人工知能学会研究会資料
SIG-SLUD-B502-07

対話破綻検出チャレンジ

The Dialogue Breakdown Detection Challenge

東中竜一郎 [1*]　船越孝太郎 [2]　小林優佳 [3]　稲葉通将 [4]
Ryuichiro Higashinaka[1]　Kotaro Funakoshi[2]　Yuka Kobayashi[3]　Michimasa Inaba[4]

[1] NTT メディアインテリジェンス研究所
[1] NTT Media Intelligence Laboratories
[2] (株) ホンダ・リサーチ・インスティチュート・ジャパン
[2] Honda Research Institute Japan Co.，Ltd.
[3] (株) 東芝 研究開発センター
[3] Toshiba Corporation
[4] 広島市立大学
[5] Hiroshima City University

Abstract: Detecting breakdowns in dialogue can be one of the promising techniques in dialogue systems. To propel the research and development for dialogue breakdown detection, we organized the dialogue breakdown detection challenge. This paper describes the background of the challenge and the dataset, and overviews the methods and results of the submitted runs of the participants. We show to what extent dialogue breakdowns can be detected by current techniques and discuss limitations and problems to be tackled in the future.

1　はじめに

音声対話エージェントが商用サービスとなるなど，対話システムがより身近になってきた．しかし，システムと人間の間で，人間同士と同じようなスムーズな対話が実現できているとは言い難い．ユーザ入力をうまく解釈できずに不適切な応答をしてしまったり，対話の流れに即していない発言をしてしまったりして，ユーザが何を話してよいか分からなくなる・話したくなくなる「対話破綻」の状況が頻発しているのが現状である．

「対話破綻検出チャレンジ」はこのような対話破綻をシステムが自動的に検出する技術（対話破綻検出と呼ぶ）の確立を目指した評価型ワークショップである．このような技術が実現すれば，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できる．また，たとえ対話破綻してしまったとしても，破綻したことが認識できれば，リカバリを試みることも可能となる．対話システムが普及するためには，少なくとも対話が破綻しないことは大前提であろう．対話破綻検出は今後の対話システムが実際に使われるための核となる技術と言える．

対話システムは，タスク指向型対話システムと非タスク指向型対話システム（いわゆる，雑談対話システム）に大きく分かれるが [1]，本チャレンジでは雑談対話システムにおける対話破綻検出を扱う．タスク指向型対話システムでは，対話内容が特定のタスク・ドメインに依存し，対話破綻の現象も限定されてしまう可能性がある．雑談対話システムは，幅広いオープンドメインの対話を扱うことから，多様な対話破綻を扱うことができ，また，汎用的な対話破綻検出技術に結び付くと考えられる．

これまで，チャレンジとして，ローブナー賞 [1]，DiaLeague [2]，spoken dialogue challenge [3], dialogue state tracking challenge [4], the REAL challenge [5] などが開催されてきた．これらは，ローブナー賞を除き，主にタスク指向型対話システムを対象にしているが，本チャレンジは，雑談対話システムの特定のモジュールに着目した初めての評価型ワークショップである．

今回のチャレンジには 6 チームが参加し，さまざまな対話破綻検出手法が検討された．本稿では，対話破綻検出チャレンジのタスク設定，対象とした対話データ，評価尺度，各チームの手法概要およびその性能比較を行い，対話破綻検出について得られた知見や現在の手法の限界，今後の展開について述べる．

2　タスク設定

対話破綻検出チャレンジのタスクは，ある対話文脈について後続するシステム発話があるとき，そのシステム発話が対話破綻となるかを検出することである．たとえば，以下では S1 から U2 までが対話文脈であり，

*連絡先：higashinaka.ryuichiro@lab.ntt.co.jp

[1] http://www.loebner.net/Prizef/loebner-prize.html

下線が引かれた S3 が後続するシステム発話である．

S1: 買い物は一人が楽です
U1: 確かに気が楽ですね
S2: 買い物は長いです
U2: まあ見るのも楽しいし
S3: 買い物は一緒が楽しいですね

ここで，S3 は S1 の発話内容と食い違ったことを言っており，おそらく対話破綻を引き起こすだろう．すなわち，このような発話について，対話破綻検出器が「対話破綻」であると検出できればよい．

