Vol.2015-NL-221 No.8
Vol.2015-SLP-106 No.8
2015/5/25

単語情報に基づく雑談対話における破綻の検出の検討

芝原 優真[1]　中野 幹生[2]　Eric Nichols[2]　山本 一公[1]

概要：
Project Next NLP の対話タスクで収集された雑談対話データを対象に，システムの発話が対話の破綻，すなわち不適当なシステムの発話かどうかを検出する手法の検討を行った．検出は，各システム発話が破綻か破綻でないかの 2 値分類をデータから学習された分類器を用いることで行った．分類に用いる特徴量として，当該システム発話に現れる単語の集合に加え，当該システム発話と，その直前のシステム発話やユーザ発話との単語の重なり，感情極性値などを検討した．その結果，単語集合のみを用いた場合より，単語の重なり情報や感情極性値を併用した方が検出精度が若干向上したものの，単語情報のみを用いた手法では限界があることが示唆された．

キーワード：雑談対話システム，対話の破綻，Project Next NLP 対話タスク

Detecting Dialogue Breakdowns in Chat Dialogues based on Word Information

Abstract: This paper presents the results of the investigation into methods for detecting dialogue breakdowns, that is, determining whether each system utterance is appropriate or not. It uses dialogues between human users and chat dialogue systems collected in the dialogue task of Project Next NLP. Detection is done by classifying each system utterance into breakdown utterances and others using a machine learning-based classifier. As the features to be used in this classification, in addition to the bag of words in the system utterance to be classified, we tried co-occurrences of words in the utterance and its preceding system and user utterances, and word polarities. It is found that the detection accuracy improved when using word co-occurrences and word polarities, but it is suggested that there is a limitation in methods using only word information.

Keywords: chat dialogue system, dialogue breakdown, Project Next NLP dialogue task

1. はじめに

ユーザがシステムと自然言語でコミュニケーションを行う対話システムの利用が注目されている．対話システムは，カーナビゲーションなど，達成するべき目的が明確な「タスク指向型対話システム」と，特定の達成するべき目的を設定しない「非タスク指向型対話システム」に大別される．非タスク指向対話では，対話を長く続けることが重要であるが，システムの発話が不適切な場合，対話が破綻してしまい，対話の継続が困難になる．したがって，対話が破綻しそうなシステム発話を事前に検出することで，破綻を防ぐことが重要である．本研究では，破綻を検出する方法を検討する．

タスク指向対話における破綻の検出の研究としては，コールルーティングシステムにおいて最初の数ターンから問題がおこることを予測する研究 [9]，ユーザの反応からシステムの間違った確認要求を検出する研究 [6]，質問応答システムにおいて問題を検出する研究 [1] などがある．これらの手法は，基本的に，問題の検出を機械学習を用いた分類問題として解いており，言語情報や韻律情報など様々な特徴量を用いている．

非タスク指向対話に関しては，Xiang ら [11] が対話行為と感情極性の情報を用いて雑談の中の質問-応答連鎖における問題を検出する手法を提案している．また，Higashinaka

[1] 豊橋技術科学大学
Toyohashi University of Technology
[2] (株) ホンダ・リサーチ・インスティチュート・ジャパン
Honda Research Institute Japan Co., Ltd.

表 1　評価者の評価の比較

		より寛容な評価者			
		○	△	×	合計
より厳しい評価者	○	4,322	334	82	4,738
	△	1,955	901	290	3,146
	×	824	1,049	703	2,576
合計		7,101	2,284	1,075	10,460

ら [5] は様々な特徴量を用いて発話間の結束性を推定する手法を提案している．

本研究では，自然言語処理におけるエラー分析プロジェクトである Project Next NLP の対話タスクで収集された雑談対話データ [3] を用いて，先行研究よりも幅広い範囲の破綻の検出を試みる．タスク参加者によってこのデータにおける対話の破綻が共同で分析されている．このデータは様々なタイプの破綻を含んでいるが，それを詳細に類型化する試みが行われている [4]．その研究が進めば，類型ごとに破綻検出器を設計することで高精度に破綻を検出できる可能性がある．しかしながら，今回は最初の試みとして，破綻かどうかを単純に判定する手法を検討した．

