

複数対話型エージェントの役割分担による ユーモア生成システムの基礎検討

中原 涼太¹ 長岡 大二¹ 呉 健朗² 大西 俊輝¹ 柴田 万里那¹ 宮田 章裕^{1,a)}

概要：

発展を続ける情報分野を支える技術の1つである対話型エージェントは、今後もより多くの場面で活躍が期待されている。一方で、エージェントとの無機質な対話に親しみを感じないユーザには、このような対話型エージェントは受け入れてもらえない可能性が懸念される。この問題を解決するために我々は、ユーザの発言の一部をわざと間違えて聞き返すような、ユーモア発話を行う対話型エージェントを提案してきた。しかし、これらは、ユーザとエージェントが1対1でコミュニケーションをすることを前提としたシステムであった。そのため、ユーザが会話を軌道修正するために、エージェントのユーモア発話を指摘（例：ツッコミ、否定）する必要があるが、これがユーザの負担となっていた。そこで本稿では、ユーモア発話を行うエージェントに加えて、ユーモア発話を指摘し、会話推進を行うエージェントを導入して役割分担させることを提案する。これは、ユーザの発話に対し、一方のエージェントがユーモア発話を行い、そのユーモア発話をもう一方のエージェントが指摘することで、ユーザが負担を感じることなく、コミュニケーションを継続させられるものである。プロトタイプシステムによる検証実験を行った結果、役割分担された複数のエージェントを用いることで、ユーザとエージェント間の対話におけるユーザの負担が軽減されることを確認できた。

A Study of a Joking System Using Role-Separated Conversational Agents

RYOTA NAKAHARA¹ DALJI NAGAOKA¹ KENRO GO² TOSHIKI ONISHI¹ MARINA SHIBATA¹
AKIHIRO MIYATA^{1,a)}

1. はじめに

現代社会において対話型エージェントは、看護やショッピングなど様々な場面で見かけられるようになった。将来的には家庭や介護などの人間との良好なコミュニケーションを求められる場面での活躍も期待されている。しかしその一方で、エージェントとの無機質な対話に親しみを持っていないユーザも一定数いると考えられ、そのようなユーザにはこのような対話型エージェントは受け入れられないとい

う問題がある。この問題を解決するためには、エージェントとユーザの間に親和的な関係を築くことが必要であると考えられる。[1]では親和的な関係の形成に笑いが欠かせないとされており、人とエージェント間の関係でも同様の報告が多数なされている[2][3][4]。これに基づき、我々はユーモアのある対話を行うエージェントを提案してきた[5][6][7]。しかしこれらの従来手法では、ユーザが会話を軌道修正するために、エージェントのユーモア発話を指摘する必要があるが、これがユーザの負担となっていた。そこで本稿では、ユーモア発話を行うエージェントに加えて、ユーモア発話を指摘し、会話推進を行うエージェントを導入して役割分担させることを提案する。これは、ユーザの発話に対し、一方のエージェントがユーモア発話を行い、

¹ 日本大学文理学部
College of Humanities and Sciences, Nihon University

² 日本大学大学院総合基礎科学研究科
Graduate School of Integrated Basic Sciences, Nihon University

a) miyata.akihiro@acm.org

そのユーモア発話をもう一方のエージェントが指摘することで、ユーザが負担を感じることなくコミュニケーションを継続させられるものである。

本稿の貢献は次のとおりである。

- ユーザとエージェントによる対話シーンにおいて、ユーザが負担を感じずに対話を行えるシステムを提案したこと。
- 上記提案のプロトタイプシステムを構築し、ユーザ実験を行って有効性を検証したこと。

2. エージェントによるユーモア生成の研究事例

2.1 ユーザとエージェントがコミュニケーションを行わない研究事例

単体のエージェントが、ユーザとコミュニケーションを行わずユーモアを生成する事例として、[8], [9] が挙げられる。[8] では、システムに入力された文章内の語（被置換語）を他の語（置換語）に置き換えた置換型駄洒落を出力するシステムを提案している。置換語の選定は、被置換語と置換語の概念ベクトルを SVM で学習することで実現している。[9] では、複合語と複合語を構成する各語の類音語を見出しとする辞書を用意し、複合語の前半部、もしくは後半部を類音語に置換することで駄洒落を生成している。

複数のエージェントが、ユーザとコミュニケーションを行わずユーモアを生成する事例として、[10], [11], [12], [13], [14] が挙げられる。[10], [11] は、漫才形式の対話文の自動生成システムである。入力された文章から単語を選び、それを音の近い単語や、その単語が修飾している句につながる別の単語に置換し、ボケを生成している。その後、ボケに対する否定や、ボケの内容を例えるツッコミ文を生成することで、漫才形式を実現している。[12], [13] では、Web ニュース記事本文の感情を表す単語を考慮し、おかしみの構造図に基づいた形式的なボケによる漫才台本自動生成手法を提案している。[14] では、エージェントがユーザに笑い感情を誘起させる手段として大喜利が用いられている。お笑い構成作家と筆者らによって得られたデータベースをもとに、機械学習のアプローチに基づいてユーモアの生成を行っている。

