

トリビア文および雑学文を利用する対話システムの構築

新名 和也^{1,a)} 嶋田 和孝^{2,b)}

概要: 近年雑談対話システムに関する研究が盛んに行われており、ニューラルネットワークや強化学習を用いた手法が一定の成果を挙げている。しかし、これらの手法は「そうですね」のような文を生成しやすいため、ユーザが対話に飽きる危険性がある。よって、より良い対話システムを構築するためには、ユーザを対話に引き込む文が必要となる。本研究ではそのような文として、トリビアを含んだ文（トリビア文）および雑学を含んだ文（雑学文）に注目する。これらの文は人の興味を引く性質があるため、対話システムに用いることでユーザを対話に引き込む効果が期待できる。本研究では、このようなトリビア文および雑学文を利用する対話システムを構築する。

A Dialogue System Using Trivia and Unusual Fact Sentences

NIINA KAZUYA^{1,a)} SHIMADA KAZUTAKA^{2,b)}

Abstract: Dialogue systems have been increasingly important these days. In particular, non-task-oriented dialogue systems have been studied because of the success of neural network approaches. However, these models tend to generate simple responses such as "yes". To construct a dialogue system that holds user's attention continuously, we need to use utterances that capture the interest of the user. In this paper, we focus on the trivia and unusual fact for the purpose. These information perhaps adds a surprise to users. Therefore, trivia and unusual fact are beneficial for dialogue systems. We construct a dialogue system using trivia and unusual fact sentences.

1. はじめに

近年、Apple の Siri や Google アシスタントなど、対話システムに対して注目が集まっている。対話システムは、ある特定のタスクの遂行を目的としたタスク指向型と、対話自体を目的とした非タスク指向型の2つに大きく分類される。このうち、非タスク指向型については近年盛んに研究が行われており、Sequence-to-Sequence (Seq2Seq) モデルを用いた手法や強化学習を用いて対話戦略を獲得する手法が一定の成果を挙げている [1][2]。しかし、これらの手法では汎用的な文（「はい」や「そうですね」など、文

脈に依存せず使用できる文）を生成しやすい傾向があるため、ユーザが対話に飽きてしまう危険性がある。ユーザを対話に飽きさせないためには、ユーザを楽しませる文を利用して対話に引き込む必要がある。

本研究では、ユーザを楽しませる文として、トリビアを含む文（トリビア文）および雑学を含む文（雑学文）に着目する。トリビア文と雑学文は、それぞれ「ある物事についての瑣末な知識であり、かつ人の興味を引くような内容を含む文」および「ある物事についての瑣末な知識やあまり知られていない知識を含む文」と定義する。例えば、トリビア文は「アメリカザリガニにサバをあげると青くなる」のような文であり、雑学文は「アメリカザリガニは体色が赤いことからマッカチンという別名もあるが、色素変異などが原因により青色や白色をしている個体もいる」のような文である。このようなトリビア文や雑学文は、人の興味を引くことや知的好奇心を駆り立てることができる他、話題展開の方法としても利用できるため、対話を弾ませるこ

¹ 九州工業大学大学院 情報工学府 先端情報工学専攻
Graduate School of Computer Science and System Engineering,
Kyushu Institute of Technology 680-4, Kawazu, Iizuka,
Fukuoka, 820-8502 Japan

² 九州工業大学 大学院情報工学研究院 知能情報工学研究系
Department of Artificial Intelligence, Kyushu Institute of
Technology 680-4, Kawazu, Iizuka, Fukuoka, 820-8502 Japan

a) k.niina@pluto.ai.kyutech.ac.jp

b) shimada@pluto.ai.kyutech.ac.jp

とができる。よって、トリビア文および雑学文を対話システムに組み込むことで、より楽しい対話を行うことができると考えられる。

これまで我々は、Web上の文書データを対象にトリビア文および雑学文抽出を試みた [3][4][5]。トリビア文については、Webから収集したトリビアデータと Wikipedia から獲得したトリビア文でない文（非トリビア文）から、人手で設計した素性と機械学習を用いて、トリビア文か非トリビア文かの2値分類およびトリビア文の「面白さ」の測定を試みた。雑学文については、IDF、単語の共起頻度、ルールの3つの基準によるスコアリングを用いて、Wikipediaからの雑学文抽出を試みた。

