

# 語句言い換えを用いた対話システムの精度向上

瀧 洸介 (指導教員 西村 俊二)

平成31年1月25日

## Accuracy improvement of dialog system using phrase paraphrase

Kosuke Fuchi (ACADEMIC ADVISOR Shunji Nishimura)

概要:機械化が進む社会において,小さな子供や高齢者といった普段機械を扱わない人を対象としたシステムとして,対話システムがある. この対話システムの精度を高めるためには膨大な学習データを扱わなければならない,時間がかかるうえ,学習データの収集も大変である. そこで,本研究では小さな学習データでも対話システムの精度を高める手段として,対話システムに入力する文章のうち,学習データにない語句を学習データにある語句に言い換えを行った場合に回答精度が向上することを明らかにした. 言い換えを行った場合と言い換えを行わなかった場合の対話システムによる回答を比較した結果,言い換えを行ったときの回答精度が言い換えを行わなかったときの回答精度に比べ0.4%向上した.

キーワード: 言い換え技術, MeCab, WordNet

## 1. 緒言

近年,機械は生活の中で重要な役割を任せられるようになり,機械が人々にとって身近な存在となっている. また,少子高齢化が進む日本において人材不足が深刻化しており,もともと人が担っていた職業や役割でさえ機械に任されることが増えている. 商品のバーコードスキャンと精算を購入者自身が行うセルフレジの導入や,飲食店での注文をタッチパネルで行うといった例が挙げられる. このように,これまでは人が行っていた単純作業から,車を運転する人工知能など高度かつ臨機応変な対応を求められる役割でさえ機械に任せようという動きがあり,こうした様々な人々の生活に対応した機械の研究,開発が進められている. しかし,機械化が進む社会において普段機械を扱わず,機械の扱いに慣れていない高齢者や小さな子供にとっては公共機関や販売店に設置されている,案内を行う機械や説明書がある家電でさえ扱うのが難しく,こうしたサービスを有効に利用できていないのが現状である.

このような,機械を扱うことが難しい人々でも機械を扱いやすくするシステムが対話システムである. 他人とのコミュニケーションをとる手段として最も基本的なものは言葉であり,機械の扱いに慣れていない人でも言葉による機械の操作は容易であり,道の案内や翻訳といった場面においてすでに利用されている. また,機械操作の簡易化だけでなく,心理的に疲弊した人へ行う対話カウンセリングや一人暮らしの高齢者等社会的に孤立している人々に対する雑談相手といった,より人々の生活に関わる重要な役割を任せられるようになり,対話システムの研究,開発も急速に進んでいる.

しかし,人工知能による対話システムは学習データにより回答精度が大きく左右される. 回答精度を高めるためにはより大きな学習データを扱わなければならないが,学習データを収集することは困難であり,大きな学習データを収集するためには時間がかかる.

そこで,学習データの量が少なくても回答精度を高める手段として,本研究では言い換え技術を提案する. 言い換えとは,ある言語表現を意味が近似的に等価な別の表現形に置換することを指し,与えられた言語表現の言い換えを生成する言い換え生成技術,および与えられた言語表現とその言い換えを行った言語表現が言い換え関係にあるか否かを判定する言い換え認識技術の総称である[1]. 言い換え技術は,難解な文章を簡易化し,読み手に読み取りやすくするように言い換える技術と,機械が受け取りやすくなるように言い換える技術がある. 後者はすでに,機械処理に適した言語表現に言い換えて機械翻訳をさせた場合に翻訳精度が向上することが報告されている[4]. また,機械学習による機械翻訳においても,目的言語を言い換えることで精度が向上することが報告されている[5]. これらのことから,対話システムの回答精度向上には言い換え技術が有効であると考えられる.