「おそらく」と書いたのは，対話破綻は主観的な要素が強く，一概に正解を決めることが難しいからである．そこで，本チャレンジでは，同じ発話について多数の評価者を用いて対話破綻のラベル付けを行い，ラベルの分布によって対話破綻らしさを表現する．参加者は，対話破綻検出対象の各システム発話について，(1) 破綻かどうかのラベルと (2) 破綻らしさの分布（○，△，×の確率分布）を提出する．そして，対話破綻検出器の評価は，多数の評価者が対話破綻とした箇所を正しく検出できるか，実際のラベル分布を正しく推定できるかといった評価尺度によって行う．なお，上記対話例において，S3 より後にユーザ発話が存在するかもしれないが，今回のチャレンジでは対話破綻を事前に回避することができる技術を主眼としたため，対象発話より後の情報は対話破綻検出に用いないこととする．また，本チャレンジでは，一つの参加チームについて 3 つまで結果（run と呼ぶ）を提出することができる．

3 配布データ

学習・開発・評価用に配布した対話データは，対話破綻検出研究のために新たに収集したものである．NTT ドコモが一般公開している雑談対話 API[2] を用いた雑談対話システムが稼動するウェブサイトを構築し，対話参加者各自にウェブブラウザでサイトにアクセス・対話してもらって収集した．各対話はシステムのプロンプトで始まり，その後ユーザとシステムが交互にそれぞれ 10 回発言したところで強制的に終了となる．

この対話データに対して，

○ **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．

[2] https://www.nttdocomo.co.jp/service/developer/smart_phone/analysis/chat/

表 1: 対話データに関する統計量

	学習用		開発・評価用
	init100	**rest1046**	**dev+test**
対話数	100	1046	20+80
アノテータ数	24	2 or 3	30
○	59.2%	58.3%	37.1%
△	22.2%	25.3%	32.2%
×	18.6%	16.4%	30.6%
Fleiss の κ	0.28	0.28*	0.20
同 κ (△+×)**	0.40	0.40*	0.27

*セット毎の κ 値のマクロ平均（詳細は [6] 参照）
**△と×を一つのラベルとみなした場合

という基準で，アノテータの主観を重視して，システム発話毎に対話破綻に関するラベルを付与した．なお，○，△，×の記号は，配布データ中ではそれぞれ O，T (Triangle の意)，X のアルファベットにより表現されている．本稿ではこれらの表記の両方を用いるため注意されたい．

表 1 に，対話データに関する統計量をまとめた．表中の「アノテータ数」は各対話に対するアノテータの数であり，延べ数ではない．

3.1 学習用データ

学習用のデータは「雑談対話コーパス」として一般公開しており [3]，二つのデータセットからなる．一つは **init100** と呼び，各対話 24 名による対話破綻アノテーションが付与された 100 対話からなる．もう一つは **rest1046** と呼び，各対話 2 名（一部 3 名）による対話破綻アノテーションが付与された 1,046 対話からなる．雑談対話コーパスの構築とアノテーションの詳細については，文献 [6] を参照されたい．

雑談対話コーパスは，自然言語処理分野の分野横断のエラー分析プロジェクトである Project Next NLP[4] の対話タスクにおいて収集したものである．対話データの収集に際しては，対話タスクに参加した研究者の他に，研究者と同じ職場で働く社会人や学生，対話システム研究とは無関係なユーザ（研究者の知人）からも協力を得た．学習用データに対するアノテーションは全て，エラー分析プロジェクトの対話タスクに参加している研究者および研究者が監督する作業者が行った．

3.2 開発・評価用データ

雑談対話コーパスを収集した際と同様のフローを用い，クラウドソーシングで募集した不特定多数のユーザから新たに 100 対話の対話データを集めた．また，対話破綻アノテーションも同じくクラウドソーシングによって不特定多数のユーザに依頼し，各対話 30 名で行った．対話データの収集には CrowdWorks[5] を用い，アノテー

