類型毎の破綻検出器におけるエラー分析
Error analysis of categorical breakdown estimators

堀井 朋　　荒木 雅弘
京都工芸繊維大学
horii@ii.is.kit.ac.jp, araki@kit.ac.jp

1　はじめに

対話システムを長く使い続けることでユーザが音声インタフェースに慣れ，高齢者などが様々なサービスを音声対話サービスで受けられるようになるというメリットがある．雑談対話はユーザに継続的な対話システムの使用を推進させ，それはシステムへの信頼感の醸成を促す．対話を長く続けるためには，対話破綻を生じさせるシステム発話を事前に検出することが有効である．

しかし，雑談対話の破綻の原因は様々であり，単一の識別器による検出は難しい．そこで，我々は破綻の類型毎に適した識別器を作成し，それらを組み合わせることで破綻検出を試み，その結果の分析を行った．

2　提案手法

対話の破綻が引き起こされる要因は一意でない．発話の文構造が誤っている場合や文脈に添えていない場合など，様々な原因が考えられる．そこで，Project Next NLP の雑談対話コーパスの分析により作成された類型化案 [1],[2] に基づき識別機を作成し，それらを組み合わせて検出を行うという手法を提案する．その破綻識別器のアーキテクチャを図 1 に示す．

類型化案の内容は表 1 の通りである．大分類は，対話のどの範囲に関連した破綻であるかを基準にして分類が行われている．また，各大分類に対し破綻要因を更に細かく突き詰めたものが小分類である．

3　実験

3.1　実験 1 : 単語ベクトルを特徴とした検出

破綻の類型化に従って検出を行うことの有効性を確かめるため，類型に則さない単一の破綻識別器と類型の 16 分類に則した破綻識別器を作成した．直前のユーザ発話とシステム発話からなる単語ベクトルを特徴とし，SVM（poly–kernel,1 次）を識別アルゴリズムに用いた．学習データは Project Next NLP 対話タスク内の類型化ワーキンググループが作成した，システム発話の破綻の有無と破綻類型がアノテーション付けされた 1000 発話である．テストデータは破綻検出チャレンジで配布された開発データ 200 発話とした．また，後者については複数の識別器のうち一つでも破綻となれば，その発話は破綻であると識別している．

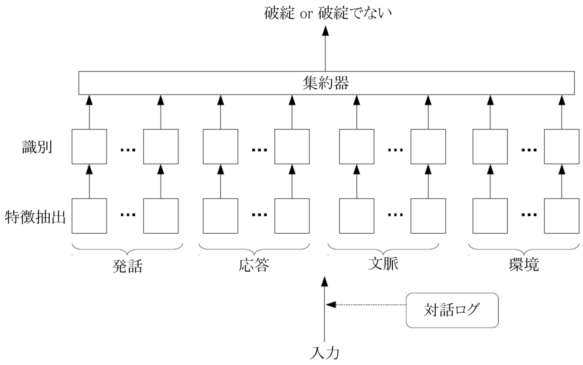

図 1: 提案システムのアーキテクチャ

結果を表 2[1] に示す．破綻分類に則した識別器の方が単一の識別器よりも精度が下回ってしまった．これは破綻類型が細分化されているため，類型毎の破綻を学習するデータの数も非常に少なくなってしまったが故に，破綻を検出できていないのはないかと考えた．つまり，学習データの数が増えたならば，より多くの破綻を検出できる可能性を考えたのである．

そこで，16 種の小分類で分けるのでなく，4 種の大分類に沿った学習データを作成し，同様の実験を行った．16 分類に則した場合と比べて Accuracy は低下したものの，F 値は上昇している．つまり，破綻をより多

[1] O・T・X：対話破綻ラベル．順に「破綻ではない」「破綻と言い切れないが違和感を感じる」「破綻」を示す．
t：しきい値．アノテータ間で評価が分かれたアノテーションの扱いを調整する．

表 1: Project Next NLP 対話タスク WG による破綻類型化

大分類	小分類	類型名	説明
発話	構文制約違反	構文的な誤り	日本語の文として正しくない
	意味制約違反	意味的な誤り	文としての意味が通らない
	不適切発話	解釈不能	著しく応答の体をなしていない
応答	量の公準違反	情報過不足	応答として内容の過不足がある
	質の公準違反	不理解	直前のユーザ発話を理解していない
	関係の公準違反	無関係	直前のユーザ発話の話題や発話意図と無関係
	様態の公準違反	意図不明	発話の意図が汲み取れない
	誤解	誤解	直前のユーザ発話の内容を誤って理解している
文脈	量の公準違反	不要情報	冗長な発話
	質の公準違反	矛盾	発話内容や態度が急転換する発話
	関係の公準違反	無関係話題	話題が文脈から逸脱
	様態の公準違反	関連性不明	文脈のどの部分と関連しているのか不明
	話題展開への不追随	不追随	話題展開後も前の話題を続けている
環境	無根拠	共通基盤欠如	根拠の無い主張
	矛盾	一般常識欠如	常識に反する主張
	非常識	社会性欠如	社会規範から外れ，相手を不快にする発話