本研究では,対話システムが受け取りやすい文章にするために,複数ある言い換え手法の中から最も簡易的である既存の語彙資源を用いた類義語への言い換えを行う. 語彙資源として用いるのは,言葉を同義語,意味上の類似関や包含関係などによって分類した辞書シソーラスであり,対話システムに入力する文章のうち学習データにない語句をシソーラスによって得られた語句に言い換える. シソーラス等語彙資源を用いた類義語への言い換えは,直感的には同義的な意味を持

つ別の語句への言い換えができると思えるが、文脈により何らかの差が生じる場合がほとんどであり、無条件で言い換えを行うことのできる語句のペアはほとんどないとされている[1, 2, 3]。ただし、本研究では文脈を細部まで読み取ることのできる人間のための言い換えではなく、対話システムが受け取りやすい文章にするための言い換えであり、文脈的同義性は重視しない。言い換え方法は、対話システムに入力する文章のうち学習データに含まれていない語句を抽出し、その語句の類義語の中から学習データに含まれている語句へ無条件に言い換える。

本論文では、言い換えを行った場合と言い換えを行わなかった場合の回答と比較し回答の精度が向上するのか、検証した。

本論文の構成は以下のとおりである。第2章では本論文で使用する諸概念と研究の具体的な方法について述べる。第3章では実験結果について述べ、第4章では実験結果についての考えをまとめる。最後に第5章で本論文の結論を述べる。

## 2. 実験方法

本章は本研究の具体的な方法を述べる。2.1 では本研究で用いた対話システムについて述べ、2.2 では言い換えに関する具体的な手法と使用する諸概念について述べ、2.3 では評価方法と評価に用いるテキストペア類似度と評価用データについて述べる。

### 2.1 対話システム

本研究で用いる対話システムは、自然言語処理の分野においてよく用いられている Sequence to Sequence モデル[10] (以下 Seq2Seq) である。本研究は入力文章に対する言い換えによる回答精度の向上を、入力文章に対し言い換えを行うことで回答精度が向上するか検証することを目的としているため、対話システムのモデルは既存のプログラム[6, 7, 8, 9]を使用し、対話システムの部分はブラックボックスとする。学習データは名大会話コーパス[11]を使用するが、もともと人工知能の学習用に作成されたコーパスではないため、Seq2Seq で学習できるよう発言者の発言を Encoder、回答者の発言を Decoder とする[12]。また、コーパス上の句読点はすべて削除し学習させる。

### 2.2 言い換え方法

第1章で述べた通り、対話システムに入力する文章のうち学習データに含まれていない語句を抽出し、その語句の類義語の中から学習データに含まれている語句へ、シソーラスを用いて無条件に言い換える。本研究で用いた語句のシソーラスを出力するシステムは

WordNet[13,14]である。対話システムへの入力文章を MeCab[15]により分かち書きし、単語ごとに分け学習データに含まれているかを調べる。なお、学習データに含まれているか否かの調査方法は、あらかじめ学習データをすべて分かち書きし、単語の頻出回数をまとめたファイルにその単語が存在するか否かで判定する。学習データに含まれていない単語があった場合、その単語を WordNet を用いて類義語を得る。得られた類義語のうち学習データに含まれている語句を抽出し、そのままの形で元の単語の位置に戻す。

#### 2.2.1 WordNet

WordNet は、言葉を同義語や意味上の類似関係、包含関係などによって分類した辞書、シソーラスによりある単語の類義語を得ることができる。python で WordNet を用いた場合、以下のリスト 2.2.1 のように出力される。

リスト 2.2.1 “悲しい” の WordNet 出力

```
{'dysphoric': ['不幸せ', '憂い', '不幸', '悲しい', '不仕合  
わせ', '不仕合せ'],  
'grieving': ['悲しい'],  
'sad': ['哀しげ',  
         '心悲しげ',  
         '哀れ気',  
         '悲しげ',  
         'もの悲しい',  
         '悲痛',  
         '心悲しい',  
         'うら悲しい',  
         '悲しい',  
         '物悲しい',  
         '沈痛',  
         '哀しい',  
         '物哀しい',  
         'うら悲しげ',  
         'もの哀しい'],  
'sore': ['悲壮',  
         '有痛性',  
         '甚い',  
         '苦しい',  
         '辛い',  
         'つらい',  
         '痛い',  
         '悲しい',  
         '心苦しい',  
         '切ない',  
         '心憂い'],  
'sorrowful': ['悲しい'],  
'unhappy': ['不幸せ', '不幸', '薄幸', '悲しい', '不仕合  
わせ', '薄倖', '不仕合せ']}
```