2.2 ユーザとエージェントがコミュニケーションを行う研究事例

ユーザとエージェントがコミュニケーションを行いユーモアを生成する事例として、[15], [16], [17], [18], [5], [6], [7] が挙げられる。[15] は、ユーザの対話継続欲求を向上させるために、対話システムにおける単語間類似度を利用したユーモア発話自動生成手法を提案している。[16], [17], [18] では、ユーザからの特定の入力に対して、あらかじめ設定されたユーモアを含む応答文を返していると思われ

る。我々も、ユーザの発言内容をわざと間違えて聞き返す方式でユーモアを生成する手法を提案してきた [5][6][7]。

3. 研究課題

人同士の親和的な関係の形成には、笑いが欠かせないとされており [1], 人とエージェント間関係においても、同様の報告が多数なされている [2][3][4]。このような背景から、笑いを通してユーザと良好な関係を築こうとしている事例はいくつか存在するが、様々な問題が見られる。

第1に、[8], [9], [10], [11], [12], [13], [14] は、ボケや漫才、大喜利をユーザに見せる披露形式でユーモア生成を行っているため、対話シーンにそのまま適用できない。第2に、[15] は、ユーザとエージェント間の対話中でのユーモア表現を試みているが、エージェントが突飛な発言を行うことがあり、ユーザがエージェントの発話をユーモアとして受容しにくい場合がある。第3に、[16], [17], [18] は、ユーモア表現を行う場合、あらかじめユーザの入力文と、その応答が決められていると思われる。このため、ユーモアがある返答を行えるのはユーザが特定の言い回しをする場合のみであり、ユーモアを提示できる機会が制限される。

これらの問題を解消するために、我々は、ユーザの入力文中のある単語を、意味が遠く音が近い単語に聞き間違えてボケることでユーモアを生成する対話型エージェントの研究を行ってきた [5][6][7]。しかし、これらは、ユーザとエージェントが1対1でコミュニケーションをすることを前提としたシステムであった。このため、エージェントがユーモア発話を行った場合、ユーザがユーモア発話に対して指摘し会話を軌道修正する必要があり、ユーザにとって負担がかかるものとなっていた。例えば、ユーザの質問に対してエージェントがユーモア発話を行った時、ユーザが質問の答えを得るには、会話を軌道修正する言葉を考えたうえで、ユーモア発話を指摘する手間がかかる。上述の問題を踏まえ本研究では、エージェントがユーザにユーモアを提示するシーンにおいて、下記の3つを研究課題とする。

課題1: ユーザとの対話中に、任意のタイミングでユーモアを提示する。

課題2: ユーザが、ユーモアに対する指摘や会話推進などを行う必要をなくす。

課題3: ユーザが、エージェントに対して親しみを感じられるようにする。

4. 提案方式

3章の研究課題を達成するにあたって、日本で広く定着しているコメディスタイルである漫才に着目する。漫才とは、ユーモア発話を行う担当（ボケ役）と、そのユーモア発話を指摘しつつ会話を推進する担当（ツッコミ役）に役割分担をし、聞き手にユーモア提示を行うというスタイルである。この役割分担により、ボケ役がユーモア発話を行

い話の腰を折っても、ツッコミ役がボケを指摘しつつ、会話の流れの軌道修正を行うことが可能である。例えば、カジュアルな情報番組において、ボケ役がユーモア発話を行い会話が脇道に逸れてしまった場合でも、ツッコミ役が指摘を行い会話の流れを軌道修正することで、番組が順調に進行していく様子がよく見られる。

以上より本稿では、このボケとツッコミのコメディスタイルをもとに、複数対話型エージェントの役割分担によるユーモア会話システムを提案する。このシステムは、ユーザの語りかけに対してユーモア発話を行うエージェント（以降、ユーモアエージェント）と、ユーモアを指摘して会話を推進するエージェント（以降、会話推進エージェント）からなる。各エージェントとの対話形式については、複数対話型エージェント環境において、各エージェントとのインタラクションが複雑化するとユーザは混乱を感じやすいことが分かっている [19]。よって本システムでは、ユーザとエージェントのインタラクションが複雑化しないよう、完全に自由な形式で対話を行うのではなく、下記テンプレートに沿ってユーザと各エージェントが対話を行うものとする。

Step1 :ユーザがエージェントたちに語りかける。

Step2 :ユーモアエージェントがユーモア発話を行う。

Step3 :会話推進エージェントがユーモアエージェントに指摘を行う。

Step4 :会話推進エージェントがユーザとの会話を継続する。

ここで、ユーモア発話に対する指摘の内容が、ユーモア発話内容の繰り返しのような単調なものであると、台本をただ読み上げているような機械的な印象をユーザに与えてしまう可能性がある。これに伴い、ユーザはエージェント間のコミュニケーションに人間味を感じられず、両エージェントに対して親しみを感じにくくなると思われる。そこで、ユーモア発話に対する指摘が単調にならないように、エージェントが言葉の意味を理解しているかのような指摘を行う。具体的には、5.3.1 項の方法で言い換えを行ったり、5.3.2 項の方法で関連する動詞を提示したりする。これにより、会話推進エージェントが、ユーモアエージェントの発話内容を咀嚼した上で指摘をしているような印象をユーザに与えられ、人間味のあるエージェント間のコミュニケーションを生み出すことができる。

