

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*

東中 竜一郎[†] NTT メディアインテリジェンス研究所
船越 孝太郎[‡] (株) ホンダ・リサーチ・インスティテュート・ジャパン
荒木 雅弘[§] 京都工芸繊維大学
塚原 裕史[¶] (株) デンソーアイティラボラトリ
小林 優佳^{||} (株) 東芝
水上 雅博^{**} 奈良先端科学技術大学院大学

1 はじめに

対話タスクは、対話システム研究に関わるタスク参加者が、対話システムのエラーを共同で分析することを目的としたタスクであり、本稿はその1年間弱の活動成果について報告する。本稿の執筆時点で、大学・企業を含む15の拠点(表1参照)が本タスクに参加している。

対話システムにおいて、エラーを共同で分析することは簡単なことではない。なぜなら、対話システムは多種多様な要素技術・モジュールで構成されるシステムであり、また、その構成方法・採用する技術もシステム毎に大きく異なる。そのため、参加者間で問題を共有・議論することは非常に困難である。

そこで、我々は、参加者間で問題を共有・議論しやすくするために「対話の破綻(文脈上不適当なシステムの応答)を検出する技術(対話破綻検出)」に焦点を絞り、その技術に関するエラー分析を行うことにした。すなわち、特定の対話システムの内部に立ち入る事は避け、その入出力である表層発話だけを対象とすることにした。これにより、システムの種類・構成に関わらず、多くの対話システム研究者・開発者にとって有益な議論がしやすい。

対話破綻検出は将来の対話システムにとって有用な技術となり得る。たとえば、システムがある発話をするると対話が破綻するという可能性を事前に推定できれば、それが回避できる可能性が高まり、よりよい対話

システムにつながる。また、たとえ不適当な応答を防ぐ事ができなくても、その後のユーザの反応から対話の破綻が検出できれば、対話の話題を変えるなど、エラーからの回復戦略を取ることも可能となる。

対話タスクは以下のステップで進めている。

1. 破綻を含む対話データを整備する。
2. 整備したデータを分析し、対話の破綻の種類とその原因・対処法を考察する。
3. 整備したデータをもとに複数の破綻検出手法を並行して研究・開発する。
4. 開発した検出器を持ち寄り、破綻検出のエラーを分析し、ボトルネックを同定する。

ステップ1は、対話データの収集に関するものである。人間と対話システムの対話で研究に自由に使って良いものは少ない。そのため、今回独自に作成することにした。本稿執筆時点で我々はステップ2の終盤にあり、データの整備・分析と、原因から見た破綻の類型化を行ったところである。今後、対処法を考察しながら、ステップ3の破綻検出手法の研究開発を進めて行く。

対話には、大きくタスク指向型対話と非タスク指向型対話(雑談対話、チャットとも呼ばれる)がある。本タスクでは、対話の破綻を網羅的に分析するために、両方の種類の対話において、対話データを確保し、対話破綻のアノテーションを行うことにした。

以降、2節で対話データについて説明する。今回、タスク指向型対話については、既存の音声対話コーパスからその一部の書き起こしデータについて提供を受けた。非タスク指向型対話については、新たに対話デー

*本稿は、[10]を基に加筆したものである。

[†]higashinaka.ryuichiro@lab.ntt.co.jp

[‡]funakoshi@jp.honda-ri.com

[§]araki@kit.ac.jp

[¶]htsukahara@d-itlab.co.jp

^{||}yuka3.kobayashi@toshiba.co.jp

^{**}masahiro-mi@is.naist.jp

表 1: 対話タスクの参加状況

参加拠点	秋田県立大学, デンソーアイテ ィラボラトリ, 広島市立大学, ホンダ・リサーチ・インスティチ ュート・ジャパン, 京都工芸繊維 大学, 京都大学, 名古屋工業大 学, 奈良先端科学技術大学院大学, NTT, 大阪大学, 首都大学東京, 東芝, 豊橋技術科学大学, Yahoo!, 山形大学
参加者数	32
アドバイザー	篠田浩一先生 (東京工業大学), 小 林哲則先生 (早稲田大学)

タ収集用の雑談対話システムを1つ用意し, 1,146 対話の雑談対話データを収集した。

3 節及び 4 節では, 上記の雑談対話データに対するアノテーションについて述べる。最初に行った 24 名のアノテータによる 100 対話へのアノテーションについて 3 節で説明し, その結果を踏まえて, 残りの 1,046 対話について, 異なりで計 22 名, 各対話約 2 名のアノテータが行ったアノテーションについて 4 節で説明する。5 節では, 4 節で説明した 1,046 対話に対するアノテーション結果の分析に基づく, 雑談対話における破綻の原因の類型について議論する。最後に 6 節で対話破綻検出の関連研究を紹介してから, 7 節でこれまでの活動をまとめつつ, いくつかの拠点で進めている破綻検出研究 (ステップ 3) の状況について簡単に紹介する。

2 対話データ

前述のように, 本タスクでは, 対話の破綻を網羅的に分析するために, タスク指向型対話と非タスク指向型対話, 両方の種類の対話において, 対話データを確保し, 対話破綻のアノテーションを行うことにした。

タスク指向型対話については, 既存の人・人あるいは人・模擬システム間の対話を, 機械的にシャッフルする処理を行い, 人工的に破綻データを生成することにした。非タスク指向型対話については, 既存の雑談対話システムを用いて, 10,000 システム発話を目標に, 人・システム間の雑談データを収集し, そのデータに対し破綻箇所のアノテーションを行うことにした。