本研究では，最も単純な手法として，単語情報を用いた破綻検出器の検討を行った．まず最初に，破綻かどうかを調べる対象のシステム発話に現れる単語の集合を利用した検出器を構築した．さらに，直前のユーザ発話やシステム発話との単語の重なりや，単語の感情極性の利用の検討を行った．これらの結果を基に，単語情報を用いた破綻検出の可能性と限界，および，破綻検出の性能向上に向けた研究の方向性について議論する．

2. 正解データ作成

Project Next NLP 対話タスクの雑談対話コーパスには，ユーザとシステムとの雑談対話が収集されているが，そのうち rest1046 と呼ばれているセットを用いた． rest1046 には 1,046 対話が含まれている．各対話では最初のシステム発話のあと，10 回のユーザとシステムとのやりとりが行われており，最初のシステム発話を除く 10 個のシステム発話に対し，2 名の評価者によって評価が付与されている．評価は「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（×）」の三段階である．表 1 に，対話コーパスの 1 対話の内容と，評価者による評価を示す．

2.1 評価者による評価の傾向の確認

2 人の評価者の評価の比較を表 1 に示す．2 人の評価者のうち，(×の数 + △の数 × 0.5) の平均の大きい方を，より厳しい評価者とした．

2 者の評価者の評価の一致度を，Cohen's κ で求めた．また，△の評価が○，×のどちらに近いかを確認するため，△を○に含めた場合，△を×に含めた場合の一致度も求めた．表 2 に κ 値を求めた結果を示す．この結果から，どの κ 値も低く，一致しているとは言い難い結果となった．また，△の評価を○または×のどちらかに分類する場合，κ 値から，×に含めたほうが一致率が良くなることが判明した．これは init100 について [3] で言及されていることと同じである．

表 2　二者の評価に対する κ 値

○，△，×の一致度	△を○に含めた場合の一致度	△を×に含めた場合の一致度
0.262	0.241	0.383

κ 値に基づく一致率が低い理由のひとつとして，各対話にアノテーションをする 2 人の評価者を選ぶ際に，予備的な評価で評価が比較的厳かった評価者と比較的寛容であった評価者から一人づつを選んだことがあげられる．

2.2 正解の決定

評価者間の一致率は高くないが，正解を決めないと破綻検出器の評価ができないことから，本研究では次のように正解を決定した．まず，上記の κ 値から，△を×に含めることにした．また，二者の評価者が共に○と評価した発話のみを「破綻ではない」，それ以外の評価は「破綻」とすることにした．二者の評価者が共に△または×とした発話のみを「破綻」とすることも考えられるが，どちらかの評価者のみが△または×とした発話にも何かしらの問題があると考え，「破綻」として扱うことにした．二者の評価者が共に△または×とした発話のみを「破綻」として実験を行うことは今後の課題とする．

2.3 破綻の例

表 3 に破綻と評価された発話の例を示す．太字で表記されているシステム発話が，×と評価されたものである．表 3 の破綻理由とは，太字のシステム発話を×と評価した評価者の評価内訳である．

3. 単語集合を用いた破綻の検出

本研究では，多くの従来研究と同様，破綻検出を分類問題として解くことにする．すなわち，破綻か破綻でないかの 2 値分類問題とする．分類器は機械学習により構築する．

分類器の特徴量として，まず手始めに学習セットに含まれる単語集合を用いることとした．これは，文書分類 [7] や対話システムの発話分類 [2] で用いられている方法である．単語頻度や tf・idf などを用いる方法も考えられるが，各発話が長くないことからあまり変化はないと考え，今回は単語集合のみを用いた．

本研究では，分類器の学習と評価を交差検定で行う．rest1046 は 11 のセットからなっていて，10 セットには 100 対話，1 セットには 46 対話が含まれている．このうち 10

話者	発話	評価コメント	評価結果	本研究での正解
ユーザ	明日の天気はどうなりそうかな			
システム	そうなのー	前の質問に答えられていない; 質問を完全に無視した応答	△, ×	破綻
ユーザ	暑いかな			
システム	晴れが多いそうですよ		○, ○	破綻でない
ユーザ	晴れだと暑くなりそうですね			
システム	晴れは多いそうです	前のシステムの発話の意味ない繰り返し	○, ×	破綻
ユーザ	夏だから. 晴れの日が多いです			
システム	晴れの日は多とか	発話の意図がわからない; 表現がおかしい	△, △	破綻
ユーザ	まあ, それはそんなところで. 気温はどうなるかな			
システム	気温は多いらしい	前の質問に答えられておらず, 日本語がおかしい; 意味的におかしい発話	×, ×	破綻

