Neural Conversational Model を用いた対話破綻の予測

Prediction of breakdowns of dialog by Neural Conversational Model

久保 隆宏[1]　中山 光樹[1]

Takahiro Kubo[1], Hiroki Nakayama[1]

[1] TIS 株式会社

[1] TIS Inc.

Abstract: This paper describes a method to detect the breakdowns of dialog by using Neural Conversational Model(NCM). When using this kinds of model , we always face the amount of external training data problem. The proposed method overcomes this issue. Learning the dialog with breakdown label and attach it according to the probabilistic distribution of annotations　enables you to use the data that includes breakdown effectively. We could show the model that is trained by this method scored high recall, low precision , and very high F-measure.

1. はじめに

本研究では，雑談対話システムが対話の文脈から外れた発話(=対話破綻)を自ら予測し，回避できるようになることを目指している．具体的には，これまでのユーザーの発話，また自らの発話と照らし合わせて，これから発話しようとしている内容が適切かどうか判断するということである．

しかし，対話が破綻している，していないの線引きは曖昧であり，またそのラベル付がされたデータを数多く取得することは困難である．

そのため，本研究ではこの曖昧性と学習データの少なさを克服することに主眼を置いている。

2. 関連研究

本研究は対話破綻検出チャレンジの一貫であるため，まずこのチャレンジにおけるタスクについて簡単に述べておく．

対話破綻検出チャレンジにおけるタスクは，対話履歴と直後のシステム発話が与えられている場合に，そのシステム発話が対話破綻を引き起こすかどうかを検出するタスクである．破綻のラベルは○△×の 3 種類であり，このラベルを予測すること，確率分布を予測することがモデルの目的となる[1].

今回は，この対話破綻の検知を行うに当たり Neural Conversational Model をベースとしている．これは，対話を行うためのモデルと対話破綻を行うためのモデルを別個に考えるのでなく，対話を行う中で破綻の予測を行うこと，そして将来的にはその予測結果を発話生成にフィードバックすることを企図している．

対話破綻を Neural Conversational Model，いわゆる Encoder-Decoder モデルで推定する方法は, 2015 年に実施された対話破綻検出チャレンジ 1 にて小林ら(2015)が提案を行っている. これは Encode 後(ユーザー発話後)，Decode 後(システム発話後)の内部状態を基に分類器を作成し推定するというものである[1]. 他に RNN(Recurrent Neural Network)を用いた例として，ユーザー発話用，システム発話用それぞれの Encoder を用意し, それらの出力結果を基に分類を行う手法も提案されている[2]. この Encoder への入力に際しては，単語を Word2Vec による分散表現へ変換する手法が取られている．

本研究では，対話モデルに近い小林ら(2015)の手法をベースとした．ここから，さらに以下の課題を克服するための工夫を加えている．

- 学習データの曖昧性
- 学習データの少なさ

これについては, 次の「モデル設計」の項で述べる．

3.モデル設計

まず，対話破綻検出チャレンジで与えられるデータは当然破綻している対話を含んでいる. そのため，これを Neural Conversational Model で学習した場合，