### 2.2.2 MeCab

MeCabは、形態素解析エンジンである。言語、辞書、コーパスに依存しない汎用的な設計であり、入力された単語ごとに分け、品詞、活用型、活用形、原型、読み、発音を出力する。以下のリスト 2.2.2 は、“私は大分高専の学生です”という文章を、MeCabを用いて形態素解析を実行した例である。また、リスト 2.2.3 はリスト 2.2.2 の文章を単語ごとに分ける分かち書きを実行した例である。

リスト 2.2.2

“私は大分高専の学生です”の形態素解析

私	名詞,代名詞,一般,*,*,*私,ワタシ,ワタシ
は	助詞,係助詞,*,*,*は,ハ,ワ
大分高専	名詞,固有名詞,組織,*,*,*大分高専,オオイトコウセン,オーイタコーセン
の	助詞,連体化,*,*,*の,ノ,ノ
学生	名詞,一般,*,*,*学生,ガクセイ,ガクセイ
です	助動詞,*,*,*特殊・デス,基本形,です,デス,デス
EOS	

リスト 2.2.3

“私は大分高専の学生です”の分かち書き

私 は 大分高専 の 学生 です
------------------

本研究では学習データを対話システムに学習させるときや、学習データ内に含まれている単語とその頻出回数を計算する際に、リスト 2.2.3 のような分かち書きを用いる。

## 2.3 評価方法

### 2.3.1 テキストペア類似度

言い換えを行った出力文章と行っていない出力文章の比較方法には、goo ラボ(<https://labs.goo.ne.jp/>)のテキストペア類似度 API[16, 17]を用いる。この API は文章や語句をベクトル化し、文章間の意味的な近似度を定量的に表す。単に文章間の一致度を表すのではなく、表現が異なる文章の類似度を測ることができる。類似度の値は 0-1 の範囲で出力される。この API により出力された値をもとに比較する。文章間の意味が近い場合はおおむね 0.6 以上の値が出力される。以下の表 2.3.1 は、“今日の天気は晴れです”という文章と異なる文章の類似度を、テキストペア類似度 API を用いて出力した値を示している。

表 2.3.1 “今日の天気は晴れです”の類似度

文章	類似度
----	-----

今日の天気は雨です	0.960
晴れだね	0.917
明日雪が降るそうですよ	0.750
好きな食べ物はハンバーグです	0.500

### 2.3.2 評価用データ

対話システムに入力する文章や評価に用いる文章は、使用する学習データから入力文章とそれも対応する出力文章を 1 セットとして無作為に抽出し、入力文章を対話システムに入力する文章とする。得られた出力文章、言い換えによって得られた出力文章と学習データから抽出した出力文章との類似度をそれぞれ出力し、値を比較する。また、言い換える前の入力文章と言い換えた後の入力文章の類似度も出力することで、言い換えを行っても意味的に近似であるか否かを確認する。

## 3. 実験結果

本章は得られた実験結果を示す。3.1 は本研究で用いた対話システムの学習データや評価用データについて述べる。3.2 は実際に得られた結果を分類してまとめ、3.3 は得られた結果の中でも特殊な事例としてまとめる。

### 3.1 事前データ

本研究に用いた対話システムが学習したデータのサイズは、Encoder が 417KB、Decoder が 354KB であり、会話文の合計は 10,722 セットであった。また、評価用データは 712 セットであり、そのうち言い換えを行った文章は 22 セットであった。