2.1 タスク指向型対話データ

名古屋大学武田一哉先生のご厚意により, 「名古屋大学 CIAIR 車内音声データベース (以降, CIAIR-ICSD)」から 30 人, 60 対話の書き起こしデータを利用する許可を頂いた。

CIAIR-ICSD は, 6 つのデータセットからなる。本データにおいて, 実験参加者は, 人・オペレータ対話 (dl), 人・模擬システム対話 (wz), 人・システム対話 (lg) の 3 種類の対話を行っている。今回, 30 人の実験参加者のそれぞれについて dl と wz の 2 対話ずつを抽出し, 計 60 対話を利用することにした。以下は, dl データの抜粋である。ここで, O はオペレータ, D はドライバを表す。

O: はいどうぞ

D: はい

D: ああファーストフードのお店を探してるんだけど

D: どうしようかな

O: はいこの近くですとマクドナルドミスタードーナツロetteriaがございませう

D: じゃあマクドナルドでお願いします

O: はいそれではマクドナルドまでご案内いたします

O: 次の次の交差点二つ目の交差点を左です

現在, 破綻対話箇所を自動生成することの妥当性などを議論中である。破綻箇所を生成した後, 対話破綻箇所の検出技術を評価するためのデータとして利用していく予定である。

2.2 非タスク指向型対話 (雑談対話) データ

NTT ドコモが一般公開している雑談対話 API[1, 11]を用いた雑談対話システムと対話が可能な雑談対話収集サイト [2] を, 拠点の一つである広島市立大学内の Web サーバに設置した。このサイトでは, ユーザが 10 発話を入力すると対話が終了し, 対話ログが出力されるようになっている。雑談対話収集サイトのスクリーンショットは図 1 の通りである。

各拠点のユーザにこの雑談システムと対話をしてもらい, 全部で 1,146 対話を収集した。対話をする際には, 付録にある「対話ガイドライン (10 箇条)」に沿うように教示をした。これは, 現状の雑談対話システム技術が成熟していないこともあって, 破綻の対話ばかり収集されてしまわないためである。話者毎の対話数は 1 から 40 までばらつきがあるが, 全体の対話者は 100 名を超えている。表 2 は収集された雑談対話デー

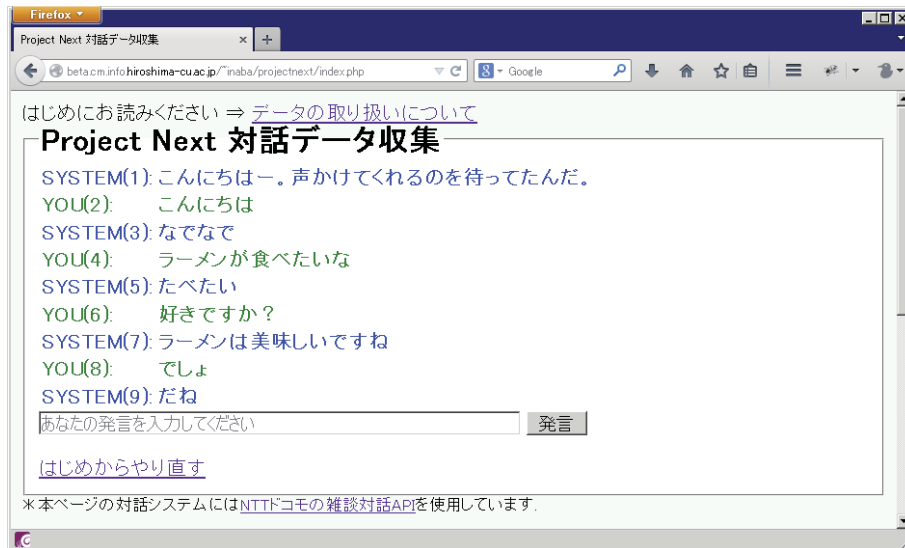


図 1: 雑談対話収集サイトの画面

タの統計情報である。収集した雑談対話の一例を以下に示す。Sはシステム発話、Uはユーザ発話を表す。

- S: こんにちは！声かけてくれるのを待ってたんだ。
 U: 今日は最高気温 36 度だって。暑いねえ
 S: 最高気温は 17 度が予想されます??
 U: いやいや猛暑ですよ
 S: 猛暑に備えていますか?
 U: うん、日傘さしたり、水飲んだり、クーラー入れたりしてる
 S: クーラーは快適ですね
 U: うん、会社だと朝はつけられないから死にそう
 S: 会社は冷房が効きますよねえ
 U: 昼間は効くね。ここはパソコンがいっぱいあるからないと大変
 S: 動きが悪いです
 U: そうなの？こっちは快適だよ
 S: パソコンは使い勝手が悪いです

3 初期アノテーション

収集した雑談対話データに対して対話破綻の初期アノテーションを行った。まず、1,146 対話からランダムに 100 対話を抽出し、全 100 対話のアノテーションを行える 24 人によってアノテーションを行った。このアノテーションの目的は、残りの 1,046 対話に対して、1 対話あたり何人のアノテータを割り当てるのが

表 2: 収集された雑談対話データの統計情報

対話数	1,146	
話者数	116	
	ユーザ	システム
発話数	11,460	12,606
異なり発話数	10,452	7,777
単語数 (形態素数)	86,367	76,235
異なり単語数	6,262	5,076

形態素解析には MeCab の version 0.996 を使用した。システム発話数は 12,606 となっているが、これは最初のプロンプト発話の 1,146 とユーザへの返答の 11,460 を合わせた値である。

