

対話破綻検出による対話システムの応答性能の向上

Improvement of Response Performance of Chat-oriented Dialogue Systems by Using Dialogue Breakdown Detection

稲葉 通将^{1*} 高橋 健一¹
Michimasa INABA¹ Kenichi TAKAHASHI¹

¹ 広島市立大学大学院情報科学研究科
¹ Graduate School of Information Sciences, Hiroshima City University

Abstract: In this paper, we experimentally show that the performance of chat-oriented dialogue systems can be improved by applying dialogue breakdown detection technique. This paper proposes three application method of this technique: classification based method, non-breakdown probability based method and probability distribution based method. The experimental results indicate that the systems' performance is improved by using dialogue breakdown detection.

1 はじめに

対話破綻検出は人と対話システムの対話において、システムの不適切な発話を発見する技術である。2015年と2016年にその競技会である対話破綻検出チャレンジ(DBDC)[1]および対話破綻検出チャレンジ(DBDC2)[2]が開催され、2017年にはDialog System Technology Challenges (DSTC6)のDialogue Breakdown Detectionトラックとして3回目の競技会が開催予定¹である。

対話破綻検出技術により、システムが自分の発話をする前に、その発話が対話破綻に繋がることが分かり、対話破綻を事前に回避することが可能となるとされている[1]。しかし、実際にまだそれを検証した例は存在しない。また、その技術を具体的にどのように適用すればより効果的に対話破綻を回避できるかということも明らかではない。

そこで本論文では、複数の対話システムの応答に対し対話破綻検出を適用し、実際に対話破綻を回避することで性能向上が可能であるかを検証する。そのため、破綻検出の適用手法を複数提案し、どのような適用手法が有効であるかの実験も行う。

2 関連研究

対話破綻検出を対話システムの応答選択に適用した研究は少数であるが存在する。Sugiyamaは対話破綻検出を用例ベースの非タスク指向型対話システムに応用する

手法を提案している[3]。本手法では、まずWord2Vecを用いて入力発話と類似した発話をデータベースから検索し、上位20件の応答候補を得る。次に、対話破綻検出手法により出力された破綻確率により応答候補をランキングし、1位の応答候補を出力するというものである。また、Moriらは複数の応答生成モジュールが生成した応答候補から、応答候補を選択する基準の1つとして対話破綻確率を用いたシステムを提案した[4]。しかし、これらの研究では、対話破綻検出を使用しない場合の評価は行っておらず、また、どのように適用すればより性能が向上するかという評価も行っていない。

3 対話システム

本研究では、対話破綻検出による対話システムの応答性能の変化を分析するため、システムは入力に対し、複数の応答候補が出力できるものを用いる。これは、多くのルールベースの対話システムのように、入力に対する応答が固定されているような対話システムでは、その応答が破綻と判定された際に別の応答ができず、性能の変化の分析が困難なためである。また、複数の応答候補が出力できる対話システムの場合、システムは各応答候補にスコアを付与し、そのスコアに基づき応答を順位付けすることが多い。そこで本研究ではこの順位、もしくはスコアそのものを対話破綻検出を適用することにより変更することで、応答候補のランキングを行い、性能の変化を分析する。

本研究では、分析のため以下の3種類の非タスク指向型対話システムを用いる。

*連絡先：広島市立大学大学院情報科学研究科
〒731-3194 広島市安佐南区大塚東 3-4-1
E-mail: inaba@hiroshima-cu.ac.jp

¹<https://dbd-challenge.github.io/dbdc3/>

3.1 用例ベース (IRS)

用例ベースの対話システムとして、DBDC2 で使用された IRS システムを用いる。IRS システムは IR-STATUS[5] に準拠した用例ベースの対話システムであり、入力されたユーザ発話と類似した用例を検索し、ヒットした用例に対する応答をシステムの応答として使用する。用例の検索には Apache Lucene を、形態素解析機には Kuromoji を用いており、使用した用例は 26972 個である。

応答候補は類似度の上位 10 件とし、応答のスコアには、Apache Lucene が出力した類似度を用いる。

3.2 ニューラル対話モデル (NCM)