### 3.2 結果

実際に得られた結果の例を分類して以下に示す。なお、結果は以下の表 3.2 のように示す。入力文章を In、それに対する評価用データの回答文章を Ans、対話システムの出力文章を Out。入力文章のうち学習データにない単語を UWord、言い換えを行った単語を CWord、言い換えを行った入力文章を CIn、それに対する出力文章を COut、Ans と Out の類似度を A-O、In と CIn の類似度を I-CI、Ans と COut の類似度を A-CO とする。つまり、言い換えを行った場合と言い換えを行わなかった場合の回答の比較とは、上記の記号を用いると A-CO と A-O の比較である。

表 3.2 結果の示し方

In	入力文章
Ans	評価用データの回答文章
Out	対話システムの出力文章

A-O	Ans と Out の類似度
UWord	入力文章内の未知単語
CWord	言い換えを行った単語
CIn	言い換えを行った入力文章
I-CI	In と CIn の類似度
COut	CIn に対する出力文章
A-CO	Ans と COut の類似度

### 3.2.1 言い換えできる単語が一つのみ

入力文章のうち言い換えできる単語が一つのみだった場合の結果を以下に示す。

表 3.2.1 結果例 1

In	密かに押してるでしょ
Ans	押してない押してない
Out	わかってるよーある
A-O	0.518
UWord	密か
CWord	秘密
CIn	秘密に押してるでしょ
I-CI	0.795
COut	わかってる出てなかった
A-CO	0.534

表 3.2.1 は、入力文章のうち言い換えを行った単語が一つのみだった例である。In の“密かに押してるでしょ”のうち“密か”という単語を“秘密”という単語に言い換えた場合の類似度は 0.518 から 0.534 と向上している。なお、“秘密”という単語の学習データ内の頻度は 2 回であった。

### 3.2.2 UWord は複数あるが言い換えできる単語が一つのみ

入力文章のうち UWord は二つ以上あったが言い換えを行った単語が一つのみだった例を示す。

表 3.2.2 結果例 2

In	私それをお金を集めてそれを贈る係の人がまたいるわけ
Ans	いるわけ
Out	あー
A-O	0.0
UWord	贈る, 係
CWord	渡す
CIn	私それをお金を集めてそれを渡す係の人が

	またいるわけ
I-CI	0.953
COut	いやのああ
A-CO	0.503
CWord	上げる
CIn	私それをお金を集めてそれを上げる係の人がまたいるわけ
I-CI	0.944
COut	あー本ああ
A-CO	0.494

表 3.2.2 は、In のうち“贈る”と“係”が UWord だが、“係”を言い換えることができなかったため“贈る”を“渡す”と“上げる”に言い換えている。A-O は 0.0 だったが A-CO は 0.503, 0.494 と類似度はいずれも大幅に向上している。

### 3.2.3 言い換えできる単語が二つ以上

入力文章のうち言い換えを行うことのできる単語が二つ以上ある場合の結果を以下に示す。

表 3.2.3 結果例 3

In	タイのさーなんかとうもろこしにミルクみたいなのかけてさーあれあれすっごいおいしかったねー
Ans	おいしかったね
Out	ほ私
A-O	0.0
UWord	とうもろこし, ミルク
CWord	豆
CIn	タイのさーなんか豆にミルクみたいなのかけてさーあれあれすっごいおいしかったねー
I-CI	0.939
COut	さー私もね一日に本当に
A-CO	0.496

表 3.2.3 は UWord が“とうもろこし”, “ミルク”と二か所ある中, “とうもろこし”を“豆”に言い換えた例である。A-CO は 0.496 と A-O の 0.0 に比べ大幅に向上した。次に“ミルク”を言い換えた例を示す。

表 3.2.4 結果例 3 の別の言い換え

CWord	牛乳
CIn	タイのさーなんかとうもろこしに牛乳みたいなのかけてさーあれあれすっごいおいし

	かったねー
I-CI	0.941
COut	私私も0見てっもう
A-CO	0.487

表 3.2.4 は表 3.2.3 の“ミルク”を“牛乳”に言い換えた例である。この場合は、表 3.2.3 の COut と異なるが、A-CO は 0.487 と向上している。次に“とうもろこし”と“ミルク”の二つの単語を言い換えた例を示す。