[3] https://sites.google.com/site/dialoguebreakdowndetection/chat-dialogue-corpus
[4] https://sites.google.com/site/projectnextnlp/
[5] http://crowdworks.jp

ションには Yahoo!クラウドソーシング [6] を用いた．どちらも性別・年齢の統制は行っていない．

学習用データへのアノテーションは専用のツールを用いたが，開発・評価用ではクラウドソーシングのシステム上それができないため，クラウドソーシングのシステムが提供するインタフェースを用いて，専用ツールによるアノテーションに可能な限り近い設定になるようにした．各発話のアノテーションに際して，対話の先頭からアノテーション対象の発話までしかアノテータに見えないようにしているのは，専用ツールによるアノテーションと同じである．このようにアノテーションした 100 対話のうち 20 対話を開発用データ（**dev**）とし，残り 80 対話を評価用データ（**test**）とした．

アノテータの人数が異なるため直接は比較できないが，開発・評価用データでは，アノテーションの一致率に関する統計量（Fleiss の κ）の値が，学習用データよりもやや低くなっている（表 1 参照）．また，○・△・×の分布をみると，○の割合が学習用データに比べて少ないことがわかる．対話システム研究者は現状のシステムの動作原理や限界を知っているためか，平均して一般人よりも破綻に対する評価が甘いようである．

4 評価方法

対話破綻検出器の評価のための確立された評価尺度は存在しない．そのため，今回は有効と考えられる尺度を複数列挙して用いることにした．具体的には，以下の二つの系統の評価尺度を用いる．

ラベル一致系統 :検出器にシステム発話が破綻かどうか O/T/X のラベルのいずれかで出力させ，アノテーション結果の多数決で決めた正解ラベルとの一致を評価する．

分布距離系統 :検出器にシステム発話が破綻かどうかを確率分布で出力させ，評価データ（各発話毎の O/T/X の頻度分布）との近さを評価する．

4.1 ラベル一致系統

各システム発話について，付与された対話破綻ラベルの多数決を取り，該システム発話が破綻かどうかのラベルを一つに決め，評価データと検出器のラベルを比較することで評価を行う．正解ラベルの決定は，閾値パラメータ t を用いて，次のように行う．まず，最も多数を占めるラベルを求める．そして，その多数ラベルが破綻ラベル (T および X) の場合，その割合が閾値 t を超える場合のみ，そのラベルを正解として採用する．そうでなければ正解は O（非破綻）とする．たとえば，ある発話に 2 人が O，3 人が T，5 人が X とアノテーションした場合，多数ラベルは X であり，その割合は $5/(2+3+5) = 0.5$ である．この場合，パラメータ t が 0.5 以下であればその発話の正解ラベルは X となるが，0.5 より大きい場合は O となる．ラベルが 3 種類なので，最多数ラベルの割合は最低でも 0.33 になる．従ってそれ以下では閾値 t は意味を持たない．

具体的な評価尺度として，以下のものを用意した．

- Accuracy : 全ラベルの一致率
- Precision, Recall, F-measure (X) : 破綻ラベル X の検出に関する精度・再現率・F 値
- Precision, Recall, F-measure (T+X) : ラベル T と X を同一の破綻ラベルとみなした場合の精度・再現率・F 値

正解を一つに決定し，それを当てることができるかだけに着目しているため，分かりやすい反面，多数決でよいのかといった問題や，どの t を用いるべきかといった問題が残る．

4.2 分布距離系統

対話破綻ラベルの分布に基づく評価尺度として，以下のものを用意した．

- JS Divergence (O,T,X) : Jensen-Shannon Divergence による分布間の距離
- JS Divergence (O,T+X) : ラベル T と X を同一の破綻ラベルとみなした場合の Jensen-Shannon Divergence による分布間の距離
- JS Divergence (O+T,X) : ラベル O と T を同一の破綻ラベルとみなした場合の Jensen-Shannon Divergence による分布間の距離
- Mean Squared Error (O,T,X) : 分布間の平均二乗誤差
- Mean Squared Error (O,T+X) : ラベル T と X を同一の破綻ラベルとみなした場合の分布間の平均二乗誤差
- Mean Squared Error (O+T,X) : ラベル O と T を同一の破綻ラベルとみなした場合の分布間の平均二乗誤差