この方式により、3章の研究課題を達成できると考える。課題1については、ユーザの入力にもとづいて動的にユーモアを生成するため、任意のタイミングでユーモアを提示できる。課題2については、会話の軌道修正をエージェントが代行するため解決できる。課題3については、人間味のあるエージェント間のコミュニケーションを生み出すことで、ユーザはエージェントに対して親しみを感じられる。

5. 実装

5.1 事前準備

Wikipedia 全文記事を形態素解析し、不要品詞などを除去して分かち書きしたものをコーパスとする。ここでの不要品詞とは IPA 品詞体系において、記号、助詞、助動詞、接続詞、副詞、連体詞、非自立語、代名詞、接尾、数、サ変・スルと分類されるものを指す。

5.2 ボケ生成

5.2.1 各 Score の計算

本項ではユーザがシステムに単語を入力した際の、出力候補単語それぞれの各 Score (s_s , s_e , s_f) の算出方法 [5][6][7] について述べる。

5.2.1.1 s_s : Semantic Score

ユーザの入力単語と出力候補単語がどれくらい意味が離れているかを表す指標として、言語モデルを用いて概念距離を測ることで s_s を算出した。入力単語と出力候補単語それぞれとの概念距離を測り、その距離が遠いほど高い s_s を与える。すなわち、 s_s は下記ようになる。

$$s_s = d_s \quad (1)$$

d_s は入力単語と出力候補単語の概念距離である。

5.2.1.2 s_e : Edit distance Score

ユーザの入力単語と出力候補単語の音がどれくらい近いかを表す指標として、編集距離^{*1}を測ることで s_e を算出した。入力単語と出力候補単語それぞれとの編集距離を測り、その距離が近いほど高い s_e を与える。すなわち、 s_e は下記ようになる。

$$s_e = \frac{1}{1 + d_e} \quad (2)$$

d_e は入力単語と出力候補単語の編集距離である。

5.2.1.3 s_f : Frequency Score

出力候補単語がどれくらいユーザから理解されやすいかを表す指標として、コーパス内での各単語の出現回数を数えることで s_f を算出した。各出力候補単語の出現回数が多いほど高い s_f を与える。すなわち、 s_f は下記ようになる。

$$s_f = \log f \quad (3)$$

f は置換候補単語のコーパス内での出現回数である。この

^{*1} 編集距離とは 2 つの文字列がどの程度異なっているかを示す距離であり、1 文字の挿入・削除・置換によって一方の文字列をもう一方の文字列に変形するのに必要な手順の最小回数と定義される。

とき単語の出現回数は、べき分布に従うため、ごく一部の単語の出現回数が極端に大きい。これらの単語が最終的な総合 Score に与える影響が大きくなりすぎないように、出現回数の対数をとったものとする。

5.2.2 出力単語の決定

人名や地名などは、ごく一部の有名なもの以外はユーザからの認知度が低い可能性があると考え、MeCab で人名・地域と判定される単語は出力候補単語から除く。極端に出現回数が少ない単語に関しても、ユーザからの認知度が低くなる可能性があると考え、出現回数が 50 回を下回る単語は出力候補単語から除く。また、エージェントの返答がオウム返しになってしまうとユーモアとして受容されない可能性が考えられる。そのため、入力単語を単にひらがな、あるいはカタカナ表記にしたものは出力候補単語から除く。最後に、概念距離が近い動詞が存在する単語でボケることで、後述のツッコミが可能なボケを生成する。これは、お笑い構成作家が漫才台本を作る際に用いる手法を参考にしている [20]。以上より、ユーザからの入力単語に対する出力候補単語となる条件は下記のようになる。

- 人名や地名に分類される固有名詞ではない。
- コーパス内での出現回数が 50 回以上である。
- 置換元単語を、ひらがな・カタカナ表記にしたものではない。
- 言語モデルを用いて、概念距離が一定以上の値の単語群を取得した時、その単語群内に品詞が動詞である単語が存在する。

ユーザの入力単語に対し、上記の条件全てを満たすコーパス内の出力候補単語の 3 つの Score (s_s , s_e , s_f) を算出し、それぞれ正規化処理を行ってから合算したものを、出力候補単語それぞれの s_t (Total Score) とする。すなわち、 s_t は下記のようになる。

$$s_t = s_s + s_e + s_f$$

全ての出力候補単語の s_t を算出し、 s_t が最も大きくなる単語を出力単語とする。

5.2.3 ボケ出力

実際の対話シーンに近づけるために、え?“ボケ単語”? と出力を行う。以下に出力例を示す。

- 入力：ミサイル
ボケ：え?味噌汁?
- 入力：バレンタイン
ボケ：え?派遣社員?

