

英語でボケる発言を行うエージェントの基礎検討

得田 舜介¹ 呉 健朗² 田中 柊羽¹ 大西 俊輝² 大串 旭¹ 宮田 章裕^{1,a)}

概要: ユーモアを生成する研究は様々な方法を用いて行われている。我々は、ユーザの発言中の単語を、その単語から発音が近く、意味が遠く、認知度が高い単語に聞き間違えるボケをするエージェントを提案してきた。しかしこの先行研究では入出力が日本語に限定されていたため、日本語を理解できない人は、エージェントのユーモアのある発言を理解することができなかった。この問題を解決するために、システムを他言語に適応する方法について検討を行う。具体的には、初期検討として、ユーザの入力が英語だった場合でも、エージェントがユーザの発言中の単語を、その単語から発音が近く、意味が遠く、認知度が高い単語に聞き間違えてボケられるようにする方法を検討する。本稿では、この検討内容について報告を行う。

1. はじめに

近年、我々の生活空間の中に多くの対話型エージェントの普及が進んでいる [1]。Apple の Siri [2]、Google Assistant [3]、Amazon の Alexa [4] などが搭載された製品を家庭内で利用するユーザも増えている。対話型エージェントは団楽・介護などの多くのシーンでの活躍が期待されており、ユーザと対話型エージェントのコミュニケーションは重要なものとなっている。多くのシーンで対話型エージェントが活躍するためには、ユーザが対話型エージェントに対して親しみやすさを感じられるようになることが重要であると考えられる。ユーザが対話型エージェントに対して親しみやすさを感じやすくするためには、対話型エージェントにユーモア性をもたせることが重要であると考えられている [5][6][7]。対話型エージェントにユーモア性をもたせるという研究は様々な方法を用いて行われている。我々はこれまで、ユーザの発言に対して 2 体のエージェントがボケとツッコミというユーモア [8] に基づいて、リアルタイムにユーモアのある発言を返す研究に取り組んできた [9][10][11][12][13]。先行研究では、ユーザがユーモアのある発言をするエージェントに対して親しみやすさを感じ、対話継続欲求が向上することを明らかにした。しかし、先行研究でのエージェントに対する入出力は日本語を前提としており、日本語を理解できないユーザはエージェントのユーモア発言を理解することができなかった。

本稿では、この問題を解決するために、先行研究のユーモア手法を英語に適応させる検討を行ったことについて報

告を行う。

本稿の貢献は、入出力が英語である際に、ユーザの発言を聞き間違えるというユーモアを生成する実装方法について検討したことである。

2. 関連研究

2.1 ユーモアに関する研究

学術的にユーモアに関する研究は、哲学、心理学、言語学など様々な観点から行われている [14]。Kim らは、ユーモアの生成と理解をコンピュータ上でモデル化することは、人がどのように言語解釈をしているかを解明するための重要な要素であると述べている [15]。Holmes は、職場におけるユーモアの検証より、心身の健康や職務遂行能力の自己評価に対して良い効果をもたらすことを明らかにしている [16]。これは人同士の関係だけでなく、人とエージェント間の関係においてもユーモアは親和的な関係の形成に有効であることが分かっている [5][6][7]。

2.2 ユーモア生成に関する研究

ユーモアを生成する研究として、画像とそれに対するユーモアタイトル 1 つ以上をデータセットとして機械学習を行うことで画像にユーモアのあるタイトルをつける研究 [17] や、文中の各単語の頭文字のアルファベットを連結することでユーモアのある省略語を作成する研究 [18] がある。また漫才形式のユーモアを生成している研究として、Web ニュース記事から自動的に漫才台本を作成しロボットに披露させる研究 [19] や、アプリ上で Web ニュース記事から自動的に作成された漫才をエージェントに行わせる研究 [20]、入力された文章から漫才形式の対話を自動生成

¹ 日本大学文理学部

² 日本大学大学院総合基礎科学研究科

a) miyata.akihiro@acm.org

している研究 [21] がある。

対話中にユーモアを生成する研究としては、会話中の文中の代名詞が指す対象の語と異なる対象の語を発見することでユーモアを生成する研究 [22] や、ユーザの発言中の語彙を異なる用法で使う駄洒落文を生成する研究 [23] がある。ユーザの発話文をシステムが擬似的に聞き間違えることで、ユーモア表現を含んだ聞き返し文を生成する方法を提案している研究もある。我々の先行研究では、エージェントが不適合解決モデル [24][25] に基づいたユーモアの生成を行ってきた。この手法では、ユーザの発言に対してエージェントがユーザの発言を聞き間違えることでユーモアを生成している。具体的には、ユーザの発言内から選定する名詞を、その名詞に対して発音が近く、意味が遠く、認知度が高い名詞に聞き間違える。

