トピック情報を考慮した用例ベース対話システム

高橋 拓誠　目良 和也　黒澤 義明　竹澤 寿幸
広島市立大学大学院 情報科学研究科
takahashi@ls.info.hiroshima-cu.ac.jp
{mera, kurosawa, takezawa}@hiroshima-cu.ac.jp

1 はじめに

人間との雑談を目的とした対話システムが，生活の中に浸透しつつある．このようなシステムは，入力発話に対する返答をルール形式で記述するルールベース手法 [1] と Twitter などから抽出した大量の応答候補を用いて統計的に応答を決定する統計的応答手法 [2] の 2 種類が代表的である．また，実際の対話事例から構成される用例ベースを用いて，ユーザ発話に最も近い用例の応答を出力する用例ベース手法 [3] と呼ばれるものもある．しかしながら，いずれの手法においてもユーザの発話に対して不自然な応答であったり，対話の継続が困難となるような対話破綻をきたした応答がしばしば行われる．

このような問題を解決するために，対話破綻検出チャレンジという評価型ワークショプが行われている [4]．対話破綻検出チャレンジでは，人間とシステムの会話の対話破綻を自動的に検出することを目標としている．このワークショップの中で船越らは対話破綻の種類について類型化を行っており，ユーザ発話に対する応答が話題や意図を無視していることが対話破綻の大きな要因の 1 つであることを報告している [5]．

本研究では，対話中における話題に着目して用例ベース手法を適用することで，より対話破綻の少ない対話システムの実装を目指す．具体的には，対話の文脈や用例に対してテキスト分類のための機械学習手法を適用することで，トピックを表すベクトルを得る．応答選択では，ユーザ発話と用例に対する文間類似度および対話の文脈と用例に対するトピック間類似度に基づく応答選択手法を提案する．

2 トピックを考慮した用例ベース対話システム

2.1 提案手法の概要

一般的な用例ベース対話システムは，入力発話 u に対して用例データベース中の適当なクエリ発話 q_i を決定することで，応答発話 $\hat{r}_i$ を選択できる．すなわち，下記の式 (1) の関数 f によって，入力発話 u に対する適当なクエリ発話 q_i と応答発話 $\hat{r}_i$ の検索が可能となる．

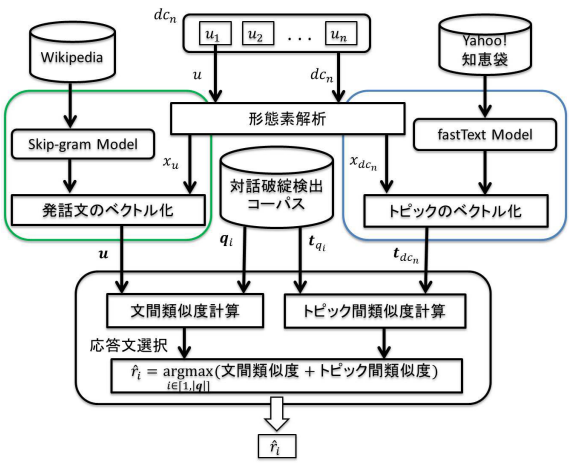

図 1: 提案手法の概要

$$\hat{r}_i = \underset{i\in[1,|\mathbf{q}|]}{\arg\max}\, f(u, q_i) \tag{1}$$

本研究では，対話中のトピック情報を利用することでより対話破綻の少ない用例ベース対話システムを提案する．式 (1) における関数 f は，入力発話 u に対する q_i の文間類似度とトピック間類似度を適用することで応答選択を行う．

提案手法の概要を図 1 に示す．以降では，2.3 節で文間類似度，2.4 節で用例データベース中の破綻発話のフィルタリング，2.5 節でトピック間類似度，2.6 節で応答選択の方法について述べる．

2.2 用例データベースの構築

用例ベース対話システムは，入力発話 u に対して適当な応答 $\hat{r}_i$ を用例データベース中から検索することで決定するシステムである．一般的に，用例データベースは対話履歴などからクエリ発話 q_i とそれに対する応答 r_i のペアを用例 $< q_i, r_i >$ として構築する．

