

# キーワード抽出を用いた ルールに基づく対話破綻自動検出器の構築

## Construction of Automatic Detector for Dialogue Breakdowns based on Rules with Keywords Extraction

谷口諒輔<sup>1</sup>

狩野芳伸<sup>1</sup>

Ryosuke TANIGUCHI<sup>1</sup>, Yoshinobu KANO<sup>1</sup>

<sup>1</sup> 静岡大学 情報学部

<sup>1</sup> Faculty of Informatics, Shizuoka University

**Abstract:** This paper presents a new method which detects dialogue breakdowns automatically for the dialogue task of Project Next NLP. Our keyword extraction is performed by a morphological analyser using our customized dictionary, regarding dictionary matched morphemes as keywords. We use these keywords as an utterance focus. We establish three rules which we found sheer variety of dialog breakdowns, based on our observation and generalization of the given dialog data. We implemented detectors by these rules which tries to exclude possible latent ambiguity of annotations.

## 1. はじめに

近年の急速な技術革新がもたらしたスマートフォンの普及と、それに伴った新しい入力インターフェースである音声認識は、日常的に対話システムを利用するシーンを提供するようになった。しかし、いずれのシステムも一問一答式の発話構成に留まるのみで、ユーザーの曖昧性を含む情報、コンテキストに依拠した会話や雑談等の非タスク指向型対話を十分に行えるものではない。

Project Next NLP 対話タスク[1]では、NTT ドコモ社が提供する雑談対話 API[2]を使用した中で獲得されたシステム発話とユーザー発話、またはそれら一連の会話について、複数のアノテーターによって付与された破綻ラベルを元に、対話システムのエラーを分析することを目的としている。しかし、そのような方法では人的コストが嵩むばかりか、それを基に機械学習を利用するには圧倒的にデータ量が足りていないことが現状である。

そこで、本論文では破綻ラベルが付与された表層的な発話内容に着目し、ヒューリスティックに導かれた独自の破綻ルールを制定、それらを組み合わせた手法を用い、学習データを必要としない極力単純化された対話破綻検出器の実現を試みる。

## 2. 提案手法

### 2.1 破綻事例の観察

対象となる対話 API においてシステムは、前段階におけるユーザ発話もしくは自身の発話内容から部分的に単語を抽出し、その単語を反復して次ターンの発話内容に利用するケースが多く存在する。以下は、雑談破綻チャレンジにおいて配布された雑談対話における一例を部分的に抜粋し示したものである。Sはシステムを、Uはユーザを表す。

S: 予防は完璧ですね

U: はい。あなたは何か熱中症対策をしていますか？

S: 予防が肝心です

U: そういえば、梅雨が明けましたね。

S: 日頃から予防を心掛けるかもしれない

このように、各発話の話題の中心はこの繰り返されて使用される単語によって決まることが多いため、会話文中のシステムとユーザの発話の前後、またはシステムの発話における前後についてキーワードを抽出できれば、焦点の遷移をある程度把握することができる。

以下では、配布された init100 から 20 対話、rest1046 より 30 対話を、破綻ルールの策定の観察の対象として無作為に選出し、使用した。これにより対話破綻と見受けられる点について、その内容と例を抜粋して紹介する。

- (1) ユーザの質問文に対して、適切な返答をシステムが行うことができない。

U: 昼ごはんは何を食べましたか  
S: ごはんはあったかいです

- (2) システムの発話について、非文ではないが文章として不適切なとき。

S: 夕食がいってないかもです  
U: 日本語として変です  
S: 日本語は早いです

- (3) システムが、自身の発話した中から特定のワードを用い、再度発話したとき。

S: こんばんは。夏といえばスイカだね。  
U: そうだね。でもまだ食べてないや。  
S: スイカで買い物できてるのですか？

- (4) ユーザの発話の直後のシステムの発話について、動作主は同一に関わらず、助詞が変換されているとき。

U: そうかな？私は旦那さんと買い物するよ。  
S: 旦那が行きますか？

- (5) ユーザの発話前後のシステムの発話について、そのどちらにも特定のキーワードが含まれているにも関わらず、ユーザ発話にはその特定のキーワードが含まれていないとき。

S: エボラ出血熱を引き起こしますよねー  
U: 何が原因なんでしたっけ  
S: エボラ出血熱を引き起こすらしいです

- (6) システムの発話における「～です・ます」または「～だ・である」といった、口調の不一致

S: 行楽シーズンにはいますか？  
U: ええ  
S: どうした

- (7) システムの発話における時勢の不一致

S: こんにちはー。ちょうど退屈してたんだ。  
U: こんにちは  
S: おはよ

- (8) システムの発話において、過去の会話中で一度断定された内容に関して、再度質問形式でユーザに発話したとき。

S: プールは水が生ぬるくないですよ  
U: 気温によるかな  
S: 水が生ぬるいとかですか？

これらのうち(6)、(7)については、破綻とみなすかどうかについてアノテーターによって異なることから、主観的な判断が強いと推測されるため、ルールとしては不適切である。(2)については、GoogleNgramを用いた単語間の共起確率を利用した手法を考案したが、第2章で述べたキーワード抽出によるルールに属さないため、今回実装は行わなかった。