表 3.2.5 結果例 3 を二か所言い換え

CWord	コーン,牛乳
CIn	タイのさーなんかコーンに牛乳みたいなのかけてさーあれあれすっごいおいしかったねー
I-CI	0.878
COut	私私も0見てっもう
A-CO	0.487

表 3.2.5 は表 3.2.3 の UWord, “とうもろこし”と“ミルク”の二つの単語を言い換えた例である。一か所言い換えた場合と比べ I-CI の値が低下したが、その他は特に変化なかった。

### 3.2.4 言い換えてできる単語が複数あるが UWord がすべて言い換えられない

以下に示す表 3.2.6 は、UWord が複数あり言い換えを行うことのできる単語も複数あるが、言い換えを行うことができない単語も含む例である。

表 3.2.6 結果例 4

In	お部屋の模様替えで棚を買うんだけどー私ほら狙ってた棚がもう載ってたカタログが期限切れでー
Ans	あこの前買ったやつやろ？
Out	でも0本当に終わりでいろいろいろいろ進んで進んで
A-O	0.495
UWord	お部屋, 模様替え, 棚, 棚, カタログ, ー
CWord	チェンジ
CIn	お部屋のチェンジで棚を買うんだけどー私ほら狙ってた棚がもう載ってたカタログが期限切れでー
I-CI	0.965
COut	で00そこ置いて置いてさー
A-CO	0.511

表 3.2.6 の例は UWord が“お部屋”, “模様替え”, “棚”, “カタログ”と“ー”であり、そのうち“お部屋”, “模様替え”と“カタログ”が言い換え可能であり、そのうち“模様替え”を“チェンジ”に言い換えた。A-O は 0.495 で A-CO は 0.511 と類似度は向上している。次に、言い換えを行うことのできる単語, “お部屋”, “模様替え”と“カタログ”を全て言い換えた例を示す。

表 3.2.7 結果例 4 を三か所言い換え

CWord	室,換える,表
CIn	室の換えるで棚を買うんだけどー私ほら狙ってた棚がもう載ってた表が期限切れでー
I-CI	0.889
COut	でもそこもいろいろ大きいの
A-CO	0.541

表 3.2.7 は表 3.2.6 を三か所すべて言い換えた例である。“お部屋”は“室”に, “模様替え”は“換える”, “カタログ”は“表”に言い換えた。A-CO は 0.541 と類似度は向上している。

### 3.3 類似度が 0 となる例

言い換えを行わない出力文章の類似度 A-O が 0 出ないときに、言い換えを行った出力文章の類似度 A-CO が 0 となる例がある。以下に示す表 3.2.8 は言い換えてできる単語が一つの場合である。

表 3.2.8 結果例 5

In	私たぶんね何か子供に対してどうもなーんかこいつらは何をやらかすかわかんないもんね
Ans	あーっ確かにそはあるけどね
Out	美容院決めて絶対
A-O	0.513
UWord	やらかす
CWord	やる
CIn	私たぶんね何か子供に対してどうもなーんかこいつらは何をやるかわかんないもんね
I-CI	0.856
COut	そうね
A-CO	0.0

表 3.2.9 結果例 5 別の言い換え

CWord	する
-------	----

CIn	私たぶんね何か子供に対してどうもなーんかこいつらは何をするかわかんないもんね
I-CI	0.909
COut	そうね
A-CO	0.0

表 3.2.8 は類似度 A-CO が 0.0 となる例を示しており、表 3.2.9 は表 3.2.8 の In のうち別の単語に言い換えた場合の結果を示している。表 3.2.8 より、“やらかす”という UWord を“やる”という単語に言い換えた。I-CI の値は 0.856 と高い類似度ではあるが A-CO は 0.0 だった。また、表 3.2.9 より“やらかす”という単語を“する”という単語に言い換えた場合も、I-CI は 0.909 であったが A-CO は 0.0 であった。“やる”と“する”はいずれも学習データ内に 79 回、331 回と高頻度であるが、類似度は極端に低下した。また、このように A-CO が 0.0 となる例は、複数言い換えを行った単語がある例にもあった。その例を以下に示す。

