
  response $\leftarrow$ Texts$[t]$
  agents $\leftarrow$ Agent(context)
  agent $\leftarrow$ Agent(response)
  **if** agent $\in$ agents & |agents| $= M$ **then**
    Samples $\leftarrow$ Samples $\cup$ (context, response)
  **end if**
**end for**
**return** Samples

図 2: コーパス作成アルゴリズム

次に，応答選択タスクのため，抽出したサンプルを訓練・開発・評価データセットに分け,「応答候補」を作成する．抽出したサンプルの「応答」を正例とし，各データセット内からランダムに抽出した「応答」を負例とする．それら正例と負例の「応答」を「応答候補集合」とする．

# 3 ニューラル応答選択モデル

## 3.1 応答選択タスクの定式化

文脈 $c$ が与えられた場合，次の応答として適切な応答 $\hat{r}$ を，応答候補集合 $R$ から予測する．

$$\tilde{r} = \underset{r \in R}{\operatorname{argmax}}\ Pr(d=1|c,r)$$

上式において，変数 $d \in \{0,1\}$ は 2 値変数であり，$d=1$ は文脈 $c$ に続く応答として正しい応答，$d=0$ は誤った応答を表す．したがって，$Pr(d=1|c,r)$ は，応答候補 $r$ が適切である確率を表し，応答候補集合 $R$ における確率最大の応答候補 $\tilde{r}$ を予測結果とする．

## 3.2 既存モデル

図 3 は，[4] で提案されたモデルを示しており，確率 $Pr(d=1|c,r)$ を以下のようにモデル化している．

$$Pr(d=1|c,r) = y_\theta(c,r) = \sigma\left(\mathbf{h}_c^{\mathrm{T}}\ \mathbf{W}\ \mathbf{h}_r\right) \quad (1)$$

ここで，$\sigma$ はロジスティックシグモイド関数，$\mathbf{h}_c \in \mathbb{R}^d$ は文脈 $c$ に対応するベクトル，$\mathbf{h}_r \in \mathbb{R}^d$ は応答候補 $r$ に対応するベクトル，$\mathbf{W}$ はパラメータ $\mathbb{R}^{d\times d}$ を表す．

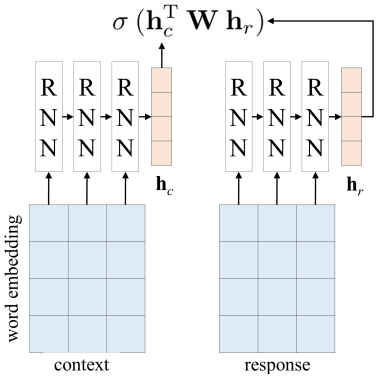


図 3: Lowe 他 (2015) のモデル

また，複数発話を結合し，$I$ 単語から成る 1 つの系列を文脈 $c = \{\mathbf{c}_i\}_1^I$ としている．$J$ 単語から成る応答候補は，$r = \{\mathbf{r}_j\}_1^J$ と定義される．$\mathbf{h}_c$，$\mathbf{h}_r$ は，RNN によってもとめられる．

$$g(\mathbf{x}_l) = f(\mathbf{W}_h \cdot g(\mathbf{x}_{l-1}) + \mathbf{W}_x \cdot \mathbf{x}_l) \quad (2)$$

上式において，各単語ベクトル $\mathbf{x}_l \in \mathbb{R}^d$ を用いて，$l$ 番目の状態ベクトル $g(\mathbf{x}_l)$ を再帰的に計算する．したがって，文脈 $c$ と応答候補 $r$ の各単語ベクトル $\mathbf{c}_i, \mathbf{r}_j \in \mathbb{R}^d$ を用いて，文脈ベクトル $\mathbf{h}_c = g(\mathbf{c}_I)$ と応答候補ベクトル $\mathbf{h}_r = g(\mathbf{r}_J)$ を計算する．(2) 式の関数 $f(\cdot)$ は任意の非線形関数であり，本研究では双曲線正接関数 $(tanh(\cdot))$ を用いる．もとめられた文脈ベクトル $\mathbf{h}_c$ と応答候補ベクトル $\mathbf{h}_r$ を用い，(1) 式で応答候補が適切である確率をもとめる．

このモデルは，複数人発話を 1 つの系列として結合して，文脈ベクトル $\mathbf{h}_c$ の計算に用いているため，発話者の以前の発話と関連のない応答を選んでしまう恐れがある．次節では，各人の発話の一貫性をモデル化するために発話者の状態を明示的に表現する手法を提案する．

## 3.3 提案モデル

図 4 は提案モデル (**発話者エンコーディングモデル**) を示している．これは，各時刻における「発話者の状態」をベクトルで表し，文脈ベクトル $\mathbf{h}_c$ を計算するのに用いている．

