単語の意味の距離から検出する対話破綻

Detection of Dialog Breakdown by Semantic Distances of Keywords

柴淳 [1]　　狩野芳伸 [1]

Atsushi SHIBA[1] ,　　Yoshinobu KANO[1]

[1] 静岡大学　情報学部
[1]Faculty of Informatics, Shizuoka University

Abstract: This paper presents a method which detects dialogue breakdowns automatically for the task of dialogue breakdown detection challenge 2. We measured semantic distances of keywords in between pairs of dialogues, detecting breakdowns together with manually created rules. Our system performance was around 75-89 points in F-measure (T+X), depending on formal run dataset. These results suggest that the semantic distance of words and our rules were effective enough to detect dialogue breakdowns in the dataset.

1. はじめに

対話破綻検出チャレンジ 2[1]では前回とは異なり、データ数が増え、また、システムの対話内容もデータごとに異なる指向性を保持している。多種多様な返答を行っているシステムの発話に対して、対話破綻検出器もその多様性に耐えうるものにならなければならない。

前回の対話破綻検出チャレンジでは、我々は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た[2]。

本論文では前回行ったアプローチの中で、有用と思われるものを用い、さらに新たなルールと意味的距離を導入することで、対話破綻検出器の性能の向上を図った。意味的距離を導入する理由として、対話の中で利用されている単語は、その単語同士の意味的な距離は近しいものであり、その意味的距離が離れているのであれば、対話破綻しているのではないかと考えられるからである。

以下では、第 2 節で提案手法を説明し、第 3 節で実験と結果、第 4 節で結果の考察を述べ、第 5 節で締めくくる。

2. 提案手法

配布された対話データにおいて、システムは、システムが発話する直前のユーザーの発話、もしくはシステムが事前に行った発話の中に含まれる単語を部分的に抽出し、次の発話に利用するケースが存在する。対話において再度利用される単語、または、その単語を利用する際に付属する単語は、その対話の中で同じ意味を持つかそれに近しい意味を持つ。今回の対話データにおいても、アノテーターの主観的な判断が下される場面は、その単語の持つ意味が事前の発話と噛み合っていない時である場合が多かった。

以下の例は、配布された対話データから破綻事例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)
S:スイカはおいしいよ
U：でも高いよね
S:いくら
U：1000 円くらいかな
S:スイカは実がなってたことあります

この対話においてシステム側は「スイカ」という単語に発話が依存しているが、そのあとに付属する単語において破綻が発生している。ユーザー側も「スイカ」について発話しているが、主語が抜けているため、その補完を事前のシステムの発話から行ったためこのような対話破綻が見られたと推測できる。

(2)
S:あなたの好きな映画について教えてください
U：サスペンスが好きです
S：映画はヒューマン系が好きですね
U：映画の題名は？

S:映画にハマるのです

この対話では、ユーザー側が投げかけた質問に対して適切な回答を行えていないとアノテーターが判断し、対話破綻していると見なされている。

(3)
S:恋人は重いですねー
U:勉強は好き？
S:「恋人たちの予感」って、あれ本当に面白い映画だよね。

この対話ではユーザー側が話題転換として投げかけた疑問文に対して、システム側が自分のひとつ前の発言を引きずってしまい、ユーザー側の疑問文に反応することができず対話破綻となっている。

(4)
U:ディズニーランドは好きですか？
S: 私が勤務している大学はディズニーランドにありますが、私が勤務している大学へは、ラスベガス、千葉県、船橋市、山梨県から通学している大学生が多いようです。

この対話では、ユーザー側が投げかけた質問に対して、対応はしているがその発話内容がおかしい文章であるとみなされたため対話破綻となっている。

(5)
S:何かお話しましょうか？
U:今日も暑いですね。雨は降っていませんか？
S:有名人で長谷部誠と栗原はるみなら、どちらに興味がありますか？
U;今日も暑いですね。雨は降ってませんか？
S:有名人でレイザーラモン RG と高橋みなみなら、どちらに興味がありますか？

この対話では、ユーザーからの疑問文に対して、システム側も疑問文を返し、その結果対話破綻とみなされている。

以上で述べた項目から、対象とするものを破綻ルールとして策定する。これは前回の我々のルールの一部と共通である。

1. システムがユーザーの発話の内容の意味を汲み取れず、システムの前の発言に依存して発話したときに破綻ラベルを付与する。

2. ユーザーの疑問文に対し、システムが疑問文を返答した時に破綻ラベルを付与する。また、システム側の文末にクエスチョンマークが二つ並んで付いていた場合は単独で破綻とみなす。

3. システムがユーザーの発話の内容の意味を汲み取った上で、意味的に異なる発話をした時に破綻ラベルを付与する。

ただし、2.において、質問発話であるかどうかの判定基準は、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定する。

