

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検討

平成30年度 卒業論文

日本大学 文理学部 情報科学科 宮田研究室

中原 涼太

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先行研究として、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、エージェントがユーモア発話を行うのみのシステムであった。そのため、エージェントがユーモア発話を行い会話の軌道が逸れた際、ユーザはエージェントのユーモア発話を指摘（例：ツッコミ、否定）し会話を軌道修正する必要があり、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、ユーザの代わりに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できるものである。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。

本稿の貢献は次の通りである。

- 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ようにす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。

目次

第1章	序論	1
1.1	研究の背景	2
1.2	研究の目的	2
1.3	本論文の構成	2
第2章	エージェントに関する研究事例	4
2.1	対話型エージェントに関する研究事例	5
2.2	ユーモア表現を行うエージェントに関する研究事例	6
2.2.1	一方的にユーモア表現を行うエージェント	6
2.2.2	対話によってユーモア表現を行うエージェント	7
第3章	研究課題	9
3.1	問題の定義	10
3.2	研究課題の設定	11
第4章	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提案	12
4.1	アプローチ	13
4.2	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提案	13
第5章	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実装	15
5.1	システムの全体像	16
5.2	事前準備	16
5.3	ボケ生成部の実装	18
5.3.1	各 Score の計算	18
5.3.2	出力単語の決定	20
5.3.3	ボケ出力	20
5.4	ツッコミ生成部の実装	21
5.4.1	言い換えツッコミの生成	21

5.4.2	動詞ツッコミの生成	22
5.4.3	ツッコミの出力	23
5.5	ボケとツッコミの出力	23
第 6 章	評価実験	25
6.1	実験の目的	26
6.2	実験の概要	26
6.3	実験の手順	27
6.4	実験結果・考察	29
6.4.1	RQ1 の実験結果・考察	29
6.4.2	RQ2 の実験結果・考察	32
6.4.3	RQ3 の実験結果・考察	36
第 7 章	結論	40
	参考文献	42
	研究業績	46

目 次

4.1	提案モデル	14
5.1	システムの全体像	16
5.2	事前準備	17
5.3	読み方辞書	17
5.4	動詞ツッコミの生成例	22
6.1	実際の実験画面	28
6.2	Q1:会話の流れは自然だと感じたか (N=10, 1人あたり5回答)	30
6.3	Q2:エージェントと苛立ちなく会話することができたか (N=10, 1人あたり5回答)	30
6.4	Q3:エージェントの発言にユーモアを感じたか (N=10, 1人あたり5回答)	30
6.5	Q6:エージェントに信頼を感じたか (N=10, 1人あたり5回答)	31
6.6	Q7:エージェントと負担なく会話できたか (N=10, 1人あたり5回答)	31
6.7	Q8:エージェントに親しみを感じたか (N=10)	31
6.8	Q1:会話の流れは自然だと感じたか (N=10, 1人あたり5回答)	33
6.9	Q2:エージェントと苛立ちなく会話することができたか (N=10, 1人あたり5回答)	33
6.10	Q3:エージェントの発言にユーモアを感じたか (N=10, 1人あたり5回答)	34
6.11	Q6:エージェント(たち)に信頼を感じたか (N=10, 1人あたり5回答)	34
6.12	Q7:エージェント(たち)と負担なく会話できたか (N=10, 1人あたり5回答)	34
6.13	Q8:エージェント(たち)に親しみを感じたか (N=10)	35
6.14	Q3:エージェントの発言にユーモアを感じたか (N=10, 1人あたり5回答)	37
6.15	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たか (N=10, 1人あたり5回答)	37
6.16	Q5:エージェントの会話に人間味を感じたか (N=10, 1人あたり5回答)	38
6.17	Q6:エージェントたちに信頼を感じたか (N=10, 1人あたり5回答)	38
6.18	Q7:エージェントたちと負担なく会話できたか (N=10, 1人あたり5回答)	38
6.19	Q8:エージェントたちに親しみを感じたか (N=10)	39

表 目 次

5.1	s_s が高い例・低い例	18
5.2	s_e が高い例・低い例	19
5.3	s_f が高い例・低い例	20
5.4	EDR 電子化辞書：単語と概念識別子の対応例	22
6.1	実験の質問一覧	28
6.2	実験者が用意した 50 単語	28
6.3	RQ1：実験の質問一覧	29
6.4	RQ2：実験の質問一覧	32
6.5	RQ3：実験の質問一覧	36

第1章 序論

1.1 研究の背景

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の我々の生活における様々なシーンへと浸透し始めており [1], 将来的にはユーザの作業の自動化や、ユーザの友人になるといった活躍が期待される。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感じることができないユーザも少なからずいると思われる。例えば、リビングルームにコミュニケーションロボットが導入されたシーンを考えたとき、ロボットが常に堅苦しい話ばかりしていたら、家族の一員にはなりにくいだろう。あるいは、介護施設で高齢者の話し相手になるロボットが、冗談1つも言わず、生真面目な会話ばかりしていたら、ロボットが施設の人気者になることは難しいだろう。このような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言える。こ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 [2][3][4][5], ユーモアと親しみの関係性が窺える。

1.2 研究の目的

先行研究として、ユーザがエージェントに対して親しみを持つことができるよう、ユーザの発言を利用してユーモア発話を行う対話型エージェントが存在する。これは、認知科学の分野において支持されている不適合-解決モデルを参考に、ユーザの発言の一部を面白く聞き間違えることで、ユーモア生成を行っている。しかし、先行研究では、エージェントがユーモア発話を行うと、会話の軌道が逸れてしまい、ユーザに会話を軌道修正する負担が生じていた。この負担のため、ユーザはエージェントを継続的に利用しづらいと考えられる。本研究では、先行研究のエージェントとの対話時に生じていたユーザの負担を解消することを目指す。ユーザの負担を解消することで、ユーザはエージェントを対話相手として継続的に利用しやすくなると考えられる。以上より、ユーザに負担をかけずにユーモア提示を行う対話型エージェントのプロトタイプシステムを実現し、検証実験からその効果を確認することを目的とする。

1.3 本論文の構成

本論文の構成は次のとおりである。

2章では、対話型エージェントを様々な分野に活用した研究事例と、エージェントによるユーモア表現を試みている研究事例について紹介する。

3章では、対話型エージェントがユーモア表現を行う際に生じる問題点について述べ、それらを踏まえたうえで本研究における課題を設定する。

4章では、日本で広く定着しているコメディスタイルである漫才をもとに、複数対話型エージェントによるユーモア生成システムを提案する。

5章では、ユーモア発話と、それに対して行う指摘内容の生成方法を中心に、複数対話型エージェントによるユーモア生成システムの実装方法について具体的に述べる。

6章では、行ったユーザ実験の目的や手順について述べ、実験結果から得られた知見についても述べる。

最後に7章にて、本論文の結論を述べる。

第2章 エージェントに関する研究事例

本章では、ソフトウェアエージェントに関する研究事例について述べる。2.1節では、対話型エージェントに関する研究事例について紹介する。2.2節では、ユーモア表現を行うエージェントについて紹介する。

2.1 対話型エージェントに関する研究事例

本節では、対話型エージェントに関する研究事例について述べる。近年では対話型エージェントを対象とした研究は多岐にわたり、様々なシーンでの活用方法が検討されている[6][7][8][9][10][11][12][13][14][15]。

[6], [7], [8]では、人と対話型エージェントが協調的に対話するための研究が行われている。[6]は、エージェントと対話を行うことでコミュニケーション不足を解決しようとする試みである。1日の中の決まった時間に、エージェントから利用者に対して会話を行うことで、継続的にコミュニケーションを促している。エージェントは利用者の発言を受け、事前に設定されたコーパス内から対応する返答をする。このとき、利用者との過去の会話を参考にしてコーパスを更新していくことで、会話の質を高めようとしている。[7]では、より自然でストレスの少ない親和的コミュニケーション制御方式を提案している。エージェントが、ユーザの発言に合わせて共感を引き起こす会話構造モデルをもとに頷きなどの身体動作を行い、会話状態をダイナミックに遷移させることでコミュニケーションを促進させている。[8]は、認知症高齢者であるユーザの日々の健康状態・認知状態を把握することを目的としている。エージェントに対するユーザの反応から、ユーザの発話間のポーズ・ピッチ・発話長・頷き時間を分析することで、健康状態・認知状態を把握することを試みている。

[9], [10]では、1人のユーザに対し、複数のエージェントが対話を行うシステムを提案している。[9]では、ユーザが1人の場合でも効果的な対話型鑑賞を行うことができる鑑賞支援システムを提案している。実験により、鑑賞や意見を促す進行役と、ユーザと同じ立場に立って感想を言う鑑賞役の2種類のエージェントを用意することで、人間による対話型鑑賞に近い話題の展開が可能であることが示唆されている。[10]では、ユーザ1人に対し、対話ドメインについて異なる意見を持つ2種類のエージェントを用意する、三者間の対話システムを提案している。実験により、三者間の対話システムは、ユーザが感じるエージェントへの親しみや、対話の雑談らしさが増すことが示されている。

[11], [12]では、エージェントをゲーム分野に利用する研究が行われている。[11]は対話によるコミュニケーションを必要とする人狼ゲームを題材とし、若者の対話教育のための支援環境として、エージェントを用いた手法を提案している。Wizard of Oz 法にて人間がエージェントを操作し議論を行うことによって、論理的な発言による議論を疑似的にユーザに体験させている。これにより、ユーザがより良い議論や対話方法について評価・修正することを可能にしている。[12]は、GPS機能付き携帯電話を端末として、エージェントによる現実感のある世界の表現を試みている。現実世界のユーザの位置情報を仮想世界へと反映することで、仮想生物からのアクションなどを引き起こすことができる。

[13], [14], [15] は、それぞれプレゼンテーション、車の運転、余暇の充実化についてユーザを支援するエージェントを提案している。[13] はエージェントが、エージェント作成者の代理として任意のユーザと会話することができる、分身エージェントの実現手法を提案している。これにより、エージェント作成者、またはエージェントと会話を行うユーザは、時間的・地理的な制約に縛られずに情報の獲得と発信ができるようになる。[14] はドライバの退屈や眠気を回避させる手段として、対話型 CG エージェントを提案している。エージェントは、車から得られる車速や経過時間、走行距離やラジオなどのデータを利用して、ユーザが飽きないような話題を提供する。[15] はスケジュール帳を情報源として、ユーザに余暇の過ごし方を提案するエージェントについて提案している。エージェントは、ユーザとテキストで会話をしながら、ユーザのスケジュールと発言に基づいて予定を提案する。