妥当かを検討するためである。ここで作成したデータののことを **init100** データと呼ぶ。

アノテーションについては、どのようなエラーがあるのかを網羅的に分析したいという目的に鑑み、トップダウンな破綻の分類は示さず、直感に従って○・×・△の3分類でアノテーションするように指示した。それぞれの意味は以下の通りである。

- **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。



図 2: 雑談データ用破綻アノテーションツール

表 3: init100 データ中の ○△× の発生割合 (発生数)

○	△	×
59.2% (14212)	22.2% (5322)	18.6% (4466)

アノテーションには、図 2 に示す専用のツールを使用した。ツールでは、非文のチェックの他に、各発話に対してコメントを記入できるようになっている。また、先行する文脈のみに基づいて対話破綻のアノテーションが出来るように、1 発話アノテーションする毎に、次のユーザ発話とシステム発話が表示されるようになっている。なお、破綻とタグをつけた後の発話をどうアノテーションするかについては、対話の先頭から、破綻とタグ付けされた発話を含むこれまでの文脈を「ありき (与えられたもの)」として、アノテーションするように教示した。

3.1 アノテータ間の一致度の分析

init100 データに対して、24 人のアノテータが付与したラベル ○, △, × の割合を表 3 に示す。24 人のアノテータ間の一致の程度を測るために Fleiss の κ を算出すると、0.276 であった。この値の解釈は「ランダムではないが、よく一致しているともいえない」とするのが妥当である。△ を × に含めて、2 値のアノテーションとして計算すると、0.396 とやや一致の具合が高まる。△ を ○ に含めると κ は 0.277 にしか改善されないため、△ は × により近いことが分かる。

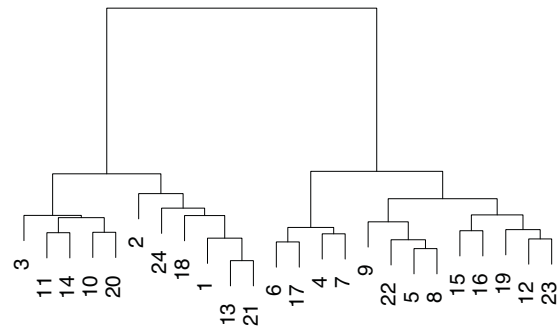


図 3: アノテータのクラスタリング結果

24 人のアノテータを Cohen の κ 値をもとに Ward 法で階層クラスタリングを行うと、図 3 のようになった。距離の定義やクラスタリングの手法を変えると、2 つのクラスタの中でのまとまり方は細かく変わるものの、大きな 2 つのクラスタ間での移動はほとんど見られなかった。図 4 に示す 24 人のアノテータの分布を見ると、○ をつける傾向の大小で、前述の 2 クラスタが分かれていることが見て取れる。2 つのクラスタの中での κ 値を求めると、それぞれ 0.414 (11 人) と 0.474 (13 人) であった。

24 人のアノテータからランダムに N 人を選び出したとき、ラベルの分布がどれだけ全体の分布から離れているのかを表したグラフを図 5 に示す。横軸は N の数で、縦軸は Kullback-Leibler divergence の対称平均の値である。黒丸が 1,000 回サンプリングした際の平

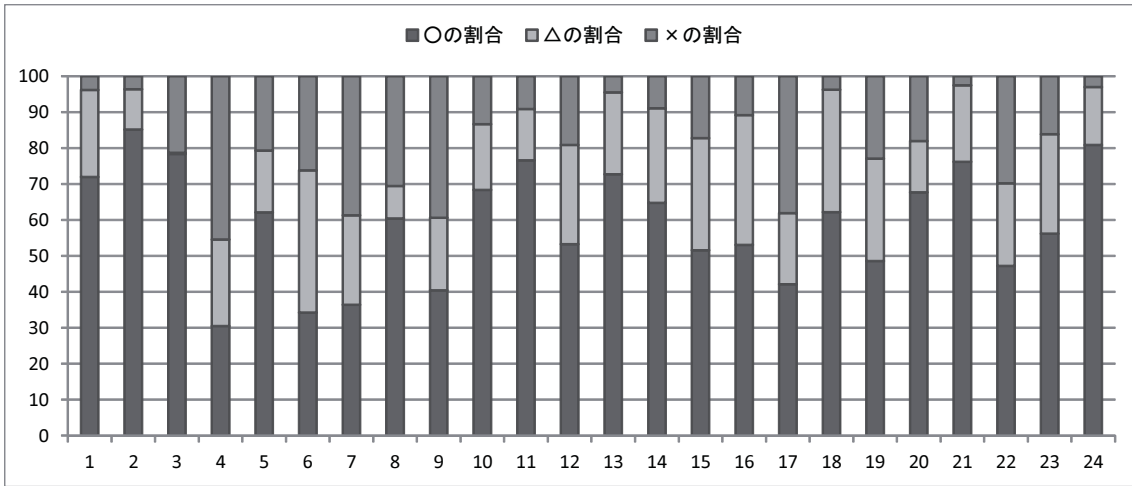


図 4: アノテータ毎の ○△× を付与した割合. 横軸はアノテータ ID.

