

2022 年度卒業論文  
OCC theory を用いた文章からの感情生起

奈良女子大学 生活環境学部 情報衣環境学科 生活情報通信科学コース 4 回生  
新出研究室 19480289 畔柳愛子

## 概要

近年、ロボットに感情を与える研究が進んでいる。ロボットが人間と対話するにあたり、相手の感情を正しく理解できるようになれば、人間らしいその場に応じた行動を選択できるようになる。従来の研究 [7][8][17][18] では、OCC theory と BDI モデルを組み合わせた Adam らの形式化に基づいて感情を形式化、実装していた。しかし、感情生起の条件となる信念は、感情を生起させる対象であるイベントを論理式に変換したり、構文解析が行える簡単な日本語に変換して与える必要があった。そこで本研究では、環境からの情報を持つ文章が与えられると想定し、与えられた文章から人手を経ずに情報を論理式の形に変換し、信念としてエージェントに与えて感情生起を行うプログラムの作成を行った。なお、本論文では、作成したプログラムについて述べ、その検証結果、課題について述べる。

## 1 はじめに

近年、人工知能の発達により、人間と意思疎通ができるロボットの開発が進んでいる。ロボットが人間と同様、感情生起を行うことができればより人間に近い行動選択をすることが期待できる。そもそも感情とは、脳内の扁桃体から生まれると言われており、扁桃体は、外の環境が危険であれば不快、安全なら快という感情を引き起こす。つまり、自分自身の危険回避のための素早い判断を下すために重要な要素である [1]。そこで、人間らしい行動選択を行うためにはエージェントにも人間らしい感情を生起させることが必要となると考えられる。

人間同士がコミュニケーションを取る際には、表情や身振り手振りなどのノンバーバルコミュニケーションも必要な要素となるが、主要な要素として言葉から相手の考えを読み取ることが挙げられる。そのため、言葉を通して適切な感情を生起させることがロボットにとっても必要になってくると考えられる。

感情生起を行い、人間らしい行動選択を行うエージェントの実現には、BDI アーキテクチャが有効である。BDI アーキテクチャとは合理的行動を行うエージェントである BDI モデルの行為決定方式を計算機上で実現したものである。これにより、BDI エージェントの構築が可能である。また、BDI モデルではエージェントの振る舞いの形式的な記述に BDI logic と呼ばれる論理体系を用いる。

一方で、人間らしい感情の理論として、OCC theory [2] が挙げられる。OCC theory とは、人間の包括的な感情を 22 種類に分類した理論であり、感情の生起条件が明確であるため比較的理解しやすいとされる。また、OCC theory は BDI モデルが持つ信念や願望などの概念を用いて感情を特徴づけているため、BDI モデルとは親和性が良く、どちらも BDI logic を用いて記号論理で扱うことができる。Adam らは、OCC theory を BDI モデルに組み込むことで感情を論理式で表現した [5]。

Adam らによる感情の表現は、後述の通り、BDI アーキテクチャの既存の実装である Jason 上での信念ベースを用いて実装が可能である。これを用いて感情を生起させる際には、bel, des, prob などの信念が必要となる。先行研究では、感情に必要な信念をシナリオから手動で信念を抽出して入力したり、形式文法による構文解析を行うことができる形で入力していた。人間とコミュニケーションをとる上で、エージェント自身が環境から必要な信念を取り出し、感情生起を行うことが望ましい。そこで、本研究では、環境からの情報を持つ文章が与えられると想定し、人手を経ずに情報を論理式の形に変換するプログラムの実装を行っている。そしてその信念を用いて適切な感情を生起できることを確認する。なお、本研究では、与えられる文章としては物語文を想定した。他者とのコミュニケーションにおいて感情生起するという目的からは、会話文を対象とすることが望まれるが、本研究ではそこまでは目指さないものとする。

## 2 関連研究

本章では、感情の理論 OCC theory と合理的行動のモデルである BDI モデル、それらを用いて感情の生起条件を形式化した Adam らの研究について述べ、従来に実装された感情生起を行う自律エージェントの研究の概要と、課題点を述べる。

### 2.1 OCC theory

OCC theory[2] とは Ortony, Clore, Collins らが提唱した、心理学的知見を元に 22 種類の感情タイプをモデル化した理論である。これは人間の感情を信念や願望といった心的状態やその時間経過による変化に基づいて特徴づけ、それによって分類している。それぞれの特徴づけが明確であり、論理モデルを用いた表現が可能のため計算機科学分野において広く用いられている。

OCC theory によって分類された 22 種類の感情のタイプは大きく 3 つのクラスに分類されており、それぞれをさらにいくつかのグループに分類している。以下にその分類を示す。