本システムは、Encoder-Decoder モデルのニューラルネットワークを用いた対話システムである。実装には KyotoNMT[6] を使用し、Encoder には各 1000 次元の LSTM を 4 層、Decoder には各 1000 次元の LSTM を 4 層のネットワークを用いた。語彙数は入力、出力ともに 80000 とし、Dropout 率は 20%、パラメータの最適化手法には Adam を用いた。学習データには約 1000 万件の Twitter のツイート・リプライペアを使用した。学習中のモデルの評価は学習データとは別の 1 万件のツイート・リプライペアにより計算した BLEU を用いた。

応答候補は、初期パラメータを変更して個別に学習した 12 個のモデルでそれぞれ生成したものを用いる。ニューラル対話モデルでは、ビームサーチにより 1 つのモデルから複数の応答候補を得ることも可能であるが、今回の実装・設定では、句点の有無や末尾の違い (例:「～です。」と「～ですよ。」) のような、ほぼ同一の応答候補が複数出力されたことから、このような設定で実験を行った。応答のスコアには各応答の生成確率を単語数で正規化した値を用いる。また、後述する対話データの作成時には、応答時間の問題から、12 個のモデルのうち BLEU が最も高かった 1 モデルを用いて対話を行った。

3.3 Neural Utterance Ranking モデル (NUR)

Neural Utterance Ranking モデル (NUR モデル) は、我々が提案したニューラルネットワークベースの対話モデル [7] であり、自動生成した複数の発話候補を適切な順にランキングし、高順位の発話を用いて対話を行うモデルである。学習設定・使用データは文献 [7] における実験で使用したものと同一である。応答のスコアは順位付けのためにモデルが出力するスコアを用いる。

4 対話破綻検出を用いた応答候補リランキング

本章では破綻検出手法を用いて、対話システムの応答選択の性能がどの程度向上できるかを実験により評価する。本実験では、対話システムが出力した応答候補を破綻検出結果に基づきリランキングし、その結果を評価する。

4.1 使用データ

本実験では対話破綻検出手法の学習用のデータと、対話破綻検出を適用し、応答性能の変化を評価するためのデータの 2 種類のデータを用いる。以下ではそれぞれについて述べる。

4.1.1 対話破綻検出学習データ

対話破綻検出手法の学習データとして、前述した 3 種類の対話システムの対話破綻アノテーション付き対話データを用いる。NCM と NUR システムのデータは対話のルール、アノテーション基準などの形式を DBDC2 に準拠し、独自に収集したものであり、IRS データは DBDC2 で配布されているデータをそのまま用いる。

データには、各対話システムとユーザとの 2 者間テキストチャット、および各システム発話に複数名のアノテータが付与した対話破綻アノテーションが収録されている。アノテーションとしては、以下の三種類の破綻ラベルが付与されている。

- 破綻ではない 当該システム発話のあと対話を問題無く継続できる。
- △ 破綻とは言い切れないが、違和感を感じる発話 当該システム発話のあと対話をスムーズに継続することが困難。
- × あきらかにおかしいと思う発話。破綻 当該システム発話のあと対話を継続することが困難。

各データの統計情報を表 1 に示す。表の△ラベルの割合と×ラベルの割合より、IRS、NCM、NUR の順で破綻を引き起こす応答を多く行う傾向があることが確認できる。

4.1.2 応答性能評価データ

対話破綻検出結果を適用した際の応答性能を評価するためのデータとして、ユーザとシステムが交互に発話した対話ログ (文脈) と文脈中の最後のユーザ発話に

表 1: 対話破綻検出学習データの統計情報

	IRS	NCM	NUR
対話数	100	100	120
ユーザ発話数	1000	1000	1200
システム発話数	1100	1100	1320
アノテータ数	30	30	34
○ラベルの割合	31.1%	47.4%	57.7%
△ラベルの割合	26.7%	32.7%	27.0%
×ラベルの割合	42.1%	17.2%	15.2%
Fleiss' κ	0.29	0.29	0.26
Fleiss' κ ($\Delta+\times$) ¹	0.38	0.43	0.42

表 2: 応答性能評価データの統計情報

	IRS	NCM	NUR
データ数	300	300	300
文脈中の平均発話数	1.37	2.14	2.04
1文脈あたりの応答候補数	11.42	10.57	10.94
1応答候補あたりの単語数	18.13	9.54	10.70
平均アノテータ数	4.61	5.92	3.88

対する複数の応答候補とスコア, および破綻ラベルが収録されたデータを用いる. データの収集方法・フォーマットは文献 [7] のテストデータと同一である, また, NUR のデータは文献 [7] のテストデータをそのまま用いる.