5.3 ツッコミ生成

4 章で述べた、ユーモアエージェントによるユーモア発話の内容を咀嚼しているような発話とは、単語の意味や日本語における主語・述語の関係を理解しているものだと考えた。この考えに基づき、ユーモア発話内容の同義語での

指摘(以下、言い換えツッコミ)と、ユーモア発話内容に関連する動詞を用いた指摘(以下、動詞ツッコミ)を実装する。以下に、具体的な生成方法を示す。

5.3.1 言い換えツッコミの生成

言い換えツッコミとは、入力単語を、その単語と意味の類似する別の単語へと言い換えて指摘を行うものである。意味が類似する単語を取得するために、入力単語の同義語を下記のような手順で取得する。

同義語の取得には、EDR 電子化辞書 [21] 内の日本語単語辞書を利用する。日本語単語辞書内に登録されている単語は、概念識別子によって結び付けられており、これを用いて入力単語の同義語群を取得することができる。取得した同義語群内で、下記の言い換え単語条件を満たす単語のうち、EDR コーパス*2内で定義付けられているその単語の頻度が最も高い単語を言い換えツッコミ単語とする。

言い換え単語条件:

- 入力単語と読み方が異なる。
- Wikipedia コーパス内での出現回数が 50 回以上。
- 入力単語と同じ品詞である。

同義語群内に、言い換え単語条件を満たす単語が存在しない場合、入力単語の上位概念語への言い換えを行う。上位概念語の取得には、EDR 電子化辞書内の概念辞書を利用する。概念辞書内では、登録されている各単語の上位下位の単語がそれぞれ結び付けられており、これを用いて上下概念語群を取得することができる。取得した上位概念語群内で、言い換え単語条件を満たす単語のうち、EDR コーパス内で定義付けられているその単語の頻度が最も高い単語を言い換えツッコミ単語とする。

5.3.2 動詞ツッコミの生成

動詞ツッコミとは、入力単語に関連する動詞を用いて指摘を行うものである。この動詞を取得するには、まず、5.1 節で作成した言語モデルを用いて、入力単語との概念距離が一定以上の値の単語群を取得する。次に、概念距離が近い順に MeCab を利用して品詞を取得する。その中で最初に品詞が動詞と判定された単語を未然形に変え、動詞ツッコミ単語とする。

5.3.3 ツッコミの出力

実際の対話を行うシーンに近づけるために、“言い換えツッコミ単語”か!、なんで“動詞ツッコミ単語”なきやいけないんだよ!と出力を行う。以下に出力例を示す。

- 入力：ミサイル
ツッコミ：誘導弾か!なんで撃ち落とさなきやいけないんだよ!
- 入力：派遣社員
ツッコミ：社員か!なんで働かなきやいけないん

*2 EDR コーパスとは、新聞や雑誌、辞典から無作為に抽出した約 20 万文の日本語コーパスと約 12 万文の英語コーパスをもとに構成される言語データである。

だよ！

6. 実験

6.1 実験目的

本研究は、ユーモア発話に対してユーザが指摘をしなければいけない負担を軽減するために、ユーモア発話を行うエージェントに加えて、ユーモア発話を指摘し、会話推進を行うエージェントを導入して役割分担させることを提案するものである。2体のエージェントが会話を推進する方式の有効性を確認するために、まず、ユーモア発話に対して、エージェントが指摘を行うことの効果を検証する必要がある。その次に、役割分担された2体のエージェントが会話を推進する方式と、1体のエージェントが会話を推進する方式を比較し、ユーザが感じる負担や苛立ちが軽減されるかどうかを検証する。最後に、ユーモア発話を咀嚼した指摘を行う方式と、ユーモア発話に対して単調な指摘を行う方式を比較し、人間味のあるエージェント間のコミュニケーションを生み出すことで、ユーザがより親しみを感じやすくなるかどうかを検証する。よって実験では以下の Research Question (RQ) を検証する。

RQ1：ユーモア発話に対して、エージェントが指摘を行うことで、ユーザが感じる苛立ちや負担が軽減されるか。

RQ2：ユーモア発話を行うエージェントに加えて、ユーモア発話を指摘し、会話推進を行うエージェントを導入して役割分担させることで、ユーザが感じる苛立ちや負担が軽減されるか。

RQ3：会話推進エージェントが、ユーモア発話を咀嚼した指摘を行うことで、ユーザがエージェントに対して親しみをより感じやすくなるか。

6.2 実験条件

本実験の被験者は20代の学生8名(男6, 女2)である。被験者が単語を入力する際には、実際のエージェントとの対話シーンに近づけるために、「“単語”について教えて?」と入力してもらう。実験には、下記の4つの方式を用いる。

- ツッコミ不在条件：1体のエージェントが、ユーモア発話のみを行う。

出力例

- ユーザ>バレンタインについて教えて?
- ボケ役>え?派遣社員?