また、1.および 3.における判断基準として Word2Vec[3]を利用して単語間の距離を測定し、その数値を判断基準として利用する。具体的には、ユーザーとシステムの発話に含まれる単語を抽出し、Word2Vec により各単語のベクトルを計算する。次にユーザーとシステム間のあらゆる単語ペアで内積を求め、すべてを合計しペア数で割ることで正規化を行い、これを単語間の距離として用いた。

3. 実験

実験では、前節で述べた 3 つのルールを別個に適用し、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)の三種類があるが、我々の手法では T を付与することが難しいため、今回の実験では O と X のみを付与することにする。

run1: Word2Vec を利用して、大規模均衡日本語コーパス[4]のデータを用い、ユーザーとシステムの発話に含まれる単語間の意味的距離を調べ、その距離が訓練データにおける平均値より大きければラベル X を付与する。また、システムが以前のシステムの発話に含まれている単語を利用した場合は、意味的距離にかかわらず優先的にラベル X を付与する。

run2: run1 と同様に単語間の意味的距離でラベルを決定する。ただし、システムが以前のシステムの発話に含まれていう単語を利用したかどうかは判定に用いない。

run3: run2 と同じことを行う。ただし、単語間の意味的距離によるラベル付与の基準を run1 で利用した値の半分に設定する。

ユーザーとシステムの発話に含まれる単語の抽出は、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[5]を改良したツールにユーザ辞書として Wikipedia データを加えたものを用いた。抽出する単語としては、Wikipedia のエントリおよび文の内容語になる名詞と動詞を抽出して用いた。ただし、Wikipedia データ内には挨拶、平仮名 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。

表 1 にフォーマルランの結果を示す。我々のシステムはラベル O と、ラベル T＋ラベル X の二値分類を行ったため、評価値には (T+X) のメトリクスのみを記載した。

4. 考察

表 1 の各評価値について run1 から run3 を俯瞰すると、X と判断するのに一番基準の緩い run1 の正答率が高いということを読み取ることができる。run1 の方法では、与えられるラベルは必然的に X が多くなる。run1 と run2 の間に差が生じたのは、システムがユーザーの発話を無視して発話した場合でも、run2 では単語の意味的距離が近ければ O とラベルを付与しているからだと考えられる。また、今回の検出結果では、T のラベルはすべて X として検出しているため、O のラベルが付与されやすくなればなるほど、T と判断すべきものも O と判断され、正答率が下がっていると予想される。

表 1. フォーマルランの評価結果

	(T+X)	Precision	Recall	F-measure
run1	DCM	72.74	89.97	80.44
	DIT	82.28	**98.05**	**89.47**
	IRS	71.35	79.55	75.23
run2	DCM	73.13	76.60	74.82
	DIT	**82.65**	89.07	85.74
	IRS	71.61	77.03	74.22
run3	DCM	71.56	63.78	67.45
	DIT	82.12	76.94	79.44
	IRS	72.53	68.06	70.23

run1 と run2 を比較したとき、F-measure および Recall はすべて run1 のほうが良い結果を出している。これは、X と判断する基準が緩い方が正確な値を出しているということである。run1 と run2 の違いからすると、以前のシステムの発話を引き継いでシステムが次の発話を行う場合は、T もしくは X とみなすことが多いということになる。一方 Precision の値は run2 よりも run1 の方が高い。言葉の意味的距離だけでは全ての破綻を検出することができず、以前のシステム発話についてのルールを追加すると網羅性が高くなるためであると考えられる。

run2 と run3 を比較すると、ほぼすべての項目で run2 のほうがよいスコアであり、特に Recall と F-measure の値は run2 の方が大幅に高くなっている。run3 において X 付与の基準値を変更した理由は、run2 の基準である訓練データの平均値に妥当性があるかを確認するためであったが、この結果からは平均値に妥当性があったと考えられる。

5. 終わりに

本研究では対話破綻検出器について、前回のルールの一部を用い、そこに新たな手法を加えることで性能の向上を試みた。フォーマルランの実験結果から判断すると、今回の手法と前回の手法を組み合わせたことでより良い結果を得ることができたと考えられる。今回の他チームの結果はまだ公表されていないため比較はできないが、全体的な結果は F-measure で 80 ポイント前後であり、前回のチャレンジの結果からすると概ね高い水準の結果を得ることができた。

今回のシステムではラベル T の判断を行っていないため、今後は、T と X とが本質的に分離可能なのかも含め、T を付与するための意味的距離の測定、ルールの改善や追加など、T の付与について検討していきたい。また、時間の制約で十分でなかったエラーの分析なども進めていきたい。

謝辞

本研究の一部は科研費若手研究(A)及び挑戦的萌芽研究の助成による。

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] 谷口諒輔, 狩野芳伸. 単語間共起及び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と評価手法の検討. 言語処理学会第 22 回年次大会（NLP2016）. 東北大学, 2016 年 3 月 8 日

[3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. ICLR Workshop, 2013.

[4] 前川喜久雄. 代表性を有する大規模日本語書き言葉コーパスの構築. 人工知能学会誌, 24(5) 616-622, 2009

[5] kuromoji: http://www.atilika.com/