このような背景を踏まえ、本研究では、トリビア文および雑学文を利用する対話システムを構築する。具体的には、ユーザの入力文に関連するトリビア文および雑学文に対して、出力として相応しいかどうかの判定を行うような対話システムを構築する。

2. 関連研究

ユーザを対話に引き込む非タスク指向型対話システムに関する研究として、稲葉ら [6] の研究がある。稲葉らは、Twitterから独語文として利用できるツイートを抽出した後、意外性、ユーモア性、トピックにおける特殊性の3つの観点からスコアを計算し、順位づけを行っている。また、他の研究として江頭ら [7] の研究がある。江頭らは、Web上のニュース記事と Wikipedia の記事を知識データとして、Markov Decision Process (MDP) を用いた対話システムを構築している。挨拶や話題提示などの各応答文生成モジュールを作成し、MDPにより獲得した対話戦略を用いて対話を行っている。

トリビアに関する研究として、Prakashら [8] の研究がある。Prakashらは、IMDBに掲載されているトリビアを教師付きデータセットとして扱い、固有表現や Superlative Words などの素性と Rank SVM を用いて、文の「トリビアらしさ」をランキングしている。また、他の研究として Fatmaら [9] の研究がある。Fatmaらは、DBpedia^{*1} に存在する RDF トリプルデータを対象に、人手で作成した教師付きデータセットと CNN を用いて、ある文がトリビアであるかどうかの判定を行っている。具体的には、畳み込みで得られた特徴量と人手で設計した素性を組み合わせて学習する Fusion Based CNN を利用して、トリビアであるかどうかの2値分類を行っている。機械学習による分類ではなく、スコアリングによってトリビアを獲得する研究として、Tsirelら [10] の研究がある。Tsirelらは、Wikipediaの記事カテゴリに注目し、意外なカテゴリ（例えば、「バラク・オバマ」に対する「グラミー賞受賞者」）

^{*1} <https://wiki.dbpedia.org/>

表 1 トリビア文データセットの例

Table 1 Examples of trivia sentences

トリビア	「へえ」数	最大「へえ」数
アフロの仏像がある	77	100
入れ歯は昔木で出来ていた	83	100
比内鶏を食べると逮捕される	55	100
ゴリラの血液型は全て B 型である	110	200

がトリビアであるとして、記事とカテゴリ間の類似度やカテゴリの凝集性に基づいたスコアリングによってトリビアを獲得している。

トリビア文は、意外性のある内容を含む。対話システムへの利用を前提に意外性のある文を獲得する研究として、太田ら [11] の研究がある。太田らは、Wikipediaの記事中の文に対して、TF-IDF、単語の共起頻度、文長の3つの指標に基づいたスコアとルールを用いて意外性のある文を抽出している。また、他の研究として杉本ら [12] の研究がある。杉本らは、Wikipediaの記事中の文に対して、対話システムに利用できる文であるかどうかの記事内における位置情報を用いて分類している。文字数や特定単語を用いたフィルタリングなどを行った後、位置情報に基づいた素性を用いて分類している。

3. データセット

本研究では、対話システムに利用するトリビア文および雑学文を Web上の文書データから獲得する。3.1節では、トリビア文のデータセットについて説明する。3.2節では、雑学文のデータセットについて説明する。

3.1 トリビア文データセット

対話システムに利用するトリビア文のデータセットは、「トリビアの泉 ～素晴らしきムダ知識～」^{*2}というタイトルの Wikipediaの記事から収集する。「トリビアの泉 ～素晴らしきムダ知識～」とは、フジテレビ系列で2002年から2012年にかけて放送された、視聴者が投稿したトリビアを紹介する番組である。この番組について書かれた Wikipediaの記事には、番組内で紹介されたトリビアが記載されているため、これをデータセットとして利用する。

データセットの例を表 1 に示す。表 1 のように、各トリビア文には「へえ」数が付随している。「へえ」数とは、番組に出演していた芸能人が、そのトリビア文に対して行った評価であり、1人最大20「へえ」までつけることができる。また、最大「へえ」数とは、そのトリビア文が紹介された放送回の中で取る事のできる最大の「へえ」数を表す。