- 自己ツッコミ条件：1体のエージェントが、ユーモア提示と単純な指摘や会話推進の両方を行う。最後に、「単語」に対する Wikipedia の概要文を出力する。

出力例

- ユーザ>バレンタインについて教えて?
- ボケ役>え?派遣社員?
- ボケ役>あ、派遣社員じゃなくてバレンタインか。バレンタインとは…(Wikipedia 概要文)

表 1 実験の質問一覧

Table1: A Question List of Experiment

Q1	会話の流れは自然だと感じましたか? 5: とても感じた ~ 1: 全く感じなかった
Q2	エージェントと苛立ちなく会話することができたと感じましたか? 5: とても感じた ~ 1: 全く感じなかった
Q3	エージェントの発言にユーモアを感じましたか? 5: とても感じた ~ 1: 全く感じなかった
Q4	ツッコミ役エージェントが、ボケ役エージェントの発言内容を理解しているように感じましたか? 5: とても感じた ~ 1: 全く感じなかった
Q5	エージェント間の会話に人間味を感じましたか? 5: とても感じた ~ 1: 全く感じなかった
Q6	エージェント(たち)に信頼を感じましたか? 5: とても感じた ~ 1: 全く感じなかった
Q7	エージェントたちと負担なく会話することができたと感じましたか? 5: とても感じた ~ 1: 全く感じなかった
Q8	エージェントたちに対して親しみを感じましたか? 5: とても感じた ~ 1: 全く感じなかった

- 単純ツッコミ条件：1体のエージェントがユーモア提示を行い、もう1体のエージェントが単純な指摘・会話推進を行う。最後に、「単語」に対する Wikipedia の概要文を出力する。

出力例

- ユーザ>バレンタインについて教えて?
- ボケ役>え?派遣社員?
- ツッコミ役>派遣社員じゃなくてバレンタインだって!バレンタインとは…(Wikipedia 概要文)

- 咀嚼ツッコミ条件：1体のエージェントユーモア提示を行い、もう1体のエージェントがユーモア発話を咀嚼した指摘・会話推進を行う。最後に、「単語」に対する Wikipedia の概要文を出力する。

出力例

- ユーザ>バレンタインについて教えて?
- ボケ役>え?派遣社員?
- ツッコミ役>社員か!なんで働かなきゃいけないんだよ!バレンタインだって!バレンタインとは…(Wikipedia 概要文)

6.3 実験手順

被験者には6.1節に示した4つの方式をそれぞれ使用してもらい、アンケートに5段階のリッカート尺度で回答してもらい。被験者への質問と選択肢を表1に示す。各方式使用時には、あらかじめ実験者が用意した5個の単語を被験者自身が入力し、出力を確認してもらう。この5単語は、ボケとツッコミが成立することを実験者が確認した50単語の中から、ランダムに選ばれたものである。最後に、被験者には出力に対しての返答を入力してもらう。入力についてはキーボードを用い、出力についてはPC画面にテキストを表示する。

被験者には、1回の入出力が終わるごとに、ツッコミ不

在条件と自己ツッコミ条件時ではQ1~3, 単純ツッコミ条件時にはQ1~5, 咀嚼ツッコミ条件時には, Q3~5に回答してもらう。上記の手順を5回終わるたびに, 全方式においてQ6~8に回答してもらう。このとき, 順序効果を相殺するために, 被験者ごとに使用するパターンの順番をランダムに入れ替える。

6.4 結果

Q1の回答結果を図1に示す。“会話の流れは自然だと感じたか”という質問に対し, “とても感じた”または“感じた”と回答した被験者は, ツッコミ不在条件では40%, 自己ツッコミ条件では30%, 単純ツッコミ条件では63%であった。Q2の回答結果を図2に示す。“エージェントと苛立ちなく会話することができたか”という質問に対し, “とても感じた”または“感じた”と回答した被験者は, ツッコミ不在条件では40%, 自己ツッコミ条件では8%, 単純ツッコミ条件では50%であった。Q3の回答結果を図3に示す。“エージェントの発言にユーモアを感じたか”という質問に対し, “とても感じた”または“感じた”と回答した被験者は, ツッコミ不在条件では31%, 自己ツッコミ条件では38%, 単純ツッコミ条件では38%, 咀嚼ツッコミ条件では78%であった。Q4の回答結果を図4に示す。“ツッコミ役エージェントがボケ役エージェントの発言内容を理解しているように感じたか”という質問に対し, “とても感じた”または“感じた”と回答した被験者は, 単純ツッコミ条件は66%, 咀嚼ツッコミ条件では65%であった。Q5の回答結果を図5に示す。“エージェントの会話に人間味を感じたか”という質問に対し, “とても感じた”または“感じた”と回答した被験者は, 単純ツッコミ条件では46%, 咀嚼ツッコミ条件では55%であった。Q6の回答結果を図6に示す。“エージェント(たち)に信頼を感じたか”という質問に対し, “とても感じた”または“感じた”と回答した被験者は, ツッコミ不在条件では25%, 自己ツッコミ条件では25%, 単純ツッコミ条件のユーモアエージェントに対しては51%, 咀嚼ツッコミ条件のユーモアエージェントに対しては63%, 会話推進エージェントに対しては76%であった。Q7の回答結果を図7に示す。“エージェントと負担なく会話できたか”という質問に対し, “とても感じた”または“感じた”と回答した被験者は, ツッコミ不在条件では38%, 自己ツッコミ条件では13%, 単純ツッコミ条件は50%, 咀嚼ツッコミ条件では63%であった。Q8の回答結果を図8に示す。“エージェントに親しみを感じたか”という質問に対し, “とても感じた”または“感じた”と回答した被験者は, ツッコミ不在条件では38%, 自己ツッコミ条件では13%, 単純ツッコミ条件は38%, 咀嚼ツッコミ条件では76%であった。