1. 結果の望ましさについて
  - (a) 自分にとっての望ましさについて
    - i. イベントに関して
      - Joy(喜び), Distress(悲しみ)
    - ii. 予想に関して
      - A. 単なる予想に対して
        - Hope(望み), Fear(恐れ)
      - B. 予想していたことが起こった
        - Satisfaction(満足), FearConfirmed(恐れていたことが確定)
      - C. 予想していたことが起きなかった
        - Relief(安堵), Disappointment(落胆)
  - (b) 他者にとっての望ましさについて
    - i. 他者が良い結果を得た
      - HappyFor(ともに喜ぶ), Resentment(憤り)
    - ii. 他者が悪い結果を得た
      - SorryFor(ともに残念に思う), Gloating(ほくそ笑む)
2. 行動の賞賛について
  - (a) 自分の行動に関して
    - Pride(誇り), Shame(羞恥心)
  - (b) 自分の望ましさに対する自分の行動に関して (Joy と Distress との混合型)
    - Gratification(満足), Remorse(後悔)
  - (c) 他者の行動に関して
    - Admiration(賞賛), Reproach(非難)
  - (d) 自分の望ましさに対する他者の行動に関して (HappyFor と Resentment との混合型)
    - Gratitude(謝意), Anger(怒り)

### 3. 対象物に対する好き嫌いについて

— Love(好き), Hate(嫌い)

## 2.2 BDI モデル

本研究は、BDI モデルを元にして行った。BDI モデルとは、信念 (Belief)、願望 (Desire)、意図 (Intention) の 3 つの心的状態を用いて人間の合理的行動をモデル化したものである [3]。達成する目標と手段を信念や願望によって選定するという意図の理論を行為決定に適用している。BDI モデルの行動様式により、自律的に行動するエージェントを BDI エージェントといい、BDI logic という様相論理体系を用いて形式的に記述できる。また、BDI モデルの行為決定方式を計算機上で実現したものである BDI アーキテクチャの既存の実装に Jason がある [4]。本研究では、Jason を用いてエージェントの感情生起を実装する。

## 2.3 Adam らによる形式化

Adam らの形式化 [5] では、OCC theory の感情を BDI モデルに組み込むことで、感情を論理式で形式化している。OCC theory の 22 種類の分類を以下の 6 グループに分類することで、グループごとに同じ形式を持つ感情生起条件が定義されている。

- Well-being emotions : Joy, Distress
- Prospect emotions : Hope, Fear
- Confirmation emotions : Satisfaction, FearConfirmed, Relief, Disappointment
- Fortunes-of-others emotions : HappyFor, Resentment, SorryFor, Gloating
- Attribution emotions : Pride, Shame, Admiration, Reproach,  
Gratification, Remorse, Gratitude, Anger
- Attraction emotions : Love, Hate

## 2.4 先行研究の課題

先行研究では、感情を生起させるのに必要な論理式を人間が何らかの形で手を加えることによって入力していた。[7] では、シナリオから感情生起を行っていたが、シナリオにおける対象のイベントを手動で論理式に変換をしていた。また、[8] では、LR 解析に基づく構文解析プログラムを用いているため、解析できる文の形が限られていたり、解析する際には主語を補うなど、構文解析を行うことができる形に文章を直す必要がある。

## 3 実装

従来の研究では、事例として物語文からの感情生起を扱っていた。それらでは、感情生起の条件となる信念を、与えられた物語文から人間が解釈して入力したり、構文解析を行うことができる簡単な文章に直してから入力する必要があった。しかし、これでは環境からの自然な入力から感情生起を行うことができない。そのため、与えられた文章からエージェント自身が必要な信念を取り出し、感情生起を行うことが望ましい。そこで、本研究では、環境からの情報を持つ文章が与えられると想定し、人手を経ずに情報を論理式の形に変換

し、感情生起を行うことができるように実装した。実装にあたり、まず Prolog の事実の形で出力することを目指した。その後必要な機能を加えつつ、des, bel などの適切なオペレータを追加することによって信念の出力を目指した。

### 3.1 構文解析

入力から得た情報を Prolog の事実の形で出力する際には、自然言語に対する構文解析を用いて文章の係り受け解析を行う必要がある。そのために、日本語係り受け解析器である CaboCha/南瓜 [9] を用いることとした。CaboCha/南瓜は形態素解析と構文解析を行うライブラリである。形態素解析とは、私たちが日頃使っている言葉である自然言語を、意味を有する最小単位である形態素にまで分割する技術のことである。そして、構文解析とは、文章の内容を解析し、その構成要素がどのような関係にあるかを明らかにすることである。

