Bidirectional LSTM-RNNを用いた対話破綻検出のエラー分析

稲葉 通将　　高橋 健一
広島市立大学 大学院情報学研究科
{inaba, takahashi}@hiroshima-cu.ac.jp

1 はじめに

2015年に開催された対話破綻検出チャレンジでは，人間と対話システムの対話ログから，対話が破綻した(対話システムが不適切な応答を行った)箇所を推定するタスクが行われた．事前に対話が破綻する可能性を推定できれば，それを回避できる可能性が高まる[1]ことから，対話破綻箇所の検出技術は対話システムの対話能力底上げのための有用な技術である．

本稿では，現状の破綻検出手法の課題と改善点を見出すため，我々が文献[2]で提案した3つの破綻検出手法のうち，最も性能の高いBidirectional Long Short-Term Memory Recurrent Neural Network (BLSTM-RNN)を用いた手法における検出エラーの分析を行う．

なお，対話破綻検出チャレンジ，および配布されている対話データの詳細は文献[3]を，我々が提案した破綻検出手法の詳細は文献[2]をそれぞれ参照されたい．また，本稿で分析を行ったデータは，チャレンジで提出したrunと同一のものであり，対話破綻検出チャレンジのホームページ[1]ですでに公開されている．

2 破綻検出手法

2.1 概要

本章では，エラー分析の対象とするBLSTM-RNNを用いた対話破綻検出手法の概要について述べる．

対話破綻検出チャレンジで提供されている対話データでは，全ての対話システムの発話に対し，対話破綻のアノテーションが行われている．アノテーションは○・△・×の 3分類で行われており，それぞれ「破綻ではない」,「破綻とは言い切れないが違和感を感じる」,「破綻」を意味する．我々の提案した破綻検出手法は，複数のアノテータが○・△・×のそれぞれに対し，どのような割合でアノテーションを行ったかという分布を推定する．

本手法では，破綻検出対象となる対話システムの発話と，その直前のユーザ発話の2発話のみを入力として用いる．まず，各発話をMecab[4]を用いて単語に分割し，単語の系列を得る．次に単語の系列を単語の分散表現の系列に変換し，この系列をBLSTM-RNNの入力とする．分散表現への変換はMikolovらの手法[5]を実装したword2vecを用いる．

2.2 Bidirectional LSTM-RNN

Recurrent Neural Network(RNN)は系列データを扱うためのモデルであり，前時刻の中間層を現時刻の入力としても用いることで，内部状態を保持しながら学習を行う．しかし，通常のRNNは逆誤差伝播による学習を行う際，勾配が減衰するという問題(勾配消失)が存在する．

Long short-term memory(LSTM)[6]は勾配消失の問題を解決するために提案されたユニットの1つである．LSTMはConstant Error Carousel(CEC)と呼ばれる記憶素子にエラーを選択的に取り込み，保持することで勾配の消失を防ぐ．

LSTMでは通常，時刻$t-1$の隠れ状態を時刻tの隠れ状態の入力として用いるが，時刻$t+1$の隠れ状態をtの隠れ状態の入力として用いる逆方向のLSTMを同時に中間層に用いるBidirectional LSTM (BLSTM)[7]が提案されている．このBLSTMは，タスクによってはLSTMよりも高い性能を示すことが知られており，本手法でもこれを用いる．

2.3 分布の推定

対話が破綻するケースは様々であるが，生成した発話文が文法的に誤っており，意味不明の発話が出力さ

[1] https://sites.google.com/site/dialoguebreakdowndetection/

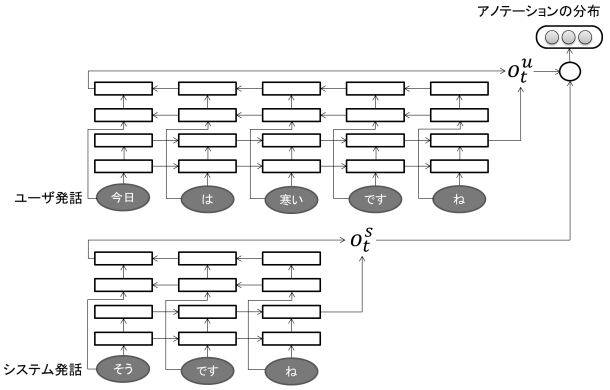

図 1: BLSTM-RNN による対話破綻検出

れてしまうというように，ユーザ発話の内容に依存せず破綻が発生する場合も多い．また，システムの過去の発言に関する質問など，システムが適切に応答するのが難しいユーザ発話も存在する．そこで提案手法では，ユーザ発話用，システム発話用の 2 つの BLSTM-RNN を用いる．それぞれがユーザ発話・システム発話を個別に学習することで，破綻検出につながる情報を効率よく処理可能となることが期待できる．また，それぞれの RNN は，順方向と逆方向の LSTM を 2 層ずつ，計 4 層の中間層を持つ構成とする．