ラベル一致系統とは異なり，一つのラベルに決めることはせず，○，△，×の分布をそのまま評価するため，直接的な評価ができると期待できるが，対話システム開発者が興味を持つような，どのくらいの精度で対話破綻が検出できるのかといったことについての値を出すわけではないため，結果が理解しづらいという問題がある．

5 手法概要と結果

ここでは対話破綻検出のための手法として，オーガナイザが用意したベースライン手法と 6 チームが考案し

[6] http://crowdsourcing.yahoo.co.jp

表 2: 参加チームの手法概要

チーム	手法	破綻検出に用いた情報・素性	各 run の違い
team1	RNN, LSTM-RNN	・単語頻度ベクトル（BoW） ・発話応答間の単語の共起頻度ベクトル ・単語頻度・共起頻度ベクトルを Sent2Vec を用いて変換したベクトル	・run1 RNN ・run2 LSTM-RNN
team2	LSTM-RNN	単語を Word2Vec を用いて変換したベクトル	・run1 最初から順に入力する LSTM-RNN と，逆順に入力する LSTM-RNN の 2 種類の RNN を統合 ・run2 単語ベクトルの系列の最後に終端を表すベクトルを付加し，この終端が入力された時点の出力を LSTM-RNN の結果として使用 ・run3 中間層に LSTM の層を 2 つ重ねた RNN を使用
team3	ルール	形態素解析器 kuromoji[7] によって抽出したキーワード	rule1 あるユーザ発話を挟む 2 つのシステム発話の両方に含まれているキーワードが該ユーザ発話に含まれていない場合に破綻ラベルを付与 rule2 ユーザの質問発話直後のシステムの発話に破綻ラベルを付与 rule3 システムの質問発話について破綻ラベルを付与 ・run1 rule1 OR rule2 OR rule3 ・run2 rule1 AND rule2 AND rule3 ・run3 rule1,2,3 で最も性能がよかったもの
team4	Poly kernel SVM	当該システム発話と 1 つ前のユーザ発話の単語ベクトル	・run1 1 次カーネル ・run2 応答の類型のみ 2 次カーネルで残りは 1 次カーネル ・run3 2 次カーネル
team5	6 層 DNN	・対象文と直前の文の対話行為 ・直前の文から予測されたシステムが出すべき対話行為 ・パープレキシティ ・自動評価値 ・システムパーソナリティを尋ねる質問か否かのフラグ	・run1 開発データ **dev** で評価した際に Precision(X),F(X) 値が最高値だったもの ・run2 開発データ **dev** で評価した際に Precision(T+X),F(T+X) 値が最高値だったもの ・run3 run1 と run2 の組み合わせ（run1 で X になったものは X とし，それ以外は run2 で X,T になったものを T とする）
team6	RNN	分散表現の単語ベクトルを NCM,LSTM エンコーダ,Bag-of-Words Embedding,NCM の拡張モデルの 4 種類の方法で変換した情報	・提出された run は一つのみ

(RNN: Recurrent Neural Network, LSTM: Long Short-Term Memory, DNN: Deep Neural Network, NCM: Neural Conversational Model)

た手法について概観する．また，それぞれのチームによる結果について触れる．各チームの手法の詳細については，それぞれの報告を参照のこと．なお，本稿においては，対話破綻検出手法とその結果のみに着目するため，各チームの名称は team1 のように匿名化していることに注意されたい．

5.1 ベースライン

本チャレンジで配布したベースラインシステムでは，Conditional Random Fields (CRF) により破綻検出を行う．対話を発話の系列とみなし，各発話に対してラベリングを行う．CRF の実装には Mallet[8] を用いた．