しかし、CaboCha/南瓜の Python バインディング 0.69 には、バッファの使い回しによるバグが存在し、想定する結果を出力することができなかった。そのため本研究では実際には日本語係り受け解析エンジン UniDic2UD の CaboCha/南瓜エミュレータ [10] を用いた。安岡 [11] によると、UniDic2UD は単語間の係り受けを解析するのに対し、CaboCha は二文節間の係り受けを解析しており、単語間の係り受けは、二文節間の係り受けとは微妙に対応しない部分がある。そのため挙動が変わる部分もあるが、同等の能力を備えていると考えられるため採用した。

「太郎は次郎に本を渡した。」という文章を CaboCha/南瓜エミュレータで解析した結果が以下の図 1 である。

```
* 0 3D 0/1 0.000000
太郎 名詞, 固有名詞, 人名, 名, *, *, タロウ, タロウ, *, PROPN B-PERSON      1<-nsubj-7
は     助詞, 係助詞, *, *, *, *, は, ハ, *, ADP 0          2<-case-1
* 1 3D 0/1 0.000000
次郎 名詞, 固有名詞, 人名, 名, *, *, ジロウ, ジロウ, *, PROPN B-PERSON      3<-obl-7
に     助詞, 格助詞, *, *, *, *, に, ニ, *, ADP 0          4<-case-3
* 2 3D 0/1 0.000000
本     名詞, 普通名詞, 一般, *, *, *, 本, ホン, *, NOUN 0          5<-obj-7
を     助詞, 格助詞, *, *, *, *, を, ヲ, *, ADP 0          6<-case-5
* 3 -1D 0/1 0.000000
渡し 動詞, 一般, *, *, *, *, 渡す, フタシ, *, VERB 0          7<-root
た     助動詞, *, *, *, *, *, た, タ, *, AUX 0            8<-aux-7
。     補助記号, 句点, *, *, *, *, 。, *, *, PUNCT 0          9<-punct-7
EOS
```

図 1 CaboCha/南瓜エミュレータでの解析結果

図 1 では、「太郎は次郎に本を渡した。」という文章を「太郎」、「は」、「次郎」、「に」、「本」、「を」、「渡し」、「た」、「。」という形態素に分解を行なっている。この形態素をトークンともいう。また、それぞれの形態素の右側にそれぞれの品詞、その分類、活用形、原型、読みなどを出力している。それぞれの形態素の右側には係り受けタグ [12] がついている。「太郎」の係り受けタグは「1<-nsubj-7」である。nsubj というのは主語を表し、形態素 1 である「太郎」は主語であり、形態素 7 である「渡し」にかかっているということを表す。

また、トークンのまとまりである文節をチャンクといい、「太郎は」、「次郎に」、「本を」、「渡した。」が当てはまる。図 1 では、アスタリスクの次に 0 から始まる文節番号が振られており、その次にそのチャンクの係り先のトークンの番号が振られている。

これらの係り受け関係を木構造として出力したのが図 2 である。図 3 は、図 2 の出力を図に表したもので



Prolog 言語では、事実を「述語名 (引数  $1, \dots, \text{引数}_n$ )」の形で記述し、「引数  $1, \dots, \text{引数}_n$ 」の間には述語名で表される関係が成り立つことを意味する。引数には定数や数値、変数などを書くことができ、「～が」という対象を表す。述語は「～である」という性質を表す。

```
male(kazuma).  
female(haruka).
```

図5 Prolog の事実

図5では、Prolog の事実の例を示す。これは、カズマは男性であること、ハルカは女性であることを示している。

```
?- female(haruka).  
true.  
  
?- female(kazuma).  
false.
```

図6 Prolog の質問

図6は図5の事実を用いて質問を行なった様子である。上の質問はハルカは女性であるかという意味の質問であり、図5よりハルカは女性であるという事実が存在するため true が出力されている。下の質問はカズマは女性であるかという意味の質問であり、図5よりカズマは女性であるという事実は存在しないため false が出力されている。

Jason でも Prolog と同様、事実は「述語名 (引数  $1, \dots, \text{引数}_n$ )」の節形式で記すことができる。

### 3.3.2 係り受け関係

先述の通り、CaboCha/南瓜エミュレータでは係り受け関係を解析することができる。その係り受け関係を用いて Prolog の事実と同様の節形式で出力を行う。出力は以下の『事実の出力順のルール』に従って行う。

基本的には、係り受け関係が続く限り、同一の事実として出力を行う。しかし、それでは不適當な場合があるので、それを回避するため以下の『述語や引数のルール』を追加した。

- 事実の出力順のルール

**引数が主語だけの時** 述語 (主語)

ex) 「太郎は歩いた。」 歩く (太郎)

**引数が主語と目的語の時** 述語 (主語, 目的語)

ex) 「太郎は本を渡した。」 渡す (太郎, 本)

**引数が主語と補語の時** 述語 (主語, 補語)

ex) 「太郎は駅まで歩いた。」 歩く (太郎, 駅)