表 2: 破綻検出スコア [O, T+X] t=0.5

	Accuracy	Precision	Recall	F-measure
類型なし	0.42	0.73	0.73	0.73
16 類型	0.56	0.76	0.29	0.42
4 類型	0.49	0.76	0.70	0.73

表 3: 応答のみ 2 次カーネルのスコア [O, T+X] t=0.5

Accuracy	Precision	Recall	F-measure
0.50	0.75	0.65	0.70

く検出できていることが分かる．また，ベースラインと比べても F 値こそ僅かに低いものの，Accuracy はこちらのほうが高い．破綻検出においては，この 2 つの値の釣り合いが重要となるため，ベースラインよりも類型毎に検出を行った方が精度が良いと考えられる．

以上のことから，学習データを充分に揃えた上で，破綻類型に従った検出器を用いれば，ある程度の有効性を持った検出精度を得られることが示された．

3.2 実験 2：類型毎に特徴を設定した検出

破綻類型毎に特徴を設定することで，検出精度の向上が見込まれる．そこで，大分類に関して有効と思われる特徴を以下のように設定し，評価実験を行った．

- 発話 : KNP によるシステム発話の構文解析スコア
- 応答 : 2 次カーネルにより単語共起情報を用いて学習
- 文脈 : システム発話とそれ以前の発話全ての単語ベクトルに対するコサイン類似度
- 環境 : システム発話の単語ベクトル

実験の結果,「発話」·「文脈」類型については，KNP[2] のスコア及びコサイン類似度に関して正例 · 負例の分布に顕著な違いは見られなかった．また,「応答」類型において 2 次カーネルを用いて学習を行い，実験 1 と同様に破綻検出を行った結果が表 3 である．

以上から，類型毎に別個の特徴を設定しても，単語ベクトルを用いて破綻検出を行った場合と変化がない，もしくは性能が低下するという結果が見られた．原因として,「発話」については 2〜3 文節程度の係り受けスコアでは文の妥当性が判定できなかったためと考えられる.「文脈」については，コサイン類似度を求める際,「車」や「自動車」と言ったような同義語を同一と見なせていなかったことが可能性として挙げられる．

[2] http://nlp.ist.i.kyoto-u.ac.jp/?KNP

表 4: 破綻アノテーション比率

	学習データ	テストデータ
発話	13 %	8 %
応答	51 %	67 %
文脈	30 %	20 %
環境	6 %	4 %

4 エラー分析

これらの結果を受け，エラー分析を行う．エラー分析の手順としては，実験結果の分析を行うためテストデータに類型アノテーションを行い，そのアノテーションの妥当性を確かめ，エラーが起きている対話の内容を分析する．それぞれのデータの対話破綻類型を集計した結果が表である．Project Next NLP によりアノテーションされた学習データと，自身でアノテーションを行ったテストデータの類型比率は各類型への破綻認識傾向が同様であるため，この類型アノテーションは妥当であると考えられる．

テストデータの 200 発話のうち正しく判定できていたものは 103 発話，破綻であると検出できてはいたが類型認識が異なっていたものは 23 発話，検出結果が誤っていたものは 74 発話であった．類型認識が異なっていたもの及び検出結果が誤っていた 97 発話におけるエラー内訳は表に示す通りである。総数は発話数の 97 を超えてしまうが，これは一つの発話を複数の識別器が破綻と検出していたからであるため問題はない．

これを見てみると，類型アノテーションの比率から考えても順当な結果であるが,「応答」に関するエラーが多く，次いで「文脈」に関するエラーが多い．今回の分析ではこれら二つについて重点的に細かい分析を行うものとする．

4.1 「応答」のエラー

「応答」について最も多く見られたエラーは，ユーザの質問に答えられていない破綻発話を破綻識別できていないというものであった．これについて詳しく見ていくと，ユーザ発話が 2 文以上ある場合，後半の文は大半質問文でありシステムは対話内容に近い返答をするのだが，質問には対応できていないというものが多数存在した．例えば，和菓子が好きという流れで，ユーザが「美味しいですね．お団子にはお茶ですが，緑茶は好きですか？」という発話をし，これにシステムは「みたらしは大好きですね」と返す．返答としてはおかしいので，この発話は破綻とするのが正解なのだが破綻と検出されずにエラーになった．もちろん，原因として考えられるのはユーザ発話に含まれる情報が多いため，そこから発話生成に使用する情報を適切に選べなかったと考えられる．例の場合だと,《お団子》という情報に引きずられ，システムは「みたらしは大好きですね」と返答したと思われる．つまり，ユーザが 2 文以上入力をしたときは，どちらかの文に重みを付けて特徴を抽出したり，別個に識別にかけるなどの対策が必要である．