^{*2} https://www.fujitv.co.jp/b_hp/trivia/

表 2 雑学文データセットの例

Table 2 Examples of unusual fact sentences

記事名	雑学文	雑学スコア
餃子	敦煌の唐代の墳墓では、副葬品として壺に入った餃子が乾燥状態で発見されている	4.45
キリン	鳴き声は牛にやや似た声で「モー」と鳴く	5.03

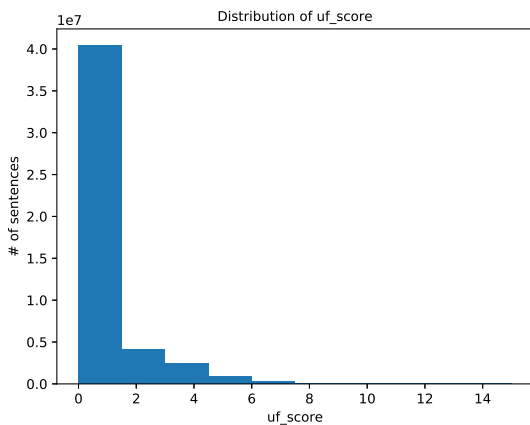


図 1 uf_score の分布

Fig. 1 The distribution of uf_score

3.2 雑学文データセット

対話システムに利用するトリビア文のデータセットは、[3]の手法を用いて獲得する。データセットの例を表 2 に示す。表 2 のように、各雑学文には [3] の手法により計算された雑学スコアが付随している。雑学スコアは文中の名詞を対象として、IDF、単語の共起頻度および独自に設定したルールに基づいて計算される。本研究では、雑学スコアの値が閾値以上の文を雑学文として獲得する。閾値の決定について、図 1 を用いて説明する。図 1 は雑学スコアのヒストグラムである。図 1 より、雑学スコアが 3 未満の文が全体の約 90% を占めていることが分かる。よって、雑学スコアが 3 以上の文は雑学文の可能性が高いといえる。したがって、本研究では、雑学スコアが 3 以上の文を雑学文として対話システムに利用する。

4. 対話システム

本研究では、トリビア文および雑学文を利用する対話システムを構築する。対話システムの概要図を図 2 に示す。図 2 のように、対話システムはトリビア・雑学文検索部、出力制御部、出力優先度計算部、雑談応答生成部の 4 つで構成される。4.1 節では、トリビア・雑学文検索部について説明する。4.2 節では、出力制御部について説明する。4.3 節では、出力優先度計算部について説明する。4.4 節では、雑談応答生成部について説明する。