**引数が主語と目的語と補語の時** 述語 (主語, 目的語, 補語)

ex) 「太郎は次郎に本を渡した。」 渡す (太郎, 本, 次郎)

- 述語や引数のルール

**修飾節**

- ・acl(連体修飾節) の係り先で一旦係り受け関係を区切る
- ・advcl(連用修飾節) の係り受け関係は考慮しない

#### 述語

- ・これ以上係り受け関係がないときにその語を述語とする
- ・acl(連体修飾節) の係り元であるときにその語を述語とする

#### 引数

- ・nsubj(主語)、obj(目的語)、obl(補語) は係り受け関係を用いて引数とする
- ・acl(連体修飾節) がある時には係り先を主語とする

名詞である体言について修飾する節を連体修飾節、動詞など用言について修飾する節を連用修飾節という。名詞を修飾する際には別の信念として出力することが適切であると考え、それ以上の係り受け関係を一旦区切って出力を行う。連用修飾節は副詞と同様の役割を果たすため、事実の出力として不要な情報であると考え、考慮せずに出力を行う。

述語に関して、基本的にこれ以上係り受け関係がないものを述語とした。しかし、連体修飾節での係り受け関係が存在する際にはその係り元を述語として出力を行う。

それ以外には基本的には係り受け関係をそのまま引数として扱うが、連体修飾節がある時にはその係り先を主語として出力を行う。

上のルールを用いて事実を出力する流れを以下に示す。例として「太郎は新しいゲームを買った」という文章を用いる。文章をチャンクに分解したものが図7であり、チャンクから必要な情報を取り出す。ここで、必要な要素というのは他のチャンクに係り受け関係があるものとし、図6では「太郎」、「新しい」、「ゲーム」、「買う」である。先述した接頭辞や接尾辞の処理もここで行う。

```
* 0 3D 0/1 0.000000
太郎 名詞,固有名詞,人名,名,*,*,タロウ,タロウ,*,PROPN B-PERSON      1<-nsubj-6
は     助詞,係助詞,*,*,*,*,は,ハ,*,ADP 0      2<-case-1
* 1 2D 0/0 0.000000
新しい 形容詞,一般,*,*,*,*,新しい,アタラシイ,*,ADJ      0      3<-acl-4
* 2 3D 0/1 0.000000
ゲーム 名詞,普通名詞,一般,*,*,*,ゲーム,*,*,NOUN      0      4<-obj-6
を     助詞,格助詞,*,*,*,*,を,ヲ,*,ADP 0      5<-case-4
* 3 -1D 0/1 0.000000
買った 動詞,一般,*,*,*,*,買う,カツ,*,VERB      0      6<-root
た     助動詞,*,*,*,*,*,た,夕,*,AUX      0      7<-aux-6
EOS
```

図7 CaboCha/南瓜エミュレータでの解析結果

取り出したチャンクとその係り受け関係を図で表現したものが、図8である。先述した係り受け関係の整理のルールを用いて、図7を分解したものが図9、図10である。ルールより、図9ではacl(連体修飾節)の係り先である「ゲーム」が主語、係り元である「新しい」が述語となる。図10ではこれ以上係り先がない「買う」が述語、「太郎」は主語、「ゲーム」が目的語である。

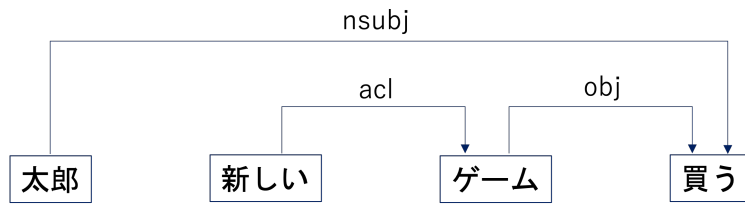


図8 図7を図式化

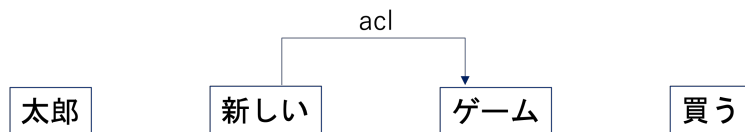


図9 図8の分解1

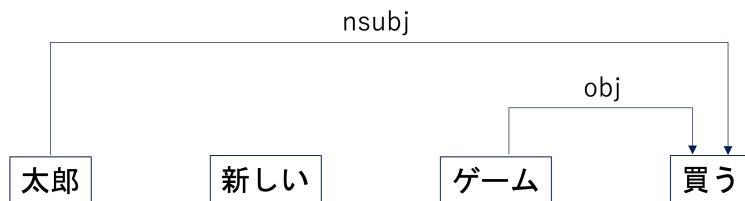


図10 図8の分解2

以上のルールを用いて事実を出力した結果が以下の図11である。

新しい ( ゲーム )  
 買う ( タロウ , ゲーム )