図 1 Project Next NLP 対話タスクで収集された対話の一例とその評価

表 3 発話の破綻例

破綻理由	話者	発話内容
同じ内容の繰り返し	システム	晴れが多いそうですよ
	ユーザ	晴れだと暑くなりそうですね
	システム	晴れは多いそうです
質問に答えていない	ユーザ	明日の天気はどうなりそうかな
	システム	そうなのー！
ユーザの発話無視	ユーザ	そうですね．午前中に行くつもりです
	システム	お昼から散髪に行くんですか？
以前のシステム発話と矛盾	システム	買い物は一人が楽です
	ユーザ	確かに気が楽ですね
	システム	買い物は一緒が楽しいですね
発話意図が不明な物	ユーザ	雨降ってくれないとお水飲めないですもんね
	システム	雨は得意ですね

セットを学習に用い，残り 1 セットを評価に用いるというプロセスを 11 回繰り返した．

3.1 特徴量

特徴量は，学習データに現れる単語の一つ一つである．各発話にそれらの単語が現れればその単語に相当する特徴の値を 1 とし，それ以外の場合 0 とした．発話を単語に分割する際には形態素解析ツール MeCab[*1] を用いた．辞書は IPA 辞書である．

分類器の構築と評価には，機械学習ソフトウェアである Weka ver. 3.6.8[10] を利用した．分類器は Weka のサポートベクタマシン (SVM) の実装である SMO を利用した．

[*1] http://taku910.github.io/mecab/

表 4 単語集合を用いた場合の分類結果

		予測結果		
		破綻でない	破綻	合計
実際の結果	破綻ではない	2,223	2,099	4,322
	破綻	1,804	4,334	6,138
合計		4,027	6,433	10,460

正解数: 6,557
正解率: 62.7%,
破綻検出の recall:70.6% precision:67.4%，F 値:69.0%

SMO のパラメータはデフォルト値を用いた (カーネルは 1 次の線形カーネルである)．なお，Random Forest など，SVM 以外の学習手法も試したが，大きな差はなかった．

3.2 分類結果

交差検定による分類の結果を表 4 に記述する．なお，すべてのシステム発話を破綻とした場合（最頻ベースライン)，正解数 6,138，正解率 58.7%となる．また，人間の 2 人の評価者の一致は，△と×を同一視すると一致発話数 7,265，一致率 69.5%であり，これが上限であると考えて良い．

3.3 破綻検出に重要な単語

上記の各単語の集合を用いて，どの単語が破綻分類に有効であったかを，Weka の特徴選択機能の一つである InfoGainAttributeEval を用いて調査した．これは，情報量の増加を調べることにより特徴量の重要性を調べる方法である．表 5 にその結果を示す．

また，ある交差検定の一つのテストセット (100 対話) とトレーニングセットの組を用い，特徴量を一つずつ取り除いた場合に判別結果がどうなるかを調査した．表 6 にその結果を示す．表 6 には，すべての特徴量を用いて学習された SVM の判別式における係数の符号から，各単語が破綻

表 5 単語の重要度確認（重要度上位 20 単語）

特徴量	information gain
ね	0.010208
うん	0.007359
を	0.006306
楽しい	0.004256
よう	0.003919
ます	0.003865
有名	0.003296
に	0.003259
よ	0.003166
の	0.003132
ない	0.003132
らしい	0.003052
美味しい	0.002486
いい	0.002471
てる	0.002303
はい	0.002272
好き	0.002262
心	0.002234
哲学	0.002164
大好き	0.002059

である発話に現れやすいのか，破綻でない発話に現れやすいのかを調べた結果も示す．

表 6 単語を取り除く前と取り除いた後の比較（重要度上位 20 単語．出現傾向の値は+が破綻発話に現れやすい単語，− が破綻でない発話に現れやすい単語）

特徴量	特徴量を取り除いた時の分類正解数の増減	出現傾向
ね	-12	−
有名	-12	+
？	-9	−
！	-9	+
気持ち	-9	−
です	-8	+
はい	-8	−
一	-8	−
え	-8	+
そういう	-8	+
塩分	-8	+
ます	-7	+
見	-7	+
いい	-6	−
最高	-6	−
湿度	-6	+
てる	-6	+
よかっ	-6	−
散歩	-6	+
DVD	-6	+