使用するラベルは破綻検出のための O/T/X のほか，ユーザ発話のみに付与する PREV-O/PREV-T/PREV-X の計 6 種類である．ユーザ発話のみに付与される 3 つのラベルは，直前のシステム発話のラベルによって決定する．つまり，直前のシステム発話のラベルが O の場合，次のユーザ発話に付与されるラベルは PREV-O となる．これは，CRF が直前のラベルを考慮したラベリングを行う手法であることから，1 つ前の破綻検出結果 (2 つ前のシステム発話のラベル) を考慮できるようにするための処理である．本チャレンジでは，検出手法は各システム発話について O/T/X のそれぞれの確率を出力する必要があるが，本手法では付与されたラベルの確率を 1.0，その他を 0 として出力する．

CRF の素性は対象となる発話，およびその直前の発話に含まれる単語とした．形態素解析器には lucene-gosen[9] を用いた．なお，本チャレンジでは破綻検出対象発話以降の発話を使用することはできないため，本手法によるラベリングの際には，対話の最初から破綻検出対象発話までの系列を逐次与えることで結果を得る．学習時は対話全体から学習を行う．

5.2 参加チームの手法概要

表 2 は参加チームの手法をまとめたものである．表からも分かるとおり，さまざまな対話破綻検出手法が試みられていることが分かる．ルールに基づく手法や Support Vector Machine (SVM) に基づく手法のほか，着目すべきなのは，近年学習手法として注目されている深層学習（Deep Learning, Deep Neural Networks）を用いた手法が多いことである．6 チーム中 4 チームが深層学習に基づく手法である．

5.3 結果：ラベル一致系統

ラベル一致系統の評価尺度に関する各チームの結果を表 3 に示す．閾値 t の値を大きくすると破綻とみなされるシステム発話の数が少なくなる．特に $t = 0.8$ 以上では X ラベルになるシステム発話の数が 29 発話となり ($t = 0.3$ なら 251 発話)，破綻システム発話の数が大きく減少する．そこで，ここでは $t = 0.5$ とし，アノテータの過半数が対話破綻としたシステム発話の検

[7] http://www.atilika.org/
[8] http://mallet.cs.umass.edu/
[9] https://github.com/lucene-gosen/

表 3: ラベル一致系統の評価尺度による各チームの結果（$t = 0.5$ の場合）

team	Accuracy	Precision (X)	Recall (X)	F (X)	Precision (T+X)	Recall (T+X)	F (T+X)
team1 run1	0.530	0.277	0.429	0.337	0.758	0.444	0.560
run2	0.628	0.429	0.015	0.029	0.857	0.011	0.022
team2 run1	**0.643**	0.444	0.444	0.444	0.824	0.361	0.502
run2	0.574	0.475	0.237	0.316	0.796	0.468	0.589
run3	0.627	0.443	0.258	0.326	0.836	0.254	0.390
team3 run1	0.460	0.167	0.035	0.058	0.725	0.510	0.599
run2	0.515	0.288	0.556	0.379	0.725	0.510	0.599
run3	0.570	0.221	0.126	0.161	0.628	0.131	0.216
team4 run1	0.599	0.313	0.232	0.267	0.741	0.201	0.316
run2	0.633	0.414	0.146	0.216	0.800	0.103	0.183
run3	0.627	0.380	0.096	0.153	0.820	0.076	0.138
team5 run1	0.515	0.366	0.551	0.440	0.790	0.670	0.725
run2	0.477	0.333	**0.783**	**0.468**	0.760	0.836	0.796
run3	0.442	0.366	0.551	0.440	0.753	**0.849**	**0.798**
team6 run1	0.631	**0.531**	0.086	0.148	**0.926**	0.138	0.240
baseline (t=0.3)	0.491	0.400	0.141	0.209	0.774	0.606	0.680
(t=0.4)	0.478	0.400	0.162	0.230	0.769	0.657	0.709
(t=0.5)	0.481	0.355	0.136	0.197	0.773	0.628	0.693
(t=0.6)	0.618	0.490	0.242	0.324	0.807	0.293	0.430
(t=0.7)	0.634	0.528	0.192	0.281	0.827	0.212	0.337
(t=0.8)	0.630	0.518	0.217	0.306	0.815	0.243	0.374
(t=0.9)	0.625	0.474	0.136	0.212	0.798	0.175	0.287