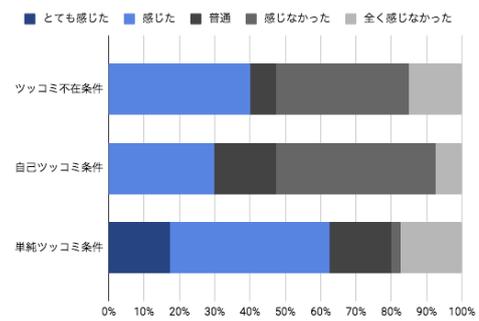


図1 Q1の回答 (N=8, 1人あたり5回答)

Fig.1: Answer to Q1

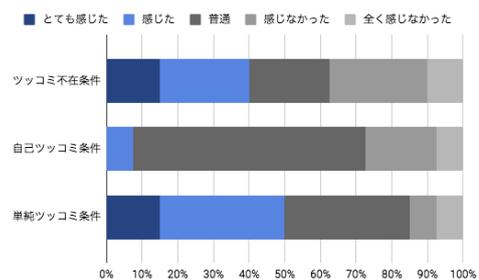


図2 Q2の回答 (N=8, 1人あたり5回答)

Fig.2: Answer to Q2

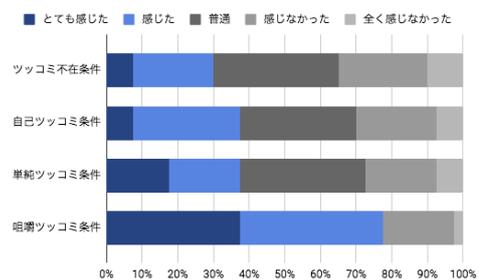


図3 Q3の回答 (N=8, 1人あたり5回答)

Fig.3: Answer to Q3

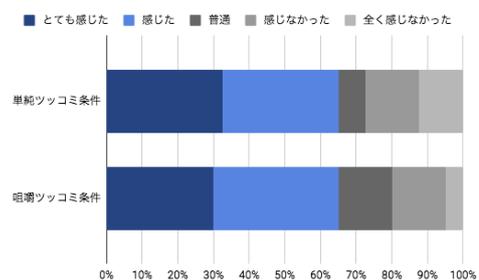


図4 Q4の回答 (N=8, 1人あたり5回答)

Fig.4: Answer to Q4

6.5 考察

6.5.1 RQ1 についての考察

RQ1では, ツッコミ不在条件と自己ツッコミ条件を比較し, エージェントがユーモア発言への指摘を代行することで, ユーザを感じる苛立ちや負担が軽減されるかについて