本研究では，対話破綻検出チャレンジ 2 の開発用および評価用データ [4] を用例データベースとして使用

する．対話破綻検出コーパスは，10 個のユーザ発話と 11 個のシステム発話を 1 対話として構成しており，1 つのデータベースにつき 50 対話収録されている．各システム発話は，直前までの対話の履歴 (以降，対話文脈と呼ぶ) に対して 30 人の被験者により 3 件法 (◯: 破綻はなく自然な発話である，△: 破綻とは言い切れないが違和感がある，×: 破綻している) で対話破綻の有無についてアノテーションされている．

本稿では，対話破綻検出コーパス中のユーザ発話 3000 個をクエリ発話の系列 $\mathbf{q} = \{q_1, q_2, ..., q_{3000}\}$ として，各 q_i の応答であるシステム発話を応答発話の系列 $\mathbf{r} = \{r_1, r_2, ..., r_{3000}\}$ として構築される用例 $< \mathbf{q}, \mathbf{r} >$ を用例データベースとして構築した．さらにコーパス中から抽出した応答発話 r_i に対して，対話破綻の有無に関してアノテーションされた評価 $a_{r_i} = (n_o, n_t, n_x)$ を付与する．なお，n_o, n_t, n_x はそれぞれ $\bigcirc, \triangle, \times$ をアノテーションした人数である．

2.3 応答選択のための文間類似度

本稿では，単語をベクトル化し，発話に含まれる全単語に対して重心ベクトルを求めることにより発話をベクトル化する．単語のベクトル化には，Mikolov らが提案した Skip-gram モデルを用いて単語ベクトル集合 $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_V)$ を生成した [6]．

学習した単語ベクトル $\mathbf{w}_j$ を用いて，入力発話 u のベクトル $\mathbf{u}$ とクエリ発話 q_i のベクトル $\mathbf{q}_i$ を式 (2), (3), (4) に従い計算する．なお，N は u に含まれる単語数，M は q_i に含まれる単語数を表す．

$$\mathbf{u} = \frac{1}{N} \sum_{\mathbf{w}_j \in u} \mathbf{w}_j \tag{2}$$

$$\mathbf{q}_i = \frac{1}{M} \sum_{\mathbf{w}_j \in q_i} \mathbf{w}_j \tag{3}$$

$$\mathbf{w}_j = \begin{cases} \mathbf{w}_j & \mathbf{w}_j \in \mathbf{W} \\ 0 & \mathbf{w}_j \notin \mathbf{W} \end{cases} \tag{4}$$

以上より計算された発話ベクトルを用いて，入力発話 u とクエリ発話 q_i の類似度は，$cos(\mathbf{u}, \mathbf{q}_i)$ により計算される．ここで，$cos(\mathbf{u}, \mathbf{q_i})$ はコサイン類似度を表す．

2.4 破綻発話のフィルタリング

本研究で使用する対話破綻検出コーパスは，システムの破綻発話を含むコーパスである．そこで，破綻発話のフィルタリングを行うために，対話破綻コーパスにアノテーションされた対話破綻の有無に基づき，応答候補 r_i に対して重み付けを行う．クエリ発話 q_i に対するシステム応答 r_i の重み付けのスコア $score_{r_i}$ は，文献 [2] を参考に式 (5)，(6) に基づき行った．

$$score_{r_i} = s_o \frac{n_o}{N} + s_t \frac{n_t}{N} + s_x \frac{n_x}{N} \tag{5}$$

$$N = n_o + n_t + n_x \tag{6}$$

式 (5) における n_o, n_t, n_x はそれぞれ $\bigcirc, \triangle, \times$ を評価した人数に対応する．また，s_o, s_t, s_x は $\bigcirc, \triangle, \times$ それぞれの重みである．本研究では，$(s_o, s_t, s_x) = (1.0, 0.5, -1.0)$ とした．