出を対象とした．$t = 0.5$ では 880 発話中、T ラベル：129 発話、X ラベル：198 発話が検出対象となる．ベースラインは内部の多数決のためのパラメータ t を使用しているため，それを 0.3〜0.9 まで 0.1 刻みで変化させた場合の性能を載せている（$t = 0.1, 0.2, 1.0$ については割愛）．学習データには **rest1046** と開発データ **dev** を用いている．

F-measure に着目すると，team5 の手法が最も優れている．team5 は DNN を用いているが，その特徴は用いている素性にあり，外部データを利用して得られる対話行為や質問分類などの情報が効果を発揮していると思われる．F-measure について最も振るわない結果と見えるのは team6 である．しかし，Precision, Recall の内訳に目を移すと，Precision については team6 が最も優れている．一方で team5 は Precision も高めであるものの，Recall が他の手法にくらべて高い．

図 1 には，t が 0.3〜0.9 の範囲の各チームの性能を Accuracy および F-masure についてプロットしたグラフを示す．t 値を高くしていくとラベル O の割合が大多数を占めていくことになるが，多くの手法はそれに従い Accuracy が高くなっていく．一方で team5 の手法だけは，t 値を高くしていくと Accuracy が下がっていく特徴があり興味深い．F-measure については，t 値を高くしていくと，Accuracy とは逆に，破綻とみなされるラベルが減るためか値が低くなっていく．t が 0.7 以上といった場合に破綻となる発話は，極めて多くの評価者が破綻と認定した箇所だと考えられるが，そのような箇所が検出しきれていないことがこの図から分かる．

5.4 結果：分布距離系統

分布距離系統の評価尺度に関する各チームの結果を表 4 に示す．分布に関しては，team2，team6 の性能が群を抜いて高い．team2 も team6 もどちらも DNN に基づく手法であり，特に，RNN に基づく手法である．文脈から対話破綻を予測するモデルとして RNN が有効であることが示唆される．なお，team6 の出力するラベル分布が評価データと近いにも関わらず，team6 がラベルの一致について高い Precision と低い Recall というアンバランスな結果となっている．また，同様に，team5 はラベル一致系統の評価尺度ではよい結果であったが，分布距離系統の尺度では振るわない結果となっている．ラベル一致系統における評価と分布距離系統における評価の違いがどのように生じているのかについて今後の分析が必要である．

6 まとめと今後の課題

雑談対話システムにおける対話破綻を自動検出することを目的とした「対話破綻検出チャレンジ」を開催し，タスク設定，対話データ，および，各参加チームの手法と結果について概観した．結果から DNN やオープンドメインに対応するための外部知識の利用などが有効らしいことが示唆されるが，今後，各対話破綻検出器の詳細を精査していきたい．また，今回，多くの評価尺度を列挙して用いたが，ラベル一致系統と分布距離系統のどちらが有用であるのかなど検証していきたい．対話破綻の要因は多種多様であり，長い文脈を見るものから，表現レベルのものまでさまざまである [7]．さらなる対話破綻検出手法の改善のため，今後もチャレ

表 4: 分布距離系統の評価尺度による各チームの結果

	JS divergence			Mean squared error		
team	(O,T,X)	(O,T+X)	(O+T,X)	(O,T,X)	(O,T+X)	(O+T,X)
team1 run1	0.200	0.122	0.106	0.104	0.133	0.105
run2	0.375	0.357	0.150	0.200	0.366	0.123
team2 run1	0.118	0.094	**0.058**	0.069	0.108	**0.058**
run2	0.143	0.106	0.076	0.083	0.118	0.075
run3	0.122	0.098	0.064	0.070	0.109	0.065
team3 run1	0.403	0.306	0.178	0.211	0.278	0.143
run2	0.440	0.307	0.313	0.229	0.280	0.289
run3	0.455	0.415	0.231	0.239	0.401	0.197
team4 run1	0.429	0.379	0.226	0.224	0.362	0.191
run2	0.420	0.398	0.190	0.218	0.383	0.152
run3	0.423	0.407	0.186	0.220	0.393	0.148
team5 run1	0.392	0.248	0.245	0.203	0.213	0.213
run2	0.394	0.212	0.297	0.204	0.172	0.271
run3	0.395	0.212	0.245	0.208	0.173	0.213
team6 run1	**0.105**	**0.080**	0.059	**0.060**	**0.090**	0.062
baseline (t=0.3)	0.400	0.266	0.192	0.213	0.234	0.153
(t=0.4)	0.398	0.254	0.195	0.211	0.220	0.156
(t=0.5)	0.399	0.261	0.194	0.212	0.228	0.156
(t=0.6)	0.399	0.339	0.187	0.206	0.317	0.148
(t=0.7)	0.403	0.361	0.181	0.209	0.342	0.142
(t=0.8)	0.402	0.352	0.184	0.208	0.333	0.145
(t=0.9)	0.411	0.375	0.182	0.213	0.357	0.143