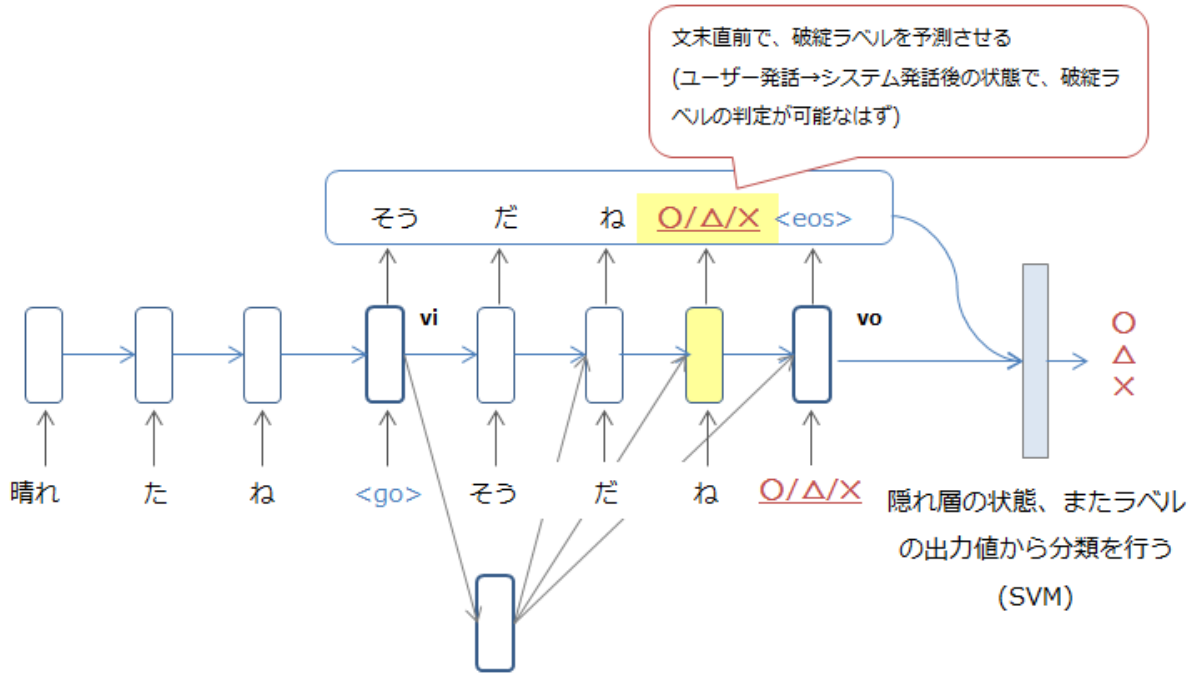

図 1: モデル設計

破綻した発話をするモデルとなってしまい，適切な返答はおろか，その検知もできなくなってしまう(適切な返答と，そうでない返答がモデル内で区別されないため).

これを避けるためには，正常な対話でまず学習を行う必要がある．しかし，これには別途相当量の対話データが必要となる．小林ら(2015)は，匿名掲示板の投稿小説から 30 万ペアの会話文を抽出し事前に学習を行っているが，こうしたデータを抽出し，学習用に整形するには非常な手間がかかる．

そこで，本研究ではシステム発話列の末尾に対話破綻ラベルを付与するという手法を用いた(図 1).

具体的には，システム発話の終端記号の前に，対話破綻の種別(○△×)を表す記号を挿入する．そして，この挿入はアノテーターの○△×の実際の分布に基づき，確率的に行う(図 2).

これにより，以下のような利点が得られる．

- 破綻した対話データの有効活用: 破綻を含んだデータであっても,「その対話が破綻しているかどうか」まで含んで学習することになるため，破綻した対話データであっても破綻した対話例として区別して(認識的に)学習することができる．
- 学習データの有効活用: 破綻ラベルを確率的に挿入するため，同じ対話データでも挿入されるラベルが異なる場合がある(図 2)．これにより，同じ対話データでも複数回学習することに意義が出るほか，ラベルの曖昧性を学習することが期待できる．

最終的な破綻ラベルの分類には，Encoder，Decoder の状態，そして破綻ラベルの出力値を基に，単純な分類器(Support Vector Machine)を作成し分類を行った. これについては, 以下の評価実験の項で述べる.

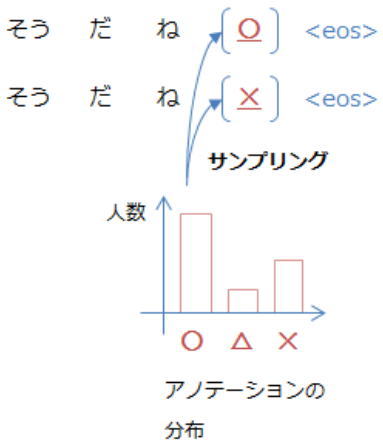

図 2: アノテーターの分布に基づく, 確率的なラベルの挿入

表 1: 評価結果

入力：単語

	E&D	D+出力	E&D(○×)	D+出力(○×)
Accuracy	.487	**.494**	.450	.480
Precision(X)	.379	**.391**	.383	.383
Recall(X)	**.784**	.609	.553	.605
F-measure(X)	**.511**	.476	.453	.469
Precision(T+X)	**.745**	.739	.724	.728
Recall(T+X)	.712	.532	**.825**	.589
F-measure(T+X)	.728	.618	**.771**	.651

入力：分散表現

	E&D	D+出力	E&D(○×)	D+出力(○×)
Accuracy	.464	**.481**	.413	.459
Precision(X)	.357	.360	**.380**	.362
Recall(X)	**.784**	.505	.525	.521
F-measure(X)	**.491**	.421	.441	.428
Precision(T+X)	**.735**	.732	.703	.721
Recall(T+X)	.745	.475	**.941**	.602
F-measure(T+X)	.740	.576	**.805**	.656

4.評価実験

実験においては，以下の点をそれぞれ変えた計 8 つのモデルについて検証を行った．

- 入力: 分散表現への変換を行うか否か
- 破綻ラベル: △を予測するか
- 特徴量
  - Encoder/Decoder 双方の隠れ層の状態
  - Decoder の隠れ層の状態と，破綻ラベルの出力値