各応答候補には 3 名以上のアノテータにより, 対話破綻検出学習データと同様の三種類の破綻ラベルが付与されている. 応答性能の評価の際には, 文脈に対して発話が自然か否かという点が重要であることから, アノテータの半分以上が○と評価した応答を正例とし, それ以外の応答を負例とする. なお, 今回の実験では, 1 つの文脈に対する応答候補の中に少なくとも 1 個以上の正例の応答が含まれているものを用いた. 文脈は Twitter から収集し, システムにより 10 個以上応答候補が生成できなかった文脈は除外した. なお, 応答候補が生成可能かどうかはシステムにより異なるため, データ中に含まれる文脈はシステムにより異なる. 各データの統計情報を表 2 に示す.

4.2 対話破綻検出手法

本実験では, 対話破綻検出手法として, DBDC2 で性能が最も高かった杉山の手法 [8] を用いる. 本手法は, システムによる急激な話題転換や対話行為の不自然なつながりなど, 対話システムに典型的な誤りのパターンを捉えた素性を用い, ExtraTreesRegressor で破

綻ラベルの分布推定を行う手法である. なお, 文献 [8] では対話行為アノテーション済みのコーパスを用いて対話行為推定器を構築し, その結果を素性の 1 つとして使用しているが, 当該コーパスは非公開である. そこで本研究では新たに対話行為アノテーション済みのコーパスを作成し, 対話行為推定器を構築した. 本コーパスで学習に使用した文は 25555 個, 対話行為クラスは 39 個であり, 単語 1~3gram を素性とした多クラス SVM により対話行為推定器を学習した. SVM のカーネルは線形カーネルを用いた. なお, DBDC2 のデータを用いて対話破綻検出を行う予備実験を行った結果, DBDC2 の結果とほぼ同等の結果が得られることを確認した. その際, 学習データとして全ての対話システムのデータを用いた場合と, システムごとにデータを分け, 破綻検出先のシステムと同じシステムのデータのみを用いて学習を行う場合とで比較を行った. その結果, 学習データは全て用いるのではなく, 対話システム別に用いたほうが性能が高いことが確認された. そこで, 本研究で用いる破綻検出の学習データはシステムごとに分け, 適用するシステムに応じて使い分けることとした.

4.3 リランキング手法

本実験では, 応答性能評価データに含まれる応答候補に対し破綻検出を行い, その結果を用いて応答候補をリランキングすることで性能評価を行う. しかし, 破綻検出手法をどのように適用すれば破綻を回避し, 応答性能を向上することができるかについては自明ではない. そこで本研究では以下の 3 つの手法によりリランキングを行う.

分類ベース 破綻検出手法の出力した分類結果を重視し, 破綻と分類された応答候補を下位にする手法である. 破綻検出手法が出力した各ラベルの確率分布から, 最大確率となったラベルを分類結果とする. 本手法では, × のみを破綻とする場合 (×) と, △と × の両方を破綻とする場合 (△+×) の 2 つで実験を行う. ここで, △と × の両方を破綻とする場合, △と分類された応答候補の方が × とされたものよりも上位になるようリランキングする. 同一ラベル中の順位は応答候補のスコアの順とする.

非破綻確率ベース 応答候補のスコアと, 破綻検出手法の出力した非破綻確率の積を新たなスコアとしてリランキングする. 破綻検出手法が出力した確率分布の ($p(\circ), p(\Delta), p(\times)$) のうち, $p(\circ)$ を非破綻確率とする場合 (○) と, $p(\circ) + p(\Delta)$ を非破綻確率とする場合 (○+△) の 2 つで実験を行う.