ンジを継続して開催していきたい.

謝辞

タスク設定についての有益な議論やデータ収集にご協力いただいた Project Next NLP 対話タスクのメンバーの諸氏に感謝いたします．開発・評価用データの作成にあたっては人工知能学会より特別補助をいただきました．また，対話データ収集には NTT ドコモの雑談対話 API を使わせていただきました．感謝いたします.

参考文献

[1] 中野幹生, 駒谷和範, 船越孝太郎, 中野有紀子, 奥村学（監修）. 対話システム. コロナ社, 2015.

[2] 橋田浩一, 伝康晴, 長尾確, 柏岡秀紀, 酒井桂一, 島津明, 中野幹生. DiaLeague: 自然言語処理システムの総合評価. 人工知能学会誌, Vol. 12, No. 3, pp. 390–399, 1997.

[3] Alan W Black and Maxine Eskenazi. The spoken dialogue challenge. In *Proc. SIGDIAL*, pp. 337–340, 2009.

[4] Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. The dialog state tracking challenge. In *Proc. SIGDIAL*, pp. 404–413, 2013.

[5] Maxine Eskenazi, Alan W Black, Sungjin Lee, and David Traum. The real challenge 2014: Progress and prospects. In *Proc. SIGDIAL*, pp. 209–216, 2015.

[6] 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: 自然言語処理におけるエラー分析, 2015.

[7] 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*, pp. 2243–2248, 2015.

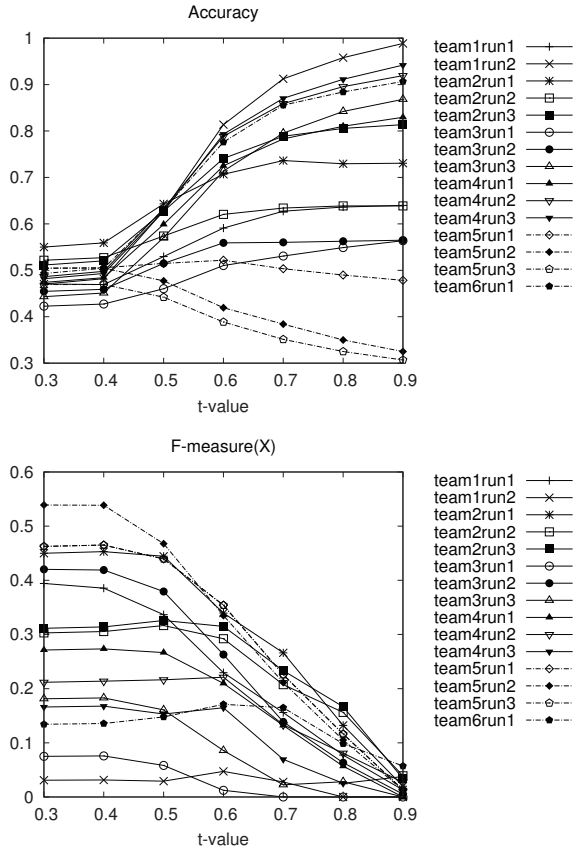

図 1: 参加チームによる各手法の，t を変動させた場合の Accuracy, および，F-measure(X)