2.2 ユーモア表現を行うエージェントに関する研究事例

本節では、ユーザとのコミュニケーションをより良いものにするために、ユーモア表現を行うエージェントについて述べる。ユーモア表現を行うエージェントは、大きく分けて、一方的にユーモア表現を行うエージェントと対話によってユーモア表現を行うエージェントに分けられる。これらは2.2.1 項、2.2.2 項においてそれぞれ紹介する。

2.2.1 一方的にユーモア表現を行うエージェント

一方的にユーモア表現を行うとは、エージェントの数に関わらず、ユーザとのコミュニケーションが生じない形式でユーモアを表現することを指す。

単体のエージェントが、一方的なユーモア表現を行う事例として、[16], [17], [18], [19] が挙げられる。[16], [17] では、エージェントが、ユーザにコンピュータの使い方をユーモアを交えて教える立場を想定している。[16] は、本来エージェントが発する文章内の語（被置換語）を他の語（置換語）に置き換えた置換型駄洒落をユーザに対して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 Support Vector Machine で学習することで実現している。[17] 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した駄洒落をユーザに対して出力している。

[18], [19] では、エージェントとユーザは、ユーモア表現を行う演者とそれを見る観客という関係にある。[18] は、各単語の頭文字のアルファベットを連結することで省略している名称を、略称そのままに各単語を置換することでユーモアを生み出す試みである。単語の置換時には、元の単語群と新たに生成された単語群の関係が、意味は遠く、音は近くなるようにしている。[19] は、画像にユーモアのあるタイトルを付ける試みである。画像とそれに対するユーモアタイトルをデータセットとして機械学習を行うことで実現している。また、地域やトレンドなども考慮することでユーモア性の向上を狙っている。

複数のエージェントが、一方的なユーモア表現を行う事例として、[20], [21], [22], [23], [24], [25] が挙げられる。いずれも、エージェントとユーザは、それぞれユーモア表現を行う演者とそれを見る観客という関係にある。

中でも [20], [21], [22], [23], [24] は、2体のエージェントをそれぞれユーモア発話を行う役（ボケ役）とユーモア発話を指摘する役（ツッコミ役）に役割分担している。[20], [21] は、漫才形式の対話文の自動生成システムである。入力された物語や小説などの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。[22], [23] では、ユーザから与えられたお題に関連する Web ニュース記事から漫才台本を自動生成する手法を提案している。Web ニュース記事本文の感情を表す単語を考慮することで、おかしみの構造図に基づいた形式的なボケを生成している。[24] では、ユーザの笑い声の大きさに応じて発話内容や発言の間、動作を変化させる2体のロボットを用いることで、人間が行う漫才に近いロボット漫才を実現している。

[25] は、複数のエージェントがユーモア表現を行う役割を担っており、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。ユーザから入力されたお題に対して、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、機械学習的アプローチに基づいてユーモアの生成を行っている。

2.2.2 対話によってユーモア表現を行うエージェント

対話によってユーモア表現を行うとは、エージェントの数に関わらず、ユーザとの対話内容をもとにしてユーモアを表現することを指す。

単体のエージェントとユーザ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、[26], [27], [28], [29], [30], [31], [32], [33], [34] が挙げられる。いずれの事例においても、エージェントとユーザは、雑談や質問を行う友人のような関係にある。[26] は、ユーザの対話継続欲求を向上させるために、対話システムによるユーモア発話の自動生成手法を提案している。ユーザから入力された文章中で、単語間類似度が最も高い修飾-名詞ペアをもとにユーモアを生成している。[27] では、文中の代名詞関係に着目することでユーモア文を生成する試みである。ユーザから入力された文中から、代名詞が指す正しい対象語とは異なる対象語を発見することで実現している。[28], [29], [30] 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。[31], [32], [33] は、エージェントが、ユーザの発言内容の一部を、意味が遠く音が離れた聞き慣れた単語にわざと間違えて聞き返すことでユーモアを生成している。[34] では、ユーザから入力された文中の単語のうち、聞き間違えたときに最もユーモア性が向上する単語を選定し置換するシステムを提案している。

複数のエージェントとユーザ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、[35] が挙げられる。[35] は、2体のエージェントを、ユーモア発話を行う役とユーザの感情分析を行う役に分担してユーモアを生成するシステムである。ユーモアの生成に

は、ユーザから入力された単語を音の近い別の単語に置き換えることで実現している。また、ユーザの感情を抽出し、その結果によってユーモア発話を行うか否か決定している。

第3章 研究課題

3.1 問題の定義

現代社会において対話型エージェントは、日常生活の様々なシーンで利用され始めている [1]. 家庭内における事例としては, Amazon の Alexa, Apple の Siri, Microsoft の Cortana, Google Assistant が有名である. 家庭外においても, ホテルやショッピングセンターなどの現場で簡単な受け答えをする役割としての利活用が進んでいる. 将来的には, 介護や福祉現場などの人間とのコミュニケーションが重要な場面での活躍も期待されており, 現在でも様々な分野での活用法が研究されている [6][7][8][9][10][11][12][13][14][15]. しかし, このような対話型エージェントによる対話の多くは, 未だ無機質なものである. そのため, 無機質な対話に親しみを持ってないユーザには, このようなエージェントは受け入れられないという問題が懸念される. 例えば, リビングルームにコミュニケーションロボットが導入されたシーンを考えたとき, ロボットが常に堅苦しい話ばかりしていたら, 家族の一員にはなりにくいだろう. あるいは, 介護施設で高齢者の話し相手になるロボットが, 冗談 1 つも言わず, 生真面目な会話ばかりしていたら, ロボットが施設の人気者になることは難しいだろう.

上述の問題を解消するために, 人とエージェント間で親和的な関係を形成する必要があると考えられる. こ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 [2], 人同士だけでなく, 人とエージェント間の関係においても, 同様の報告が多数なされている [3][4][5]. 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, [16], [17], [20], [21], [22], [23], [25] 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, [26] 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言をユーモアとして受容しにくい場合がある. 第 3 に, [28], [29], [30] 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, 先行研究として, 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 [31][32][33][34]. しかし, これらの先行研究は, ユーモア発言を行うのみのシステムであった. このため, エージェントがユーモア発言を行った場合, ユーザが会話を継続するためには, ユーモア発言に対して指摘し会話を軌道修正する必要がある. ユーザにとって負担がかかるという問題が存在していた. 例えば, ユーザがエージェントに対して「味噌汁について教えて?」と質問し, それに対してエージェントが, 先行研究の手法をもとに「え? ミサイル?」とユーモア発言を行ったシーンを考える. このとき, ユーザが質問の答えを得るには, 「なんでミサイルなんだよ! 味噌汁だって!」のような会話を軌道修正する言葉を考えたうえで, ユーモア発言を指摘する手間がかかる. 次に, ユーザがエージェントに対して雑談を持ちかけ, それに対してエージェントがユーモア発言を行ったシーンを考える. 雑談の場合, 質問と違いユーザが答えを得る必要はない

が、会話を継続するためには、ユーザがユーモア発話を指摘しなければならない。先行研究では、ユーザがエージェントと長期間コミュニケーションを行う事が想定されているため、上記のような負担が継続的にユーザにかかる。そうすると、ユーザはエージェントとの対話に疲れ、エージェントを対話相手として受け入れられなくなってしまうことが考えられる。

3.2 研究課題の設定

3.1節で定義した問題を踏まえ、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、ユーザが、ユーモア発話に対する指摘や会話推進などを行う必要をなくすことを研究課題とする。

第4章 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提案

4.1 アプローチ

3.2節で設定した課題を達成するために、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や娯楽番組において、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。

4.2 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提案

4.1節で述べた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 [36]。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 [37]。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。本提案システムのモデルを図 4.1 に示す。

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、常に同じものであったり、ユーモア発話内容の繰り返す単調なものであったりすると、決められた言葉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、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、5.4.1 項の方法でユーモア発話内容の言い換えを行ったり、5.4.2 項の方法でユーモア発話内容に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と考えられる。