また,「こんにちは」に対して「こんちゃっす」とシステムが返したところ，これが破綻であると検出されエラーになるというような，言語のブレに弱いという面も見られた．このエラーに対しては，システムに日本語の話し言葉の柔軟性が認識できていなかったことが原因として考えられる．言語というものは日々少しずつ変化が生じるものであるので，そういった方向にも対応していかなければならないことが明らかになった．

4.2 「文脈」のエラー

「文脈」に対するエラーとして最も目立ったのは，検出器が話題展開についていけずに破綻でない発話を破綻である，と検出してしまうものである．例えば，「こんにちは」という挨拶の応酬の後に「ところで〜」と話題を展開させようとシステムが発話するのだが，それを破綻であると判定する。これは発話対の持つ情報が大きく離れたものになってしまうことから起きたエラーであると考えられる。つまり,「ところで」や「むしろ」といった接続詞の役割を検出器が認識することが必要である．とは言え，会話内容が突然飛躍するとその発話が破綻している可能性は少なくないので，そのあたりの調整は注意しなければならない．

もう一つ特徴的だったのは，ユーザ発話の真意を汲めずにシステムが前自発話の内容を引きずった発話をする，あるいは会話の流れを考慮していないものを検出できていないといったエラーである．例を挙げると「〜は好きですか？」というシステムに質問に対し，ユーザが「好きです」と返答するとシステムが自分への告白であると捉え,「ありがと」と返事をする.「話したいと思ってました」というシステム発話に対して「なんですか？」と言われ，システムが「なんでもないです」と返す．これらを破綻と検出できていないのである．こういった典型的な「文脈」のエラーに関しては，検出器に渡している特徴が直前のユーザ発話と当該システム発話の単語ベクトルのみであることから，ある種必然的に生じたエラーであると考えられる．今

表 5: 破綻検出エラー内訳

正解＼検出結果	発話	応答	文脈	環境	破綻でない
発話	0	7	2	0	2
応答	1	0	2	1	26
文脈	1	8	0	0	12
環境	0	1	2	0	3
破綻でない	6	25	14	1	0

後，以前の発話情報も含めた特徴を与え識別方法に適切なものを選べば，このエラーは解消するのではないかと考えている．

4.3 他に見られた特徴的なエラー

「応答」「文脈」の両者に関して見られたエラーで特徴的なものが一つあった．システム発話に対してユーザの反応が「まぁいいや」,「はいはい，わかりました」などポジティブでもネガティブでもない曖昧なものであると，システム発話が明らかに破綻であっても破綻であると検出できない，というものである．原因として，ユーザ発話から得られる情報が極端に少なくなってしまうため，破綻を検出できないのではないかと考えた．

また，特殊な事例ではあったがユーザの方が対話を破綻させたとき，それ以降システム側が上手く発話生成できずに多数の破綻を生んでしまい，それを検出できずにエラーとなるといった事例も見られた．このことから，システム側の発話の質はユーザ側の発話に大きく左右されるものであるため，ユーザ側がしっかりとした返答をすることが必要ということが確認できた．

5 おわりに

破綻を類型化し，その分類に則して破綻を検出する手法を検討した．結果として，類型毎の破綻検出手法に一定の有効性があることが示されたが，破綻類型のそれぞれに適した特徴を発見するまでには至らなかった．

そして，実験結果に対するエラー分析を行った．その結果，エラーを生み出す発話の特徴が多数見受けられた．今後は，これらの発話的な特徴への対策を考え，破綻検出の精度向上を目指す．

参考文献

[1] Ryuichiro Higashinaka, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, Yuka Kobayashi and Masahiro Mizukami: Towards Taxonomy of Errors in Chat-oriented DialogueSystems, *In Proc. SIGDIAL 2015, pp.87-95. (2015)*

[2] Ryuichiro Higashinaka, Masahiro Mizukami, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara and Yuka Kobayashi: Fatal or not? Finding errors that lead to dialogue breakdowns in chat-oriented dialogue systems, *In Proc. EMNLP 2015, pp.2243-2248. (2015)*

[3] Yang Xiang, Yaoyun Zhang, Xiaoqiang Zhou, Xiaolong Wang, Yang Qin: Problematic SituationAnalysis and Automatic Recognition for Chinese Online Conversational System, *In Proc. of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, pp.43-51. (2014)*

[4] Ryuichiro Higashinaka, Toyomi Meguro, Kenji Imamura, Hiroaki Sugiyama, Toshiro Makino, Yoshihiro Matsuo: Evaluating Coherence in Open Domain Conversational Systems, *In Proc. of Interspeech 2014, pp. 130-134.*