各 BLSTM-RNN は入力系列の最後の要素が読み込まれた時点で固定長のベクトルを出力する．ユーザ発話用 BLSTM-RNN の出力を o_t^u，システム発話用の出力を $o_{t'}^s$ とすると，アノテーションの分布 y は以下の式により求める．

$$y = softmax(W_u o_t^u + W_s o_{t'}^s + b_{us}) \qquad (1)$$

損失関数には，正解の分布との間の Mean squared error を用いる．

表 1 に提案する BLSTM-RNN を用いた破綻検出の例を示す．ユーザ発話とシステム発話はそれぞれ別の BLSTM-RNN に順に入力される．それぞれの出力は式 (1) により統合され，最終的に○・△・×のそれぞれに対応する 3 次元の確率分布を得る．出力ラベルは最も確率の高いものとする．以降，本稿ではこの出力ラベルを用いて分析を行う．

3　エラー分析

本稿で分析対象とするデータは，対話破綻検出チャレンジで配布された評価用データ 80 対話である．1 対話につき対話システムは 11 回発話しているため，破綻検出対象の発話は合計 880 個である．これらの発話に対し，前章で述べた破綻検出手法により破綻ラベルを推定し，そのデータを分析する．

表 1: 推定したラベルと正解ラベルの混同行列

		推定ラベル ○	△	×	合計
正解ラベル	○	467	18	68	553
	△	76	11	42	129
	×	99	11	88	198
	合計	642	40	198	880

3.1　検出傾向

BLSTM-RNN による破綻検出手法の出力傾向の分析のため，検出手法の出力から決定したラベルと正解ラベルとの間の混同行列を表 1 に示した．表より，提案手法は○，×，△の順に多くラベルを出力していることがわかる．これは，正解ラベルの分布とも一致しているが，正解と比べると，提案手法は△を出力する割合が小さい．よって，提案手法は正解がラベル△である場合，誤って○もしくは×と推定することが多い事がわかる．また，提案手法は○を多く出力する傾向にあり，推定エラーが発生した場合は，正解が×であるのにもかかわらず，○と推定してしまうケースが最も多くなっている．

3.2　破綻分類別の分析

提案手法の推定エラーがどのような状況で発生しやすいのかを確認するため，破綻の種類を分類し，それに基づいて分析を行った．破綻の分類は，文献 [8] で提案された対話破綻の類型化案に基づいて行った．この類型化案では，発話，応答，文脈，環境の 4 つの大分類と，大分類に紐づく 3〜5 つの小分類が提案されており，破綻はそれらを組み合わせた 16 種類のいずれかに分類される．

そこで我々は，パラメータ $t = 0.5$ における正解ラベルが△および×の 327 発話に対し，分類を行った．1 つの破綻箇所について，複数の分類が当てはまると考えられる場合は，その中で最も妥当と考えられる 1 つに分類することとした．

表 2: 大分類

	分類数	エラー数	エラー率
発話	53	31	0.585
応答	227	108	0.476
文脈	40	30	0.750
環境	7	6	0.857

表 3: 小分類：発話

	分類数	エラー数	エラー率
構文制約違反	17	7	0.412
意味制約違反	34	23	0.676
不適切発話	2	1	0.500

表 4: 小分類：応答

	分類数	エラー数	エラー率
量の公準違反	30	18	0.600
質の公準違反	12	10	0.833
関係の公準違反	175	73	0.417
様態の公準違反	8	5	0.625
誤解	2	2	1.000

3.2.1 大分類

表 2 に 4 つの大分類の分類発話数，提案手法における検出エラー数，およびエラー率を示した．検出エラー数とは，正解ラベルが△および×である発話に対し，提案手法により○と推定された発話の数である．

まず，分類数を見ると，直前のユーザの応答に関する破綻である「応答」が全体の 7 割程度を占めていることがわかる．その次に多いのがシステム発話の生成過程に問題があるため生じた破綻である「発話」，次に話の流れに関する破綻である「文脈」，その他の原因で生じた破綻である「環境」と続く．エラー率を見ると，分類数と同じ順でエラー率が低くなっているのがわかる.「応答」が最もエラー率が低いのは，応答に関する破綻は学習データ中でも多く存在し，適切な学習ができたためと考えられる．そのため「発話」に関してはより多くの学習データを用いることで，性能が改善する可能性があると考えられる．

一方,「文脈」と「環境」に関しては,「発話」と「応答」に比べ，エラー率が非常に高い．その最大の原因は，提案手法では破綻検出対象となるシステム発話と，その直前のユーザ発話のみを素性としており，文脈や環境に関する素性を全く使用していないためであると考えられる．これらの検出率を高めるためには，文脈と環境を考慮した素性を導入することが必須である．