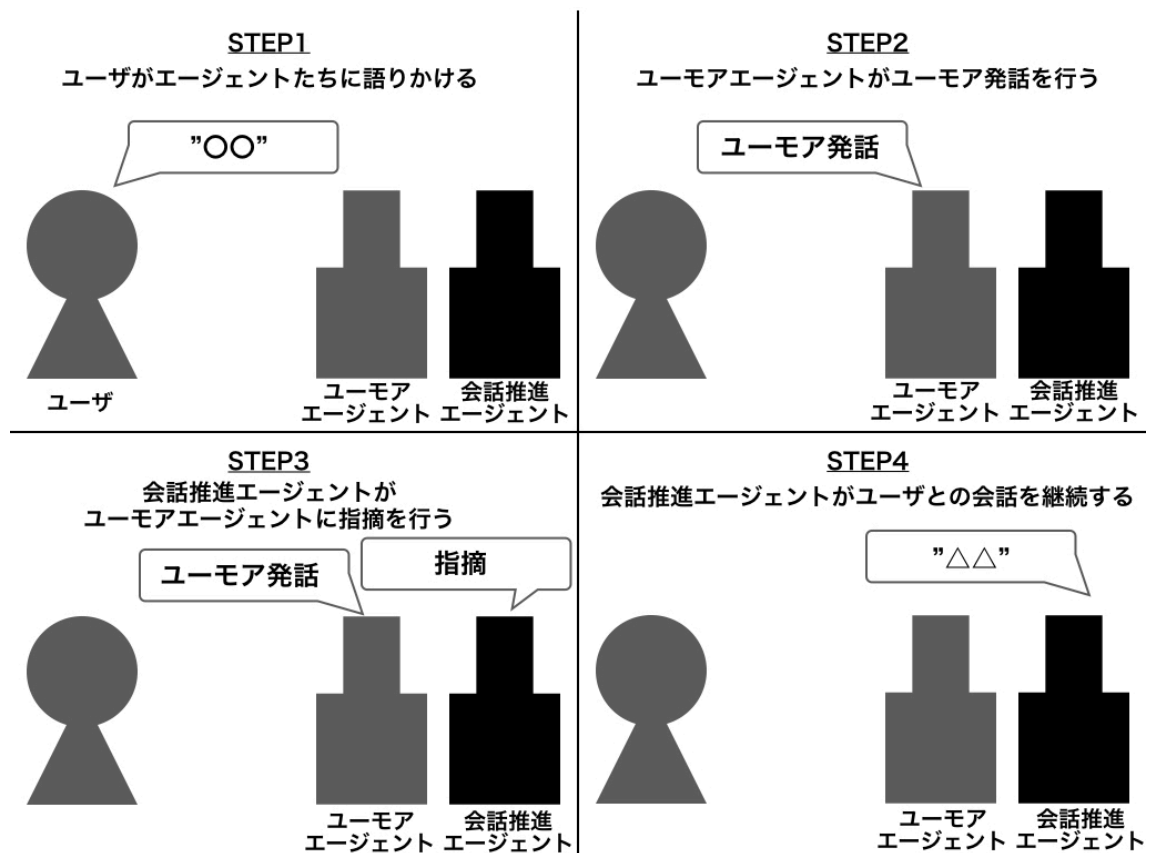


図 4.1: 提案モデル

上記の方式では、ユーザの負担となっていた会話の軌道修正を、代わりにエージェントが行うため、3.2 節の研究課題を達成できると考えられる。

第5章 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実装

5.1 システムの全体像

提案システムは、ユーモア発話を行うボケ生成部と、指摘・会話推進を行うツッコミ生成部からなる（図 5.1）。

ボケ生成部では、ユーザからの入力単語に対し、意味が離れていて、音が近く、ユーザが理解できる単語をユーモア発話として出力する。具体的には、ユーザの入力単語に対して、どれくらい意味が離れているかを表す s_s (Semantic Score), どれくらい音が近いかを表す s_e (Edit distance Score), どれくらいユーザから理解されやすいかを表す s_f (Frequency Score) をもとに出力単語を決定する。

ツッコミ生成部では、ボケ生成部で生成したユーモア発話に対して、指摘を行う文を出力する。指摘の内容は、ユーモア発話内容の言い換え語や、関連する動詞から自動生成する。

実装は、ボケ生成部、ツッコミ生成部ともに Python(バージョン 3.6.3) を用いて行う。

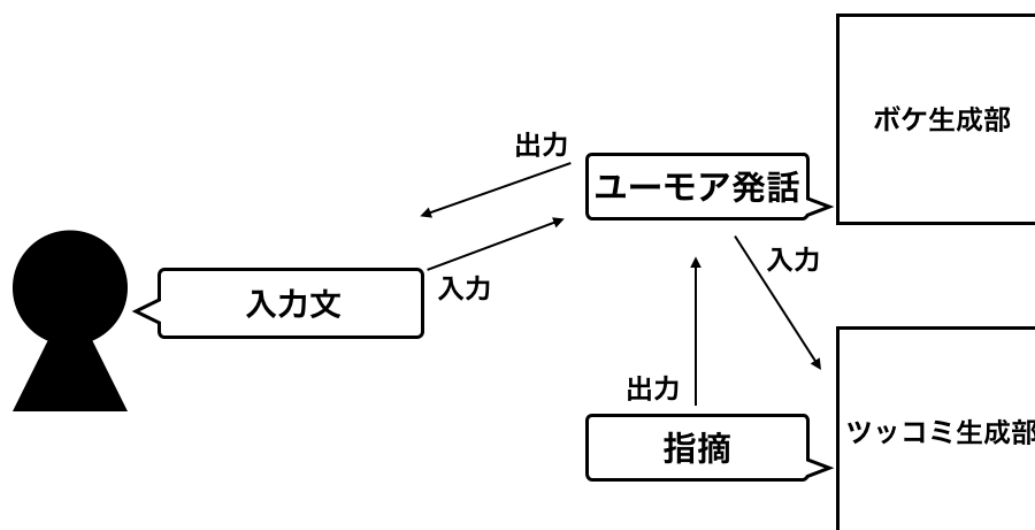


図 5.1: システムの全体像

5.2 事前準備

Wikipedia 日本語記事全文（取得日時 2017.10.12）を MeCab[38] を用いて形態素解析し、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（1,004,870 単語）。ここでの不要品詞とは、語彙の意味を担っていなかったり、Wikipedia 日本語記事での出現回数が多すぎたりするため、後述する言語モデルの作成や、5.3.1.3 目で述べる s_s の算出に悪影響を及ぼすものである。具体的には、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．また，コーパスを元に読み方リストと言語モデルの作成を行う（図 5.2）．読み方リストとは，コーパス内での出現回数が 50 回以上かつ日本語である単語の標準形，MeCab で取得した標準形の読み方（カタカナ），標準形の読み方をローマ字表記に変換したのち子音を削除して母音のみにしたもの，これら 3 つを単語ごとにまとめたリストである（図 5.3）．これらは，後述の 5.3.1.2 目で用いる．本研究における言語モデルとは，単語間の概念距離を計算できるよう自然言語の単語をベクトル表現したものであり，word2vec[39] を用いてコーパスから獲得する．このとき，次元数はデフォルトの値である 100 を使用する．Window サイズは，同じトピックに現れる単語を取得しやすいように，デフォルトの値より大きい 15 に設定する，使用するライブラリは gensim(バージョン 3.0.1) とする．

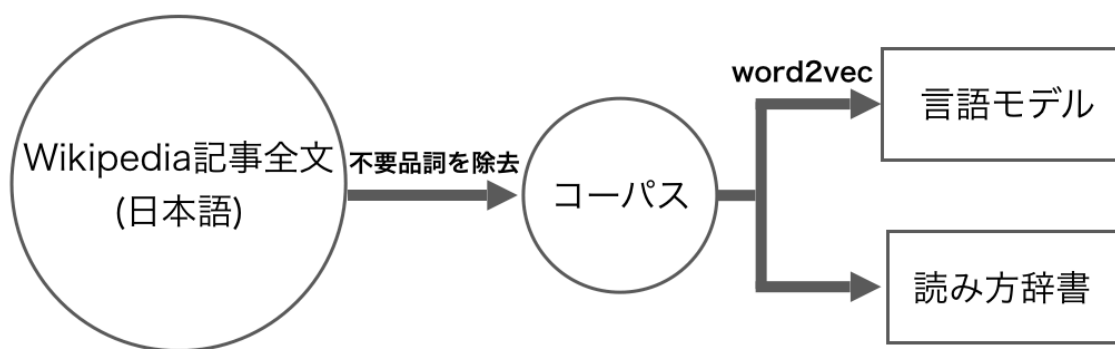


図 5.2: 事前準備

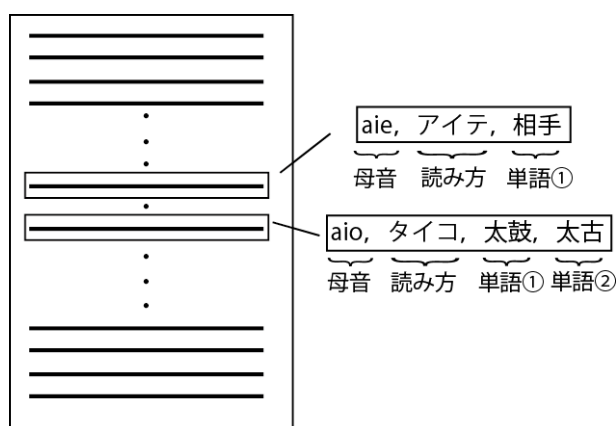


図 5.3: 読み方辞書

5.3 ボケ生成部の実装

5.3.1 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) の算出方法について述べる. この各スコアの算出は, 先行研究で用いられているユーモア発話の生成手法をもとに行う [31][32][33][34].

5.3.1.1 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 を算出する. 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 (表 5.1). すなわち, s_s は下記のようになる.

$$s_s = d_s$$

d_s は入力単語と出力候補単語の概念距離である. d_s の算出には gensim ライブラリ (バージョン 3.0.1) で実装されている word2vec の similarity メソッドを用いる. similarity メソッドで導き出される類似度の値域は -1.0 ~ 1.0 であり, 値が大きいほど単語間の概念距離が近いことを意味する. そのため, 実際には得られる類似度の逆数を取得することで s_s を算出する.

表 5.1: s_s が高い例・低い例

元の単語	出力候補単語	s_s
りんご	死後	1.243140
	いちご	0.619789
探偵	割合	1.293864
	犯罪	0.689766

5.3.1.2 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*を測ることで s_e を算出する. 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 (表 5.2). すなわち, s_e は下記のようになる.

$$s_e = \frac{1}{1 + d_e}$$

*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.

d_e は入力単語と出力候補単語の編集距離である． d_e の算出には python-Levenshtein ライブラリ（バージョン 0.12.0）内の distance メソッドを用いる．導き出される d_e の値域は $0 \sim \infty$ であり，値が大きいほど単語間の編集距離が遠いことを意味する．

また，試行錯誤の過程で，入力単語との編集距離が同じである単語の中でも聞き間違いに聞こえる単語とそうでない単語があることが分かった．例えば，“戦闘機（enoui）”という入力単語に対し，“倦怠期（enaii）”，“関東（anou）”という単語の各母音のみを考慮した編集距離はともに 2 であるが，前者は聞き間違いに聞こえ，後者はそうでないと感じられる人が多かった．多くの試行を行った結果，入力単語と出力単語の頭文字の母音と末尾の文字が一致する場合は，聞き間違いに聞こえやすいということが分かった．この知見に基づき， s_e が高い場合でも，入力単語と先頭の文字の母音・末尾の文字が異なる単語は出力候補単語から除く．

表 5.2: s_e が高い例・低い例

元の単語	出力候補単語	s_e
工場	向上	1.000000
	今週	0.333333
機会	未来	1.000000
	地方自治体	0.200000

5.3.1.3 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 を算出する．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（表 5.3）．すなわち， s_f は下記のようになる．

$$s_f = \log f$$

f は出力候補単語のコーパス内での出現回数である．このとき，取りうる f の値域は $50 \sim 3,362,984$ となる（下限が 50 となる理由については後述する）．ごく一部の極端に出現回数の多い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を s_f とする．また，同様の理由から s_f の最大値を 4.82 とする．この値は，コーパス内からランダムに選んだ 1000 単語を対象に出現回数の調査を行い，99.5 パーセンタイル値である単語の出現回数に基づき決定した． s_f の値が 4.82 となる出現回数は，66,069 回である．一方，極端に出現回数が少ない単語はユーザに理解されない可能性が高いと考え，出現回数が 50 回を下回る単語は出力候補から除く．

上記より， s_f の値域は $1.698970 \sim 4.820000$ となる（出現回数 50 回の単語の s_f は 1.698970 である）．

表 5.3: s_f が高い例・低い例

単語	s_f
行う (875,205 回)	4.820000
共通 (41,803 回)	4.621207
歪曲 (1,012 回)	3.005181
赤信号 (346 回)	2.539076

5.3.2 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。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。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。

- 頭文字の母音と末尾の文字が入力単語と一致する。
- コーパス内での出現回数が50回以上である。
- 人名や地名に分類される固有名詞ではない。
- 入力単語を、ひらがな・カタカナ表記にしたものではない。

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。

$$s_t = w_s s_s + w_e s_e + w_f s_f$$

w_s, w_e, w_f は重み係数である（本稿では重み係数は全て1.0とする）。

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。

5.3.3 ボケ出力

エージェントが聞き間違いを行うシーンを想定し、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?