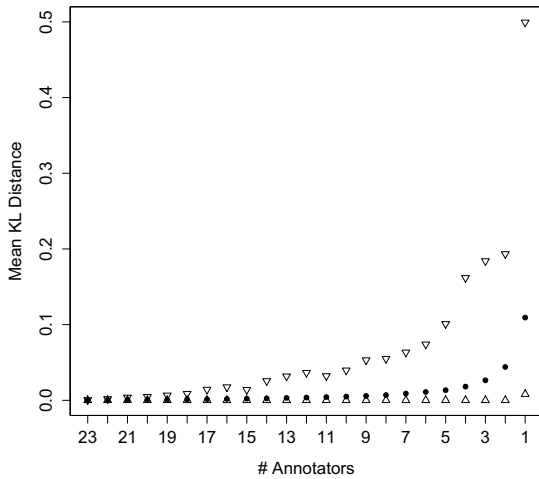


図 5: 24 人のラベル分布とランダムサンプリングした N 人のラベル分布の異なり

均値を示す。下向き三角は 1,000 回中の最大値, 上向き三角は 1,000 回中の最小値を表す。アノテータが 1 人から 2 人になる段階で, 平均値からの乖離は半分近く縮まり, あとは, 自然に 24 人の分布に近寄っていくことが分かる。

○の数と △と ×の数が半分ずつアノテーションされた発話の分析も行った。すると, 表面的には間違っているとは言えないけれども, 違和感がありそうな応答が多かった。「こういうことは普通ここでは言わない」という印象は持つものの, なぜそれを言わないかの説明が難しいものが多い。また, 発言の意図が読みにくそうな発話, ユーザの話聞いていないと感じられる発話, 社会的に問題がある発話, ふざけているように

も見える発話なども判断が割れていた。これらの判定を分析することは, 対話における認知の研究として学術的に面白い課題となり得る。以下は, 評価が割れた発話 (下線部) を含む対話の例である。

U: 他に好きな楽器はありますか?
 S: サックスがいいですね
 U: サックスは木管楽器なんですよ
 S: 相性がいいですね
 U: 台風は大丈夫でしたか
 S: 台風は必要です
 U: 雨は必要ですよ
 S: 朝から雨が降るので

これは発話の意図が読みにくいから, 「それがどうしたのか」という印象を受ける発話である。

U: 夏って感じがします
 S: 海は騒がしいですよ
 U: 海に行くと盛り上がりますもんね
 S: 海は静かなんですよ
 U: 騒がしいのか静かなのかどっちですか
 S: 海はサイコーですよ
 U: サイコーですね

これは人によっては, 質問に答えずにごまかしているように感じられるだろう発話である。

3.2 コメントに基づく破綻の分類

init100 データ中の一部の対話について, アノテータによって対話破綻箇所が付与されたコメントを手で

表 4: rest1046 データ中の○△×の発生割合 (発生数)

○	△	×
58.30% (13363)	25.33% (5805)	16.37% (3752)

分類したところ、大きく「同じ内容の繰り返し」「矛盾した内容」「発話として唐突」「ユーザ発言内容の無視」「質問に答えていない」に分類できることが分かった。

同じ内容の繰り返し 表層としては少し異なっていたとしても内容として同じものを繰り返す場合、破綻とアノテーションされることが多かった。たとえば、「美味しいですね」「いいですね」などと同じような発話を繰り返す場合である。

矛盾した内容 システム発話間で矛盾が見られる場合は破綻とされることが多かった。たとえば、「イチゴが好き」という発言の直後に「リンゴが好き」と発言するなど、一貫性を欠く発話は問題視された。

発話として唐突 「おはようございます」に対して「明けましておめでとうございます」のように、文脈とは関係のない発言を突然行うことがあり、このような発話は破綻とされていた。

ユーザ発言内容の無視 対話はお互いが協調して進めていくものであるため、ユーザ発話を全く受けずにシステムが発話を行った場合には対話の破綻とみなされることが多かった。たとえば、旅行の話をしていて「車で行きましょう」とユーザが話しかけたのに「車はかっこいいですね」と車そのものについて言及したりする場合である。

質問に答えていない ユーザ発言内容の無視に近いが、特に質問に答えていないものが破綻とされていた。たとえば、「チワワは欲しいですね」とシステムが話し、それに応じてユーザが「飼う予定はあるの?」と質問したが、システムは「チワワはいらしいですよ」と答えたような場合である。

上記以外にも口調の唐突な変化などが、問題のある現象として観察された。さらに詳しい分類については5節で述べる。

4 残りの対話へのアノテーション

init100 データに対するアノテーション結果について、タスク参加者で議論を行った結果、残りの1,046

対話 (以後、rest1046 データと呼ぶ) のアノテーションについては、1対話につき2人で実施するという結論に至った。2名とした理由は以下の通りである。

- 人的・経済的コストの面から、アノテーションにかかる作業量は最小限が望ましい。
- アノテーションのコストを最小化できるのは1名でアノテーションを行う場合であるが、この場合、アノテータ間の揺れのために、破綻とされるべき発話が見逃されてしまう可能性がある。よって、複数名が望ましい。
- 前述の分析でアノテータは大きく2つのクラスタに分かれることが分かっている。これらの2つのクラスタから1名ずつ割り当てることによって、見逃しを最も効率的に減らせる可能性がある。

1,046対話をランダムに11個のサブセット(a-k)に分割した。a-jの10個のサブセットはそれぞれ100対話を含み、最後のサブセットkだけが46対話を含む。

アノテーションには、22名のアノテータの協力が得られることになった。22名のうち19名が、init100データに対するアノテーションに参加していたアノテータである。まずこの19名について、図3のクラスタに基づき、2つの大クラスタからなるべく1名ずつのアノテータが割り当てられるように、サブセットkを除く10サブセットに割り当てた。その後残りの3名を同10サブセットに割り当てた。1名当りの分担当量を2サブセットと固定して22名を10サブセットに割り当てたので、i,jの2つのサブセットだけ3名のアノテータを割り当てた。サブセットkについては、余力のある2名に割り当てた。