図11 出力結果

「太郎」が「タロウ」として出力されているのは、CaboCha/南瓜エミュレータの解析結果において「太郎」の原型が「タロウ」であるからである。

### 3.4 否定語の処理

山下ら [14] は、「自然言語処理の分野では、何を持って否定表現とするか厳密な決まりはなく、各々で定義している。」とし、菅沼ら [15] は「ない」、「ぬ」、「まい」の活用形を抽出し、字面のみから否定表現の検出を試みている。これらに基づき本研究では、以下の『否定表現の判定を行う語句』に含まれているとき否定語がかかっている語句の原型に not を付与して出力した。

- 否定表現の判定を行う語句  
 「ない」「ぬ」「ん」「ず」「非」「不」「無」「未」「反」「異」

### 3.5 事実の出力形式の整理

Jason の信念ベースに入力する際には、述語やアトムの名前に非 ASCII 文字が使えないこと、それらの先頭は英小文字でなければならないことなどの制約が多くある。Jason の信念ベースの文法に適合するように処理を行なった。

#### 3.5.1 英字での出力

まず、日本語文字では Jason の信念ベースの述語やアトムに使用できない。そのため前章までに記した Prolog の事実にある単語を英訳して出力をする。

Google は Google Translate(Google 翻訳) と呼ばれる多機能翻訳サイトを提供しており、Google Trans API を実装した Python ライブラリである googletrans[16] というものが存在する。これは Google Translate と同じサーバを利用しており、簡単に正確な翻訳を行うことができるため採用した。

図 11 を Python の googletrans ライブラリを用いて英訳したものが図 12 である。

```
new ( game )  
buy ( Taro , game )
```

図 12 図 11 を googletrans を用いて英訳

#### 3.5.2 その他の処理

図 12 の出力結果では、引数の先頭が大文字である。これでは Jason に変数記号と解釈されるため、述語または引数の先頭が大文字であるときには小文字に変換を行う。これ以外にも、ハイフンや空白がある時には削除してから事実として出力を行う。

### 3.6 程度を表すアノテーションの追加

石川 [17] は、程度の強さを取り入れた論理体系 [18] をもとに Des, Prob に度合いを表すオペレータをアノテーションとして追加した。以下がその表現であり、 $A \equiv B$  は「論理式  $A$  を Jason の節  $B$  で表現する」を意味する。

$$Bel^i\varphi \equiv \text{bel}(X,Z) \quad Des_d^i\varphi \equiv \text{des}(X,Z)[\text{degOfCert}(D)] \quad Prob_i^i\varphi \equiv \text{prob}(X,Z)[\text{degOfCert}(D)]$$

その上で、生じた感情の度合いによって異なる行動を選択する例を、以下のように実現している。

- 0.1 ≤ D < 0.4 のとき、「すこしの感情を生じたことを宣言する」行動を選択
- 0.4 ≤ D < 0.8 のとき、「感情を生じたことを宣言する」行動を選択
- 0.8 ≤ D ≤ 1 のとき、「とても感情を生じたことを宣言する」行動を選択

また、中山 [19] は、程度副詞とは山田文法以来の通説となった 副詞三分類「情態・程度・陳述」のうちの一つであるとし、程度副詞の下位分類を試みた。

本研究では中山の分類と、石川のアノテーションの定義を用いて、程度副詞に 0 から 1 までの数値を相対的に数値を割り当てた。程度副詞が存在するときには、程度副詞に対応する数値の情報を事実を持たせた。以下

の『付与する数値と判定を行う程度副詞』が、本研究で判定を行なった程度副詞とその相対的な数値である。程度副詞が含まれていないときには 0.5 を付与させる。

- 付与する数値と判定を行う程度副詞
  - 0.1 「少し」、「多少」
  - 0.2 「より」
  - 0.3 「やや」
  - 0.4 「比較的」、「わりあい」、「なかなか」、「けっこう」
  - 0.6 「かなり」、「よほど」、「すこぶる」、「そうとう」、「だいぶ」
  - 0.7 「ずいぶん」、「はなはだ」、「とても」、「たいへん」、「たいそう」、「非常に」、「いっそう」、「ずっと」、「もっと」、「一段と」、「ひときわ」
  - 0.8 「きわめて」、「きわめて」、「すこぶる」、「ごく」
  - 0.9 「あまりに」、「いたって」
  - 1.0 「最も」、「一番」

### 3.7 オペレータの追加

先述の通り、Adam らの論文では Bel, Des, Prob のオペレータを用いて感情生起の条件を表現している。そこで、3.2.2 節で述べた事実の出力の時に判定した述語や引数を用いて以下の『信念の出力ルール』に基づいて論理式として出力する。