2.5 応答選択のためのトピック間類似度

対話中のトピックを考慮して応答選択を行うために，発話数 n の対話文脈 dc_n に対してトピック推定を行う．トピック推定は，テキスト分類の機械学習手法である fastText[7] を用いて行う．

学習データは，Yahoo!知恵袋のデータセットを使用する．Yahoo!知恵袋は，すべての質問と回答が必ず 1 つのカテゴリに所属する．この時，カテゴリ集合 $C = \{c_1, c_2, ..., c_N\}$ を正解ラベルとして学習を行う．

学習された fastText のモデルを用いることで，発話数 n の対話文脈 dc_n に対してトピックベクトル $\mathbf{t}_{dc_n} = (p_1, p_2, ..., p_N)$ を得ることができる．なお，p_i は対話文脈 dc_n がクラス c_i に所属する確率，N はクラス数を表す．同様に，クエリ発話 q_i に対してトピックベクトル $\mathbf{t}_{q_i} = (p_1, p_2, ..., p_N)$ を得ることができる．以上より，対話文脈 dc_n とクエリ発話 q_i のトピック間類似度は $cos(\mathbf{t}_{dc_n}, \mathbf{t}_{q_i})$ によって計算される．

2.6 文間類似度とトピック間類似度に基づく応答選択

入力発話 u を含む発話数 n の対話文脈 dc_n に対して，適当なシステム応答 $\hat{r_i}$ を式 (7) に従って決定する．ここで，λ は文間類似度とトピック間類似度の重要度を調整するための $[0, 1]$ の実数値である．

$$\begin{aligned} \hat{r_i} = \underset{i \in [1, |\mathbf{q}|]}{\arg\max} (&\lambda (cos(\mathbf{u}, \mathbf{q}_i) * score_{r_i}) \\ &+ (1 - \lambda)(cos(\mathbf{t}_{dc_n}, \mathbf{t}_{q_i}))) \end{aligned} \tag{7}$$

3 評価実験

本実験では，対話破綻検出コーパスの中にある対話文脈を入力として対話システムに与え，出力される応答候補の上位 10 件が対話文脈に対して妥当であるかを評価する．評価用データセットは，対話破綻検出コーパス中の対話文脈から対話破綻のない発話が 5 発話連続している対話文脈を 10 個用意した (評価用データ:

ts_1〜ts_{10})．なお，対話破綻に関するアノテーションが $n_x < 10$ の発話を対話破綻のない発話と定義し，ts_i を評価する際には ts_i を応答選択のための用例データベースから除外した．

3.1 実験設定

本実験で比較する手法は以下の通りである．各手法は応答選択式 (式 (7)) のパラメータ n と λ を調整することで実現する．n はトピック間類似度の計算で考慮する対話文脈の発話数，λ は文間類似度とトピック間類似度の重要度を調整するパラメータである．また，CNT は文間類似度 (2.3 節)，SCR は破綻発話フィルタリングのスコア (式 (5))，TPC はトピック間類似度 (2.5 節) を考慮した対話システムであることを表す．

- CNT(Baseline): 文間類似度のみ考慮する手法 ($\lambda = 1$, $score_{r_i} = 1$)
- CNT+SCR: 文間類似度とスコアを考慮する手法 ($\lambda = 1$)
- CNT+SCR+TPC(1): 文間類似度とスコアとトピックを考慮する提案手法 ($\lambda = 0.5, n = 1$)
- CNT+SCR+TPC(3): 文間類似度とスコアとトピックを考慮する提案手法 ($\lambda = 0.5, n = 3$)
- CNT+SCR+TPC(5): 文間類似度とスコアとトピックを考慮する提案手法 ($\lambda = 0.5, n = 5$)

なお，実験に際して，文間類似度に用いる単語ベクトルは，次元数 100，ウィンドウサイズ 5，最小出現頻度は 5 に設定して学習を行った．また，トピック間類似度に用いるトピックベクトルは，次元数 290，各クラスタには Yahoo!知恵袋の質問とそれに対する回答を 1 文書として 10,000 文書用意した．