5.4 ツッコミ生成部の実装

4.2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られる。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。例えば、“え？ミサイル？”というユーモア発話が出力された場合、言い換えツッコミでは、“誘導弾か！”，動詞ツッコミでは、“なんで撃ち落とさなきゃいけないんだよ！”と出力を行う。5.4.1項に言い換えツッコミ、5.4.2項に動詞ツッコミの具体的な生成方法を示す。

5.4.1 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方法で取得する。

同義語の取得には、EDR 電子化辞書 [40] 内の日本語単語辞書を利用する。日本語単語辞書内の単語には、かな表記や発音、品詞や活用情報、概念説明や概念識別子といった様々な情報が付与されている。この中の、概念識別子という情報を利用することで、ある単語の同義語群を取得する。概念識別子とは、単語の意味ごとに付与されている16進数の番号であり、同じ意味の単語には同じ概念識別子が付与されている。これを利用することで、ある単語の同義語群を取得できる。単語と概念識別子の対応例を表5.4に挙げる。“デスク”という単語は、“読み書き用の机”や“主任”といった複数の意味が存在するため、それぞれに対応した複数の概念識別子が付けられている。取得した同義語群内で、EDR コーパス[†]内で定義付けられている単語の頻度が最も高い単語を、言い換えツッコミ単語として出力する。このとき、入力単語が表記揺れした単語は、言い換えとは言えないため、出力候補から削除する。同様に、Wikipedia コーパス内での出現回数が50回以下の単語は、ユーザが理解しにくいと考え、出力候補から削除する。よって、言い換えツッコミ単語の出力候補単語となる条件は下記の通りである。

言い換え単語条件:

- 入力単語が表記揺れした単語ではない（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。

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概念の単語がそれぞれ結び付けら

[†]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。

表 5.4: EDR 電子化辞書：単語と概念識別子の対応例

単語	概念識別子
衣料	0e504a
衣類	0e504a
デスク	3bb892, 3bee23, 3d05cf
ランチ	3bec74, 3c0457, 3c0458, 3c58fd
昼食	3bec74

れており，これを用いて上位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．

5.4.2 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，5.3 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．概念距離の算出には，gensim ライブラリ (バージョン 3.0.1) で実装されている word2vec の similarity メソッドを用いる．次に，取得した単語群から，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の品詞を未然形に変え，動詞ツッコミ単語とする (図 5.4)．品詞の変更は，MeCab の辞書をもとに行う．

	単語	品詞	概念距離
入力単語： ミサイル	空対地ミサイル	名詞	(0.819)
	巡航ミサイル	名詞	(0.807)
	ミサイル発射	名詞	(0.770)
	兵器	名詞	(0.563)
	撃ち落とす	動詞	(0.562)

図 5.4: 動詞ツッコミの生成例

5.4.3 ツッコミの出力

エージェントがユーモア発話に対して指摘を行うシーンを想定し、「“言い換えツッコミ単語”か！なんで“動詞ツッコミ単語”なきゃいけないんだよ！」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ゃいけないんだよ！
- 入力：派遣社員
ツッコミ：社員か！なんで働かなきゃいけないんだよ！

5.5 ボケとツッコミの出力

5章で行った実装による出力例を下記に示す。

- ユーザ：味噌汁について教えて？
ボケ：え？ミサイル？
ツッコミ：誘導弾か！なんで撃ち落とさなきゃいけないんだよ！ミサイルだって！
- ユーザ：バレンタインについて教えて？
ボケ：え？派遣社員？
ツッコミ：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！
- ユーザ：ペニシリンについて教えて？
ボケ：え？警備員？
ツッコミ：ガードマンか！なんで取り押さえなきゃいけないんだよ！ペニシリンだって！
- ユーザ：じゃんけんについて教えて？
ボケ：え？晩年？
ツッコミ：余生か！なんで没しなきゃいけないんだよ！じゃんけんだって！
- ユーザ：賠償金について教えて？
ボケ：え？太陽神？
ツッコミ：自然神か！なんで崇めなきゃいけないんだよ！賠償金だって！
- ユーザ：土俵について教えて？
ボケ：え？旅行？
ツッコミ：旅か！なんで出かけなきゃいけないんだよ！土俵だって！
- ユーザ：戦列艦について教えて？
ボケ：え？検察官？
ツッコミ：検事か！なんで取り調べなきゃいけないんだよ！戦列艦だって！

- ユーザ : 舞鶴について教えて？
ボケ : え？ライフル？
ツッコミ : ライフル銃か！なんで撃たなきゃいけないんだよ！舞鶴だって！
- ユーザ : タオルについて教えて？
ボケ : え？バトル？
ツッコミ : もめ事か！なんで挑まなきゃいけないんだよ！タオルだって！
- ユーザ : コンセントについて教えて？
ボケ : え？コメント？
ツッコミ : 講評か！なんで語らなきゃいけないんだよ！コンセントだって！

第6章 評価実験

6.1 実験の目的

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。まずは、1体のエージェントが、ユーモア発話に加えて指摘を行うことの効果を検証する。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条件と、1体のエージェントが会話を推進する条件を比較し、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条件と、ユーモア発話に対して単調な指摘を行う条件を比較し、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。

以上をまとめると、本実験では以下の 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 実験の概要

本実験の被験者は20代の学生10名(男8, 女2)である。被験者が単語を入力する際には、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つの条件を用いる。

- 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)

- 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 実験の手順

被験者には6.2節に示した4つの条件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う。被験者への質問と選択肢を表6.1に示す。各条件使用時には、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、一定のユーモア性がある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（表6.2）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自由入力してもらう。これは、ユーザが会話を軌道修正する際の負担を検証するためである。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。実際の実験画面を図6.1に示す。テキストの色を使い分けることで、発話を行っているエージェントの区別が付くようにする。

表 6.1: 実験の質問一覧

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

表 6.2: 実験者が用意した 50 単語

じゃんけん	裁判	神社	領域	役割
超能力	舞鶴	科目	シマウマ	仏壇
占い	消防車	幼稚園	戦国時代	馬車
割れ物	魔物	タオル	生身	ヨガ
成績	トランプ	標本	バレンタイン	放送局
味噌汁	救急車	森林公園	ペニシリン	犬
買収	ゴジラ	忠誠	魔物	磁石
化学者	脱出	電流	マフィア	故障
戦列艦	味噌汁	伝説	土俵	目撃者
ガメラ	栄養士	天使	中学校	ゾンビ

ユーザ＞味噌汁について教えて？
 ボケ役＞え？ミサイル？
 ツッコミ役＞誘導弾か！なんで撃ち落とさなきゃいけないんだよ！味噌汁だって！
 ツッコミ役＞味噌汁は、日本料理における汁物の一つで、だしを味噌で調味した汁に、野菜や豆腐、麴や魚介類などの食品を実としたスープ様の料理だよ
 ユーザ＞教えてくれてありがとう

図 6.1: 実際の実験画面

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、全条件において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 実験結果・考察

6.4.1 RQ1の実験結果・考察

RQ1は、エージェントがユーモア発話への指摘を代行することで、ユーザが感じる苛立ちや負担が軽減されるかである。そのため、6.2節で示したツッコミ不在条件と自己ツッコミ条件間で比較を行う。また、6.2節で示した質問のうち、Q1～Q3、Q6～Q8への回答を確認する。Q1～Q3、Q6～Q8を表6.3に再掲する。

表 6.3: RQ1：実験の質問一覧

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
Q6	エージェント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

Q1の回答結果を図6.2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38％、自己ツッコミ条件では44％であった。Q2の回答結果を図6.3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42％、自己ツッコミ条件では22％であった。Q3の回答結果を図6.4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30％、自己ツッコミ条件では36％であった。Q6の回答結果を図6.5に示す。“エージェントに信頼を感じたか”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20％、自己ツッコミ条件では20％であった。Q7の回答結果を図6.6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40％、自己ツッコミ条件では30％であった。Q8の回答結果を図6.7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40％、自己ツッコミ条件では20％であった。

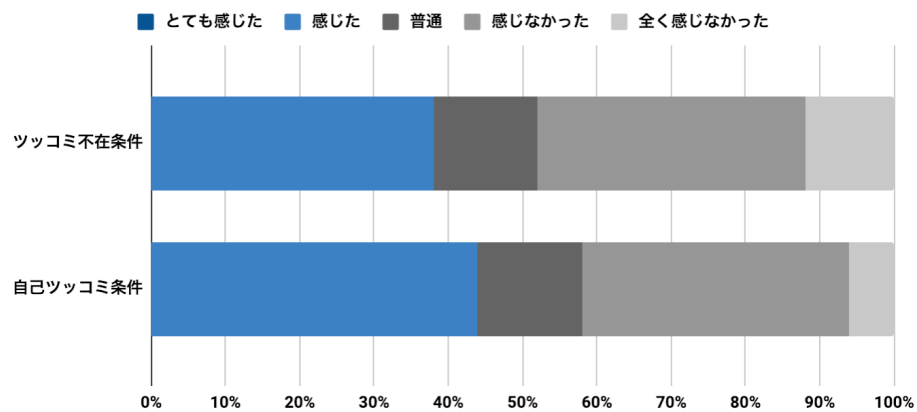


図 6.2: Q1:会話の流れは自然だと感じたか (N=10, 1人あたり5回答)