- 事実の出力順のルール
  - 引数が主語だけの時** bel( 主語 , 述語 )  
ex) 「太郎は歩いた。」 bel( taro , walk )
  - 引数が主語と目的語の時** bel( 主語 , 目的語 + 述語 )  
ex) 「太郎は本を渡した。」 bel( taro , bookhandover )
  - 引数が主語と補語の時** bel( 主語 , 補語 + 述語 )  
ex) 「太郎は駅まで歩いた。」 bel( taro , stationwalk )
  - 引数が主語と目的語と補語の時** bel( 主語 , des ( 補語 , 目的語 + 述語 ) )  
ex) 「太郎は次郎に本を渡した。」 bel( taro , des( girou , bookhandover ) )  
googletrans を用いているため、次郎は jiro にならない

また、「～してしまった」という文言が含まれている時にはその物事が望ましくない物事であると判断できる。その場合には以下の『望ましくないときの出力ルール』に基づいて出力する。

- 望ましくないときの出力ルール
  - 引数が主語だけの時** des( 主語 , not 述語 )
  - 引数が主語と目的語の時** des( 主語 , not 目的語 + 述語 )
  - 引数が主語と補語の時** des( 主語 , not 補語 + 述語 )
  - 引数が主語と目的語と補語の時** des( 主語 , not des ( 補語 , 目的語 + 述語 ) )

### 3.8 感情を表す言葉に対応する信念の出力

人が話す言葉や、物語文では感情を表現する言葉が最初から含まれていることがある。そのように、感情を表す言葉が直接含まれる場合は、その感情の生起条件に対応する論理式を出力することで正しく感情を生起することが可能である。

OCC theory が定義している感情と、その感情を表すと判定する言葉を以下の『感情を表すと判定する単語』に示す。感情を表すと判定する単語としては、OCC theory で表される感情である単語を日本語訳したものを選定する。ここで、fearconfirmed(恐れていたことが確定する)、happyfor(誰かと共に喜ぶ)、sorryfor(誰かと共に残念に思う)は、感情として思っていたとしても言葉として表現する可能性は低いと考え、それ以外の感情のみ判定を行なった。

ここに示した特定の単語が含まれているときには、その語によって感情が直接表現されているとし、その生起条件に対応する論理式を出力する処理を行なう。

例えば、「おばあさんは、散歩できて喜んだ。」という文章には喜ぶという感情を表現する言葉が含まれている。喜ぶという感情は、以下の論理式で表現できる。

$$Joy_i\varphi \equiv Bel_i\varphi \wedge Des_i\varphi$$

すなわち、 $\varphi$  について成立を信じるときかつ、 $\varphi$  は望ましいことだと思っているときに  $\varphi$  に対して喜ぶという感情を生起する。そのため、この文章に対しては「おばあさんは散歩をした」、「おばあさんは散歩をすることが望ましいと思っている」という信念を出力した。図 13 がその結果であり、この信念があることにより「喜ぶ」という感情が正しく生起される。

```
des( grandmother , stroll ) [degOfCert( 0.5 )]  
bel( grandmother , stroll )
```

図 13 感情語に対応する信念の出力

- 感情を表すと判定する単語

joy 「喜び」、「喜ぶ」、「嬉しさ」、「歓喜」、「嬉しい」

distress 「嘆く」、「嘆き」、「苦悩」、「悲嘆」、「悲しむ」、「悲しい」

hope 「望む」、「望み」、「希望」、「期待」

fear 「恐れ」、「恐怖」、「不安」、「心配」、「懸念」、「心配」、「怖」

satisfaction 「実現」、「達成」、「成就」、「充足」、「満足」

relief 「安堵」、「安心」

dissapointment 「落胆」、「失望」、「失意」

resentment 「憤る」、「恨み」、「憤り」

gloating 「ほくそ笑む」、「嘲笑う」

pride 「誇る」、「誇り」

shame 「恥羞」、「恥」、「不名誉」

gratification 「満足」

remorse 「後悔」、「痛恨」、「悔恨」、「自責」

admiration 「称賛」、「感嘆」、「感心」、「憧れ」

reproach 「非難」、「責める」、「叱責」、「叱る」  
gratitude 「感謝」、「報恩」、「謝意」  
anger 「怒」  
love 「好」  
hate 「嫌」、「憎む」

## 4 実験

本章では作成したプログラムが期待通りの信念を抽出するかどうか、また、正しく感情生起を行うことができるかの検証を行う。先行研究 [8] と同様のシナリオ、『はじめてのおつかい』 [20] を用いて、本研究にて作成されたプログラムから生成される信念が、[8] にて生成された信念と一致するかを見ることで検証を行う。シナリオは以下の通りである。