表 5 から，「ね」「よう」「ます」のような文末に使用される単語や，「楽しい」「有名」のような肯定的な単語が重要であることが判明した．

表 7 単語集合と単語重なりを用いた場合の分類結果

		予測結果		
		破綻でない	破綻	合計
実際の結果	破綻でない	2,223	2,099	4,322
	破綻	1,778	4,360	6,138
合計		4,001	6,459	10,460

正解数: 6,583
正解率: 62.9%,
破綻検出の recall:71.0% precision:67.5%，F 値:69.2%

表 6 に現れた「塩分」「湿度」のような単語は，たまたま今回用いたコーパスに多く現れたと考えられる．そのため，いつもこれらの単語が有効とは考えにくく，破綻検出のためのより一般的な特徴量を発見する必要がある．

4. 特徴量の追加

破綻の分類を基に，特徴量の追加を検討した．本研究では，着目した発話より以前の発話との単語の重なり，発話に含まれる単語が持つ感情極性値を追加し，分類精度の変化を調べた．

4.1 単語重なりを扱った特徴量

破綻かどうかを調べるには，文脈情報を用いることが有効だと考えられる．話題を特定しない雑談対話において，文脈を適切にモデル化することはまだ容易ではない．そこで，簡単な方法として，当該システム発話とそれ以前の発話の単語の重なり情報が利用できないかと考えた．それにより，同じ内容の繰り返しや，矛盾した内容の発話の検出が行えることを期待した．

加えた特徴量を以下に示す．

- ひとつ前のシステムの発話と現在のシステムの発話との重なり単語数
- ふたつ前のシステムの発話と現在のシステムの発話との重なり単語数
- 直前のユーザの発話と現在のシステムの発話との重なり単語数

分類実験を行った結果を表 7 に示す．

単語集合情報のみを利用した結果と，単語重なりを追加した結果とを比較したものを表 9 に示す．分類精度はわずかに向上しているものの，正確二項検定による有意差検定を行ったところ，片側検定結果が $p = 0.13$ であり，統計的に有意な差はなかった．

表 8 に単語重なりを追加することによって正しく分類した発話例，誤分類した発話例を示す．単語重なりを特徴量に追加したことで，重なった単語が多い場合破綻と分類され，重なりが少ない場合破綻ではないと分類されている．これにより，以前の発話内容と同じ内容の繰り返しを破綻と分類することができるが，単語の重なりが多い場合，適

表 8 単語重なりの特徴量を追加による破綻検出への影響例

破綻であると正しく分類した発話	システム	待ち時間は暇です ね
	ユーザ	何して待ちますか.
	システム	待ち時間は暇です
破綻であると誤分類した発話	システム	花火大会はいいらしい
	ユーザ	風流ですよね
	システム	花火大会は 浴衣が いいらしい

表 9 単語重なり特徴量の追加による分類精度の変化

		単語集合のみ		
		正しく分類	誤分類	合計
単語集合 + 単語重なり	正しく分類	6,327	256	6,583
	誤分類	230	3,647	3,877
合計		6,557	3,903	10,460

切な発話であったとしても破綻と分類されることがある.

4.2 感情極性値を扱った特徴量

分類しようとしているシステム発話がポジティブかネガティブか，また，ポジティブ，ネガティブの度合いがそれ以前の発話から変化しているかどうかを見ることで発話内の矛盾や，以前のシステム発話の内容との矛盾による破綻を検出できることを期待し，感情極性値を用いた特徴量を追加した.

感情極性値とは，単語がポジティブな意味を持つものなのか，ネガティブな意味を持つものなのかを数値的に表したものである．今回は，単語感情極性対応表 [8] を使用した．この辞書は，感情極性値を-1 から+1 の実数値を割り当て， -1 に近いほどネガティブ，+1 に近いほどポジティブとしている.

感情極性値に基づく特徴量として，以下を用いた.

- システム発話に含まれる単語の感情極性値の最大値
- システム発話に含まれる単語の感情極性値の最小値
- システム発話に含まれる単語の感情極性値の平均値
- システム発話に含まれる単語の感情極性値の最大最小の差
- システム発話に含まれる単語の感情極性値の平均値と，直前のユーザ発話に含まれる単語の感情極性値の平均との差
- システム発話に含まれる単語の感情極性値の平均値と，1 つ前のシステム発話に含まれる単語の感情極性値の平均との差

分類の結果を表 11 に示す.

単語集合のみを利用した結果と，感情極性値を追加した結果とを比較したものを表 12 に示す．分類精度は若干低下した.