¹△を × とみなした場合

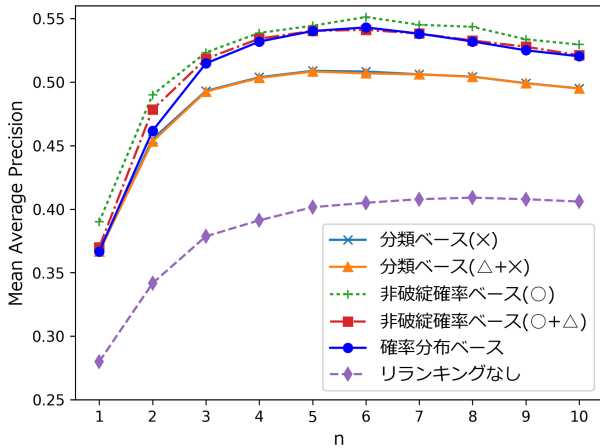


図 1: 順位 n を変化させた際の MAP(IRS)

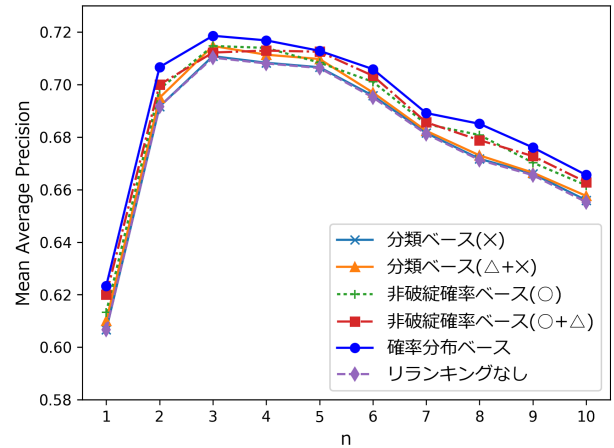


図 3: 順位 n を変化させた際の MAP(NUR)

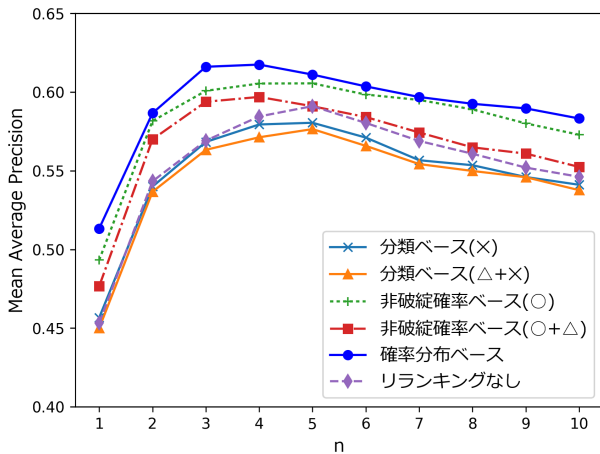


図 2: 順位 n を変化させた際の MAP(NCM)

確率分布ベース 破綻検出手法が出力した各ラベルの確率 $(p(\circ), p(\Delta), p(\times))$ と応答スコア s を用いて、以下の式により新たなスコア s_{new} を算出し、リランキングを行う。

$$s_{new} = w_1 p(\circ) + w_2 p(\Delta) + w_3 p(\times) + w_4 s \quad (1)$$

式中の $w_1 \sim w_4$ はパラメータであり、確率的勾配降下法により求める。損失関数は正例の応答候補の教師スコアを 1.0、負例を 0.0 とした際の s_{new} との平均二乗誤差とした。このように確率分布ベース手法はパラメータの最適化を伴うため、10 分割交差検証により評価を行う。