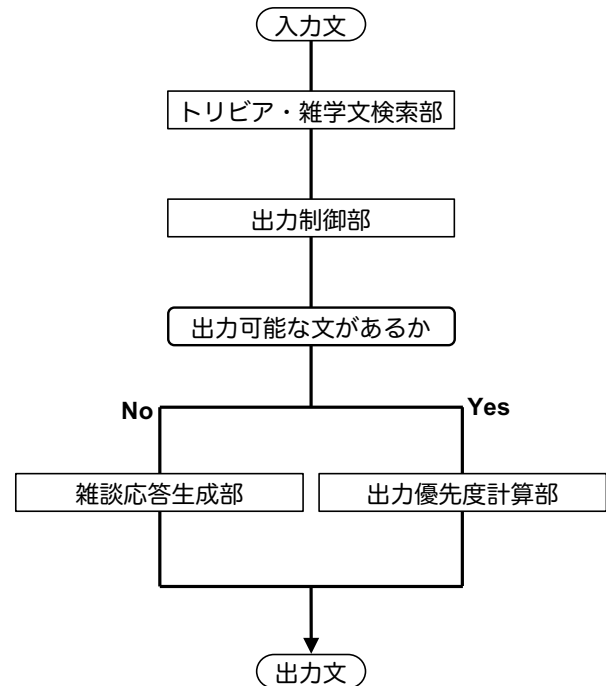


図 2 対話システムの概要図

Fig. 2 The outline of the dialogue system

4.1 トリビア・雑学文検索部

トリビア・雑学文検索部では、ユーザの入力文に含まれるキーワードから、そのキーワードに関するトリビア文および雑学文を検索する。まず、ユーザの入力文に含まれる名詞をキーワードとして抽出する。この時、抽出する名詞は形態素解析器 MeCab^{*3} によって一般名詞および固有名詞と判断された名詞のみとする。次に、Wikipedia のリダイレクト情報を用いて、キーワードをリダイレクト先の単語へ置換する。例えば、「クラッカーボール」は「癩癩玉」に置換される。キーワードを置換する理由は、「ネコ」と「猫」のように、表記揺れが起きた際に関連するトリビア文や雑学文を検索可能にするためである。最後に、置換したキーワードに関連するトリビア文および雑学文を検索する。トリビア文は、置換したキーワードを含むトリビア文を獲得する。検索の際は、トリビア文に含まれる名詞も、キーワード同様 Wikipedia のリダイレクト情報を用いて置換した後に検索する。雑学文については、置換したキーワードとタイトルが一致する Wikipedia の記事を獲得し、その記事の雑学文を獲得する。

*3 <http://taku910.github.io/mecab/>

4.2 出力制御部

出力制御部では、4.1 節で獲得したトリビア文および雑学文に対して、システムの出力として用いた場合に対話が破綻するかどうかの判定（対話破綻検知）を行う。対話破綻検知については、Sugiyama[13]の手法に基づき行う。Sugiyama は、対話行為や内容語の一致率などの素性と機械学習の手法を用いて、対話破綻検知器を構築している。本研究では、Sugiyama の利用した素性に基づき、以下の素性を用いて対話破綻検知器を構築する。

- Content Word Matching Rate
- Word Mover's Distance
- Similarity
- Number of Elapsed Turns

4.2.1 節以降で、各素性の詳細について説明する。

4.2.1 Content Word Matching Rate

人間同士の対話において、話し相手へ返答する際、話し相手が直前に発話した内容に言及することが多い。例えば、「今日も寒いですね」に対して、「こうも寒いと、肉まんでも食べたくなりますね」のように返答することが一般的である。よって、ある発話文とその直前の発話文との内容語の一致率（Content word matching rate）は、対話破綻検知に有効であると考えられる。したがって、本研究では、4.1 節で獲得した各トリビア文および雑学文と直前のユーザの入力文および1つ前のシステム発話との Content word matching rate を素性として利用する。なお、内容語は名詞、動詞、形容詞の3つとする。

4.2.2 Word Mover's Distance

Word Mover's Distance[14]は、ある2文間の編集距離の計算時に、編集距離のコスト部分を単語の分散表現[15]による類似度で置き換えた手法である。4.2.1 節で言及したように、一般的にある連続した2発話は、トピックやその内容が似ている。よって、2文間の類似性を測る Word Mover's Distance は対話破綻検知に有効であると考えられる。したがって、本研究では、Word Mover's Distance を素性として利用する。Word Mover's Distance を計算する文の組み合わせは以下である。

- (1) 各トリビア文および雑学文 & 直前のユーザの入力文
- (2) 各トリビア文および雑学文 & 1つ前のシステム発話文
- (3) 各トリビア文および雑学文 & 直前のユーザの入力文 + 1つ前のシステム発話文

なお、Word Mover's Distance の計算は、名詞および動詞のみを用いて計算する。

4.2.3 Similarity

4.2.2 節で2文間の類似度を Word Mover's Distance で計算するように、別の尺度による2文間の類似度も対話破綻検知に有効であると考えられる。したがって、本研究では、単語の分散表現に基づく類似度を素性として利用する。まず、各トリビア文および雑学文とユーザの入力文に対し

て、単語の分散表現に基づく文の分散表現を計算する。文の分散表現は、文中に含まれる全単語の分散表現の平均値とする。次に、算出した文の分散表現に基づき、各トリビア文および雑学文とユーザの入力文とのコサイン類似度を計算する。

4.2.4 Number of Elapsed Turns

Sugiyama は、対話システムは対話の序盤では比較的適切な発話を行うが、対話が進むにつれて不適切な発話が増加する傾向があると述べている。よって、発話の経過ターン数は対話破綻検知に有効であると考えられる。したがって、本研究では発話の経過ターン数を素性として利用する。

4.3 出力優先度計算部

出力優先度計算部では、4.2 節で非破綻であると判定された各トリビア文および雑学文に対して、どの文を優先して出力すべきかを表す出力優先度（output priority）を計算する。ある文 S に対する出力優先度 $OP(S)$ は式 (1) を用いて計算する。

$$OP(S) = i_score(S) \times \frac{\sum_{n \in Noun_S} k_score(S, n)}{|Noun_S|} \quad (1)$$

$$k_score(S, n) = \sum_{S \neq UFS \in A_n} i_score(UFS) + \sum_{S \neq TS \in TS_n} i_score(TS) \times 10 \quad (2)$$