3. 研究課題

対話型エージェントに対して親しみやすさを感じやすくするために、対話型エージェントにユーモア性をもたせることが重要であると考えられている。我々は、ユーザの発言に対し、発言中の特定の単語を、その単語から意味が遠く、発音が近く、認知度の高い単語に言い換えるようなボケをするエージェントを提案してきた [9][10][11][12][13]。しかし、これらの先行研究は入出力が日本語を前提としていたため、他言語に適用できず、日本語を理解できないユーザはエージェントのユーモア発言を理解することができない問題がある。

先行研究では、単語中の文字の母音の違いを指標にして単語間の発音の近さを算出していた。これは、日本語では“ん”を除く全ての文字について発声時の音が a, i, u, e, o のいずれかの母音で終わることから、日本語の発音は母音に依存すると考えたためである。しかし、英語の単語を発声する際には、単語を構成している各文字の発音が母音で終わるとは限らないことから、英語の発音は母音に依存するとは限らないと考えられる。このことから、先行研究の、単語中の文字の母音の違いを指標にする実装方法では、任意の英単語に対して発音の近い単語を探すことができないという問題がある。

この問題を踏まえ、本稿では、先行研究のユーモア生成方法を英語に適応することを研究課題として設定する。

4. 検討

本稿では、我々の先行研究である、ユーザの発言に対してエージェントがボケる発言を行うまでの入出力を英語にするための検討を行う。具体的にはユーザの発言した英文内の英単語に対して、発音が近く、意味が遠く、認知度の高い英単語を見つけ、ユーザの発言を聞き間違えたかのように出力する実装方法の検討をする。ユーザの発言内の英単語を置換元単語とし、聞き間違える際に使用する英単語

をボケ単語とする。本研究で想定される対話例を3つ挙げる。

対話例 1

ユーザ I like beer.

エージェント What? I like bear?

対話例 2

ユーザ You are funny robot.

エージェント What? You are funny rabbit?

対話例 3

ユーザ I have been to the desert.

エージェント What? I have been to the dessert?

4.1 先行研究との差異

実装は、下記の3段階に分けて行う。

- 発音が近い単語を見つけるための実装
- 意味が遠い単語を見つけるための実装
- 認知度の高い単語を見つけるための実装

前述のとおり、日本語と英語の発音には違いがあるため、発音が近い単語を見つけるためのアルゴリズムは先行研究と同じものは使えない。入出力が日本語であった先行研究では、比較する単語の母音を文字列にし、その文字列間の編集距離を発音の近さとして測っていた。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して定義される。3章より、入出力を英語にする本研究では、先行研究の実装方法では発音の近さを算出できない。そこで英単語の発音の近さを測る方法として、英単語をその単語の発音を表す文字列に変換する Metaphone[26] を利用する。Metaphone とは、単語を、その単語の子音と発音記号を考慮して、単語の発音を表す文字列に変換するアルゴリズムである。ここで Metaphone の出力例をいくつか挙げる。

- BEER → BR
- BEAR → BR
- ROBOT → RBT
- RABBIT → RBT
- DESERT → TSRT
- DESSERT → TSRT

4.2 事前準備

事前準備として、2つのデータベースを構築する。これらのデータベースは、Redis[27] を使用して構築する。

1つ目のデータベースは発音の近い単語を選ぶためのデータベースである。以降、発音 DB とする。発音 DB は、英語 Wikipedia 記事全文の英単語を使って構築されたセット型のデータベースである。発音 DB の key は Metaphone コード、value は Metaphone で変換した際に key になる単語である。

2つ目のデータベースは認知度の高さを測るためのデータベースである。以降、認知度 DB とする。認知度 DB は、英語 Wikipedia 記事全文の英単語を使って構築された文字列型のデータベースである。認知度 DB の key は英単語、value は key の英単語が英語 Wikipedia 記事全文中で出現する回数である。出現する回数を数える際に単語の活用形は考慮せず、同一単語として数える。この際の活用形の判定には NLTK[28] を用いている。

4.3 入力

ユーザがエージェントに対してテキストで発言を行う。ここで入力された文章からユーモアを生成する。

4.4 ボケ単語の生成方法

ボケ単語は先行研究と同じく、ユーザの発言内のある単語に対して、発音が近く、意味が遠く、認知度の高い単語とする。しかし、今回入出力を英語に対応させるためにアルゴリズムは異なっている。

4.4.1 置換元単語の決定

先行研究では、ユーザの入力文中の名詞の中で文のトピックに、最も近いと算出された名詞を置換元単語としていた。しかし、本研究では先行研究を英語に対応する初期検討ということで、ユーザの入力文の最後の名詞単語を置換元単語とする。I や You などの主語が置換元単語となることを避けるため、文中の最後の名詞を置換元単語とした。