分散表現への変換には, Facebook の発表した fastText を用い[4]，日本語 Wikipedia で学習させたものを用いた．

△のラベルについては，○か×か判断できない場合に自然につくものととらえ，分類器を○か×の二値分類にし，何れの確度も高くない場合に△にするという手法も検証した．

特徴量については，小林ら(2015)の提案を参考にそのまま Encoder/Decoder の状態を入力としたものと，今回の学習の結果得られる破綻ラベルの出力値を使ったもの，双方について検証した．

学習データとしては，対話破綻検出チャレンジ 1 の学習用データ(1046 対話)，開発用データ(20 対話)，そして今回の対話破綻検出チャレンジ 2 の開発用データ(3 つの異なる雑談対話システムから，それぞれ 50 対話ずつ，計 150 対話)を用いた．そして，評価用データとしては対話破綻検出チャレンジ 1 の評価用データ(80 対話)を用いた．

学習に際しては，Neural Conversational Model を学習後に，分類器の学習を行った．分類器の学習に際しては，対話破綻検出チャレンジ 1 の学習用データはラベルに大きな偏りがあったため, これを除外した．

実験における仮定としては，分散表現を用い，特徴量として破綻ラベルの出力値を用いたものが最も良いと考えた．

5.実験結果

実験結果についてまとめたものが，表 1 となる．
結果として，分散表現の利用による精度の向上は見られなかった．この原因としては，分散表現による意味の汲み取りが逆効果として働いてしまったことが考えられる．具体例として，以下は分散表現モデルで破綻の検知に失敗したもの(ラベルは破綻だが，正常とみなしたもの)の一例である．

- 梅雨明けましたね

- 梅雨に突入するのでしょうか？

- どんな小説を読んでるんですか？
- 小説は早いです.

こうした同じ単語，また同意味の単語(食事とカレー，病院と風邪など)が使用されているケースは，対話が破綻していないパターンで多くみられる(例として，「夏だからバーベキューもアリだよね」「バーベキューは楽しいですね」など).

しかし，上記の例のように，同意味の単語を使用しているからといって破綻がないわけではない．分散表現を利用したモデルは利用していないモデルよりも同意味の単語が検知されやすくなるため，結果として同意味の単語を用いているが応答文として破綻しているパターンを検知できず，精度が向上しないという現象が考えられる．よって，さらなる精度の向上のためには，応答文としての文構造・体裁をチェックするための特徴量が必要ではないかと思われる.

特徴量については，出力値は Accuracy，Encode の状態は Recall に大きく寄与するとの結果が得られた．△ラベルの省略による寄与はあまりなかったが，△のラベル自体がかなり曖昧な 位置づけであり，何れのモデルも△に関する精度は低い結果だった.

モデル全体としては高 Recall 低 Precision のモデルとなった. F-measure としては高い値を記録できており，別途のデータセットの用意がなくとも十分なモデルを構築できることを証明できた．しかし，Accuracy と Precision については十分とは言えず，前述のとおり文構造に関する特徴量を導入するなど，さらなる工夫が必要と考えている.

6.結論

本研究では，Neural Conversational Model による対話破綻の検知を行った．事前学習を行わずに，対話破綻のデータのみでモデルの学習を行うため，システム発話末尾にアノテーションの分布に基づき確率的に対話破綻ラベルを挿入するという手法を用いた.

これにより対話破綻のデータを有効的に活用することが可能になり，F-measure ベースで高い精度を持つモデルの構築を行うことができた．しかし，Accuracy/Precision ベースではまだ十分な値が出せておらず，この改善のためには応答文としての構造・正当性を評価できる特徴量が必要と考えている.

参考文献

[1] 東中竜一郎，船越孝太郎，稲葉通将，荒瀬由紀，角森唯子，対話破綻検出チャレンジ 2，第 78 回言語・音声理解と対話処理研究会（第 7 回対話システムシンポジウム），(2016)

[2] 小林 颯介, 海野 裕也, 福田 昌昭: 再帰型ニューラルネットワークを用いた対話破綻検出と言語モデルのマルチタスク学習，SIG-SLUD, Vol. B5, No. 02, pp.41-46, (2015)

[3] 稲葉 通将，高橋 健一: Long Short-Term Memory Recurrent Neural Network を用いた対話破綻検出，SIG-SLUD, Vol. B5, No. 02, pp.57-60, (2015)

[4] J Bojanowski P., Grave E., Joulin A., and Mikolov T.: Enriching Word Vectors with Subword Information, (2016)