4.4 実験結果

リランキング性能の評価には、ランキングの上位にどれだけ正例の応答候補が出現しているかを示す指標

である Mean Average Precision (MAP) により行った。MAP の計算に用いる順位 n を上位 1 位~10 位まで変更させた場合の IRS, NCM, NUR のそれぞれの結果を図 1~3 に示した。結果より、非破綻確率ベース (○)、非破綻確率ベース (○+Δ)、確率分布ベースについては全てのシステムに対して性能向上が確認できた。一方、分類ベース (×)、分類ベース (Δ+×) についてはシステムによって有効か否かが異なる結果となった。個別に見ていくと IRS(図 1) では、非破綻確率ベース (○) が最も大きく性能が向上し、次いで非破綻確率ベース (○+Δ)、確率分布ベースという順であった。分類ベース (×) と分類ベース (Δ+×) はほぼ同じ結果となり、他の手法と比べ性能向上の幅は小さかったが、それでもなおリランキングを行わない場合よりも大きく性能が向上した。NCM(図 2) では、確率分布ベース、非破綻確率ベース (○)、非破綻確率ベース (○+Δ) の順で性能向上の幅が大きくなり、分類ベース (×) と分類ベース (Δ+×) に関してはリランキングを行わない場合よりも性能が悪化する結果となった。NUR(図 3) では、確率分布ベースが他の手法と比べ性能向上の幅が大きかったが、いずれの手法も他のシステムに破綻検出手法を適用した場合よりも、性能向上の幅は小さくなった。

4.5 考察

前節の結果より、対話破綻検出を適用することにより応答性能の向上が可能であることが確認できたが、システムによって有効な適用手法や性能向上の度合いが異なっていた。この原因を調査するため、対話破綻検出手法の出力結果について分析を行った。

表 3 は、応答性能評価データに対し対話破綻検出手法を適用した結果、最大確率となったラベルの分布をシステム別に示したものである。ここから、IRS では

表 3: 破綻検出手法の出力ラベルの分布 (括弧内は件数)

	○	△	×
IRS	20.7% (711)	0.3% (10)	79.0% (2706)
NCM	84.9% (2862)	9.4% (317)	5.7% (193)
NUR	96.6% (3173)	2.1% (69)	1.3% (42)

× ラベルとなった応答候補が他と比べて非常に多いことがわかる。したがって、IRS の場合、より多くの応答候補のスコアが破綻検出手法により変更されることになり、性能向上の幅が大きくなったといえる。一方、NCM と NUR では多くが○ラベルと判定されており、性能向上の余地が小さかった結果、性能向上の幅も小さくなった。特に、NUR では 96.6% が○ラベルであり、△と × ラベルのみを用いる分類ベースの手法は有効でないことがわかる。このように、性能の高い対話システムにおいては、破綻を引き起こす応答が少ないことから、分類ベースの手法は有効ではなく、○ラベルの情報も用いる非破綻確率ベース、確率分布ベースの手法が有効であるといえる。

また、NCM では分類ベース (×) と分類ベース (△+×) はリランキングを行わない場合よりも性能が悪化した。そこで、破綻検出手法が × ラベルを出力した応答候補のうち、正例であった割合を調査した。その結果、IRS が 11.6%、NUR が 19.0% であったのに対し、NCM では 39.4% と大きな値であったことが確認された。したがって、NCM では破綻検出手法の性能が低かったため、リランキング後の性能が悪化したと考えられる。

次にリランキングを行った場合の応答の変化を分析するため、破綻検出手法を適用した場合としない場合における、順位が 1 位の応答に含まれる単語数と語彙数を調査した。リランキング手法は全システムにおいて安定して性能の良かった確率分布ベースを用いた。結果を表 4 に示す。表より、全てのシステムにおいて破綻検出手法の適用により、単語数と語彙数の両方が小さくなる傾向があることがわかる。特に、IRS はその変化量が大きい。

また、確率分布ベースを用いた場合の実際の応答の変化の例を表 5~7 に示した。表 5 から、IRS では適用前と比べ簡潔な応答となっていることがわかる。このような応答を行えば、対話破綻は減らすことが可能であるが、システム側からは何の情報も提供しない退屈な対話となる危険性がある。応答候補の中に簡潔な応答が含まれるシステムに対し対話破綻検出を適用する場合には、そのような応答ばかりが選ばれないよう注意する必要があるといえる。

表 4: 対話破綻検出の適用による 1 応答あたりの単語数と語彙数の変化

	単語数 (適用前)	単語数 (適用後)	語彙数 (適用前)	語彙数 (適用後)
IRS	17.24	11.16	1219	761
NCM	10.13	8.47	363	332
NUR	9.69	9.54	885	881