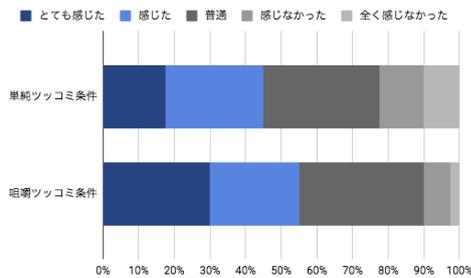


図 5 Q5 の回答 (N=8, 1 人あたり 5 回答)
Fig.5: Answer to Q5

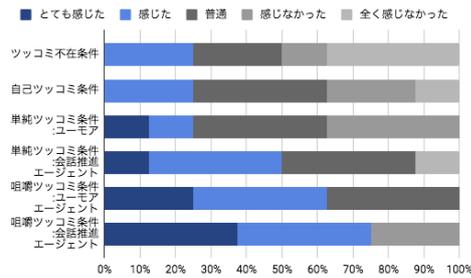


図 6 Q6 の回答 (N=8)
Fig.6: Answer to Q6

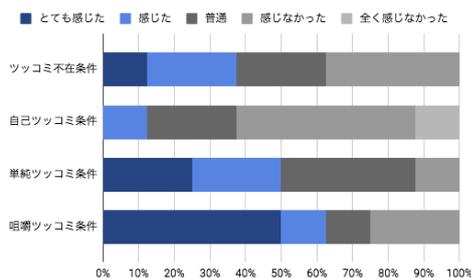


図 7 Q7 の回答 (N=8)
Fig.7: Answer to Q7

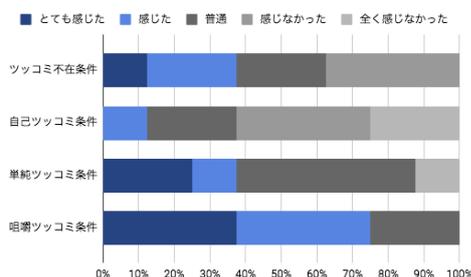


図 8 Q8 の回答 (N=8)
Fig.8: Answer to Q8

検証する。つまり、Q1~3, Q6~8 への回答を確認する。Q1~3, Q6~8 の回答に対して、2つの方式間で Wilcoxon の符号順位検定を行ったが、有意差は確認できなかった。この結果から、1体のエージェントがユーモア発話に加えて指摘や会話推進を行うことは、指摘を行わない場合と同程度ユーザに苛立ちや負担を与えてしまうと考えられる。

これは、エージェントが行ったユーモア発話を、そのエージェント自らが指摘すること、会話の流れの違和感を感じたユーザが多く存在したためだと考えられる。また、両方式の全質問において、“とても感じた”または“感じた”と回答した被験者数は半分以下であった。ここから、両方式ともに、ユーモア対話中に用いる方式として適切でないと考えられる。

6.5.2 RQ2 についての考察

RQ2 では、自己ツッコミ条件と単純ツッコミ条件を比較し、役割分担された2体のエージェントを用いることで、ユーザを感じる苛立ちや負担へが軽減されるかについて検証する。つまり、Q1~3, Q6~8 への回答を確認する。

Q1, Q2 への回答に対し、2つの方式間で Wilcoxon の符号順位検定を行ったが、どちらも有意差は確認できなかった。しかし、両質問ともに、“とても感じた”、“感じた”と回答した被験者が、単純ツッコミ条件では、自己ツッコミ条件と比べて多くの割合を占めていた。ここから、役割分担された2体のエージェントを用いて対話を行うことで、自然な会話の生成や、ユーザの苛立ちを軽減できる傾向が見られる。

Q3 への回答に対し、2つの方式間で Wilcoxon の符号順位検定を行ったが、有意差は確認できなかった。また、“とても感じた”、“感じた”と回答した被験者の割合についても差が見られなかった。ここから、役割分担された2体のエージェントを用いることは、ユーザに与えるユーモアについては影響がないと考えられる。

Q6 への回答に対し、自己ツッコミ条件と、単純ツッコミ条件下のエージェント間で Wilcoxon の符号順位検定を行ったが、有意差は確認できなかった。しかし、“とても感じた”、“感じた”と回答した被験者が、単純ツッコミ条件の会話推進エージェントに対しては51%おり、自己ツッコミ条件への回答の約2倍であった。ここから、役割分担された2体のエージェントを用いることで、ユーザからの信頼を得られる傾向が見られる。これは、会話推進エージェントが、被験者の入力に対して正しい返答をするためであると考えられる。

Q7 への回答に対し、2つの方式間で Wilcoxon の符号順位検定を行った結果、5%水準で有意差が確認できた。ここから、役割分担された2体のエージェントを用いることで、ユーザの負担を軽減できることが示唆される。これは、6.5.2 項の Q1, Q2 への考察で示した通り、役割分担された2体のエージェントを用いることで、少なからず自然な会話の生成や、ユーザの苛立ちを軽減できたためだと考えられる。

Q8 への回答に対し、2つの方式間で Wilcoxon の符号順位検定を行った結果、5%水準で有意差が確認できた。ここから、役割分担された2体のエージェントを用いることで、ユーザに、より親しみを与えられることが示唆される。

6.5.3 RQ3 についての考察

RQ3 では、単純ツッコミ条件と咀嚼ツッコミ条件を比較し、会話推進エージェントがユーモア発話を咀嚼した指摘を行うことで、ユーザがより親しみを感じられるかを検証する。つまり、Q3~5, Q6~8 への回答を確認する。

Q3 への回答に対し、2つの方式間で Wilcoxon の符号順位検定を行ったが、有意差は確認できなかった。しかし、“とても感じた”、“感じた”と回答した被験者が、咀嚼ツッコミ条件の会話推進エージェントに対しては 78 % おり、単純ツッコミ条件への回答の約 2 倍であった。ここから、会話推進エージェントがユーモア発話を咀嚼した指摘を行うことで、ユーザにユーモアを与えやすい傾向が見られる。

Q4, Q5 への回答に対し、2つの方式間で Wilcoxon の符号順位検定を行ったが、どちらも有意差は確認できなかった。ここから、会話推進エージェントがユーモア発話を咀嚼した指摘を行うことは、ユーザがエージェント間の会話に対して感じる人間味について影響がないと考えられる。とはいえ、被験者からの自由記述に、“ツッコミ役が言葉の意味を理解してツッコんでいて親しみが持てた”といった意見があったため、この傾向が普遍的なものかどうか今度検証を行っていく。