アノテータが各対話にアノテーションを行う方法は、init100の場合(3節)と同じである。アノテーションの結果の分布を表4に示す。init100よりも、△の割合が増えているが、△と×を併せて見た場合には、init100のときとほぼ同じ分布と考えられる。また、各サブセット毎のFleissのκ値を表5に示す。

rest1046データのアノテーションに際しては、担当する対話の最初の5対話と最後の5対話、計10対話だけ、△、×をつけた箇所には、必ずその判断理由をコメントとして書くことを求めた。これにより、総数で3,748個、異なりで2,468個のコメントを得た。

5 対話破綻の類型化

△、×の破綻アノテーションに付随するコメントを分析し、雑談対話における対話破綻の類型化を行った。

表 5: サブセット a-k 毎の Fleiss の κ 値 (i,j のみ 3 名でのアノテーション, その他は 2 名づつ)

サブセット	a	b	c	d	e	f	g	h	i	j	k	計
対話数	100	100	100	100	100	100	100	100	100	100	46	1046
○の数	1271	1159	1222	1174	1186	693	1150	975	2162	1781	590	13363
△の数	550	522	474	258	400	732	543	633	567	863	263	5805
×の数	179	319	304	568	414	575	307	392	271	356	67	3752
κ	0.31	0.38	0.19	0.30	0.37	0.36	0.23	0.14	0.24	0.29	0.27	0.28*

(*マクロ平均)

表 6: 類型化原案

	誤り	矛盾	冗長	ずれ
発話	非文	矛盾を含む発話		その他の破綻
応答	発話対制約違反	矛盾を含む応答	直前発話と全く同じ内容	同上
文脈	文脈から逸脱した発話	文脈的情報との矛盾	執拗な繰り返し	同上
環境	一般常識に反した発話	一般常識との矛盾		同上

具体的には、以下の手順に基づき、類型化案を作成した。

- 3.2 節の分析結果を叩き台とし、類型化原案を作成
- 類型化原案に基づいて、コメントを分類
- 分類結果を基に議論し、類型化案を改善

5.1 類型化原案の作成

5.1.1 コメントの分類の方針

対話が、ある発話によって破綻するとき、原因はその発話だけにあるとは限らない。もちろん、その発話が文法的におかしなものであったり、意味がわからなかったりする場合もある。しかし、その発話が文として正しいものであったとしても、「相手の発話に対して、このように応答するのはおかしい」場合や、「前に言ったことと矛盾している」という場合においても、対話の継続が困難となる。このように、対話の破綻を分析するに当たっては、当該発話そのものに原因があるのか、または広い意味での文脈（直前の発話、対話履歴、状況なども含む）に原因があるのかを特定する必要がある。

また、破綻が生じた原因が存在する範囲が同じであっても、その内容は様々である。必要な情報の欠落や曖昧性のために意味が特定出来ない場合や、意味が特定できても文脈と矛盾する場合、矛盾はしなくても冗長な場合などがある。

従って、対話破綻分類の方針として、データのどの範囲を見て破綻と判断したかを大分類、破綻の内容を小分類として、表 6 に示した 2 次元で分類するものとした。

まず、破綻の根拠となっている情報に基づき大分類を決定し、その後、破綻の種類を表す小分類を決定した。この分類に当てはまらない、もしくは複数の分類に当てはまると思われる破綻については、その都度、問題点を記録しておくものとした。

5.1.2 大分類の詳細

大分類は、破綻を認定する際に、どの範囲に関連した破綻であるかという基準で分類を行った（図 6 参照）

- 発話
 - 当該システム発話のみから破綻が認定できるケース。典型的には非文が該当する。「意味不明」というコメントの場合でも、この発話単独で意味がわからないのではなく、前の発話や文脈との関係で意味が取れない、というケースがあるので注意した。
- 応答
 - 直前のユーザ発話と当該システム発話から破綻が認定できるケース。典型的には、発話対制約違反や、前発話の話題を無視した応答などが該当する。あくまでもそれまでの対話の流れは無視して、1つ前の発話との関係だけで判断した。

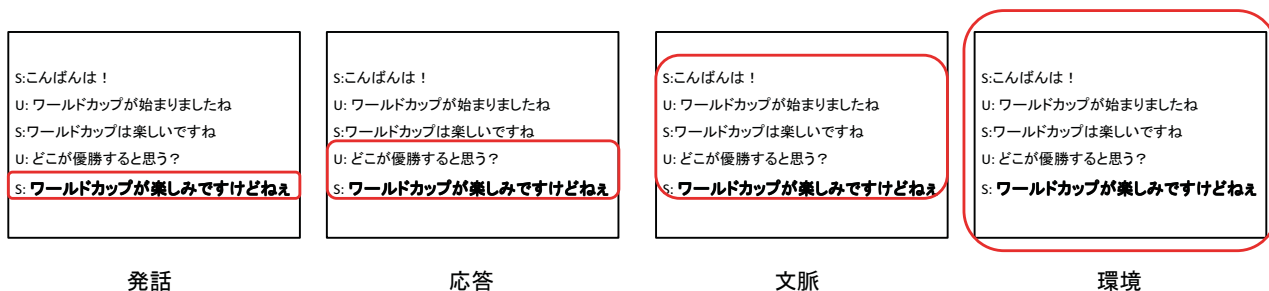


図 6: 大分類を決める基準（範囲の違いを模式化した図であり、図中の発話は必ずしも各ケースに実際に該当する発話ではない）。太字は破綻と認定された発話。

● 文脈

対話開始時点から当該システム発話までの情報から破綻が認定できるケース。典型的には、対話の流れから判断できる不適切な発話・矛盾する情報の提供・不要な繰り返しなどが該当する。