3.2 評価指標

本実験の評価対象は，10 個の対話文脈 (ts_1〜ts_{10}) に対して各システムが選択した上位 10 件の応答候補である．5 つの対話システムの出力を比較するため，合計 500 個 (異なり発話数: 365 個) の応答を評価する．応答候補の評価は，下記の項目について被験者 6 名 (男性 4 名，女性 2 名，22〜27 歳) に依頼した．

- 対話破綻の有無: ◯: 破綻していない，△: 破綻とは言い切れないが違和感がある，×: 破綻している
- 話題の適合性: ◯: 適切な話題である，△: 不適切な話題とは言い切れないが違和感がある，×: 不適切な話題である

上記の評価項目について，nDCG(normalized Discounted Cumulated Gain)[8] を用いて各システムを評価する．第 k 位までの順位付けの評価を行う場合，式 (8)，(9)，(10) 従い計算する．

$$R_i = g_o \frac{n_o}{N} + g_t \frac{n_t}{N} + g_x \frac{n_x}{N} \tag{8}$$

$$DCG_k = R_1 + \sum_{i=2}^{k} \frac{R_i}{\log_2 i} \tag{9}$$

$$nDCG_k = \frac{DCG_k}{iDCG_k} \tag{10}$$

上記の式において，k は順位付けを行う最大数，R_i は第 i 位の応答の関連度，DCG_k は第 k 位までにおいて提案された順位付けの正しさの指標，$iDCG_k$ は第 k 位までにおいて提案される順位付けの最大得点を表す．式 (10) において，n_o は ◯ と評価したアノテータの人数，n_t は △ と評価したアノテータの人数，n_x は × と評価したアノテータの人数を表し，g_o, g_t, g_x は ◯, △, × それぞれの得点を表す．本実験では，$(g_o, g_t, g_x) = (2, 1, 0)$ とした．また，N はアノテータ全体の人数 ($N = 6$) を表す．

表 1 に実験で使用した対話文脈の一例を，表 2 に表 1 の対話文脈に対して各システムにより 1 位に順位付けされた応答候補を示した．ここで，関連度は式 (8) によって計算した値である．

3.3 実験結果

対話破綻の有無に関する nDCG による評価結果

対話破綻の有無に関する各システムの評価結果を図 2 に示す．図 2 は，全応答候補に対する nDCG の評価の平均値を表している．なお，図中の“$*$”は有意水準 5%，“$**$”は有意水準 1%の t 検定を行い有意差のあった部分を示している．

図 2 より，トピックを考慮しないシステム (CNT, CNT+SCR) と比較してすべての提案手法が良い結果を示していることが分かる．しかし，本研究ではトピックベクトルを生成する際にトピックの切り替わりまでは考慮できていない．そのため，直前のユーザ発話 ($n = 1$) のトピックのみ考慮する提案手法が最も良い結果を示したと考えられる．

話題の適合性に関する nDCG による評価結果

話題の適合性に関する各システムの評価結果を図 3 に示す．図 3 より，トピックを考慮しないシステム (CNT，CNT+SCR) と比較してすべての提案手法が良い結果を示していることが分かる．さらに 2 つの評価項目における nDCG の評価結果に対してピアソ

表 1: 実験で用いた対話文脈の例

話者	発話内容
U	その地域のお祭りなどにも行きたいですね。
S	お祭りに行きたいんですね？お祭りは夜が楽しいですよねえ
U	私は特に打ち上げ花火が見たいです。
S	打ち上げ花火は見たいんですね？お祭りは楽しいですね
U	あなたも一緒にお祭りに行きませんか？
S	(このタイミングで生成される応答候補を評価する)

表 2: 比較手法における 1 位の応答候補と関連度

比較手法	応答候補	関連度
CNT(Baseline)	私が旭川市の旭山動物園の中で気に入っている場所は、やきとり弁当です。	0.00
CNT+SCR	フリーターです	0.00
CNT+SCR+TPC(1)	盆踊りで盛り上がりますよねえ	1.50
CNT+SCR+TPC(3)	お祭りに行きたいんですね？お祭りは行くのがいいですよねえ	1.00
CNT+SCR+TPC(5)	雰囲気が楽しいですね	0.83