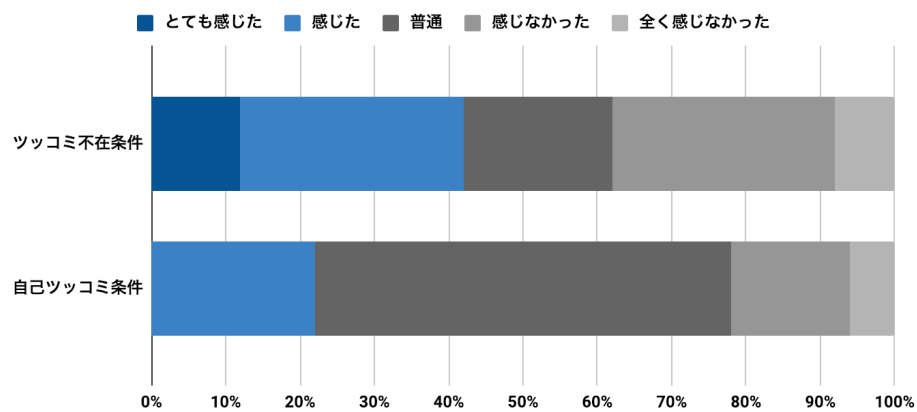


図 6.3: Q2:エージェントと苛立ちなく会話することができたか (N=10, 1人あたり5回答)

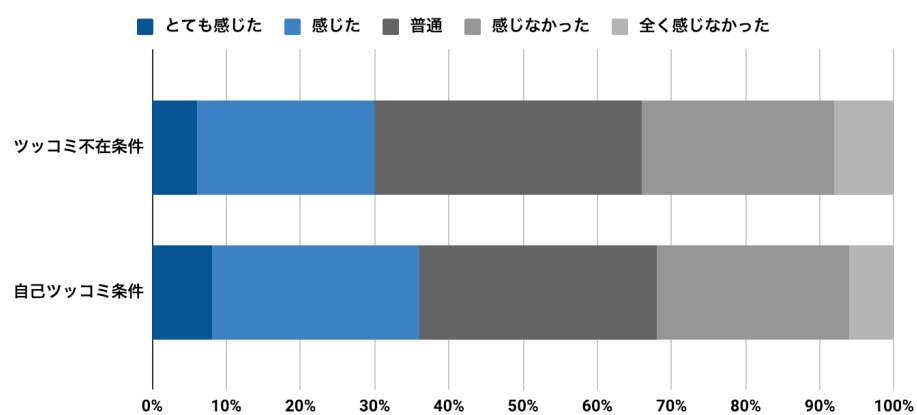


図 6.4: Q3:エージェントの発言にユーモアを感じたか (N=10, 1人あたり5回答)

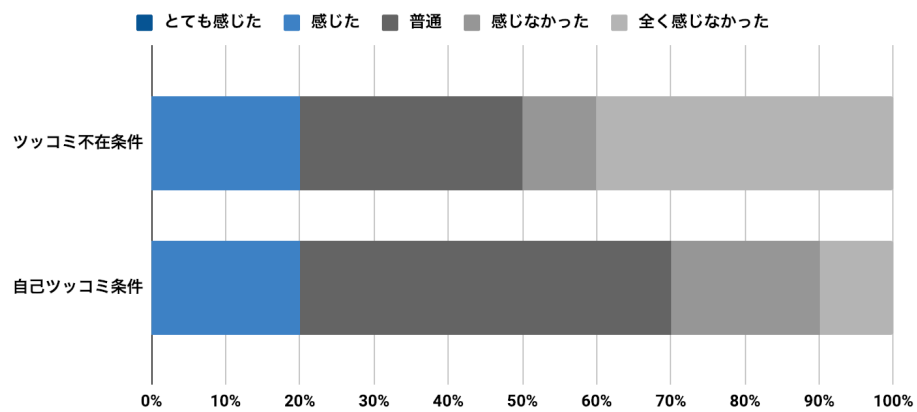


図 6.5: Q6:エージェントに信頼を感じたか (N=10, 1人あたり5回答)

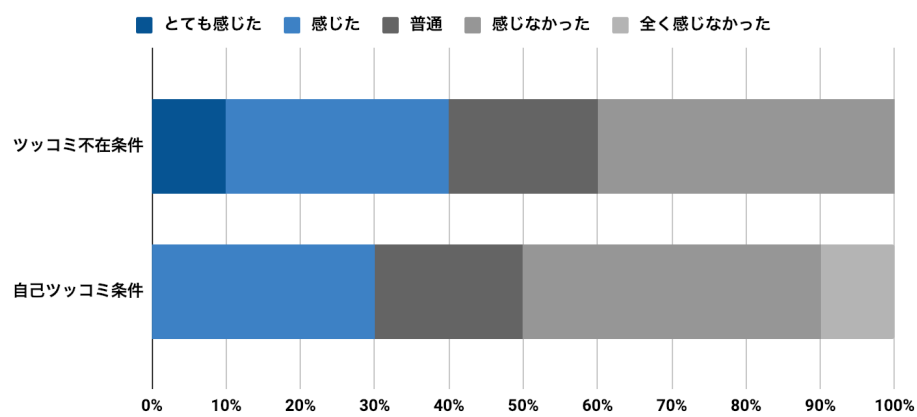


図 6.6: Q7:エージェントと負担なく会話できたか (N=10, 1人あたり5回答)

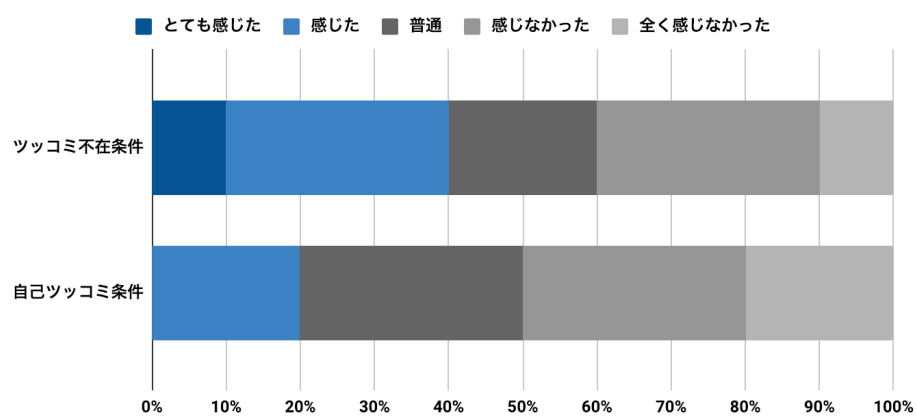


図 6.7: Q8:エージェントに親しみを感じたか (N=10)

Q1～Q3, Q6～Q8 の回答に対して, 2つの条件間で 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られられる. 被験者からの自己ツッコミ条件に対しての自由記述を見ると, “一人芝居感が強い”や“自分で間違いに気づき不自然”といった意見が見られた. このことから, 会話の流れの違和感が, ユーザの苛立ちや負担に繋がってしまったと考えられる. また, 両条件の全質問において, “とても感じた”または“感じた”と回答した被験者数は半数以下であった. ここから, 両条件ともに, ユーモア対話中に用いる条件として適切でないと考えられる.

6.4.2 RQ2の実験結果・考察

RQ2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で, ユーザが感じる苛立ちや負担が軽減されるかである. そのため, 6.2節で示した自己ツッコミ条件と単純ツッコミ条件間で比較を行う. また, 6.2節で示した質問のうち, Q1～Q3, Q6～Q8への回答を確認する. Q1～Q3, Q6～Q8を表6.4に再掲する.

表 6.4: RQ2 : 実験の質問一覧

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
Q6	エージェント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	エージェントに対して親しみを感しましたか？ 5: とても感じた ～ 1:全く感じなかった

Q1 の回答結果を図 6.8 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, 自己ツッコミ条件では 44 %, 単純ツッコミ条件では 58 %であった. Q2 の回答結果を図 6.9 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, 自己ツッコミ条件では 22 %, 単純ツッコミ条件では 58 %であった. Q3 の回答結果を図 6.10 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, 自己ツッコミ条件では 36 %, 単純ツッコミ条件では 38 %であった. Q6 の回答結果を図 6.11 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, 自己ツッコミ条件では 20 %, 単純ツッコミ条件のユーモアエージェントに対しては 30 %, 会話推進エージェントに対しては 51 %であった. Q7 の回答結果を図 6.12 に示す. “エージェント (たち) 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, 自己ツッコミ条件では 30 %, 単純ツッコミ条件は 60 % であった. Q8 の回答結果を図 6.13 に示す. “エージェント (たち) 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, 自己ツッコミ条件では 20 %, 単純ツッコミ条件は 40 % であった.

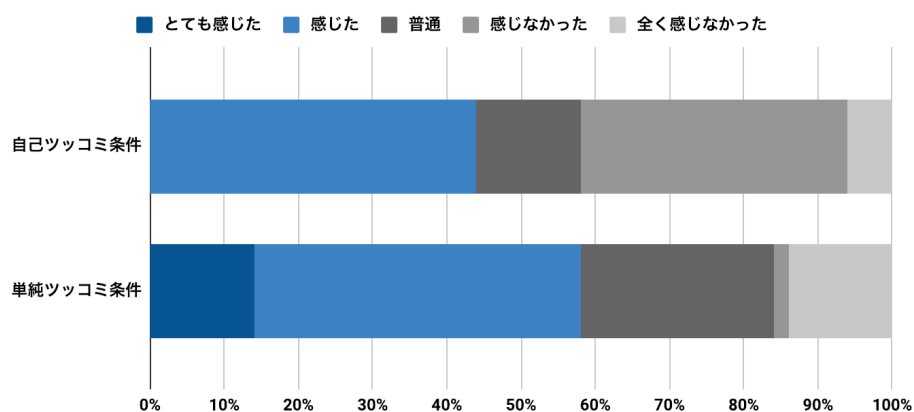


図 6.8: Q1:会話の流れは自然だと感じたか (N=10, 1 人あたり 5 回答)