● 環境

破綻原因が、上記の3分類には当てはまらないケース。典型的には、一般常識に反するシステム発話が該当する。

5.1.3 小分類の詳細

小分類は、破綻の内容について分類を行う。

● 誤り

先に認定した大分類の単位を成立させている規則に違反しているもの。

- 「発話」：構文制約，意味制約に違反
- 「応答」：発話対制約，発話の結束性 (cohesion) に違反
- 「文脈」：対話の首尾一貫性 (coherence)，焦点維持に違反
- 「環境」：一般常識，コミュニケーションの制約に違反

● 矛盾

認定した大分類の範囲内で、矛盾する情報が提供されている場合。例えば、あるものが好きであると言った後に、それが嫌いである、というようなケース。なお、「発話」単位での矛盾はほとんどないと思われる。

● 冗長

不要な繰り返しがなされているケース。大分類

表 7: 分類結果

	誤り	矛盾	冗長	ずれ	総計
発話	274	1	2	37	314
応答	1,466	65	32	32	1,595
文脈	760	199	183	29	1,171
環境	96	12	0	6	114
総計	2,596	277	217	104	3,194

「文脈」においては、適度な繰り返しは対話の要素として成立するので、執拗な繰り返しに限定する。

● ずれ

上記以外の破綻。適切な類型化が可能であれば、記録しておく。

5.2 対話破綻アノテーションの分類

表7に分類結果を集計したものを示す。ただし、これらは全コメントをカバーするものではない。

この結果から、大分類に関しては「応答」・「文脈」が大半であるという、妥当な結果が得られた。また、「発話」・「環境」もある程度存在することから、この大分類の設定は、ある程度妥当なものであると結論づけることができる。

一方、小分類は「文脈」以外の大分類で「誤り」の数が圧倒的に多く、さらなる分類の必要性があることがわかった。分析の過程で得られた細分類候補には、必須要素の欠如・曖昧な発話・解釈の誤り・共有信念レベルの誤り・話題の飛躍・相手発話の無視など、大分類の結果に依存するものが多く、表6の類型化原案で提示された2次元による分類の見直しを行うこととなった。

類型化が今回データ収集に利用した雑談対話システムの振る舞いのみに依存しないように、破綻の原因

の範囲と、その破綻内容を組み合わせて類型化を行うことは、重要である。しかし、破綻内容の一般化にこだわるあまり、実際のカテゴリ分け作業が困難になってはいかない。

これらの検討の結果、提案する類型化案は、大分類毎に適切な小分類を設定することとした。

5.3 対話破綻の類型化案

前節で述べた検討の結果、対話破綻の類型化案を表8のように提案する。大分類は維持し、それぞれの小分類に対して、適切な小分類を設定した。特に、「応答」・「文脈」の大分類においては、対話における協調の原則である Grice の公準 [4] に基づき、システム側のエラーに起因する破綻を追加する方針で、小分類を設定した。

5.3.1 「発話」の小分類

基本的に、システム発話の生成過程に問題がある場合が該当する。「非文」の概念に近い。

- 構文制約違反
必須格の欠如や助詞の誤りなどが原因で、発話内容が理解できないもの。
例)「熱中症に気をつけないな」、「なにができませんよねー」
- 意味制約違反
意味素性の異なる語の組合せなどが原因で、発話内容が理解できないもの。
例)「仕事は真面目ですね」、「便乗に重ねますよねえ」
- 不適切発話
発話として機能を果たさないもの。
例)「みんっ」、「ちゅっちゅ」

5.3.2 「応答」の小分類

前発話との関係において、協調の原則が守られていない破綻に関しては、Grice の公準に基づく分類を行う。また、それ以外の何らかの処理において生じたエラーについての破綻をこれらに追加する。

- 量の公準違反
前発話に対して、応答に含むべき内容が欠落している、あるいは、不要な内容が含まれており、

これが会話の進行を阻害する場合。

「ラーメンが好きです」→「豚骨で有名ですね」

- 質の公準違反
前発話と矛盾する内容の発話。前発話の話題に追随しながらも、前発話で示された命題内容と矛盾する命題内容を含む場合。なお、意味理解の失敗などによって生じる「ずれ」を含む発話は、関係の公準違反とする。
例)「最近、山登りを始めました」→「山登りが出来ないのですか？」
 - 関係の公準違反
発話対として不自然な発話で、これによって会話の継続ができなくなっている状況。あるいは、前発話の意味内容と関係しない内容の発話を行った場合。相手の発話を無視している状況を含む。
例)「それでは、趣味はなんですか？」→「最後に旅行されたのはいつですか？」
 - 様態の公準違反
発話意図の曖昧性のため、会話の継続ができなくなっている状況。
例)「退屈は嫌いですか？」→「退屈は心ですね」
 - 誤解
内容の解釈エラーによって、不適切な発話になっている状況。多義語の意味の取り違いなどが該当する。
例)「スイカを食べたらお腹を壊します」→「スイカで買い物できてるのですか？」
- #### 5.3.3 「文脈」の小分類
- 基本的には、「応答」の小分類と同様の考え方で、協調の原則に対する違反と、エラーによって生じた破綻で分類する。
- 量の公準違反
与える情報が何もない無情報発話、あるいは、これまでの文脈に出現した発話の不要な繰り返し。
例) 海に行っていないという文脈において:「海に行かないなあ」
 - 質の公準違反
これまでの文脈と矛盾する内容の発話。また、これまでの話題に対する Positive/Negative の極性が、急に転換する発話。
例) 人間関係が薄いという文脈において:「人間関係は深いです」