表 10 に感情極性値を追加することによって正しく分類した発話例，誤分類した発話例を示す．感情極性値を特徴量に加えることによって，以前の発話との矛盾を検出することができた．表 10 の「破綻であると正しく分類した発話」では，以前のシステム発話と現在のシステム発話との感情極性値の平均の差が大きくなったため，破綻と分類することができた．しかし，「破綻であると誤分類した発話」では，以前のユーザ発話と現在のシステム発話との感情極性値の平均の差が大きくなってしまい，正しく分類することができなかった．対話では一つ一つの発話が短いため，一つの単語の感情極性値に大きく左右されてしまうという問題がある.

表 10 感情極性値の特徴量の追加による破綻検出への影響例 (括弧内は感情極性値)

破綻であると正しく分類した発話	システム	荷物 (-0.915009) は最低限がいい (0.999314) ですよねー
	ユーザ	何を持っていくつもりですか
	システム	荷物 (-0.915009) は無理 (-0.921332) です
破綻であると誤分類した発話	システム	こんちわ
	ユーザ	階段 (-0.678903) は好き (0.93204) ですか？
	システム	階段 (-0.678903) は欲しい (-0.795071) ですね

表 11 単語集合と感情極性値の両方を用いた場合の分類結果

		予測結果		
		破綻でない	破綻	合計
実際の結果	破綻でない	2,208	2,114	4,322
	破綻	1,809	4,329	6,138
合計		4,017	6,443	10,460

表 12 感情極性値を追加したことによる分類精度の変化

		単語集合のみ		
		正しく分類	誤分類	合計
単語集合 + 感情極性値	正しく分類	6,475	62	6,537
	誤分類	82	3,841	3,923
合計		6,557	3,903	10,460

正解数: 6,537
正解率: 62.5%,
破綻検出の recall:70.5% precision:67.2%，F 値:68.8%

4.3 単語重なりと感情極性値の両方を利用した破綻検出

単語集合に単語重なり，感情極性値の両方を追加して分類実験を行った結果を表 13 に示す．単語集合のみを利用した結果と，単語の重なり，感情極性値を追加した結果とを比較したものを表 14 に示す．分類精度はわずかに向上しており，正確二項検定による有意差検定を行ったところ，片側検定結果が $p = 0.06$ となり，有意傾向が見られた．表 15 に，単語集合に単語重なりと感情極性値の両方を追加することによって正しく分類した発話例，誤分類した発話例

表 13 単語集合，単語重なり，感情極性値を用いた場合の分類結果

		予測結果		
		破綻でない	破綻	合計
実際の結果	破綻でない	2,228	2,094	4,322
	破綻	1,773	4,365	6,138
合計		4,001	6,459	10,460

正解数: 6,593
正解率: 63.0%,
破綻検出の recall:71.1% precision:67.6%，F 値:69.3%

表 14 単語重なり，感情極性値を特徴量へ追加したことによる分類結果の変化

		単語集合のみ		
		正しく分類	誤分類	合計
単語集合 + 重なり +感情極性値	正しく分類	6,321	272	6,593
	誤分類	236	3,631	3,867
合計		6,557	3,903	10,460

表 15 単語重なり，感情極性値の特徴量を追加による破綻検出への影響例 (括弧内は感情極性値)

破綻であると正しく分類した発話	システム	午後(-0.567759)から雨(-0.319582)が 降り出し ますよねえ
	ユーザ	それは残念 (-0.995963) です
	システム	午後(-0.567759)から雨(-0.319582)が 降りますよねえ
破綻であると誤分類した発話	システム	犬は愛想が いいですね
	ユーザ	そうだね．賢いよね．
	システム	勘 (-0.365702) が いい(0.999314)ですね

を示す．

4.4 特徴量の有効性の検証

使用した特徴量セットの有効性を検証するため，それぞれの特徴量を取り除いた場合に，正解数，正解率，F 値がどのように変化したのかを確認した．その結果を表 16 に示す．特徴量のリストは表 17 に示した．

この結果から，どれか一つの特徴量が大きく寄与しているのではないことがわかった．なお，単語集合を用いず，単語の重なり情報と単語極性値のみを用いた場合も試したが，すべての発話が破綻と分類された．