4.4.2 ボケ単語の決定

置換元単語からボケ単語を生成する方法について述べる。ボケ単語は、置換元単語に対して、発音が近く、意味が遠く、認知度の高い英単語である。

まず、置換元単語を Metaphone で変換し、置換元単語の発音を表す文字列を取得する。この文字列を発音 DB で検索することで、Metaphone で変換した際に置換元単語の発音を表す文字列と同じになる単語群を取得することができる。取得した全ての単語に対して、NLTK と認知度 DB を用いて、各単語の品詞と英語 Wikipedia 記事全文内での出現回数を求める。このとき、単語が名詞かつ出現回数が一定回数以上の単語をボケ単語の候補とする。出現回数の下限は実際にアルゴリズムを実行して決定する。次に、置換元単語から意味の遠い単語を選ぶために、Word2Vec[29] を利用する。Word2Vec により求められる値から概念距離を求め、ボケ単語の候補全てと置換元単語の概念距離を測る。この際に概念距離が離れている単語を意味の遠い単語とする。

取得したボケ単語の候補それぞれの出現回数と概念距離を用いて、認知度を表すスコア (s_f)、意味の遠さを表すスコア (s_s) を算出する。まず s_f を算出する方法について述べる。認知度 DB によって求められた単語の出現回数はべき分布に従うので、極端に出現回数が多いとボケ単語を

決定する際に影響が大きくなってしまう。そこで出現回数の対数をとったものを s_f とする。 s_f は下記のように算出する。

$$s_f = \log f \quad (1)$$

次に s_s を算出する方法について述べる。Word2Vec によって求められた単語間の類似度は、値が大きいほど意味が似ていて、値が小さいほど意味が似ていないということを表している。本稿では、意味の遠い単語を求めたいので、置換元単語とボケ単語候補の類似度 $u_{(k)}$ の逆数を用いて、 s_s を算出する。逆数を用いることで、 s_s の値が大きい単語が意味の遠い単語として求められる。 s_s は下記のように算出する。

$$s_s = \frac{1}{n} \sum_{k=1}^n \frac{1}{1 + u_{(k)}} \quad (2)$$

上記で求めた s_f と s_s を、最小値 0.0、最大値 1.0 となるように正規化をする。正規化を行うことで、 s_f 、 s_s の最終的なスコアを算出する際の影響を均等にしている。正規化された s_f と s_s を用いて、ボケ単語の候補ごとの最終的なスコア (s_h) を求め、 s_h が最も高くなる単語をボケ単語とする。 s_f 、 s_s へ重み付け・合算を行い、 s_h を算出する。 s_h は下記のように算出する。

$$s_h = w_f s_f + w_s s_s \quad (3)$$

w_f 、 w_s は重み係数であり、実際にアルゴリズムを実行して値の大きさを決定する。

4.5 出力

得られたボケ単語をユーザの入力文に対して、聞き間違えたように出力をする。出力内容は、”What? WORD(入力文の置換元単語をボケ単語に変更したもの)?” と出力することで、聞き間違えていることを表す。

5. おわりに

本稿では、先行研究で行ってきたユーザの発言に対して、聞き間違えるというボケを行うエージェントの実装を、どのように英語に適応させるかを検討した。その結果、入出力が日本語である先行研究の実装方法から、発音の近い単語を探すための実装を変える必要があることがわかった。英単語を Metaphone を利用して発音を表す文字列に変換し、事前にデータベースを作ることを実装した。本稿で、実装したアルゴリズムを用いると、ユーザの入力が英語だった場合でも、エージェントが、ユーザの発言中の単語を、その単語から発音が近く、意味が遠く、認知度の高い単語に聞き間違えてボケることが可能であると考えられる。