式 (1) の n は名詞、 $Noun_S$ は文 S に含まれる名詞（一般名詞および固有名詞）の集合、 $i_score(S)$ は文 S の「面白さ」を表すスコア（Interestingness score）、 $k_score(S, n)$ は「知識の豊富さ」を表すスコア（Knowledge score）である。 $i_score(S)$ は、文 S が雑学文であった場合、表2にあるように、雑学文に付随する雑学スコアを最大値で正規化した値を返す。文 S がトリビア文であった場合、表1にあるように、各トリビア文に付随している「へえ」数を最大「へえ」数で正規化した値を返す。また、式 (2) の UFS は雑学文、 A_n は名詞 n とタイトルが一致する Wikipedia の記事における雑学文集合、 TS はトリビア文、 TS_n は名詞 n を含むトリビア文集合である。ここで、 A_n および TS_n は、既に対話中で出力したトリビア文および雑学文は含まないものとする。

4.4 雑談応答生成部

雑談応答生成部では、4.1 節でトリビア文および雑学文の検索結果が0文であった場合と、4.2 節で全てのトリビア文および雑学文が破綻であると判定された場合に、対話を継続させるために雑談応答を生成する。本研究では、雑談応答生成部は NTT docomo が提供している自然対話 API*4 を用いて構築する。

*4 https://dev.smt.docomo.ne.jp/?p=docs.api.page&api_name=natural_dialogue&p_name=api_usage_scenario

表 3 対話破綻検知の実験結果

Table 3 The result of the dialogue breakdown detection

	Precision	Recall	F-measure
O	0.63	0.57	0.60
X	0.75	0.79	0.77
Average	0.70	0.71	0.70

表 4 実験の対話例

Table 4 Example dialogue of the examination

発話者	発話文
ユーザ	こんにちは
システム	こんにちは。早速ですが、あなたの趣味は何ですか
ユーザ	私の趣味は読書です
システム	いいですね！読書は心を豊かにします・・・

5. 実験

構築した対話破綻検知器および対話システムについて実験を行った。5.1 節では、対話破綻検知の実験について説明する。5.2 節では、対話システムの実験について説明する。

5.1 対話破綻検知実験

4.2 節の対話破綻検知器について実験を行った。5.1.1 節では、実験設定について説明する。5.1.2 節では、実験結果について説明する。5.1.3 節では、5.1.2 節の内容を踏まえた考察について説明する。

5.1.1 実験設定

4.2 節で利用したデータセットは、対話破綻検出チャレンジ [16] で配布されているデータセット 550 対話を用いた。対話破綻検出チャレンジのデータセットは、1 対話に 21 発話が含まれており、ユーザ発話が 10 発話、システム発話が 11 発話となっている。また、システム発話には 30 人のアノテーターにより対話破綻タグ (X: 破綻, T: おそらく破綻, O: 非破綻) が付与されている。そのため、各システム発話に対して、タグを一意に決定する必要がある。本研究では、システム発話に付与する最終的なタグを X タグと O タグのみとした。システム発話に付与する最終的なタグは、X タグと O タグのうちタグの付与数が多い方とした。この時、T タグの付与数の半分を X タグの付与数に加えた上で、X タグと O タグの付与数を比較した。その結果、X タグが付与された発話は 3,709 発話、O タグが付与された発話は 2,341 発話となった。このデータセットを用いて、対話破綻検知器を構築した。

4.2.3 節で用いる単語の分散表現については、Word2Vec^{*5} を用いて獲得した。なお、次元数は 200 とした。

対話破綻検知器に用いる機械学習手法は Support Vector Machine (SVM) とした。カーネルは RBF カーネルを利用した。また、X タグと O タグの発話数が不均衡であるため、O タグに対するペナルティの重みを X タグの約 1.5 倍^{*6}に設定した。

評価については、10 分割交差検定で評価した。評価指標は、Precision, Recall, F-measure とした。

5.1.2 実験結果

対話破綻検知に対する実験結果を表 3 に示す。表 3 よ

り、Average の F-measure が 0.70 と比較的高い値を示していることが分かる。また、X タグの Recall が 0.79 と比較的高い値を示していることから、破綻する発話に関しては比較的高い精度で認識できていることが分かる。