## 2.2 破綻ルールの策定

前節で述べた項目のうち、対象とするものを破綻ルールとして策定する。内容的に重複しているところもあるため、下記に示される3点として一般化し、ルールとして策定する。

(I) ユーザが話題の転換の後、そのターニングポイント前のワードをシステムが発話したとき。言い換えるに、システム1、ユーザ、システム2の順の発話内容の内、システム1とシステム2には同一のキーワードが含まれているにもかかわらず、ユーザにはそれに一致するキーワードが含まれていない場合に破綻ラベルを付与する。

(II) 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する。

(III) システムの質問発話について破綻ラベルを付与する。

ただし、(II)、(III)に関して、質問発話であるという判断基準は、各発話の文末にクエスチョンマークが含まれているかのみで判断する。

## 3. 実験

実験では、前節で述べた3つのルールを次のように適用し、ラベルを付与した。ラベルはO(破綻ではない)、T(破綻とは言い切れないが違和感を感じる発話)、X(明らかにおかしいと思われる発話)の3種類が与えられる。

**run1:**( I )(II)(III)のルールを OR で接続し判定する。3 つ全ての、または 3 つのうち 2 つのルールに該当する場合は破綻ラベル X を付与する。ただし 2 つのルールに該当する場合は、3 つのルール適用時よりも破綻可能性を軽減させる。1 つだけ該当する場合はラベル T を付与する。

**run2:**( I )(II)(III)のルールを AND で接続し判定する。いずれかのルールについて一つでも該当するものがあれば破綻ラベル X を付与する。

**run3:** ( I )(II)(III)のルールを単体で運用した際に、最も判定正答率が高かった(III)のみを使用、該当のものに破綻ラベル X を付与する。

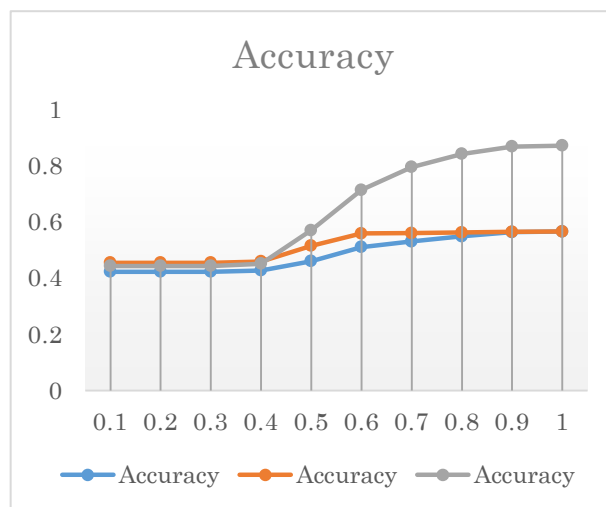
キーワードの抽出については、Java で実装されたオープンソースの日本語形態素解析器 `kuromoji`[3]と、`kuromoji` のユーザ辞書として Wikipedia データを用いた。我々の手法では、形態素解析時に、ユーザ辞書に登録された文字列については最長文字列マッチ相当によって品詞分解されるため、通常形態素解析結果よりも矯正的に単語が区切られることになる[4]。またユーザ辞書に登録された単語が検出された場合、Wikipedia という属性値を与えた。本検出器では、この属性値が与えられた単語をキーワードとして扱い、これがシステムとユーザの各発話の焦点であると考えられる。ただし、Wikipedia データ内には挨拶、ひらがな 2 文字の単語、また文末の助動詞などの本タスクの目的にとっては有害なエントリがある。これらのエントリについては、一度与えられた対話データについて形態素解析を行い、必要に応じて目視で辞書から除外した。