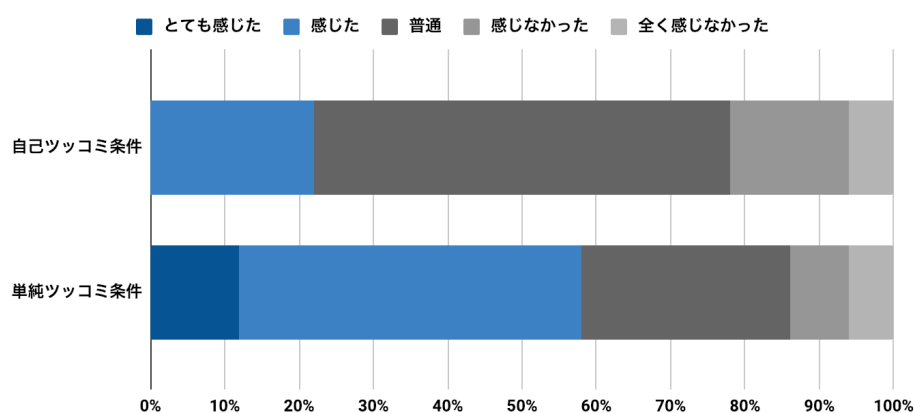


図 6.9: Q2:エージェントと苛立ちなく会話することができたか (N=10, 1 人あたり 5 回答)

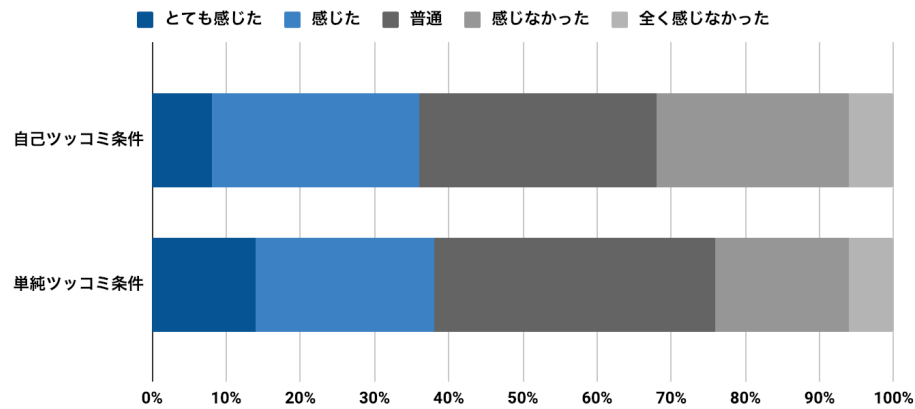


図 6.10: Q3:エージェントの発言にユーモアを感じたか (N=10, 1人あたり5回答)

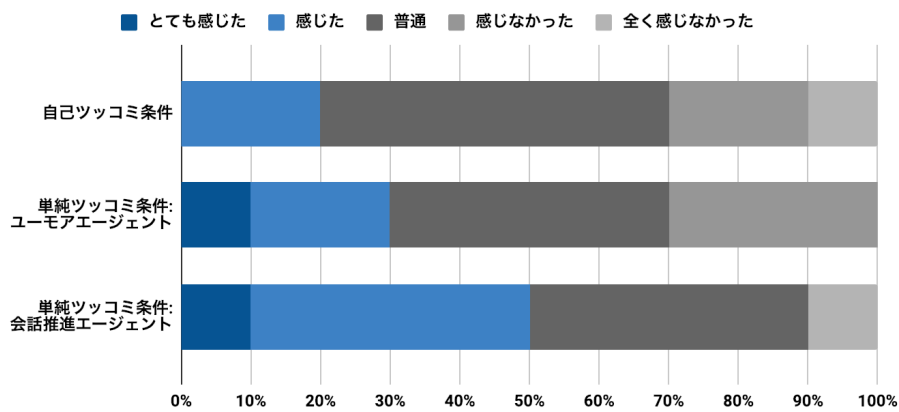


図 6.11: Q6:エージェント（たち）に信頼を感じたか (N=10, 1人あたり5回答)

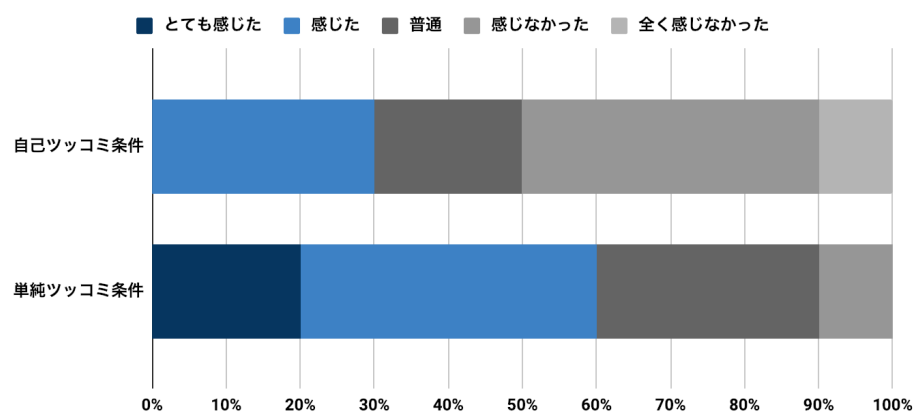


図 6.12: Q7:エージェント（たち）と負担なく会話できたか (N=10, 1人あたり5回答)

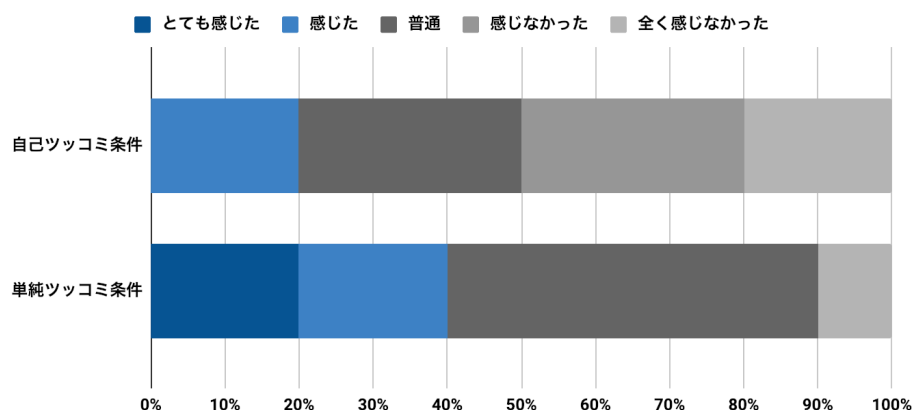


図 6.13: Q8:エージェント（たち）に親しみを感じたか（N=10）

Q1 への回答に対し、2つの条件間で 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、単純ツッコミ条件は、自己ツッコミ条件に寄せられた“一人芝居感が強い”や“自分で間違いに気づき不自然”といった意見を改善した条件であるため、会話の流れの自然さに、全く影響を及ぼさないことは考えにくい。今後さらに被験者を増やして追加実験を行い、検証をしていく。

Q2 への回答に対し、2つの条件間で 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、ユーモア発話を行ったエージェントとは別のエージェントがユーモア発話に対して指摘を行うことで、ユーザの苛立ちを軽減できることが示唆される。これは、ユーモア発話を行ったエージェントと別のエージェントが指摘を行うことで、一人芝居感がなくなり、苛立ちが軽減されたと考えられる。

Q3 への回答に対し、2つの条件間で 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。これは、ユーモア性と役割分担することには関連がないためだと思われる。また、本実験では実世界のエージェントを用意していないことが要因として考えられるため、今後検証を深めていく。

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のみ半数を超えている。ここから、役割分担された2体のエージェントを用いたとき、指摘を行うエージェントは、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のみをするためであると考えられる。

Q7 への回答に対し、2つの条件間で Wilcoxon の符号順位検定を行った結果、1%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、ユーザがユーモア発話に対して指

摘を行う必要がなくなったことに加え、本項の Q2 への考察で示した通り、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つの条件間で 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。これは、本項の Q2, Q7 への考察で示した通り、ユーザの苛立ちや負担が軽減されたことで、エージェントへの親しみが向上したと考えられる。

6.4.3 RQ3 の実験結果・考察

RQ3 は、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である。そのため、6.2 節で示した単純ツッコミ条件と咀嚼ツッコミ条件間で比較を行う。また、6.2 節で示した質問のうち、Q3～Q8 への回答を確認する。Q3～Q8 を表 6.5 に再掲する。

表 6.5: RQ3 : 実験の質問一覧

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

Q3 の回答結果を図 6.14 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、単純ツッコミ条件では 38 %，咀嚼ツッコミ条件では 74 % であった。Q4 の回答結果を図 6.15 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 64 %，咀嚼ツッコミ条件では 64 % であった。Q5 の回答結果を図 6.16 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 56 % であった。Q6 の回答結果を図 6.17 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、単純ツッコミ条件のユーモアエージェントに対しては 30 %，会話推進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 60 %，会話推進エージェントに対しては 70 % であった。Q7 の回答結果を図 6.18 に示す。“エージェントたち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、単純ツッコミ条

件は60%，咀嚼ツッコミ条件では70％であった．Q8の回答結果を図6.19に示す．“エージェントたち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，単純ツッコミ条件は40％，咀嚼ツッコミ条件では80％であった．

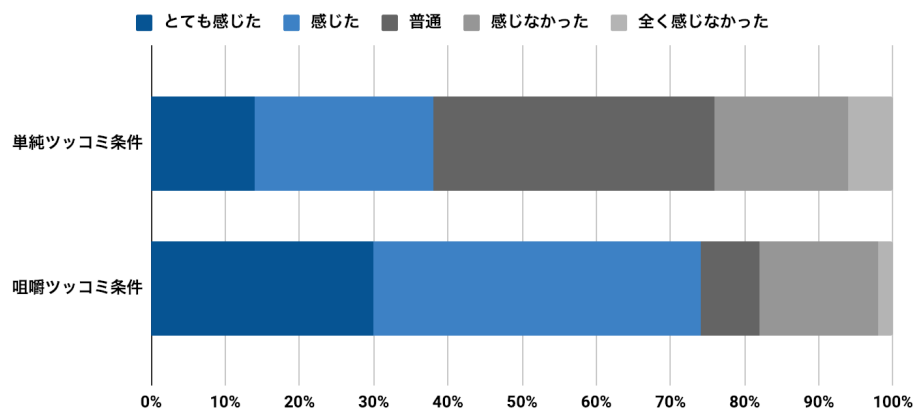


図 6.14: Q3:エージェントの発言にユーモアを感じたか (N=10, 1人あたり5回答)