3.2.2 小分類：発話

発話の小分類に関する結果を表 3 に示した．小分類はそれぞれ,「構文制約違反」は文法エラーを,「意味制約違反」は文法エラーは無いが，語の組み合わせにより意味が理解できないもの,「不適切発話」は「みんっ」などの無意味な発話を意味する．3 つの小分類のうち，文法エラーを意味する構文制約違反のエラー率が最も小さくなった．これは，提案手法は単語の系列を順に BLSTM-RNN に入力していく手法であることから，単語の前後関係の不備を捉えやすく，比較的精度よく破綻を検出できたと考えられる．例えば，評価データ中には「自身はゆいます」という発話が存在するが,「は」と「ゆい」という単語が並ぶことは日本語ではあまり見られないことから，正しく破綻が検出できている．一方，意味制約違反の場合，隣り合う単語だけでは破綻が検出し辛いため，エラー率が大きくなったと考えられる．

3.2.3 小分類：応答

応答の小分類に関する結果を表 4 に示した．量，質，関係，様態の公準違反はいずれも Grice の公準 [9] に基づく分類である．また，誤解は多義語の意味の取り違いなどによる破綻である．

表より,「関係の公準違反」の分類数が非常に多く，またエラー率も最も低いことがわかる．関係の公準違反の典型的なパターンは，ユーザの質問に対し，システムが適切に応答できないというものである．対話破綻検出チャレンジで採用された対話システムは，ユーザの質問に適切に応答できないことが多い．そこで提案手法は，ユーザの質問の後のシステム発話に対して△・×のラベルを推定することが多くなり，結果的にそれがエラー率が低くなった要因と考えられる．

3.2.4 小分類：文脈・環境

文脈の小分類に関する結果を表 5 に，環境の小分類に関する結果を表 6 に示した．文脈の小分類の「話題展開不追随」はユーザからの話題の展開が起こっているにも関わらず，それに追随できていないことから生じる破綻である．また，環境の小分類「無根拠」は根

表 5: 小分類：文脈

	分類数	エラー数	エラー率
量の公準違反	14	11	0.786
質の公準違反	6	5	0.833
関係の公準違反	11	7	0.636
様態の公準違反	2	2	1.000
話題展開不追随	7	5	0.714

表 6: 小分類：環境

	分類数	検出エラー数	エラー率
無根拠	3	2	0.667
矛盾	4	4	1.000
非常識	0	0	0.000

拠の無い受け入れがたい断定を,「矛盾」は一般常識との矛盾を,「非常識」は悪口など，社会規範から外れる発話により生じる破綻をそれぞれ意味する．

表より，それぞれどの小分類も分類数が少なく，エラー率が大きい．これは前述したように，提案手法は文脈・環境を考慮していないことが原因であると考えられる．また，破綻を特定するために参照しなければならない範囲が大分類の発話・応答と比べ広く，より検出が困難であると考えられる．一方，完全に同一の発話をシステムが何度も繰り返すことで破綻となっている箇所もあり，比較的簡単に検出可能なものも含まれている．

4 おわりに

本稿では，Bidirectional Long Short-Term Memory Recurrent Neural Network (BLSTM-RNN) を用いた対話破綻検出手法における検出エラーの分析を行った．分析のため，対話破綻の発生した発話を 16 種類の類型に分類し，検出エラーとの関係を調査した．分析の結果，提案した対話破綻検出手法は，直前のユーザの発言とそれに対する応答の関係性に関する破綻の検出精度が高いことが判明した．一方で，過去の発言との矛盾など，文脈に関する破綻や，会話の文脈以外で発生した破綻に関しては，検出精度が非常に低いことも明らかとなった．

今後は今回の分析を踏まえ，文脈に関する素性を扱えるように手法を改良し，対話破綻検出の精度を高めていく予定である．

参考文献

[1] 東中竜一郎, 船越孝太郎. Project next nlp 対話タスクにおける雑談対話データの収集と対話破綻アノテーション. 言語・音声理解と対話処理研究会, Vol. 72, pp. 45–50, 2014.

[2] 稲葉通将, 高橋健一. Long short-term memory recurrent neural network を用いた対話破綻検出. 言語・音声理解と対話処理研究会 第 75 回 (第 6 回対話システムシンポジウム), pp. 57–60, 2015.

[3] 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. 言語・音声理解と対話処理研究会 第 75 回 (第 6 回対話システムシンポジウム), pp. 27–32, 2015.

[4] T. Kudo. Mecab: Yet another part-of-speech and morphological analyzer. *http://taku910.github.io/mecab/*, 2005.

[5] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

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

[8] 東中竜一郎, 船越孝太郎, 荒木雅弘, 塚原裕史, 小林優佳, 水上雅博. Project next nlp 対話タスク：雑談対話データの収集と対話破綻アノテーションおよびその類型化. 言語処理学会第 21 回年次大会ワークショップ, 2015.

[9] H Paul Grice. *Studies in the Way of Words*. Harvard University Press, 1991.