時刻 $n \in \{1,2,\cdots,N\}$ における発話者 $m \in \{1,2,\cdots,M\}$ の発話に対応するベクトル表現 $\mathbf{u}_{n,m}$ を，次のように計算する．

$$\mathbf{u}_{n,m} = g(\mathbf{c}_{n,m,L})$$

上式において，$L$ 個の単語から構成された発話 $c_{n,m} = \{\mathbf{c}_{n,m,l}\}_1^L$ の各単語 $\mathbf{c}_{n,m,l} \in \mathbb{R}^d$ を (2) 式 $g(\cdot)$ により計

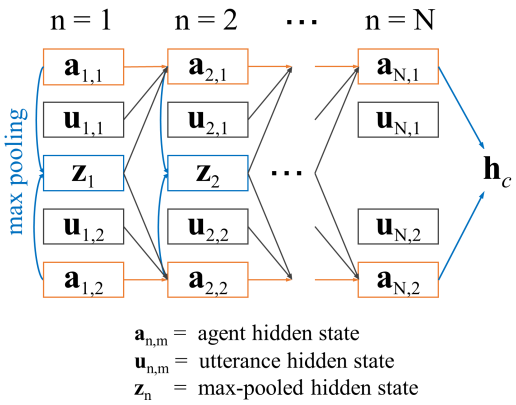


図 4: 発話者エンコーディングモデル: 2 人の例

算している．次に，時刻 $n$ における発話者 $m$ の状態を表すベクトル $\mathbf{a}_{n,m}$ を計算する．

$$\mathbf{a}_{n,m} = f(\mathbf{W}_a \cdot [\mathbf{u}_{n,m}, \mathbf{a}_{n-1,m}, \mathbf{z}_{n-1}])$$

$$\mathbf{z}_n = \max_{m \in \{1,2,\cdots,M\}} \mathbf{a}_{n,m}$$

ここで，発話者全員の状態ベクトルに対して max-pooling を行ったベクトル $\mathbf{z}_n$ と，発話ベクトル $\mathbf{u}_{n,m}$，1 時刻前の発話者状態ベクトル $\mathbf{a}_{n-1,m}$ を結合し，発話者状態ベクトル $\mathbf{a}_{n,m}$ を計算する．関数 $f(\cdot)$ は (2) 式と同様に任意の非線形関数であり，本研究では双曲線正接関数 ($tanh(\cdot)$) を用いる．発話者全員の状態ベクトルに対して max-pooling を行って計算されたベクトルを，各発話者の状態ベクトル更新に利用することによって，他の発話者の状態も考慮した更新が行われる．これは，ある発話者の状態は，他の発話者からも影響を受けて変化しうるという直感をモデリングしたものである．最後に，$N$ 時刻における全発話者の状態ベクトルに対して max-pooling を行い，文脈ベクトル $\mathbf{h}_c$ を得る．

$$\mathbf{h}_c = \max_{m \in \{1,2,\cdots,M\}} \mathbf{a}_{N,m}$$

得られた文脈ベクトル $\mathbf{h}_c$ と，既存モデルと同様にもとめた応答候補ベクトル $\mathbf{h}_r$ を用いて (1) 式を計算し，応答候補 $r$ の確率をもとめる．

## 3.4 モデルの訓練

交差エントロピー誤差関数を最小化することによって，モデルを訓練する．

$$E(\theta) = -\sum_n [d_n \log y_\theta(c_n, r_n) + (1 - d_n) \log (1 - y_\theta(c_n, r_n))] + \frac{\lambda}{2}||\theta||^2$$

表 1: 複数人テキスト会話コーパスのサンプル数

| | 発話者数 | サンプル数 | 単語数 | 単語/発話 |
|---|---|---|---|---|
| 訓練 | 2 | 41,769 | 4,3 M | 9.57 |
| | 3 | 52,938 | 5,9 M | 10.15 |
| | 4 | 42,296 | 5,0 M | 10.84 |
| 開発 | 2 | 2,457 | 258,4 k | 9.56 |
| | 3 | 3,114 | 346,9 k | 10.13 |
| | 4 | 2,488 | 298,5 k | 10.91 |
| 評価 | 2 | 4,919 | 524,8 k | 9.70 |
| | 3 | 6,246 | 687,4 k | 10.00 |
| | 4 | 4,978 | 592,4 k | 10.82 |

[1] のモデルにおいて訓練を行うパラメータは，$\theta = \{\mathbf{W}_x, \mathbf{W}_h, \mathbf{W}\}$ であり，提案モデルは，$\theta = \{\mathbf{W}_x, \mathbf{W}_h, \mathbf{W}, \mathbf{W}_a\}$ である．訓練に関連するハイパーパラメータなどの詳細は，4.1.2 節で述べる．