今後の課題として、英語に対応させたときに、入出力が日本語であった先行研究と同等のユーモア性がみられるか

の検証を行う。加えて、先行研究では実装されていたツッコミを英語に対応させる予定である。

参考文献

- [1] McDuff, D. and Czerwinski, M.: Designing emotionally sentient agents, In Communications of the ACM, Vol.61, pp.74–83 (2018).
- [2] Apple Inc. : Siri, <https://www.apple.com/ios/siri/> (last visited on 2021/12/18).
- [3] Google LLC.: Google assistant, <https://assistant.google.com/> (last visited on 2021/12/18).
- [4] Amazon Inc.: Alexa, <https://www.amazon.co.jp/meet-alexa/>(last visited on 2021/12/18).
- [5] Khooshabeh, P., McCall, G.C.S., Gratch, J. and Blascovich, J.: Does it matter computer jokes, In Proc. International Conference on Human Factors in Computing Systems (CHI '11), pp.77–86 (2011).
- [6] Lugar, E., and Sellen, A.: Like Having a Really Bad PA:The Gulf between User Expectation and Experience of Conversational Agents, In Proc. International Conference on Human Factors in Computing Systems (CHI '16), pp.5286–5297(2016).
- [7] Liao, Q.V., Hussain, M.M., Chandar, P., et al.: All Work and no Play? Conversations with a Questionand-Answer Chatbot in the Wild?, In Proc. International Conference on Human Factors in Computing Systems (CHI '18), pp.1–13 (2018).
- [8] Tsutsumi, H.: Conversation Analysis of Boke-tsukkomi Exchange in Japanese Comedy, New Voices, Vol.5, pp.147-173(2011).
- [9] 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, pp.231–238 (2018).
- [10] 呉 健朗, 長岡大二, 中原涼太, 宮田章裕: 文のトピックを考慮した単語置換によるユーモア発話を行う対話型エージェント, 情報処理学会論文誌, Vol.61, No.1, pp.113–122 (2020).
- [11] 呉 健朗, 富永詩音, 武藤佑太, 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, 情報処理学会論文誌, Vol.61, No.8, pp.1353–1362 (2020).
- [12] 呉 健朗, 大西俊輝, 大串旭, 宮田章裕: ノリツッコミを行う対話型エージェント, 情報処理学会インタラクシオン 2022 論文集 (2022)
- [13] Go, K., Onishi, T., Ogushi, A. and Miyata, A.: Conversational Agents Replying with a Manzai-style Joke, Proc. 33rd Australian Conference on Human-computer-interaction (OzCHI '21) (2021)
- [14] Marta, D.: Beyond a Joke: Types of Conversational Humour, Language and Linguistics Compass, Volume3, pp.1284–1299 (2009).
- [15] Kim, B., Benjamin, B., Seana, C., et al.: Computational Humor, IEEE Intelligent Systems, Vol.21, pp.59–69 (2006).
- [16] Holmes, J.: Sharing a laugh: Pragmatic aspects of humor and gender in the workplace, Journal of Pragmatics, Vol.38, pp.26–50 (2006).
- [17] Lee, D., Han, S., Oh, K. and Choi, H.: A temporal community contexts based funny joke generation., In Mobile Data Management (2017).
- [18] Stock, O. and Strapparava, C.: Getting serious about the development of computational humor, In International Joint Conference on Artificial Intelligence, pp.59–64 (2003).
- [19] Umetani, T., Aoki, S., Akiyama, K., et al.: Scalable Component-Based Manzai Robots as Automated Funny Content Generators, Journal of Robotics and Mechatronics, Vol.28, No.6, pp.862–869(2016).
- [20] Haraguchi, K., Yane, K., Sato, A., et al.: Chat-type Manzai application: mobile daily comedy presentations based on automatically generated manzai scenarios, Proc. 18th International Conference on Advances in Mobile Computing & Multimedia, pp.143–152(2020)
- [21] 吉田 裕介, 萩原 将文: 漫才形式の対話文自動生成システム, 第 13 回日本感性工学大会 (2011).
- [22] Tinholt, W.H., and Nijholt, A.: Computational humour:Utilizing crossreference ambiguity for conversational jokes, In International Workshop on Fuzzy Logic and Applications, Vol.4578, pp.477–483 (2007).
- [23] Dybala, P., Ptaszynski, M., Maciejewski, J., et al.: Multiagent system for joke generation: Humor and emotions combined in human-agent conversation, Journal of Ambient Intelligence and Smart Environments, Vol.2, pp.31–48 (2010).
- [24] Coulson, S. and Williams, R.: Hemispheric asymmetries and joke comprehension, Neuropsychologia, Vol. 43, No. 1, pp. 128–141 (2005).
- [25] Samson, C. A., Hempelmann, F.C., Huber,O. and Zysset, S.: Neural substrates of incongruity-resolution and nonsense humor, Neuropsychologia, Vol. 47, No. 4, pp.1023–1033 (2009).
- [26] Metaphone:
<https://docs.oracle.com/cd/B72202.01/Content/processor.library/transformation/metaphone.htm> (last visited on 2021/12/19).
- [27] Redis: Redis | AWS, <https://aws.amazon.com/jp/redis/> (last visited on 2021/12/19).
- [28] NLTK: Natural Language Toolkit, <https://www.nltk.org> (last visited on 2021/12/19).
- [29] Mikolov,T., Chen,K., Corrad,G. and Dean,J.: Efficient estimation of word representations in vector space, In Proc. ICLR Workshops Track (2013).