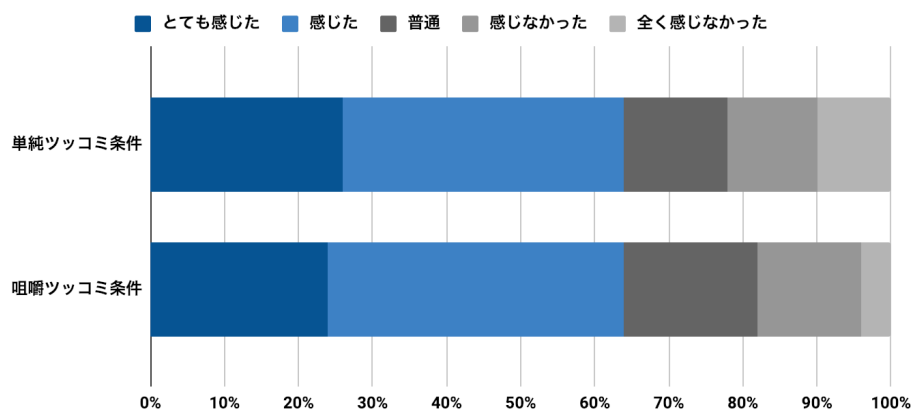


図 6.15: Q4:ツッコミ役エージェントがボケ役エージェントの発言内容を理解しているように感じたか (N=10, 1人あたり5回答)

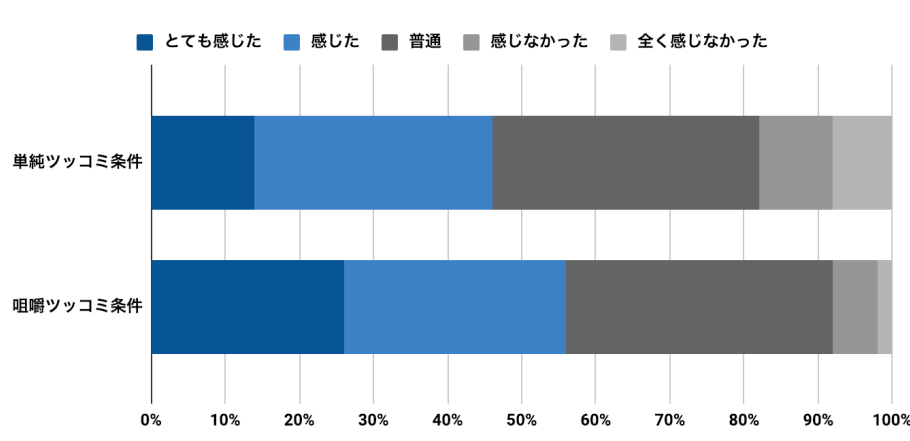


図 6.16: Q5:エージェントの会話に人間味を感じたか (N=10, 1人あたり5回答)

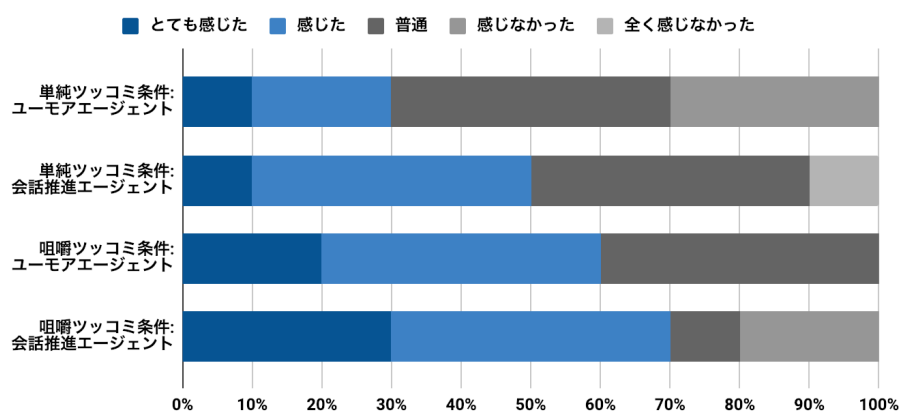


図 6.17: Q6:エージェントたちに信頼を感じたか (N=10, 1人あたり5回答)

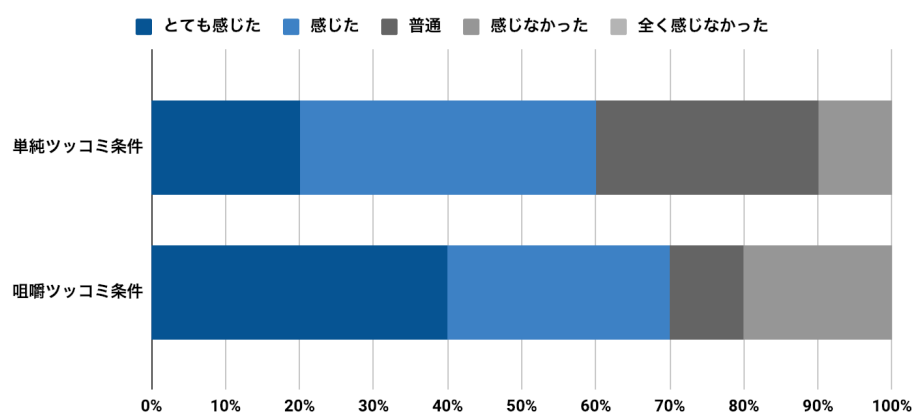


図 6.18: Q7:エージェントたちと負担なく会話できたか (N=10, 1人あたり5回答)

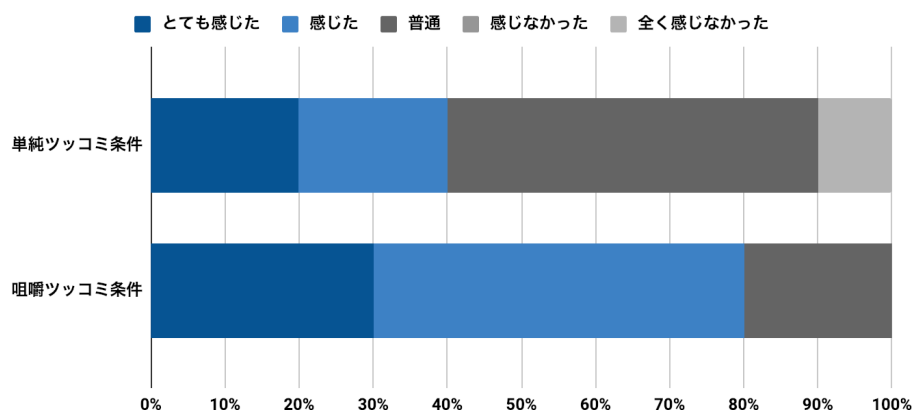


図 6.19: Q8:エージェントたちに親しみを感じたか (N=10)

Q3 への回答に対し、2つの条件間で Wilcoxon の符号順位検定を行った結果、10 %水準の有意傾向が見られた。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。また、被験者からの咀嚼ツコミ条件に対しての自由記述を見ると、“単語の意味を踏まえてツコミでいて良かった”、“もうひと押しあれば面白そう”といった肯定的な評価を得られた。今後は、実装の改善や被験者を増やして追加実験を行い、検証をしていく必要がある。

Q4, Q5 への回答に対し、2つの条件間で 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。被験者からの咀嚼ツコミ条件に対しての自由記述を見ると、“突拍子もないツコミを行うことがあった”という意見が見られ、実装の改善が必要であると考えられる。

Q6, Q7 への回答に対し、2つの条件間で 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つの条件間で 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、本項の Q3 への考察で示した通り、ユーモア発話を咀嚼した指摘を行うことは、人とエージェント間の親和的な関係の生成に必要なユーモアを与えやすい傾向がある。このため、被験者を増やして追加実験を行うことで、有意差を確認できる可能性があり、今後検証を行っていく必要がある。

第7章 結論

本稿は、ユーザと対話型エージェント間の対話シーンにおいて、先行研究である [31][32][33][34] のシステムがユーモア発話を行うと、会話の軌道が逸れるため、ユーザはユーモア発話を指摘し、会話の軌道修正を行う必要があるという問題の解消を狙ったものである。この問題を解決するために、本研究では、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した。この着想のもと、ユーモア発話を行うエージェントに加え、ユーザの代わりにユーモア発話を指摘し、会話の軌道修正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。これは、一方のエージェントがユーモア発話を行い、もう一方のエージェントが、ユーモア発話に対して、言葉の意味を理解しているような人間味のある指摘を行うシステムである。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感じやすさの向上が見られた。また、人間味のある指摘を行うことで、ユーザが感じるユーモア性が向上する傾向が見られた。

本研究の期待される活用法として、ユーザと良好な関係を築く必要がある場面での活用が期待される。例えば、将来的に家庭内にロボットが導入されたシーンにおいて、ユーザが持ちかけてきた雑談に対して、エージェントがユーモア発話を行うことで、ユーザにエージェントへの親しみを抱かせることができる。さらに、ユーモア発話に加えて、エージェントが会話推進を行うことで、ユーザの負担が軽減される。これらにより、ユーザはエージェントと良好な関係を築くことができ、家庭内においてもエージェントを導入しやすくなるだろう。

参考文献

- [1] D. McDuff and M. Czerwinski. Designing emotionally sentient agents. *Communications of the ACM*, Vol. 61, No. 12, pp. 74–83, 2018.
- [2] 井上宏. 「笑い学」研究について. 笑い学研究, No. 9, pp. 3–15, 2002.
- [3] Peterr Khooshabeh, Cade McCall, Sudeep Gandhe, Jonathan Gratch, and James Blascovich. Does it matter computer jokes. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pp. 77–86, 2011.
- [4] Ewa Luger and Abigail Sellen. Like having a really bad PA: the gulf between user expectation and experience of conversational agents. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 5286–5297, 2016.
- [5] Vera Q Liao, Muhammed Masud Hussain, Praveen Chandar, Matthew Davis, Marco Crasso, Dakuo Wang, Michael Muller, Sadat N Shami, and Werner Geyer. All work and no play? conversations with a question-and-answer chatbot in the wild. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI'18)*. ACM, New York, NY, USA, Vol. 13, 2018.
- [6] 武田風太, 鈴木慎太郎, 矢島敬士. 対話型キャラクターエージェントを用いたコミュニケーションによる高齢者の見守りシステム支援. 情報処理学会第78回全国大会 2016, 第1巻, pp. 401–402, 2016.
- [7] 速水達也, 佐野睦夫, 向井謙太郎, 神田智子, 宮脇健三郎, 笹間亮平, 山口智治, 山田敬嗣. 交替潜時と韻律情報に基づく会話同調制御方式と情報収集を目的とした会話エージェントへの実装. 情報処理学会論文誌, 第54巻, pp. 2109–2118, 2013.
- [8] 野中裕子, 酒井洋一, 安田清, 林佑樹, 中野有紀子. 語りかけエージェントとの会話における韻律情報の分析. 情報処理学会第75回全国大会講演論文集 2013, 第2013巻, pp. 179–180, 2013.
- [9] 小倉拓人, 徳永隼人, 松村冬子, 原田実. 進行役と鑑賞者のエージェントを用いた対話型鑑賞システム. じんもんこん 2015 論文集, 第2015巻, pp. 43–50, 2015.