## 4. 考察

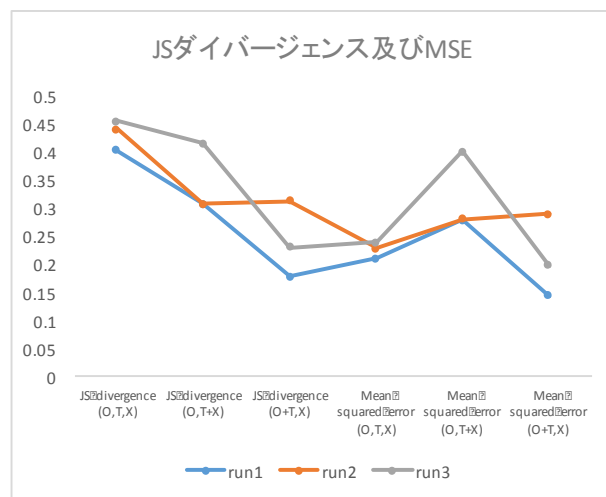
Accuracy (全ラベルの一致率) について run1 から run3 を俯瞰すると、ルールを単体運用した run3 の正答率が高いということを読み取ることができる。しかし、雑談対話データの収集の際の各ラベルの発生数を参考とするに、破綻ではないラベル O をアノテーターは最も多く与えていることから、しきい値を 0 から 1 に近づけ正解アノテーションをラベル O にスライドさせていくことによって、そもそもラベル O を多く与えた run3 の一致率が高いことは必然であり、Accuracy を用いた一概の評価は有意であるとは言えない。これより、以下に記された評価指標について各検出器を検討する。

### 4.1 Recall(再現率)と Precision(精度)

run1、run2 のラベル X と T の評価総数に変化はないため、しきい値の変化に伴う再現率及び精度の推移は一致している。両者の評価結果での最大の差異はラベル X の再現率であり、しきい値が 0.5 のとき、その差は約 50%と run2 の方が網羅性が高いと読み取れる。同時に、精度に関しても run2 が上回っていることから、本来破綻ラベルを付与すべき対話について run1 では読み過ぎられていると言える。run3 に関しては評価ラベルの偏りを前述した通り、再現率と精度は共に低い水準となった。



### 4.2 JS ダイバージェンスと MSE



上図より、run1 と run3 は近似した精度評価指標を有しているのに対し、run2 は異なった結果を示していることがわかる。run1、run3 はラベル T が O に近

い場所にあるとしているが、run2 では T は X 寄り付与されている。run1 は 3 つの検出器で唯一 T を付与できるものであるが、単体のルール(III)を適用した run3 が近似した理由は以上の二つが考えられる。

- ① ルール(III)で破綻と評価される発話総数が、ルール(I)(II)よりも相対的に少なかったため。
- ② ルール(III)で破綻と評価される発話が、ルール(I)(II)に多く内包されているため。

アノテーターによって付与された母集団のラベルが 0 に偏在していること、また上記の理由から run3 の一致率が上昇した原因を推測できる。

対して run2 は、元来 run1 でラベル T として扱われた発話をすべて破綻と認定しているためラベル X は T と大きく乖離する結果となった。

またこの二値的な分類方法は、結果として X と T 間、T と 0 間の曖昧さを排除し、しきい値が大きくなるほど、つまり厳格な評価になればなるほど評価値が良くなることにつながったと考える。

## 5. まとめ

本稿では対話破綻検出器について、どのような対話破綻の形式があるかを対話データから読み取り、それに基づいた一般化したルールを作成し自動化を試みた。ひとまずの評価結果は下すことはできたが、現状のルールベースに則った対話破綻検出器では、各ラベル間の評価が厳しく行われてしまいということにより、複数のアノテーターによって付与された幅のあるラベル評価に対して、当システムは対応できていないことがわかる。

この点の解決策は第一に、run1 で実行された OR で組み合わせられた結果をもとに、自身もつそれぞれ 8 つの集合に最適化された確率変数を振り分けるというチューニングを施すことが挙げられる。より判定がよりシームレスになることを期待する。第二に、導入するルールを増やすことがあるが、ルールが増えれば増えるほど、集合の数は増大するため、本論文で目指す極力シンプルな検出器という条件を満たすためには取捨選択を行い、最大効率のものを目指さなければならない。

また、ルール(I)は各対話の焦点となりえるキーワードを抽出するが、これには様々は各要素が一緒くたになっていることから、一つの対話に複数のキーワードが設定されることもあれば空になることもある。コンテキスト上の推移を明瞭化するという意味で、一対話に一つのキーワード、また空である場合は一つ前の焦点を利用するといったセンタ理論の

部分的な実装は有意に働くのではないか。加えて、今回利用を見送った GoogleNgram を用いた単語間共起による破綻検出器は、文脈に則った判定は不可能であるものの、発話した内容についてそれが日本語的に正しいかどうかの判定をルールベースの検出器と併用することによって、より高い精度でアノテーションが行えるようになるかもしれない。

## 参考文献

- [1] 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, 第 6 回対話システムシンポジウム, 2015
- [2] 雑談対話 API:  
[http://www.nttdocomo.co.jp/service/developer/smart\\_phone/analysis/chat/](http://www.nttdocomo.co.jp/service/developer/smart_phone/analysis/chat/)
- [3] kuromoji:  
<http://www.atilika.com/ja/products/kuromoji.html>
- [4] 狩野 芳伸: 大学入試センター試験歴史科目の自動解答, 2014 年度人工知能学会全国大会 (第 28 回), 2014