5. 考察

単語集合のみを用いたベースライン手法に対し，単語の重なり情報と感情極性値の利用を試みたが，大きな改善は得られなかった．表 9，表 11，表 14 からわかるように，特徴量を追加した際に，新たに検出できた破綻もあるが，それと同程度の数の誤分類の増加がある．これは，新たに追加した特徴量が，特定のタイプの破綻のみをうまくとらえる特徴量になっていないと考えられる．

表 16 使用する特徴量を変更した場合の分類結果の比較

特徴量	正解数	正解率 (%)	破綻検出の F 値 (%)
0 (単語集合のみ)	6,557	62.7	69.0
0–3 (単語集合+単語重なり)	6,583	62.9	69.2
0,4–9 (単語集合+感情極性値)	6,537	62.5	68.8
0–9 (全特徴量)	6,593	63.0	69.3
1 以外	6,597	63.1	69.3
2 以外	6,577	62.9	69.1
3 以外	6,578	62.9	69.2
4 以外	6,585	63.0	69.2
5 以外	6,594	63.0	69.3
6 以外	6,593	63.0	69.3
7 以外	6,596	63.1	69.3
8 以外	6,578	62.9	69.2
9 以外	6,79	62.9	69.2

表 17 特徴量のリスト

ID	特徴量
0	単語集合
1	ひとつ前のシステムの発話と現在のシステムの発話との重なり単語数
2	ふたつ前のシステムの発話と現在のシステムの発話との重なり単語数
3	直前のユーザの発話と現在のシステムの発話との重なり単語数
4	システム発話に含まれる単語の感情極性値の最大値
5	システム発話に含まれる単語の感情極性値の最小値
6	システム発話に含まれる単語の感情極性値の平均値
7	システム発話に含まれる単語の感情極性値の最大最小の差
8	システム発話に含まれる単語の感情極性値の平均値と，直前のユーザ発話に含まれる単語の感情極性値の平均との差
9	システム発話に含まれる単語の極性値の平均値と，ひとつ前のシステム発話に含まれる単語の感情極性値の平均との差

この問題を解決するためには，詳細な破綻の分類 [4] を行い，そのタイプの破綻だけをうまく検出することが必要である．しかしながら，[4] で提示されている破綻の分類の多くは，詳細な意味理解の結果を必要とするものであり，本稿で試したような単語情報のみを用いた手法では不十分であると考えられる．

6. おわりに

本稿では，Project Next NLP 対話タスクの雑談対話コーパスを用い，単語情報を用いて破綻を検出する手法を検討した結果を示した．単語集合のみを用いたベースラインの手法に比べ，単語の重なり情報や感情極性値を併用した方法は，検出精度は向上したもののその差はわずかであった．この結果から，単語情報のみを用いて，すべてのタイ

プの破綻を検出しようとする手法の限界が示唆された．さらに，分類精度，破綻の詳細な分類に基づき，特定のタイプの破綻のみを検出するような手法を構築することが必要であることも示唆された．

今後は，単語情報だけではなく，構文意味情報も用いて破綻の分類毎の検出器を構築する方法を検討して行く予定である．また，破綻の評価の評価者間一致率が高くなかったことから，どのように正解ラベルを決定すべきかについても再検討を行う．

謝辞

Project Next NLP 対話タスクの関係者の皆様，単語感情極性対応表の利用をご快諾いただいた高村大也氏，本稿にコメントを頂いた，荒木雅弘氏，駒谷和範氏，東中竜一郎氏，船越孝太郎氏に深く感謝いたします．

参考文献

[1] 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.

[2] Jennifer Chu-Carroll and Bob Carpenter. Vector-based natural language call routing. *Computational linguistics*, Vol. 25, No. 3, pp. 361–388, 1999.

[3] 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. 人工知能学会 第 72 回言語・音声理解と対話処理研究会, pp. 45–50, 2014.

[4] 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. 言語処理学会大 21 回年次大会ワークショップ論文集, 2015.

[5] 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.

[6] Diane Litman, Julia Hirschberg, and Marc Swerts. Predicting user reactions to system error. In *Proc. ACL*, pp. 370–377, 2001.

[7] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, Vol. 34, No. 1, pp. 1–47, 2002.

[8] 高村大也, 乾孝司, 奥村学. スピンモデルによる単語の感情極性抽出. 情報処理学会論文誌ジャーナル, Vol. 47, No. 02, pp. 627–637, 2006.

[9] 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.

[10] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques, 3rd Edition*. Morgan Kaufmann, San Francisco, 2011.

[11] 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, 2004.