表 8: 検討後の類型化案

大分類	小分類	内容
発話	構文制約違反	構文的な誤り
	意味制約違反	意味的な誤り
	不適切発話	発話としての機能を持たない
応答	量の公準違反	応答としての情報の過不足
	質の公準違反	前発話との矛盾
	関係の公準違反	発話対を形成しない応答（意味的側面も含む）
	様態の公準違反	発話意図が曖昧
	誤解	内容の解釈エラー
文脈	量の公準違反	繰り返し
	質の公準違反	文脈との矛盾
	関係の公準違反	文脈からの飛躍
	様態の公準違反	発話内容が曖昧
	話題展開への不追随	話題展開を無視
環境	無根拠	受け入れがたい断定
	矛盾	一般常識との矛盾
	非常識	社会規範から外れる発話

● 関係の公準違反

これまでの文脈の意味内容と関係しない内容の発話を行った場合。

例) 服の話の文脈において:「水の色が青緑色に変化なんですよねー」

● 様態の公準違反

発話内容の曖昧性のため、会話の継続ができなくなっている状況。

例) 薬の話題において:「薬は大丈夫ですね」

● 話題展開への不追随

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状況。

例) ドイツの話題の後:「あなたは海外経験ありますか?」→「ドイツは小売価格が高いです」

● 非常識

罵詈雑言など、対話相手との社会的関係を破壊する発話。

5.4 検討課題

今回の分類作業において、複数の大分類に渡る破綻がいくつか見られた。例えば、非文・発話対制約違反・話題からの逸脱のように、複数の大分類に渡る破綻が同時に起こることがあり得る。

また、自分のことなのに伝聞で話すなどの不自然な様態や、対話相手のキャラクタが突然変わるなど、人間同士では起こらないタイプの破綻の扱い対話システム特有の現象も、破綻原因として取り上げておく必要があることがわかった。

5.3.4 「環境」の小分類

会話の文脈以外の原因で生じた破綻を分類する。

● 無根拠

根拠のない、一方的な主張。

例) 「マグロは鮮度が悪いですよええ」

● 矛盾

一般的に正しいと信じられている常識と矛盾する発話。

例) 「熱中症はいいですね」

6 関連研究

タスク指向型の音声対話システムの文脈では、音声認識、発話理解、対話管理などの各モジュールから得られる特徴量から対話に破綻が起きているかどうかを判定する手法がいくつか提案されている。たとえば、Walker ら [8] や Herm ら [5] は、コールセンタにおける通話について、問題が起こっているかどうかを数ターンで判定する判定器を機械学習の手法で構築し

ている。対話中のユーザの満足度の遷移を推定する研究もされている [7]。これらは雑談対話を扱ってはいないが、目的意識は本稿での取り組みと近い。

雑談対話においては、Chai らがユーザの対話行為の系列の情報を用いて、問題のある質問応答ペアかどうかの判別を行っている [3]。Xiang らは、対話行為に加え、感情の系列を用いることで、雑談対話における問題発話の検出を行っている [9]。Higashinaka らも、雑談対話システムの発話の結束性をさまざまな素性から推定する手法を提案している [6]。しかしながら、これらの研究は精度がまだまだ高いとは言えず、また、対話破綻の類型化なども行われていない。対話タスクでは、エラー分析を詳細に行うことで、対話破綻の原因を明らかにし、高精度な破綻検出を実現したいと考えている。

7 おわりに

本稿では、Project Next NLP における対話タスクの目標やステップ、そして、これまでに行った雑談対話データの収集と対話破綻アノテーション、そしてアノテーション時のコメントに基づく破綻原因の類型化について報告した。

本稿の執筆時点において、いくつかの拠点では収集したデータを使って破綻検出に取り組み始めている。例えば、京都工芸繊維大では、2.1 節で述べたタスク指向型対話データを用いた破綻検出に取り組んでいる。ここでは、1 対話中のあるシステム発話を、ランダムに選んだ他の対話中のシステム発話と置き換えることで、破綻を 1 カ所含む 400 個の対話データを作成し、その破綻箇所の検出に取り組んだ。まだ取り組みの初期段階であるが、少なくとも単語の出現を基にした特徴量だけでは 400 カ所の破綻をほとんど検出できず、雑談対話の場合よりも難しい可能性があることがわかって来ている。一方で、生成した破綻箇所のほとんどは、人が見れば容易に見分けることができる。このギャップをどのように埋めるかは、現在の自然言語処理技術の問題点を探るといって Project Next NLP の観点からも、興味深い課題ではないかと考える。

対話破綻のアノテーションがされたデータが多くなれば、そのデータを用いて様々な機械学習の手法が試されるようになるし、また、対話破綻の類型化が進めば、より破綻しない対話システムに向けた指針が明らかになっていくだろう。対話タスクの営みは、現状で特定の対話システムのエラー分析であるが、同じ目的意識を共有しにくい対話システムという分野において、