5 むすび

本研究では 3 種類の対話システムの応答に対し、対話破綻検出手法を適用することで、応答性能を向上できるかを実験的に評価した。対話破綻検出の適用手法として、破綻ラベルの分類結果を用いる分類ベース手法、破綻ではない確率を用いる非破綻確率ベース手法、各ラベルの確率分布を用いた確率分布ベースの 3 つを提案した。実験の結果、非破綻確率ベース手法と確率分布ベース手法については全てのシステムの応答性能の向上が確認できた。分類ベース手法に関してはシステムの応答性能を悪化させるケースが確認された。また、対話破綻検出の適用によるデメリットとして、短く簡潔な応答が増加することが確認された。

謝辞

本研究は JSPS 科研費 16H05880 の支援を受けて実施されたものである。

参考文献

- [1] 東中竜一郎, 船越孝太郎, 小林優佳, 稲葉通将. 対話破綻検出チャレンジ. 第 75 回言語・音声理解と対話処理研究会 (第 6 回対話システムシンポジウム), 人工知能学会研究会資料 SIG-SLUD-75-B502, pp. 27-32, 2015.
- [2] 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 78 回言語・音声理解と対話処理研究会 (第 7 回対話システムシンポジウム), 人工知能学会研究会資料 SIG-SLUD-B5025-19, pp. 64-69, 2016.
- [3] Hiroaki Sugiyama. Utterance selection based on sentence similarities and dialogue breakdown detection on ntcir-12 stcv task. pp. 552-553, 2016.
- [4] Hideaki Mori and Masahiro Araki. Selection method of an appropriate response in chat-oriented dialogue systems. In *Proceedings of the*

表 5: 対話破綻検出の適用による応答の変化 (IRS)

文脈	対話破綻検出適用前	対話破綻検出適用後
A:ロシアンコーラうますぎィ!	バイオリン! 8年やって今弾けないってもったいないですね。再開はしないんですか?	うまいね
A:明日カードリーダーとファイル買いたいんだけどめんどい	そうなってるといいですね～。精進します。	私もです。笑
A:ダメだ今はタイミングが悪い B:大丈夫? A:だめ	そこまでしっかりと育てられるとは思いません。	いいですよ。それに、だめじゃないですよ。

表 6: 対話破綻検出の適用による応答の変化 (NCM)

文脈	対話破綻検出適用前	対話破綻検出適用後
A:こたつ出してから固い床にずっと座ってるから腰がヤバい	床に座るのは危険だと思っているのだが…	それはあかんやつやな
A:研究室に枕持ってこよ…	研究室に持ってきてないんですか?	枕にしてください!
A:これは一海老名市民としてもな B:横浜から海老名って結構遠いよね A:意外とそうでもないんだよな。そう、特急ならね	そうなんだよね、そうなんだよね。	そうなんですよね。

表 7: 対話破綻検出の適用による応答の変化 (NUR)

文脈	対話破綻検出適用前	対話破綻検出適用後
A:今日できることは明日でもできるってそう言ってこころを落ち着かせよう!	こころと体は別だよ	こころのゆとりが欲しいかも
A:もっと頑張ろう…頑張らなきゃ……(レッドブルを買いに行く)	レッドブルとか眠眠打破とかきかないんだよ	エナジードリンクはレッドブルが一番だね
A:そもそも何を持って迷惑行為ってとこよりも損したくないならちよっとは考えなきゃだめよ B:そーだねー A:気合いれろオラァ	普段使っていない子だと気合入れるよ	気合じゃ世界は変わらない

17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 228–231, 2016.

- [5] Alan Ritter, Colin Cherry, and William B Dolan. Data-driven response generation in social media. In *Proceedings of the conference on empirical methods in natural language processing*, pp. 583–593, 2011.
- [6] Fabien Cromieres. Kyoto-nmt: a neural machine translation implementation in chainer. In *Proceedings of COLING 2016: System Demonstrations*, pp. 307–311, 2016.

[7] Michimasa Inaba and Kenichi Takahashi. Neural utterance ranking model for conversational dialogue systems. In *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 393–402, 2016.

[8] 杉山弘晃. 発話生成における誤りパターンの分析に基づく対話破綻検出. 第78回言語・音声理解と対話処理研究会(第7回対話システムシンポジウム), 人工知能学会研究会資料 SIG-SLUD-B505-23, pp. 81–84, 2016.