- [10] 藤堂祐樹, 西村良太, 山本一公, 中川聖一. 複数の対話エージェントを用いた音声対話システムの分析と評価. 情報処理学会研究報告ヒューマンコンピュータインタラクション (HCI) , 第 2013-HCI-151 巻, pp. 1–7, 2013.
- [11] 片上大輔, 小林優, 大澤博隆, 稲葉通将, 篠田孝佑, 鳥海不二夫. 擬人化エージェントを用いた人狼対戦システムの開発. ゲームプログラミングワークショップ 2014 論文集, 第 2014 巻, pp. 133–134, 2014.
- [12] 松原和也, 林勇輔, 水久保勇記, 鶴見悠子, 垂水浩幸. 携帯電話向けインタラクティブエージェントと NPC への応用. 情報処理学会研究報告エンタテインメントコンピューティング (EC) , 第 2005 巻, pp. 71–78, 2005.
- [13] 久保田秀和, 黒橋禎夫, 西田豊明. 知識カードを用いた分身エージェント. 電子通信情報学会論文誌, No. 8, pp. 600–607, 2003.
- [14] 廣瀬有紹, 加藤誠巳. ドライバの退屈・眠気回避を目的とした会話型 CG エージェントに関する検討. 情報処理学会第 65 回全国大会講演論文集, 第 2003 巻, pp. 411–412, 2003.
- [15] 近藤亮太, 畑中聖二, 加藤誠巳. 余暇の過ごし方を提案する会話型エージェントシステムに関する検討. 情報処理学会第 69 回全国大会講演論文集, 第 2007 巻, pp. 543–544, 2007.
- [16] 中谷仁, 岡夏樹. ロボットの日常会話におけるユーモア生成の試み. 人口知能学会 2009 年全国大会論文集, 2009.
- [17] キムビンステッド, 滝澤修. 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.
- [18] O. Stock and C. Strapparava. Getting serious about the development of computational humor. In *International Joint Conference on Artificial Intelligence (IJCAI)*, Vol. 3, 2003.
- [19] Dongkeon Lee, Seung-Ho Han, Kyo-Joong Oh, and Ho-Jin Choi. A temporal community contexts based funny joke generation. In *Mobile Data Management (MDM)*, 2017.
- [20] 吉田裕介, 萩原将文. 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.
- [21] 竹越智也, 萩原将文. 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 —. Vol. 15, No. 1, pp. 47–54, 2016.

- [22] 真下遼, 梅谷智弘, 北村達也, 瀧本明代. 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. データ工学と情報マネジメントに関するフォーラム (DEIM Forum), pp. F4-4, 2015.
- [23] 青木哲, 義尚晃, 原口和貴, 梅谷智弘, 北村達也, 瀧本明代. 理解しやすい対話を用いた漫才台本の自動生成. データ工学と情報マネジメントに関するフォーラム (DEIM Forum), pp. C1-5, 2018.
- [24] Kotaro Hayashi, Takayuki Kanda, Takahiro Miyashita, Hiroshi Ishiguro, and Norihiro Hagita. Robot manzai: Robot conversation as a passive-social medium. In *International Journal of Humanoid Robotics*, Vol. 5, pp. 67-86, 2008.
- [25] 伊勢崎隆司, 小林明美, 有賀玲子, 山田智広. 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.
- [26] 藤倉将平, 小川義人, 菊池英明. 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.
- [27] Hans Wim Tinholt and Antron Nijholt. Computational humour: Utilizing cross-reference ambiguity for conversational jokes. In *International Workshop on Fuzzy Logic and Applications*, Vol. 4578 of LNCS, pp. 477-483, 2007.
- [28] Apple inc. : Siri, <https://www.apple.com/ios/siri/> (last visited on 2019/1/29).
- [29] Google inc. : Google assistant, <https://assistant.google.com/> (last visited on 2019/1/18).
- [30] Softbank. : Papper, <https://www.softbank.jp/robot/> (last visited on 2019/1/29).
- [31] 鈴木奨, 呉健朗, 瀧田航平, 堀越和, 中辻真, 宮田章裕. 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) , 第 2017-GN-102 巻, pp. 1-6, 2017.
- [32] 呉健朗, 鈴木奨, 瀧田航平, 中原涼太, 中辻真, 宮田章裕. 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, 第 22 巻, pp. 15-18, 2017.
- [33] 呉健朗, 鈴木奨, 瀧田航平, 堀越和, 中辻真, 宮田章裕. 対話型エージェントにおけるボケる返答機能の実装. 情報処理学会インタラクション 2018 予稿集, pp. 1015-1019.
- [34] 呉健朗, 中原涼太, 長岡大二, 宮田章裕. 文に対してボケて返す対話型エージェントの基礎検討. 日本バーチャルリアリティ学会 VR 学研報, 第 23 巻, pp. 13-17, 2018.

- [35] Pawel Dybala, Michal Ptaszynski, Rafal Rzepka, and Kenji Araki. Multi-humoroid: joking system that reacts with humor to humans' bad moods. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pp. 1433–1434, 2010.
- [36] 中原涼太, 長岡大二, 呉健朗, 大西俊輝, 柴田万里那, 宮田章裕. 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. ワークショップ 2018 (GN Workshop 2018) 論文集, 第 2018 巻, pp. 1–8, 2018.
- [37] A. P. Chaves and M. A. Gerosa. Single or multiple conversational agents? an interactional coherence comparison. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems.*, 2018.
- [38] Mecab: Yet another part-of-speech and morphological analyzer, <http://taku910.github.io/mecab/> (last visited on 2019/1/29).
- [39] Tomas Mikolov, Kai Chen, Greg Corred, and Jef frey Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*, 2013.
- [40] 情報通信研究機構. EDR 電子化辞書.

研究業績

査読付き論文誌

- (1) 呉健朗, 中原涼太, 長岡大二, 中辻真, 宮田章裕: ボケて返す対話型エージェント, 日本バーチャルリアリティ学会論文誌, Vol.23, No.4, 2018, pp.231-238 (2018 年 12 月).
-

査読付き国内会議

- (1) 中原涼太, 長岡大二, 呉健朗, 大西俊輝, 柴田万里那, 宮田章裕: 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, 情報処理学会グループウェアとネットワークサービスワークショップ 2018 論文集, pp1-8 (2018 年 11 月).
-

研究会・シンポジウム

- (1) 柴田万里那, 大西俊輝, 呉健朗, 長岡大二, 中原涼太, 宮田章裕: 柔らかい物体の動きによる共感表現方法の基礎検討, 情報処理学会インタラクション 2019, (2019 年 3 月発表予定).
- (2) Genki Kurita, Ryota Nakahara, Daiji Nagaoka, Akihiro Miyata: A Basic Study of a Conversational Agent Speaking with Ambiguity, Kanagawa Institute of Technology International Symposium 2018 (2018 年 9 月).
- (3) 長岡大二, 中原涼太, 呉健朗, 鈴木奨, 宮田章裕: 文脈を考慮してボケる対話型エージェントの実装と評価, 情報処理学会シンポジウム論文集, マルチメディア、分散、協調とモバイル (DICOMO2018), Vol.2018, pp.627-632 (2018 年 7 月).
- (4) 呉健朗, 中原涼太, 長岡大二, 宮田章裕: 文に対してボケて返す対話型エージェントの基礎検討, 日本バーチャルリアリティ学会 VR 学研報, Vol.23, No.CS-2, pp13-17 (2018 年 6 月).
- (5) 長岡大二, 中原涼太, 小林舞子, 鈴木奨, 呉健朗, 宮田章裕: 文脈を考慮してボケる対話型エージェントの基礎検討, 情報処理学会インタラクション 2018 予稿集, pp.882-884 (2018 年 3 月).
- (6) 呉健朗, 鈴木奨, 瀧田航平, 中原涼太, 中辻真, 宮田章裕: 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, pp15-18 (2017 年 12 月).
-

メディア掲載

- (1) 笑いがわかる、ボケる AI, TBS ラジオ THE FROGMAN SHOW A.I. 共存ラジオ 好奇心家族, <https://www.tbsradio.jp/225340> (2018 年 2 月 12 日).
 - (2) AI、笑いを学ぶ, 国際イベントニュース, 第 25 号 16 面 (2017 年 10 月 10 日).
 - (3) 「電話番号教えて」「え、ケンカ番長？」日大生が作った“ボケる AI”, ITmedia, <http://www.itmedia.co.jp/news/articles/1709/21/news093.html> (2017 年 9 月 21 日)
 - (4) 「ボケる AI」初公開。スゴイ技術で、あえて間違える【TGS2017】, 日経トレンドイネット, <http://trendy.nikkeibp.co.jp/atcl/pickup/15/1003590/092101200/> (2017 年 9 月 21 日).
-

展示

- (1) ボケる AI, 東京ゲームショウ 2017 (2017 年 9 月 21～22 日).
-

受賞

- (1) 情報処理学会グループウェアとネットワークサービスワークショップ 2018 ベストペーパー賞, 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, 受賞者: 中原涼太, 長岡大二, 呉健朗, 大西俊輝, 柴田万里那, 宮田章裕: (2018 年 11 月).
 - (2) 情報処理学会グループウェアとネットワークサービスワークショップ 2018 ベストプレゼンテーション賞, 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, 受賞者: 中原涼太(2018 年 11 月).
 - (3) 第 20 回シンポジウム サイバースペースと仮想都市研究会シンポジウム, 2017 年サイバースペース賞, ボケて返す対話型エージェントの実装, 受賞者: 呉健朗, 鈴木奨, 瀧田航平, 中原涼太, 中辻真, 宮田章裕 (2018 年 6 月).
-