表 3.2.10 結果例 6

In	そいでああいうやっぱり先生はねあの今金八先生やってるけど情熱がね子どもに伝わるんですよ
Ans	うんうん
Out	あそうですことですね
A-O	0.0
UWord	金八先生, 熱, 伝わる
CWord	パッション, 浸透
CIn	そいでああいうやっぱり先生はねあの今金八先生やってるけどパッションがね子どもに浸透んですよ
I-CI	0.882
COut	あそうのんですね
A-CO	0.0
CWord	熱心, 普及
CIn	そいでああいうやっぱり先生はねあの今金八先生やってるけど熱心がね子どもに普及んですよ
I-CI	0.884
COut	あそうなんですね
A-CO	0.0

表 3.2.10 は UWord の“情熱”と“伝わる”をそれぞれ“パッション”, “浸透”と“熱心”, “普及”に言い換えた例である。I-CI の値は 0.882, 0.884 と高いが、A-CO の値はどちらも A-O と変わらず 0.0 であった。

次に示す表 3.2.11 は、表 3.2.2 の結果例 2 の UWord, “贈る”をほかの類義語に言い換えたときに類似度が 0.0 となる例である。

表 3.2.11 結果例 2 の別の言い換え

CWord	くれる
CIn	私それをお金を集めてそれをくれる係の人がまたいるわけ
I-CI	0.957
COut	あーなんやああです
A-CO	0.0
CWord	与える
CIn	私それをお金を集めてそれを与える係の人がまたいるわけ
I-CI	0.943
COut	ええええじゃんああ
A-CO	0.0

表 3.2.11 は UWord を“くれる”と“与える”に言い換えた例である。A-CO は A-O と変わらず 0.0 と表 3.2.2 の 0.503 に比べ類似度が極端に低下した。

表 3.2.2 の例では、A-O が 0.0 の場合でも A-CO の値が向上した。しかし、同様に A-O が 0.0 の文章 13 セットのうち A-CO の値が向上したのは上記の文章例も含めた 3 セットのみであった。その 3 セットについても複数行った言い換えのうち A-CO の値が向上したのはごくわずかであり、大半は A-CO も 0.0 と変化しなかった。

以上のことから、言い換える語句によっては類似度が 0 となる、つまり回答精度が極端に低下する例も少なからずあった。その原因として考えられることは、対話システムが出力した出力文章と評価用データの回答文章の意味的な差である。しかし、用いた評価用データが少なく、回答精度が極端に低下する法則性を発見できなかったため、特定できなかった。よって本研究では、この現象は用いた評価用データに問題があると仮定し、次の 4 章で示すデータには A-O が 0.0 だった場合と A-CO が 0.0 だった場合の結果を除いたデータを示す。

## 4 実験結果のまとめと考察

本章は得られた実験結果のまとめと考察を述べる。4.1 では得られた実験結果をまとめたデータとそれに対する考察を、4.2 では類似度が 0 となった原因に関する考察を述べる。

## 4.1 実験結果のまとめ

以下の表4は、入力文章と言い換えを行った入力文章間の類似度 I-CI、評価用データの回答文章と対話システムの出力文章間の類似度 A-O と、評価用データの回答文章と言い換えを行った入力文章に対する対話システムの出力文章間の類似度 A-CO の平均値を示している。