5.1.3 考察

5.1.2 節の内容を踏まえ、トリビア文および雑学文を利用する対話システムへの有用性について考察する。5.1.2 節で言及したように、対話破綻検知としての結果は比較的高い精度となっていることが分かる。しかし、本研究で用いたデータセットには、トリビア文および雑学文が含まれていない。一般的にトリビア文および雑学文は雑談応答と比べ、文の長さや含まれている単語の品詞数などが大幅に異なる。よって、トリビア文および雑学文まで考慮した対話破綻検知器を構築するためには、トリビア文および雑学文を含んだ対話破綻データセットの作成や雑談応答文とは異なる特徴を表す素性の検討などを行う必要がある。

5.2 対話実験

構築した対話システムについて実験を行った。5.2.1 節では、実験設定について説明する。5.2.2 節では、実験結果について説明する。5.2.3 節では、5.2.2 節の内容を踏まえた考察について説明する。

5.2.1 実験設定

対話システムに利用するトリビア文は、3.1 節で説明した 1,031 文のトリビア文データセットを利用した。雑学文については、3.2 節で説明した 3,665,321 文の雑学文データセットを利用した。雑学文を獲得する Wikipedia の記事については、2017 年 5 月 21 日付でウィキメディア財団より提供されているデータベース・データ^{*7}から入手したものをを用いた。

比較対象については、提案手法から出力制御部と出力優先度計算部をそれぞれ片方のみ用いたモデル (図 2 参照) と、雑談応答生成部のみモデルとした。被験者の数は 5 人とした。

対話システムの実験方法について、各被験者とそれぞれの対話システムとの間で対話を行った。対話設定について、対話例を表 4 に示す。まず最初のユーザからの入力「こんにちは」とした。次に、システムの最初の出力は、雑談応答生成部で生成した雑談応答と表 5 中のいずれかの質問

^{*5} <https://radimrehurek.com/gensim/models/word2vec.html>

^{*6} 正確には 3709/2341 倍

^{*7} <https://dumps.wikimedia.org/jawiki/>

表 5 質問文

Table 5 The question sentences

質問文	応答文
あなたの趣味は何ですか	私の趣味は○○です
好きな食べ物は何ですか	私の好きな食べ物は○○です
旅行で行きたい国はどこですか	○○へ旅行に行ってみたいです
好きな動物は何ですか	好きな動物は○○です
休日によくすることは何ですか	休日にはよく○○をします

表 6 各評価項目の平均値

Table 6 The result of the dialogue system

モデル	Q1	Q2	Q3	Q4
提案手法	2.80	3.60	3.00	2.80
出力制御なし	3.20	3.20	2.40	3.60
出力優先度なし	3.20	3.20	3.00	3.40
雑談応答のみ	3.60	3.80	3.60	3.40

文とした。最後に、ユーザはシステムからの質問文に対して、その質問に応じた文（表 5 の応答文の項目）を入力してもらった。なお、表 5 の各質問文に対する応答文の○○については、ユーザに任意に埋めてもらった。以降の発話から、ユーザは任意の文を入力することとした。また、1 回の対話におけるユーザの発話数については最低 5 発話とした。対話を終了するタイミングは、被験者に一任した。なお、対話するモデルの順番は各被験者毎にランダムとした。また、質問文は被験者毎にランダムに割り振った。

評価については、対話を終了する毎に、対話したモデルに対して、Q1「対話の流れは自然であったか」、Q2「対話は楽しかったか」、Q3「話し相手として満足なものであったか」、Q4「また対話したいと思うか」のそれぞれの項目を 5 段階で評価してもらった。

5.2.2 実験結果

対話システムに対する実験結果を表 6 に示す。表 6 より、全ての評価項目に対して、提案手法が雑談応答のみのモデルを下回っていることが分かる。また、出力制御なしのモデルと雑談応答のみのモデルを比較すると、Q4 の項目で出力制御なしのモデルが雑談応答のみのモデルを上回っていることが分かる。このことから、トリビア文および雑学文を対話破綻を考慮せず出力する場合、再使用性の観点で有効であることが分かる。