『お母さんからお使いを頼まれ家を出たみいちゃんは、坂で転んでしまい、その拍子に持っていたお金を落とした。落ちたお金を無事見つけたみいちゃんは元気に坂を登って行った。無事にお使いを済ませたみいちゃんをお母さんが迎えにきた。』

### 4.1 先行研究

先行研究 [8] ではこのシナリオを構文解析プログラムで信念抽出可能な形に変えて入力をしている。入力した文章は以下の通りである。

- 『みいちゃんはお母さんにお使いを頼まれた』
- 『みいちゃんは転んでしまった』
- 『みいちゃんはお金を落としてしまった』
- 『みいちゃんはお金を探した』
- 『みいちゃんはお金を見つけた』
- 『みいちゃんはお母さんにお使いを伝えた』

実験の結果は以下に示す。入力の各文のイベントからは以下の信念が抽出された。文章の左側に振られている番号は、後の参照に用いるイベント番号である。

- ① 『みいちゃんはお母さんにお使いを頼まれた』  
→ `bel(haha,des(michan,"お使い"))[degOfCert(0.5)]`
- ② 『みいちゃんは転んでしまった』  
→ `bel(michan,"転ぶ")`  
→ `des(michan,not "転ぶ")[degOfCert(0.5)]`
- ③ 『みいちゃんはお金を落としてしまった』  
→ `bel(michan,"お金を落とす")`  
→ `des(michan,not "お金を落とす")[degOfCert(0.5)]`
- ④ 『みいちゃんはお金を探した』  
→ `bel(michan,"お金探す")`

⑤ 『みいちゃんはお金を見つけた』  
→ des(michan, "お金見つける")[degOfCert(0.5)]

→ bel(michan, "お金見つける")

⑥ 『みいちゃんはお母さんにお使いを伝えた』

→ bel(haha, bel(michan, "お使い"))

先行研究では、これらの結果から以下の感情が生じた。

② → Distress(嘆き) を生起 (度合い 0.5)

③ → Distress(嘆き) を生起 (度合い 0.5)

⑤ → Joy(喜び) を生起 (度合い 0.5)

⑥ → 母の Joy(喜び) を生起 (度合い 0.5)

## 4.2 実験結果

本研究では、先述のシナリオを人手を経ずに入力を行なった。入力する文章は以下の通りである。

- 『お母さんからお使いを頼まれ家を出たみいちゃんは、坂で転んでしまい、その拍子に持っていたお金を落とした。落ちたお金を無事見つけたみいちゃんは元気に坂を登って行った。無事にお使いを済ませたみいちゃんをお母さんが迎えにきた。』

実験の結果は以下に示す。上の文を入力した結果、以下の信念が出力された。

bel(miichan, des(mom, errandask)[degOfCert(0.5)])

bel(miichan, slopefall)

des(miichan, not slopefall)[degOfCert(0.5)]

bel(money, beathold)

bel(miichan, moneydrop)

bel(money, dropdown)

bel(miichan, moneybefound)

bel(miichan, slopeclime)

bel(miichan, becompleted)

bel(mom, miichanwelcome)

これらより、bel(miichan, slopefall)、des(miichan, not slopefall)[degOfCert(0.5)] の2つの信念から転ぶというイベントに対して Distress(嘆き) が生起 (度合い 0.5) した。

今回の実験結果と先行研究 [8] の結果を比較し、イベントごとに出力結果をまとめたものが表 1 である。

先行研究の結果に加え、そのほかにも文章から多くの信念を出力することができた。一方、③, ⑤のイベントについて、先行研究で出力されていた、お金を落とすことが望ましくないことやお金を見つけることが望ましいことは、本研究では出力されない。しかしこれらは、人間として生起する根本的な信念であるが、みいちゃんのように思っているかどうかは入力する文章から必ずしも読み取ることはできない。そのため、出力しなくても問題がない信念であると考えられる。

イベント	先行研究での結果	本研究での結果
①	bel(haha,des(michan,"お使い"))[degOfCert(0.5)]	bel(miichan,des(mom,errandask)[degOfCert(0.5)])
②	bel(michan,"転ぶ") des(michan,not "転ぶ")[degOfCert(0.5)]	bel(miichan,slopefall) des(miichan,not slopefall)[degOfCert(0.5)]
		bel(money,beathold)
③	bel(michan,"お金を落とす") des(michan,not "お金を落とす")[degOfCert(0.5)]	bel(miichan,moneydrop)
		bel(money,dropdown)
④	bel(michan,"お金探す")	
⑤	des(michan,"お金見つける")[degOfCert(0.5)] bel(michan,"お金見つける")	bel(miichan,moneybefound)
		bel(miichan,slopeclime)
⑥	bel(haha,bel(michan,"お使い"))	bel(miichan,becompleted)
		bel(mom,miichanwelcome)