表4 類似度の平均

I-CI	0.896
A-O	0.505
A-CO	0.507

表4より、A-O と A-CO の平均値はそれぞれ 0.505、0.507 となり、わずかではあるが言い換えを行った方が類似度は 0.4% 向上している。

よって、対話システムに入力する文章のうち学習データに含まれていない語句を抽出し、その語句の類義語の中から学習データに含まれている語句へ無条件に言い換えた場合、言い換えを行わなかった場合の回答と比較し回答の精度が向上するといえる。

だが、予想していたほど回答の精度は向上しなかった。その原因として考えられることは言い換え方法である。本研究で用いた言い換え方法は、2.2 で述べたように語彙資源を用いた無条件での言い換えである。言い換えの対象となる単語とその類義語の品詞や活用形が一致していないときに言い換えを行うと、言い換え後の文章が正しい構成で成り立っていない。例えば、表 3.2.10 の結果例 6 のような“伝わる”という動詞を“浸透”という名詞に言い換えたため、言い換え後の文章が正しい構成で成り立っていない。正しい構成でない文章に言い換えたときに精度が向上した例もあるが低下した例もあり、このような言い換えに対して言い換えを行わないことや、品詞、活用形を統一することでさらに精度が向上すると考えられる。

また、学習データに含まれていない単語があるがその類義語が存在しない、もしくは類義語も学習データに含まれていないために言い換えることができない例があった。具体的な例としては表 3.2.2 の結果例 2

“係”のような類義語が無いような単語や表 3.2.10 の結果例 6 “金八先生”のような固有名詞である。このような単語に対して本研究ではそのまま言い換えを行わなかったが、固有名詞や類義語がない単語の言い換え方法は今後検討する必要がある。

## 4.2 類似度の変化について

3.3 でも述べたように、言い換える語句によっては類似度が 0 となる、つまり回答精度が極端に低下する場合も少なからずあった。この原因として考えられるの

が、評価用データである。本研究では、対話システムに用いた学習データから会話文のセットを無作為に抽出し、それを評価に用いた。しかし、学習データは会話文となっており、前の会話文の内容を踏まえた会話文となっている。抽出した会話文のセットには、それぞれ前の会話文の内容を踏まえた入力文章と出力文章となっているため、前の会話文を記憶できない対話システムは、流れを把握していない回答となる。つまり、評価用データと意味的にも差が生じたと考えられる。よって、このような対話システムの出力文章と評価用データの出力文章に意味的な差が生じないためにも、評価用データは単独の入力文章に対応する出力文章のセット、つまり、一つの入力文章に対して一つあるいは複数の出力文章をセットにした対話人工知能用の評価用コーパスを用意した方が、より目的に適していることが明らかになった。

## 5. 結言

本論文では、対話システムに学習させるデータ量が少なくとも回答精度を高める手段として、対話システムに入力する文章のうち学習データに含まれていない語句を抽出し、その語句の類義語の中から学習データに含まれている語句へ無条件に言い換える、語句言い換えの手法を提案した。また、言い換えを行った場合の対話システムの回答と言い換えを行わなかった場合の対話システムの回答と比較し回答の精度が向上するか実験を行った。なお、評価方法として、あらかじめ学習データの会話文の中から抽出した入力文章に対応する出力文章を評価用データとし、評価用データと対話システムの回答文章の二文章間の意味的な類似度が、言い換え前より言い換え後の方が高い場合、精度が向上したと評価した。

言い換え前の類似度の平均値に比べ言い換え後の類似度の平均値の方が高く、0.4% 向上していた。この結果から、対話システムに入力する文章のうち学習データに含まれていない語句を抽出し、その語句の類義語の中から学習データに含まれている語句へ無条件に言い換えを行った場合、回答精度が向上することがわかった。

以上のことから、言い換えを行う方が回答精度は向上する。しかし、言い換えを行うことで回答精度が極端に低下することもあり現実的な手法であるとは言えないため、対話人工知能用の評価用コーパスを作る等の課題を解決することが回答精度をさらに向上させるためにも必要である。