# 4 実験

提案モデルの効果を調べるため，発話者数の異なる 3 つのコーパスを作成し実験を行った．以下では，実験設定，および実験結果について述べる．

## 4.1 実験設定

### 4.1.1 データ

2015 年 7 月中の Ubuntu IRC Logs のデータを用いた．文脈とする発話数を $N = 10$，発話者数を $M = 2, 3, 4$，応答候補数を 2(正例=1，負例=1) に設定し，3 つのコーパスを作成して評価実験を行った．作成した各コーパスをシャッフルし，全体の 85%を訓練，5%を開発，10%を評価データセットとした．表 1 に各コーパスの抽出したサンプル数 (文脈と応答のペアの数)，単語数，発話平均単語数を示す．実際の訓練では計算速度を考慮し，これらの訓練サンプルの中から，各発話が 50 単語以下で構成されているサンプルのみを用いた．開発・評価データはすべてのサンプルを用いた．

### 4.1.2 訓練詳細

パラメータの最適化は，ミニバッチ (バッチサイズ = 32) を利用した確率的勾配降下法 (SGD) で行った．学習係数は Adam[3] を用いて自動調整し，エポック数は 100 に設定した．各ハイパーパラメータは，[4] の実験設定を参考に以下のように選んだ．

**単語ベクトル** ： GloVe[6] によって事前に学習された 300 次元のベクトル[3]を用いた．

[3] http://nlp.stanford.edu/projects/glove/

表 2: 応答選択タスクの予測結果: 正解率

| 発話者数 | データ | 既存モデル [4] | 提案モデル |
|---|---|---|---|
| 2 | 開発 | 75.34 | 78.14 |
| | 評価 | 73.92 | 77.17 |
| 3 | 開発 | 69.97 | 76.23 |
| | 評価 | 71.42 | 76.00 |
| 4 | 開発 | 70.90 | 72.07 |
| | 評価 | 68.98 | 70.75 |

**パラメータ次元数** ：単語ベクトルに対するパラメータ $\mathbf{W}_x$ は $50 \times 300$，発話者の状態に対するパラメータ $\mathbf{W}_a$ は $50 \times 150$，その他のパラメータ $\mathbf{W}$，$\mathbf{W}_h$ は $50 \times 50$ 次元に設定．

**パラメータ初期値** ：[1] で提案されているパラメータ初期化法を採用した．これは，$n_j \times n_{j+1}$ のパラメータ行列の値を $\left[-\frac{\sqrt{6}}{\sqrt{n_j+n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j+n_{j+1}}}\right]$ から一様分布に従ってサンプリングする．

**正則化項** ：正則化項のハイパーパラメータ $\lambda$ は [0.01, 0.005, 0.001, 0.0005, 0.0001] の中から開発データの正解率が最大となるものを選んだ．

### 4.2 実験結果

表 2 に予測結果の正解率を示す．発話者数がいずれの場合も，提案モデルが既存モデルを上回った．特に発話者が 3 人の場合に両者の差は 4.5 ポイント程度に広がり，提案モデルの有効性が示唆された．また，どちらのモデルも，発話者数が多くなると正解率が下がっている．これは，会話への参加者が多い場合，応答可能な人数も増え，許容可能な応答が多くなることによって，適切な応答を選ぶことが困難になるためであると考えられる．

## 5 おわりに

本研究では，複数人テキスト会話における応答選択のためのデータ作成と，発話者の状態を考慮したモデルの提案に取り組んだ．評価実験では，提案モデルが既存モデルを上回る性能を達成した．また，発話者の人数が多い会話のほうが，適切な応答を選ぶことが困難であることが分かった．今後の課題として，発話者の人数を限定しない会話での応答選択タスクに取り組む．提案モデルは発話者の人数が可変の発話にも対応可能であるため，より現実的な会話における応答選択の性能を評価する．

## 参考文献


[1] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of International conference on artificial intelligence and statistics*, pages 249–256, 2010.

[2] Rudolf Kadlec, Martin Schmid, and Jan Kleindiest. Improved deep learning baselines for ubuntu corpus dialogs. *arXiv preprint arXiv: 1510.03753*, 2015.

[3] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv: 1412.6980*, 2014.

[4] 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 *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294. Association for Computational Linguistics, 2015.

[5] Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL/HLT*, pages 380–390. Association for Computational Linguistics, 2013.

[6] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543. Association for Computational Linguistics, 2014.

[7] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. *arXiv preprint arXiv: 1507.04808*, 2015.

[8] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of NAACL/HLT*, pages 196–205. Association for Computational Linguistics, 2015.

[9] Oriol Vinyals and Quoc V. Le. A neural conversational model. In *Proceedings of International Conference on Machine Learning*, 2015.