一定の方向性を与える重要なものであり、これを機に、国内の対話システムコミュニティのさらなる活性化も期待される。

なお、本プロジェクトで作成する対話データおよびアノテーションデータは、当面は、プロジェクトに参加するメンバーに限定して配布するが、一定期間の後、一般に公開する予定である。ただし、CIAIR-ICSD については、公開予定のデータはテキストだけで、音声等については配布元から別途入手が必要なので注意されたい。

謝辞

対話データの収集にご協力下さったみなさま、対話破綻アノテーションにご協力頂いた拠点参加者のみなさま、対話データ収集のためのシステム構築とサーバ運営にご協力いただいた広島市立大の稲葉通将氏、タスク指向型対話データをご提供頂いた名古屋大学の武田一哉先生に感謝いたします。

本稿の著者は、タスク共同リーダー 2 名と、5 節の類型化に直接的に貢献したワーキンググループのメンバーに限っていますが、その他の拠点参加者の方々におかれなくても、電話会議やメーリングリストでの議論を通じて本稿の執筆に様々な貢献していただきました。一人一人お名前を挙げるのは控えさせていただきますが、改めて拠点参加者の皆さまのご協力にお礼申し上げます。

最後に、タスクの中間報告書に対して有益なコメントをいただいたアドバイザーの方々に感謝いたします。

参考文献

- [1] 雑談対話 API. https://www.nttdocomo.co.jp/service/developer/smart_phone/analysis/chat/.
- [2] 雑談対話収集サイト. <http://beta.cm.info.hiroshima-cu.ac.jp/~inaba/projectnext/>.
- [3] Joyce Y Chai, Chen Zhang, and Tyler Baldwin. Towards conversational QA: automatic identification of problematic situations and user intent. In *Proc. COLING/ACL*, pp. 57–64, 2006.
- [4] H. P. Grice. Logic and conversation. In P. Cole and J. Morgan, editors, *Syntax and Semantics 3: Speech Acts*, pp. 41–58. New York: Academic Press, 1975.

- [5] Ota Herm, Alexander Schmitt, and Jackson Liscombe. When calls go wrong: How to detect problematic calls based on log-files and emotions? In *Proc. Interspeech*, 2008.
- [6] Ryuichiro Higashinaka, Toyomi Meguro, Kenji Imamura, Hiroaki Sugiyama, Toshiro Makino, and Yoshihiro Matsuo. Evaluating coherence in open domain conversational systems. In *Proc. Interspeech*, pp. 130–133, 2014.
- [7] Alexander Schmitt, Benjamin Schatz, and Wolfgang Minker. Modeling and predicting quality in spoken human-computer interaction. In *Proc. SIGDIAL*, pp. 173–184, 2011.
- [8] Marilyn Walker, Irene Langkilde, Jerry Wright, Allen Gorin, and Diane Litman. Learning to predict problematic situations in a spoken dialogue system: Experiments with How May I Help You? In *Proc. NAACL*, pp. 210–217, 2000.
- [9] Yang Xiang, Yaoyun Zhang, Xiaoqiang Zhou, Xiaolong Wang, and Yang Qin. Problematic situation analysis and automatic recognition for chinese online conversational system. In *Proc. CLP*, pp. 43–51, 2014.
- [10] 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第72回研究会(第5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-B402, pp. 45–50, 2014.
- [11] 大西可奈子, 吉村健. コンピュータとの自然な会話を実現する雑談対話技術. NTT DoCoMo テクニカル・ジャーナル, Vol. 21, No. 4, pp. 17–21, 2014.

付録：対話ガイドライン (10 箇条)

1. たまたま待合室や飛行機などで隣り合った見知らぬ人と話すイメージで対話しましょう。特定の個人を想定して対話を行わないように注意してください。
2. システムの発話はなるべく好意的に解釈し、対話を続けるように努力しましょう。システムの力量

を試すような発話は控えてください。また、長文の入力や、人間でも答えにくいような難しい問いかけは避けてください。

3. 対話毎に新しい気持ちで話しましょう。残念ながら、システムは前のあなたとの対話を覚えていません。新しい対話セッションでは、前回の対話のことは忘れて対話してください。
4. 自分から話題を開始しましょう。システムは、話題となる単語（主に名詞）があると、その単語に基づいて対話を行うことができますが、そのような単語が見つからないとうまく対話できないことがあります。システムから話題を振られるのを待つのではなく、自分から話題を振りましょう。なお、マニアックな話題にはうまく対応できないことがあります。
5. 自分から話題を変えましょう。システムは、自分から話題を変えることをあまりしません。対話が進展しなかったり、システムがよく分からない応答を繰り返したりするようなときは、自分から新しい話題に変えてみましょう。
6. 何度も同じことを繰り返さないようにしましょう。システムの理解力が対話中に突然改善することはありません。ある発話について満足のいく回答が得られなくても、システムに対して、同じ質問や発話を繰り返したりしないようにしましょう。
7. 誹謗中傷は控えましょう。ここで収集しているデータは公開されます。データを見た人が不快になるような発言は避けて下さい。誹謗中傷とは、システムに対するもの、特定の個人に対するものの両方を含みます。
8. 個人情報を入力しないようにしましょう。公開されるデータであることを念頭において、名前、住所、電話番号などの個人情報は発話に含めないで下さい。第三者の個人情報の入力は絶対にやめてください。
9. 日本語で入力しましょう。システムは日本語しか理解できません。また、大阪弁や博多弁などの方言にも対応していませんので、標準語で入力してください。顔文字・フェイスマークなども理解できませんので使用しないでください。
10. 上記を守っていれば、あとは自由です。対話を楽しみましょう。