表1 実験結果の比較

## 5 まとめ

本研究では、外部から与えられた文章から、感情生起に必要な信念を出力するプログラムの作成を行った。先行研究では物語から推測される論理式を与えたり、文章を構文解析を行うことができる簡単な形に変換してから与える必要があった。しかし、本研究では、人手を経ずに与えた物語文から正しく信念が出力できるようになった。

今後の課題としては、意味解析を行った上で信念を出力することがあげられる。現在は、否定語の判定を単に字面のみから定めているが、反語などの判定にも意味解析は必要である。また、比較的体裁の整った物語文から事実を出力しているが、将来的には人間同士が用いる口語から信念を出力する際にも重要になると考えられる。ほかにも、現在は bel, des オペレータのみを追加しているが、より多くの感情を生起させるには Prob のオペレータを追加するルールを定める必要があると考えられる。

## 6 謝辞

本研究の遂行及び本論文の執筆にあたり、指導教員の新出尚之准教授には、丁寧なご指導、ご助言を賜りました。心からの感謝の気持ちとお礼を申し上げたく、謝辞にかえさせていただきます。

## 参考文献

- [1] 岡本英生. 奈良女子大学 2021 年度 授業 感情・人格心理学—配布資料 [感情・人格心理学 第 10 回目 (12 月 13 日) 感情の生物学的基礎]. 講義で配布 (非公開).
- [2] A. Ortony, G. L. Clore, and A. Collins. The Cognitive Structure Of Emotions. *Cambridge University*

- Press, 1998.
- [3] Anand S. Rao, Munindar P. Singh, and Michael P. Georgeff. Formal methods in DAI: Logic-Based Representation and Reasoning. *Massachusetts Institute of Technology*, 1999.
  - [4] Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. Programming Multiagent Systems in AgentSpeak using Jason. *John Wiley and Sons*, 2007.
  - [5] Carole Adam, Andreas Herzing, and Dominique Longin. Logical formalization of OCC theory of emotions. *Synthese*, Vol. 168, No. 2, pp. 201–248, 2009.
  - [6] 今井那緒, 浅井沙良, 塚本麻衣, 新出尚之. OCC theory に基づくエージェントの感情表現と時間経過に関する論理モデル. 日本ソフトウェア科学会第 34 回大会 (2017 年度) 講演論文集, 2017.
  - [7] 吉井優佳. 自律エージェントの感情表現の度合いと他者からの情報の信頼性. 2018 年度卒業論文, 2019. 奈良女子大学生生活環境学部情報環境学科生活情報通信科学コース.
  - [8] 塚本麻衣. 論理モデルに基づく感情生起と感情推測を行うエージェント. 2020 年度修士論文, 2021. 奈良女子大学大学院人間文化研究科.
  - [9] 工藤拓, 松本裕治. チャンキングの段階適用による日本語係り受け解析. 情報処理学会論文誌, Vol. 43, No. 6, pp. 1834–1842, 2002.
  - [10] 安岡孝一. UniDic2UD 2.9.7, 2022. <https://pypi.org/project/unidic2ud/>.
  - [11] 安岡孝一. yasuoka の日記: UniDic2UD による CaboCha エミュレータ, 2019. <https://srad.jp/~yasuoka/journal/634405/>.
  - [12] 安岡孝一. 世界の Universal Dependencies と係り受け解析ツール群. 第 3 回 Universal Dependencies 公開研究会, pp. 1–20, 2021.
  - [13] 加藤暢, 高田司郎, 新出尚之. 数理論理学—合理的エージェントの応用に向けて—. コロナ社, 2014.
  - [14] 山下紗苗, 上泰, 奥村紀之. 品詞情報とルールベースによる否定表現有無の判定. 言語処理学会第 25 回年次大会発表論文集, 2019.
  - [15] 菅沼明, 倉田昌典, 牛島和夫. 日本語文章推敲支援ツール『推敲』における否定表現の抽出法. 情報処理学会論文誌, Vol. 31, No. 6, pp. 792–800, 1990.
  - [16] SuHun Han. googletrans 3.0.0, 2020. <https://pypi.org/project/googletrans/>.
  - [17] 石川葉子. 感情に程度の強さを取り入れたエージェントの実現. 2014 年度卒業論文, 2015. 奈良女子大学理学部情報科学科.
  - [18] 池ノ内彰子, 新出尚之. OCC theory に基づくエージェントの感情表現の論理モデルについて. *In Proc. of JAWS2014*, 2014.
  - [19] 中山恵利子. 程度副詞の分類の試み—その程度量基準により—. 阪南論集 人文・自然科学編, Vol. 31, p. 75, 1996.
  - [20] 筒井頼子 (作), 林明子 (絵). はじめてのおつかい (こどものとも傑作集). 福音館書店, 1977.