5.2.3 考察

5.2.2 節の内容を踏まえ、提案手法における実験結果について考察する。まず、それぞれの対話モデルにおけるユーザ発話数やユーザの発話長について調査を行った。調査結果を表 7 および表 8 に示す。表 7 より、提案手法が他の手法に比べてユーザ発話数が 1 発話長いことが分かる。また、表 8 より、提案手法が他の手法に比べてユーザ発話長が 1 文字から 2 文字長いことが分かる。この結果から、提

表 7 各モデルの平均ユーザ発話数

Table 7 The values of the average number of user utterances

モデル	平均ユーザ発話数
提案手法	7.40
出力制御なし	6.20
出力優先度なし	5.80
雑談応答のみ	6.20

表 8 各モデルの平均ユーザ発話長

Table 8 The values of the average number of user utterance length

モデル	平均発話長
提案手法	12.97
出力制御なし	11.45
出力優先度なし	11.62
雑談応答のみ	11.00

案手法が他の手法に比べ、ユーザを対話により引き込むことができているといえる。しかし、長く対話を続けたことで、システムの欠点などが露呈しやすかったために、提案手法の評価が比較的低くなったのではないかと考えられる。

次に、実際の対話例から定性的な分析を行った。提案手法のモデルにおける対話例を表 9 に示す。雑談応答のみのモデルにおける対話例を表 10 に示す。表 9 における「○○といえば、『』と聞いたことがあります」という文の『

』中の文がトリビア文および雑学文である。表 9 の対話例では、「ヘビ」についての雑学文を出力している。また、表 9 の「物事をうまく進行させるのって、なかなか難しいよね。」のように、提案手法のモデルでは最初のトピックとは違うトピックで話を進行するケースが多く見られた。一方で、表 10 より、雑談応答のみのモデルではトピックが「オムライス」で一貫していることが分かる。これらの結果から、トピックの激しい遷移が提案手法に対する低い評価の原因であると考えられる。トピック遷移が激しい原因として、ユーザの入力文に含まれるキーワードからトリビア文および雑学文を検索したことと、雑談応答生成部でトリビア文および雑学文の出力を考慮できないことが考えられる。したがって、対話トピックの推定機構を導入することで、この問題を解決できると考えられる。また、トリビア文や雑学文は、一般的に話のネタが無くなる対話の中盤あたりで使用されると考えられる。よって、トリビア文および雑学文を出力しても良いターン数を考慮することで、より良い対話システムを構築できると考えられる。

6. むすび

本研究では、ユーザを楽しませる対話システムとして、トリビア文および雑学文を利用する対話システムを構築した。ユーザの入力文から得られたキーワードに関するトリビア文と雑学文を獲得し、対話破綻検知と出力優先度の 2

表 9 提案モデルの対話例

Table 9 Example dialogue of our dialogue system

発話者	発話文
ユーザ	こんにちは
システム	こんにちは 早速ですが、好きな動物は何ですか
ユーザ	私の好きな動物はヘビ
システム	いいですね! ヘビといえば、『日本においてもヘビは太古から信仰を集めていた』という事を聞いたことがあります
ユーザ	太古から進行を集めてたのか~
システム	物事をうまく進行させるのって、なかなか難しいよね。
ユーザ	確かに難しいことですね...
システム	日本語って上手くできてるよね。漢字で「ものごと」っていう意味がよく伝わるもん。
ユーザ	ものごと
システム	それは残念!
ユーザ	悲しい!
システム	どうしたの . . .

表 10 雑談応答のみのモデルの対話例

Table 10 Example dialogue of the baseline

発話者	発話文
ユーザ	こんにちは
システム	こんにちは 早速ですが、あなたの好きな食べ物は何ですか
ユーザ	私はオムライスが好きです
システム	いいですね! おいしいオムライスは、やっぱり、ふわふわトロトロだよね。
ユーザ	そうですね! 作るのが難しいですけど美味しいですね
システム	卵に牛乳を入れてオムライスを作ると、やわらかく仕上がるんだって。
ユーザ	そうなんですか! 初めて知りました!
システム	たしかにオムライスはそうですね
ユーザ	オムライスはチキンライスも美味しいので好きです
システム	オムライスは大好きよねー . . .