Q6, Q7 への回答に対し、2つの方式間で Wilcoxon の符号順位検定を行ったが、どちらも有意差は確認できなかった。ここから、会話推進エージェントがユーモア発話を咀嚼した指摘を行うことは、ユーザがエージェントに対して感じる信頼や負担について影響がないと考えられる。

Q8 への回答に対し、2つの方式間で Wilcoxon の符号順位検定を行ったが、有意差は確認できなかった。しかし、“とても感じた”、“感じた”と回答した被験者が、咀嚼ツッコミ条件の会話推進エージェントに対しては 76 % おり、単純ツッコミ条件への回答の 2 倍であった。ここから、会話推進エージェントがユーモア発話を咀嚼した指摘を行うことで、ユーザに親しみを与えやすい傾向が見られる。これは、6.5.3 項の Q3 で示した通り、ユーザに、よりユーモアを与えられたためだと考えられる。

7. おわりに

本稿は、[5], [6], [7] のシステムに見られる、ユーザがユーモア発話に対して指摘をしなければいけない負担の解消を狙ったものである。その手段として、ユーモア発話を行うエージェントに加え、ユーモア発話を指摘し、会話推進を行うエージェントを導入して役割分担させることを提案した。プロトタイプシステムによる実験を行った結果、役割分担された 2 体のエージェントを用いることで、ユーザが感じる負担の減少や、エージェントに対する親しみの向上が見られた。今後の課題として、ユーザがよりエージェントに対して親しみを感じられるように、指摘の内容について実装の改善を行っていききたい。

参考文献

- [1] 井上: 「笑い学」研究について, 笑い学研究, No.9, pp.3-15 (2002).
- [2] Peter: Does it Matter if a Computer Jokes?. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp.77-86, 2011.
- [3] Luger et al.: Like Having a Really Bad PA”: The Gulf Between User Expectation and Experience of Conversational Agents. Proc. the 2016 CHI Conference on Human Factors in Computing Systems (2016).
- [4] Liao et al.: All Work and no Play? Conversations with a Question-and-Answer Chatbot in the Wild. Proc. the 2018 CHI Conference on Human Factors in Computing Systems (2018).
- [5] 鈴木他: ボケて返す対話型エージェントの基礎検討, 情報処理学会研究報告グループウェアとネットワークサービス (GN), Vol.2017-GN-102, No.3, pp.1-6 (2017).
- [6] 呉他: ボケて返す対話型エージェントの実装, 日本バーチャルリアリティ学会 VR 学研報, Vol.22, No.CS-4, pp.15-18 (2017).
- [7] 呉他: 対話型エージェントにおけるボケる返答機能の実装, インタラクシオン 2018 インタラクティブセッション (2018).
- [8] 中谷他: ロボットの日常会話におけるユーモア生成の試み, 人工知能学会 2009 年全国大会論文集, 1J1-0s2-5 (2009).
- [9] キム他: 日本語駄洒落なぞなぞ生成システム”BOKE”, 人工知能学会誌, Vol.13, No.6, pp.920-927 (1998).
- [10] 吉田他: 漫才形式の対話文自動生成システム, 日本感性工学会論文誌, Vol.11, No.2, pp.265-272 (2012).
- [11] 竹越他: ロボット漫才自動生成システム - 動作が漫才に与える影響の考察 -, 日本感性工学会論文誌, vol.15, No.1, pp.47-54(2016).
- [12] 真下他: 文の感情を考慮した漫才ロボット台本自動生成手法の提案, DEIM Forum2015 F4-4 (2015).
- [13] 青木他: 理解しやすい対話をういた漫才台本の自動生成, DEIM Forum2018 C1-5 (2018).
- [14] 伊勢崎他: お題に対してユーモアを生起する回答文選択の検討, マルチメディア, 分散協調とモバイルシンポジウム 2018 論文集, pp.643-648, 2018.
- [15] 藤倉他: ユーモア発話の自動生成における単語間類似度導入によるユーモア受容性の向上, HAI シンポジウム 2014 (2014).
- [16] Apple Inc.: Siri, <https://www.apple.com/ios/siri/> (Last visited on 2018/08/31).
- [17] Google Inc.: Google Assistant, <https://assistant.google.com/> (Last visited on 2018/08/31).
- [18] SoftBank.: Papper, <https://www.softbank.jp/robot/> (Last visited on 2018/08/31).
- [19] Chaves et al.: Single or Multiple Conversational Agents? An Interactional Coherence Comparison. Proc. the 2018 CHI Conference on Human Factors in Computing Systems (2018).
- [20] 元祖爆笑王, “漫才入門ウケる笑いの作り方, ぜんぶ教えます.”, Rittor-Music, 2008.
- [21] [情報通信研究機構 02] 独立行政法 情報通信研究機構: EDR 電子化辞書, (2002).
- [22] MeCab: Yet Another Part-of-Speech and Morphological Analyzer, <http://taku910.github.io/mecab/> (Last visited on 2018/08/31).
- [23] Tomaset al.: Efficient Estimation of Word Representations in Vector Space, In Proceedings of Workshop at ICLR (2013).