## 謝辞

本研究に際して、様々なご指導を頂きました西村俊二講師に深く感謝いたします。また、この研究の機会を

くださった情報工学科の先生方, そして多くの知識やご指摘を下さいました同研究室の先輩・同期の皆様へ厚く御礼申し上げます。

## 参考文献

- [1] 乾健太郎, 藤田篤: 言い換え技術に関する研究動向, 自然言語処理, Vol. 11, No. 5, pp.151-198, 2004.
- [2] 藤田篤, 乾健太郎: 語積分を利用した普通名詞の洞概念後への言い換え, 言語処理学会第7回年次大会発表論文集, pp.331-334, 2001.
- [3] Edmonds, P. Semantic : Representations of Near Synonyms for Automatic Lexical Choice. Ph.D. thesis, published as technical report CSRI-399, Department of Computer Science, University of Toronto, 1999.
- [4] 白井諭, 池原悟, 河岡司, 中村行宏: 日英機械翻訳における原文自動書き換え型翻訳方式とその効果, 情報処理学会論文誌, Vol.36, No.1, pp.12-21, (1995).
- [5] 関沢祐樹, 梶原智之, 小町守: 目的言語の低頻度語の高頻度語への言い換えによるニューラル機械翻訳の改善, 言語処理学会第23回年次大会発表論文集, pp.982-985, 2017.
- [6] MNIST for ML Beginners., “Tensorflow”, [オンライン]. Available: <https://www.tensorflow.org/>, [最終アクセス日:19 12 2018].
- [7] sanshirookazaki, “GitHub – sanshirookazaki/chat: Tensorflow: lstm, seq2seq model”, [オンライン]. Available: <https://github.com/sanshirookazaki/chat>. [最終アクセス日:20 12 2018].
- [8] Pu-of-Parari, “GitHub - Pu-of-Parari/Japanese-Neural-Conversational-Model”, [オンライン]. Available: <https://github.com/Pu-of-Parari/Japanese-Neural-Conversational-Model>, [最終アクセス日:20 12 2018].
- [9] 田村和弘: リカレントニューラルネットワークを用いた対話システムの試作と評価, The 32<sup>nd</sup> Annual Conference of the Japanese Society for Artificial Intelligence, 2018, 2018.
- [10] Ilya Sutskever, Oriol Vinyals, Quoc V. Le. : Sequence to Sequence Learning with Neural Networks, 2018.
- [11] Fujimura, Itsuko, Shoji Chiba, Mieko Ohso : Lexical and Grammatical Features of Spoken and Written Japanese in Contrast: Exploring a lexical profiling approach to comparing spoken and Written corpora, Proceedings of the VIIth GSCP International Conference. Speech and Corpora,, 393-398, 2012.
- [12] NOKUBI Takatsugu, “make-meidai-dialogue: Get Japanese dialogue corpus”, [オンライン]. Available: <https://github.com/knok/make-meidai-dialogue>, [最終アクセス日: 19 12 2018].
- [13] Kyoko Kanzaki, Francis Bond, Noriko Tomuro and Hitoshi Isahara : Extraction of Attribute Concepts from Japanese Adjectives., In LREC-2008, Marrakech, 2008.
- [14] 宗定洋平, “[自然言語] Wordnet × Python で類義語を抽出する”, [オンライン]. Available: <https://www.yoheim.net/blog.php?q=20160201>, [最終アクセス日:19 12 2018].
- [15] Taku Kudo, Kaoru Yamamoto, Yuji Matsumoto : Applying Conditional Random Fields to Japanese Morphological Analysis, Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004), pp.230-237 (2004.).
- [16] K.Bessho, O.Furuse, R.Kataoka, and M.Oku : Kanshinji Antenna : A Japanese –language Concept Search System, IJHCI, Vol.23, Issue 1&2, pp.25-49, 2007.
- [17] NTT Resonant Inc., “テキストペア類似度 API”, [オンライン]. Available: [https://labs.goo.ne.jp/api/textpair\\_doc](https://labs.goo.ne.jp/api/textpair_doc). [最終アクセス日:20 12 2018].