つに基づき各文の出力をコントロールする対話システムを構築した。実験の結果、対話破綻検知については、比較的高い精度で対話破綻検知を行うことができた。しかし、本研究で利用したデータセットにはトリビア文および雑学文が含まれていなかったため、トリビア文および雑学文に関する対話破綻検知モデルの構築が必要であることが分かった。対話システムについては、全ての評価項目において、提案手法が雑談応答のみのモデルを下回る結果となった。分析の結果、トピック遷移の激しさなどが原因で、提案手法の評価が低いものとなった可能性が示唆された。

今後の課題として、対話破綻検知については、新たな素性の検討やトリビア文および雑学文を含む対話破綻検知データセットの作成が挙げられる。対話システムについては、対話トピックの推定機構の導入や対話戦略の再検討が挙げられる。具体的には、対話の大筋のトピックを推定する機構の導入や対話の経過ターン数を考慮したトリビア文および雑学文出力モデルの検討などが挙げられる。

参考文献

- [1] Vinyals, O. and Le, Q.: A neural conversational model, *arXiv preprint arXiv:1506.05869* (2015).
- [2] Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J. and Jurafsky, D.: Deep reinforcement learning for dialogue generation, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, D16-112*, Valencia, Spain, pp. 1192-1202 (2016).
- [3] 新名和也, 嶋田和孝: シズルワードから想起される料理の雑学を話す対話システム, 電子情報通信学会, 言語理解とコミュニケーション研究会 (NLC), 第4回自然言語処理シンポジウム, Vol. 117, pp. 77-82 (2017).
- [4] 新名和也, 嶋田和孝: トリビア文抽出のためのトリビア度合いの推定, 電子情報通信学会, 言語理解とコミュニケーション研究会 (NLC) (2018).
- [5] Kazuya, N. and Kazutaka, S.: Trivia Score and Ranking Estimation Using Support Vector Regression and RankNet, *Proceedings of PACLIC32* (2018).
- [6] 稲葉通将, 吉野友香, 高橋健一: 対話システムが話し手役になるためのオープンドメイン独話生成, 人工知能学会論文誌, Vol. 31, No. 1, pp. DSF-F1-9 (オンライン), DOI: 10.1527/tjsai.DSF-505 (2016).
- [7] 江頭勇佑, 柴田知秀, 黒橋禎夫: 雑談対話システムにおける強化学習を用いた応答生成モジュールの選択, 言語処理学会第18回年次大会発表論文集, D3-7, Vol. 18, pp. 654-657 (2012).
- [8] Prakash, A., Chinnakotla, M. K., Patel, D. and Garg,

- P.: Did You Know?—Mining Interesting Trivia for Entities from Wikipedia, *Twenty-Fourth International Joint Conference on Artificial Intelligence*, pp. 3164–3170 (2015).
- [9] Fatma, N., Chinnakotla, M. K. and Shrivastava, M.: The Unusual Suspects: Deep Learning Based Mining of Interesting Entity Trivia from Knowledge Graphs, *Thirty-First AAAI Conference on Artificial Intelligence*, pp. 1107–1113 (2017).
- [10] Tsurel, D., Pelleg, D., Guy, I. and Shahaf, D.: Fun Facts: Automatic Trivia Fact Extraction from Wikipedia, *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, ACM, pp. 345–354 (2017).
- [11] 太田知宏, 鳥海不二夫, 石井健一郎: 発話生成を目的とした Wikipedia からの文抽出, 人工知能学会第 23 回全国大会, 2G1-NFC5-11 (2009).
- [12] 杉本俊, 植木拓, 林宏幸, Eric, N., 中野幹生: Wikipedia からの特定ドメインの雑談対話システムのための発話候補文集合の獲得, 人工知能学会第 31 回全国大会, 3A1-3 (2017).
- [13] Sugiyama, H.: Dialogue breakdown detection based on estimating appropriateness of topic transition, *Dialog System Technology Challenges (DSTC6)* (2017).
- [14] Kusner, M., Sun, Y., Kolkin, N. and Weinberger, K.: From word embeddings to document distances, *International Conference on Machine Learning*, pp. 957–966 (2015).
- [15] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient Estimation of Word Representations in Vector Space, *CoRR*, Vol. abs/1301.3781, pp. 1–12 (online), available from <http://arxiv.org/abs/1301.3781> (2013).
- [16] 東中竜一郎, 船越孝太郎, 小林優佳, 稲葉通将ほか: 対話破綻検出チャレンジ, *SIG-SLUD*, Vol. 5, No. 02, pp. 27–32 (2015).