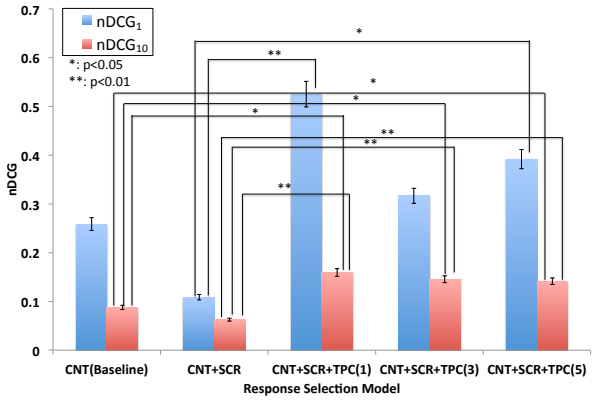

図 2: 対話破綻の有無に関する nDCG による評価結果

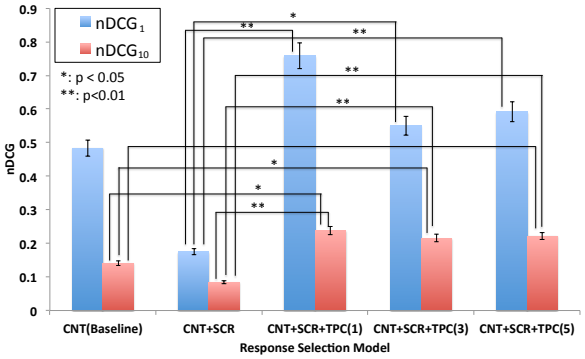

図 3: 話題の適合性に関する nDCG による評価結果

ンの相関係数を求めたところ，すべての手法において 0.95 以上の強い相関関係が確認できた．これは，対話破綻の有無が返答に適切な話題が含まれるかどうかということに強く依存していることを表している．

4 おわりに

本研究では，入力発話とクエリ発話の文間類似度および対話文脈とクエリ発話のトピック間類似度に基づき応答選択することで，より対話破綻の少ない用例ベース対話システムを実装した．トピック間類似度の計算に用いるトピックベクトルは，fastText を用いることで生成した．実験で $nDCG_{10}$ による評価結果に対して有意水準 5%の t 検定を行ったところ，対話中のトピック情報を考慮しない手法と比べて，提案手法すべてに有意差が確認できた．

今後は，トピックベクトルを生成する際に話題の切り替わりを認識できるような枠組みを検討する予定である．

参考文献

[1] Richard S Wallace. *The Anatomy of A.L.I.C.E.* Springer, 2009.

[2] Michimasa Inaba and Kenichi Takahashi. Neural Utterance Ranking Model for Conversational Dialogue Systems. In *SIGDIAL*, pp. 393–403, 2016.

[3] 水上雅博, Lasguido Nio, 木付英士, 野村敏男, Graham Neubig, 吉野幸一郎, Sakriani Sakti, 戸田智基, 中村哲. 快適度推定に基づく用例ベース対話システム. 人工知能学会論文誌, Vol. 31, No. 1, 2016.

[4] 東中竜一郎, 船越孝太郎, 稲葉通将, 荒瀬由紀, 角森唯子. 対話破綻検出チャレンジ 2. 第 7 回対話システムシンポジウム, pp. 64–69, 2016.

[5] 船越孝太郎, 東中竜一郎, 稲葉道将, 小林優佳, 菅原朔, 高梨克也, 大塚裕子, 小磯花絵, 坊農真弓. 対話破綻検出チャレンジにおける対話破綻データと破綻検出の分析. 言語処理学会第 22 回年次大会, pp. 433–436, 2016.

[6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representation in vector space. In *International Conference on Learning Representations (ICLR)*, 2013.

[7] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[8] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, Vol. 20, No. 4, pp. 422–446, 2002.